Se a instalação correu normalmente, o comando git deve mostrar uma saída como a abaixo, mostrando como usar a ferramenta e quais os comandos mais comuns.

usage: git [--version] [--help] [-C <path>] [-c name=value]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

reset Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept.

git blame "arg.xxx": Este comando, seguido pelo nome de um arquivo, mostra a última pessoa que alterou cada linha do mesmo e a data da alteração.

git add pode ser executado de modo interativo com a opção -i. git add -i, o modo interativo inicia um prompt de comando específico, aguardando as decisões do usuário sobre quais as alterações detectadas no working directory devem ser adicionadas ao index.

1. Rode o comando git add -i

$ git add -i

staged unstaged path

1: unchanged +1/-0 index.html

\*\*\* Commands \*\*\*

1: status 2: update 3: revert 4: add untracked

5: patch 6: diff 7: quit 8: help

What now>

1. Para adicionar arquivos alterados, utilize a opção update. Você pode digitar tanto o número, quanto a palavra escrita como também somente a primeira letra dela. No caso, utilizamos aqui a abreviação u e pressione enter.

What now> u

staged unstaged path

1: unchanged +1/-0 index.html

1. Aparecerá uma listagem dos arquivos modificados, com um número atribuído a cada arquivo. Digite o número que representa o arquivo index.html e pressione enter .

Update>> 1

staged unstaged path

\* 1: unchanged +1/-0 index.html

1. Aparecerá um asterisco ao lado do número do arquivo, indicando que ele será adicionado. Pressione enter novamente para confirmar e voltar ao menu principal.

Update>>

updated one path

\*\*\* Commands \*\*\*

1: status 2: update 3: revert 4: add untracked

5: patch 6: diff 7: quit 8: help

1. Se quiser, confira as modificações. Utilize a opção status para conferir que o arquivo index.html foi adicionado corretamente, digitando a opção e pressionando enter.

What now> status

staged unstaged path

1: +1/-0 nothing index.html

\*\*\* Commands \*\*\*

1: status 2: update 3: revert 4: add untracked

5: patch 6: diff 7: quit 8: help

1. Utilize a opção quit para terminar a interação!

What now> 7

Bye.

$

TECLAS DE ATALHO GIT BASH

Tab ⇆ : Autocompleta o comando a partir da posição do cursor.

Ctrl+a : Move o cursor para o inicio da linha, equivalente a tecla Home.

Ctrl+b : Move o cursor um caractere atrás.

Ctrl+c : Envia o sinal SIGINT para tarefa atual, então esta é cancelada e fechada.

Ctrl+d : Envia o marcador EOF, este fecha a sessão atual do shell, igual ao exit.

Ctrl+d : Deleta o caracter corrente, se existirem caracteres digitados.

Ctrl+e : Move o cursor para o final da linha, equivalente a tecla End.

Ctrl+f : Move o cursos para o caracter seguinte.

Ctrl+g : Aborta a pesquisa e restaura a linha original.

Ctrl+h : Deleta o caractere anterior, equivalente a tecla Backspace.

Ctrl+i : Equivalente a tecla Tab.

Ctrl+j : Equivalente a tecla Enter.

Ctrl+k : Limpa o conteúdo da linha que está após o cursor e copia

este conteúdo para área de memoria.

Ctrl+l : Limpa o conteudo da tela, equivalente ao comando clear.

Ctrl+n : Recupera o próximo comando (equivalente a tecla ↓).

Ctrl+o : Executa o comando encontrato do histórico.

Ctrl+p : Recupera o comando anterior (equivalente a tecla ↑).

Ctrl+q : Adiciona o próximo caractere digitado à linha textualmente.

Ctrl+r : Executa o ultimo comando digitado.

Ctrl+s : Volta ao próximo comando digitado recentemente.

Ctrl+t : Avançar os dois últimos caracteres.

Ctrl+u : Limpa o conteúdo da linha anterior ao cursor e copia o conteúdo para

área de memoria.

Ctrl+v : Adiciona o próximo carácter à linha textualmente.

Ctrl+w : Apaga a palavra depois do cursor e copia para área de memória.

Ctrl+y : Comando colar.

Ctrl+x Ctrl+e : Edita a linha atual no programa $EDITOR, ou vi.

Ctrl+x Ctrl+r : Ler o conteúdo do arquivo inputrc, e incorporar todas as

ligações ou atribuições de variáveis ​​encontradas lá.

Ctrl+x Ctrl+u : Desfazer, separado por linha.

Ctrl+x Ctrl+v : Exibe informações sobre a versão da instancia corrente do bash.

Ctrl+x Ctrl+x : Alterar a posição do cursor para sua antiga posição.

(C-x, porque x tem uma forma de passagem).

Ctrl+z : Envia o sinal para SIGTSTP a tarefa atual, para suspende-lo. Para

executá-lo em segundo plano pode-se entrar no bg. Para trazer de

volta através de fundo ou suspensão fg ['nome do processo ou

identificação de trabalho "] (primeiro plano) pode ser emitido.

Ctrl+\_ : Desfazer, separado por linha.

Alt+b : move o cursor para palavra anterior.

Alt+c : Capitaliza o caracter sob o cursor e move para o final da palavra.

Alt+d : Corta a palavra depois do cursor.

Alt+f : move o cursor para próxima palavra.

Alt+l : Transforma para caixa baixa todo caractere a partir do cursor

em toda palavra.

Alt+r : Cancela as alterações e colocar de volta a linha de como era na história.

Alt+u : Transforma para caixa alta todo caractere a partir do cursor em toda

palavra.

Alt+. : Insira o último argumento para o comando anterior (a última palavra da

entrada história anterior).