ANEXO IC

GUIA DE UTILIZAÇAO (GIT)

|  |
| --- |
| Objetivo/Introdução |

Esse documento tem por objetivo oferecer um guia padronizado de utilização do GIT para usuários internos e externos.

|  |
| --- |
| Pré-requisitos para Execução das Atividades |

O usuário do GIT deve ter acesso prévio criado pelo administrador.

|  |
| --- |
| Referências |

<https://git-scm.com/>

|  |
| --- |
| Descrição das Atividades |

## Instalação

O GIT deve ser instalado na máquina localmente a partir dos link abaixo:

<https://git-scm.com/downloads>

Para realizar a configuração corretamente, deve ser o guia padrão de instalação do GIT a partir do link:

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

## Clonando projetos

**Passo 1:** Para ter acesso à URL do projeto acesso o repositório na Azure Dev ops e clique em "Clone" no canto superior direito.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

**Passo 2:** Para copiar a URL do projeto basta clonar no botão "*Copy clone URL to clipboard*" com na imagem abaixo.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

**Passo 3:** Para baixar o projeto para a máquina local é necessário acessar o diretório desejado por linha de comando e executar o seguinte comando.

git clone + url do repositório remoto

## Fluxo de *commit* e *push*

O GIT possui três áreas para três estágios diferentes no processo de *commit*.

Uma imagem contendo placar

Descrição gerada automaticamente

A primeira é o **Working Directory** onde ficam as alterações no código.

Para verificar os arquivos que foram alterados digite o comando:

git status

Os arquivos alterados serão destacados na cor vermelha conforme imagem abaixo:

Uma imagem contendo screenshot

Descrição gerada automaticamente

### GIT add

Antes de fazer o commit é necessário adicionar os arquivos alterados no **Index**, que funciona como uma área pré-commit.

**Adicionando arquivos individuais ao index:**

Para adicionar um único arquivo ao index na intenção de fazer o commit apenas dele execute comando abaixo:

**git add + path do arquivo**

A imagem abaixo mostra como o path deve ser informado.

Uma imagem contendo comida

Descrição gerada automaticamente

O arquivo adicionado ficará destacado na cor verde conforme imagem abaixo.

Uma imagem contendo screenshot

Descrição gerada automaticamente

Adicionando todos os arquivos alterados ao index ao mesmo tempo:

Para adicionar todos os arquivos alterados ao mesmo tempo basta digitar o comando:

git add

Todos os arquivos ficarão destacados na cor verde conforme a imagem abaixo:

Uma imagem contendo screenshot

Descrição gerada automaticamente

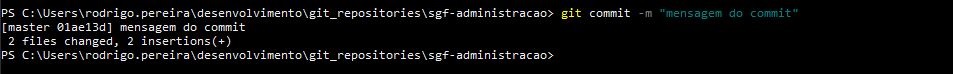
### GIT commit

Agora que todos os arquivos estão no index ou seja estão prontos para o commit, basta digitar o comando abaixo:

git commit -m + "mensagem do commit"

Todo commit precisa ter uma mensagem resumida do contexto das alterações e isso é possível através da flag -m seguido da mensagem entre aspas.

Após a execução do commit será informada a quantidade de arquivos alterados e inserções no código conforme a imagem abaixo.



### GIT push

Por enquanto os arquivos estão no Local Repository e é necessário subir para o repositório na Azure Dev Ops.

Para fazer isso se utiliza o comando push acompanhado do nome repositório e do nome da branch.

git push + nome repositorio + nome da branch

O nome default para os repositórios do git é "origin" porém esse nome pode ser outro. No caso do exemplo abaixo o default foi utilizado.

Uma imagem contendo screenshot, texto

Descrição gerada automaticamente

## Atualizando o Código com o Pull

É muito importante manter o repositório local atualizado com o remoto. Para esse procedimento é necessário utilizarmos o seguinte comando:

git pull + nome do repositório + nome da branch

No exemplo abaixo estamos atualizando nossa branch "master" local de acordo com a branch master no repositório remoto.

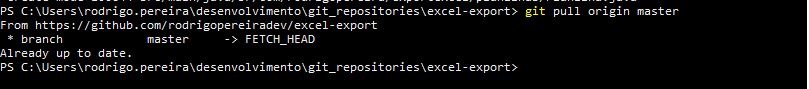


Caso existam atualizações, os arquivos atualizados serão listados indicando a inclusão ou remoção de linhas de código. A imagem abaixo nos dá um exemplo dessa situação.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Caso não existam atualizações a mensagem "Already up to date" será apresentada conforme exemplo abaixo.



## Criação e mudanças de branches

### Criando uma nova branch

Para criarmos uma branch é bem simples:

git branch + nome da nova branch

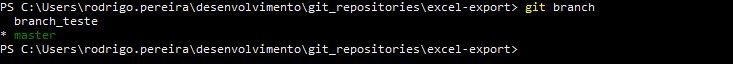
A imagem abaixo tem um exemplo prático da execução do comando.



Para verificar a lista de branchs locais existentes basta executar o seguinte comando:

git branch

Após a execução do comando uma lista de branchs será exibida. A branch na qual o usuário estiver será destacada com uma asteriscos e na cor verde conforme exemplo abaixo.



### Mudando de branch

Para mudar de branch basta digitar o comando abaixo.

git checkout + nome da branch

A imagem abaixo exemplifica o comando.



Uma mensagem informando que houve a mudança de branch será informada.



### Criando e mudando para a nova branch ao mesmo tempo

É possível o usuário criar uma branch e mudar para ela ao mesmo tempo. Para isso basta utilizar a flag -b ao executar o comando abaixo.

git checkout -b + nome da nova branch

O exemplo abaixo demonstra como o comando é executado com uma mensagem informando que houve uma mudança para a nova branch.



### Criando branches no repositório remoto

Agora que criamos uma branch local é preciso criar uma versão remota dela. Para isso basta executar o comando abaixo.

git push + nome do repositorio remoto + nome da branch local

Na imagem abaixo mostra um exemplo de como dever ser executado o comando. Ressaltamos que "origin" é o nome default do repositório remoto e ele pode mudar de acordo com o projeto.



Após execução alguma informações serão apresentadas e uma delas é de que uma nova branch foi criada no repositório.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Para verificarmos a lista de branch remotas basta digitarmos o comando normal acompanhado da flag **-a**.

git branch -a

As branchs remotas serão apresentadas destacadas na cor vermelha conforme o exemplo abaixo.

Texto preto sobre fundo branco

Descrição gerada automaticamente

Caso o comando não esteja listando todas as branchs remotas você pode rodar o comando abaixo para atualizar essa consulta.

git fetch

A imagem abaixo mostra a execução do fetch.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Observe que o comando mapeou duas novas branches.

## Erros GIT

Caso ocorra o erro "*File name to long*" abrir o power shell como administrador e executar o seguinte comando: git config --system core.longpaths true. Tente clonar o repo novamente.

|  |
| --- |
| Utilização de Dados Pessoais (LGPD) |

Não contempla a utilização de Dados Pessoais, conforme a Lei Geral de Proteção de Dados Pessoais (LGPD).

|  |
| --- |
| Anexos |

N/A.

|  |
| --- |
| Controle de Versão |

Histórico (para uso da Gerência de Governança):

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data Publicação | Elaborador | Motivo da Versão |
|  |  |  |  |

Aprovação da Versão Atual:

|  |  |
| --- | --- |
| Nome | Cargo |
|  |  |