|  |  |
| --- | --- |
| **Leitura de Software**  **Docente: Prof. Fábio Nogueira de Lucena**  **Discente: Alexandre José Messias de Lara**  **Matrícula: 201206315** |  |

# **Fábrica de Software**

# **Documento: fs-mobile-2**

# **Git (exercícios adicionais)**

1. **Siga atentamente os *slides* disponíveis** [**AQUI**](https://docs.google.com/presentation/d/183OiLbly9Q80Pv7gl19bZyrGtGwu21r-7s9OIb07xSM/edit?usp=sharing)**. Experimente os comandos, ambiente-se com eles. Esclareça suas dúvidas com o professor. Faça uso de repositórios locais e, para boa parte dos comandos, terá que interagir com o GitHub (serviço sugerido na disciplina). Faça isto antes de prosseguir com os demais itens.**
2. **Qual o comando para que as “marcas” ou *tags* sejam enviadas para o repositório remoto? (um simples git push não produz este efeito)**

git push origin <nome\_da\_tag>

1. **Qual o nome do *branch* padrão do Git?**

master

1. **O que o comando git branch <branchname> realiza?**

Cria uma nova branch com o nome especificado

1. **Como criar um *branch* a partir de um *commit* específico?**

git branch <nome> <commit>

1. **Em um repositório, qual o efeito do comando git branch erro1234?**

Cria uma nova branch com o nome erro1234

1. **Qual o comando para se alternar para um *branch* de nome experimento2?**

git checkout experimento2

1. **Em um repositório com dois *branches*, b1 e b2, onde b1 é o corrente, qual o efeito do comando git branch?**

Mostrará ambos os branches, porém o b1 estará com um asterisco na frente.

1. **O que o comando git checkout -b novobranch faz?**

Cria um novo branch e alterna pra ele, tornando-o o branch corrente.

1. **Qual a função do comando git branch -d teste?**

Deleta o branch de nome teste

1. **Durante o desenvolvimento de um software é comum, por exemplo, utilizar um novo recurso por meio de experimentação. Talvez uma nova tecnologia, uma nova biblioteca que pode ser útil ao que está em desenvolvimento, ou até mesmo uma nova versão de um produto já empregado. Para que o uso deste novo recurso não interfira com o que é considerado pronto, um *branch* pode ser criado para a experimentação. Código que for criado para a experimentação existirá apenas no *branch* criado. Se eventualmente o experimento demonstrar um resultado satisfatório, as alterações realizadas no *branch* poderão ser incorporadas no que é considerado pronto, ou seja, no *branch* principal (*master*). Esta última ação é conhecida por *merge.* Neste item, apresente uma sequência de comandos que simula um caso simples de criação e uso seguido de *merge* empregando um *branch* para ilustrar uma experimentação conforme acima.A sequência deve incluir, obrigatoriamente: (a) criação de um ou mais *branches;* (b) chaveamento para pelo menos dois *branches* e (c) *merge*. Para simular alteração em um arquivo, basta simplesmente fornecer algo como Arquivo <nome> é alterado. O que foi fornecido em negrito representa uma ação que altera um arquivo cujo nome é fornecido entre o sinal de menor e o de maior.**

git branch develop

git checkout develop

arquivo teste.txt é criado

git add teste.txt

git commit -m “Commit inicial”

git branch develop2

git checkout develop2

arquivo teste.txt é alterado

git add -u

git commit -m “Alteração no arquivo teste.txt”

git checkout master

git merge develop

git merge develop2