

ATIVIDADE: IGNORANDO ARQUIVOS

1. Descreva o resultado esperado para um repositório com um arquivo .ignore contendo:

Criado arquivo .ignore, contendo:

```
!lib.a
/TODO
build/
doc/*.txt
doc/**/*.pdf
```

Interpretação de cada linha:

- !lib.a
→ Não ignorar o arquivo lib.a, mesmo que um padrão anterior diga para ignorá-lo.
- /TODO
→ Ignora apenas o arquivo ou pasta TODO localizado na raiz do repositório.
- build/
→ Ignora qualquer pasta chamada build e todo o conteúdo dentro dela, em qualquer nível.
- doc/*.txt
→ Ignora arquivos .txt diretamente dentro da pasta doc/, mas não em subpastas.
- doc/**/*.pdf
→ Ignora todos os arquivos .pdf dentro de doc/ e suas subpastas recursivamente.

2. Inclua um arquivo hello.tmp no repositório criado na atividade anterior.

```
hellogit> hello.tmp
```

3. Observe o "status" deste arquivo, sem incluí-lo no repositório.

Comando executado:

```
git status
```

O resultado deve mostrar:

```
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .ignore
    hello.tmp
nothing added to commit but untracked files present (use "git add" to track)
```

ATIVIDADE: IGNORANDO ARQUIVOS

Ou seja, o Git está monitorando que o arquivo existe, mas ainda não está versionando (untracked).

4. Altere o repositório para que passe a ignorar este arquivo.

1. Editado o arquivo .ignore:
2. Adicionou-se, a seguinte linha no final: *.tmp
3. Salvou e fechou o arquivo.
4. Execute git status novamente:

On branch main

Untracked files:

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

.gitignore

.ignore

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

Agora, o arquivo hello.tmp não está mais, sendo exibido como "untracked", pois o Git está ignorando esse tipo de arquivo.

Confirmando:

git check-ignore -v hello.tmp

```
projetos_git\hellogit>git check-ignore -v hello.tmp
```

```
.gitignore:1:*.tmp    hello.tmp
```

Isso mostra que uma linha do .gitignore está fazendo o Git ignorar o arquivo.