Git Ignore

Use o .gitignore para ignorar os arquivos que você quiser.

No projeto clonado na última aula prática, os arquivos que clonamos correspondiam ao projeto completo do NetBeans. O problema de se armazenar num repositório git todos os arquivos de um projeto do NetBeans é que o repositório conterá não apenas códigos fonte, mas também códigos específicos do NetBeans de quem criou ou está usando o projeto. Consequentemente, teremos problemas como o da última aula prática, quando, ao clonar o projeto e tentarmos abrí-lo no NetBeans, este indicou que a JDK referenciada não existia.
Mais uma inconveniência que pode acontecer é relacionada ao git merge. O git não faz merge de arquivos binários. No repositório clonado havia arquivos .class, que são binários. Quando for necessário fazer merge e arquivos binários estiverem sido modificados no merge, o git não conseguirá fazer o merge.
Para solucionar esses problemas, contamos com o arquivo .gitignore. Este arquivo diz ao git o que podemos ignorar no projeto. Para um projeto NetBeans, podemos usar, por exemplo, um arquivo com o conteúdo abaixo:

```
nbproject/private/
build/
nbbuild/
dist/
nbdist/
nbactions.xml
.nb-gradle/
```

Isso diz ao git para ignorar todas as pastas listadas acima e também o arquivo listado acima.

Tarefas

1. Crie um repositório git, numa pasta que você deve criar para este roteiro.
2. Crie dois arquivos texto no repositório, cada um com um conteúdo diferente. Um `git status` deve mostrar algo parecido com o abaixo:

```
On branch master
Initial commit
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arq1.rtf
    arq2.rtf
nothing added to commit but untracked files present (use "git add"
to track)
```

3. Adicione um arquivo .gitignore na sua pasta de trabalho. Há duas formas de se fazer isso, uma é criando um arquivo .gitignore na pasta, a outra é digitar o comando touch .gitignore.
4. Após criar o arquivo .gitignore, adicione o nome do arquivo que quer ignorar, no arquivo .gitignore (abrindo-o em qualquer editor de texto) e salve o arquivo.
5. Execute um git status e veja se o arquivo que você quer ignorar realmente está sendo ignorado.
6. Crie um commit inicial para seu arquivo e o arquivo .gitignore.
7. Crie um branch, mude para esse branch e altere algo no seu arquivo. Crie um commit para essa alteração.
8. Volte para o master.
9. Altere algo no seu arquivo, na mesma linha que você alterou no outro branch. Crie um commit para essa alteração.
10. Tente fazer o emerge, no master, com o branch que você criou.

11. Se você fez todos os passos corretamente, deve aparecer a seguinte mensagem de erro:

```
Auto-merging arq1.rtf
CONFLICT (content): Merge conflict in arq1.rtf
Automatic merge failed; fix conflicts and then commit the result.
```

Resolva o conflito. O seu arquivo deverá estar mais ou menos assim:
<<<<< HEAD
informação que você mudou no HEAD
>>>>> novobranch
informação que você mudou no novobranch

Decida qual é a informação que você quer e salve o arquivo. Por exemplo, o arquivo poderá ficar assim:
informação que mudei no HEAD

Detalhe: você pode colocar qualquer informação. Não apenas as duas mudanças indicadas pelo git no conflito.

12. Para que as mudanças tenham efeito, é necessário fazer um commit. Faça o commit.
Use o comando `git log` para demonstrar o que você faz para o professor.