Tutorial básico de utilização do Git

Utilizando no Visual Studio 2015

## Criando um repositório local

Para dar início aos trabalhos, vamos levar em conta que estamos começando um projeto novo e que a única coisa que já temos é um **repositório** **remoto vazio**. Sendo assim a primeira atividade que deve ser feita é a criação de um **repositório local**.

<acao>

## Ligando a um repositório remoto

Como já existe um repositório remoto vazio e agora também existe um repositório local. O próximo passo que deve ser feito a criação de uma ligação entre os dois repositórios para que o código que está sendo versionado localmente possa ser compartilhado na rede.

<acao>

## Clonando um repositório remoto

A outra forma de iniciar um trabalho de versionamento de código usando Git seria se já existisse um projeto no repositório remoto. Esse cenário é, talvez, o mais comum dos casos, pois isso acontece quando vamos dar continuidade a um trabalho que tinha sido iniciado anteriormente. Nesse caso o que será feito é um **clone do repositório remoto**. Quando um repositório remoto é clonado, o usuário que fez o clone passa a ter um **repositório local idêntico ao repositório que foi clonado**. Vale lembrar que quando se clona um repositório remoto, ele automaticamente já possui uma ligação entre o local e o remoto.

<acao>

## Verificando o status dos arquivos do repositório

Como já foi visto inicialmente, quando foi falado sobre o ciclo de vida dos arquivos versionados, um arquivo não é adicionado automaticamente a ***staging area*** de modo que é necessário indicar quais arquivos devem ser adicionados. Para saber quais arquivos foram adicionados, alterados ou excluídos é necessário ver o ***status*** dos arquivos que estão no *working directory* do repositório

<acao>

## Gravando o arquivo no repositório

A partir do momento em que um ou mais arquivos foram adicionados a *staging area*, os mesmos já estão sendo rastreados pelo Git e, portanto é possível gravar efetivamente no repositório local. O nome dessa ação é ***commit***. Vale lembrar que, no Git, todos os *commits* precisam ter uma mensagem descrevendo o que foi feito.

<acao>

**Verificando o histórico de alterações**

Para verificar todo o histórico de alterações que foram feitas no repositório local, é possível ver através do log do Git.

<acao>

## Enviando alterações para um repositório remoto

Como já existe um repositório ligado ao repositório local que foi criado. Agora é o momento de enviar para a rede todas as alterações que foram feitas localmente. Para isso, será utilizado o ***push***.

<acao>

Lembrando que como a partir do momento em que um repositório remoto está sendo utilizado, acredita-se que não existe apenas uma única pessoa fazendo alterações nesse repositório remoto. Sendo assim, para que não haja conflito com eventuais trabalhos feitos por outras pessoas, é recomendado que antes do *push*, seja feito um ***pull*** para pegar eventuais alterações feitas por terceiros.

<acao>