

ATIVIDADE: ESTADOS E DIFERENÇAS

1. Altere o arquivo hello.java para que gere dez saídas "Hello Git!"

Edite o conteúdo do arquivo hello.java para conter:

```
public class Hello {  
    public static void main(String[] args) {  
        for (int i = 0; i < 10; i++) {  
            System.out.println("Hello Git!");  
        }  
    }  
}
```

2. Observe o "status" e diferenças, a seguir registre a versão atualizada no repositório

Execute os seguintes comandos no terminal:

git status

```
On branch main  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git restore <file>..." to discard changes in working directory)  
        modified:   hello.java  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

git diff hello.java

```
diff --git a/hello.java b/hello.java  
index 536712d..656eb77 100644  
--- a/hello.java  
+++ b/hello.java  
@@ -1,5 +1,7 @@  
 public class Hello {  
     public static void main(String[] args) {  
-        System.out.println("Hello Git!");  
+        for (int i = 0; i < 10; i++) {  
+            System.out.println("Hello Git!");  
+        }  
     }  
-}  
 \ No newline at end of file  
+}
```

git add hello and git commit -m "Atualiza hello.java para imprimir 10x 'Hello Git!'"

```
On branch main  
nothing to commit, working tree clean
```

ATIVIDADE: ESTADOS E DIFERENÇAS

3. Observe a diferença entre os arquivos. Repita o processo fazendo novas modificações

- ✓ Modificado, o arquivo hello.java: `System.out.println("Hello GitHub!");`

On branch main

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: hello.java

no changes added to commit (use "git add" and/or "git commit -a")

- ✓ `git diff hello.java`

`diff --git a/hello.java b/hello.java`

`index 656eb77..b762d16 100644`

`--- a/hello.java`

`+++ b/hello.java`

`@@ -1,7 +1,7 @@`

`public class Hello {`

`public static void main(String[] args) {`

`for (int i = 0; i < 10; i++) {`

`- System.out.println("Hello Git!");`

`+ System.out.println("Hello GitHub!");`

`}`

`}`

`}`

- ✓ `git add hello.java` and `git commit -m "Alteração: mensagem trocada para 'Hello GitHub!'"`

`[main de4c723] Alteração: mensagem trocada para 'Hello GitHub!'`

`1 file changed, 1 insertion(+), 1 deletion(-)`

On branch main

nothing to commit, working tree clean

✓ 4. Crie um arquivo "hello.txt", inclua um texto com algumas orientações sobre o Git e adicione ao repositório

1. Crie o arquivo com o conteúdo:

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

hello.txt

nothing added to commit but untracked files present (use "git add" to track)

2. Modificação e commit de hello.txt

Mensagem: "Git é um sistema de controle de versões distribuído."

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

hello.txt

nothing added to commit but untracked files present (use "git add" to track)

ATIVIDADE: ESTADOS E DIFERENÇAS

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: hello.txt

[main 27c67aa] Adição do arquivo de texto com mensagem

1 file changed, 1 insertion(+)

create mode 100644 hello.txt

On branch main

nothing to commit, working tree clean

5. Renomeie o arquivo para "hello.md"

git mv hello.txt hello.md

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

renamed: hello.txt -> hello.md

git commit -a -m "Renomeia hello.txt para hello.md"

[main 21d0366] Renomeando hello.txt para hello.md

1 file changed, 0 insertions(+), 0 deletions(-)

rename hello.txt => hello.md (100%)

On branch main

nothing to commit, working tree clean

6. Remova o arquivo do repositório

git rm hello.md

rm 'hello.md'

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

deleted: hello.md

git commit -a -m "Remove o arquivo hello.md do repositório"

unknown option: -m

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

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

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]

[--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]

[--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]

<command> [<args>]