

1. Instalar o *software* Git nos computadores pessoais.
 - (a) Aceder ao endereço <https://git-scm.com/downloads> e escolher o sistema operativo para o qual se pretende descarregar a aplicação Git.
 - (b) Seguir as instruções de instalação específicas para o sistema operativo.
 - (c) Criar uma pasta com o nome `labp` numa localização à escolha. Será dentro desta página que se vai criar repositórios e exercitar os comandos Git.
 - (d) Em ambiente *Windows* abrir o programa `gitbash` nesta pasta. Para sistemas operativos *Linux* e *MacOS* abrir uma janela de terminal nesta pasta.
2. Criar um repositório no *BitBucket* e cloná-lo para dentro da pasta `labp` utilizando a interface de linha de comandos do Git.
 - (a) Criar uma conta no *BitBucket*. Para tal aceder ao endereço <https://bitbucket.org>, apertar o botão *Get started* e completar os passos requeridos.
 - (b) Criar um repositório no *BitBucket* com o nome `labp20N...`, em que `N...` é o vosso número de aluno. Para tal, apertar o botão *+* à direita e escolher *Repository*.
 - (c) Usar o botão *Create a README* para adicionar um ficheiro ao repositório remotamente. Tendo um repositório não vazio facilita a interação futura com este.
 - (d) No topo da página de *Overview* do repositório aparece o comando que devem executar para clonar o repositório, ou seja, para fazerem uma cópia integral do repositório para as vossas máquinas. Existem dois protocolos que podem ser usados: *SSH* e *HTTPS*. Para agora, escolher *HTTPS*, copiar o comando e colar na linha de comandos que está aberta na pasta `labp`.
3. Exercitar a interação entre o repositório e o diretório de trabalho utilizando a interface de linha de comandos do Git.

- (a) Adicionar a esta pasta dois ficheiros de texto com nomes `LEIA_ME` e `LEIA_ME_A_SEGUIR` contendo a informação que achem adequada, mas com pelos menos 4 linhas de texto.
- (b) Verificar o resultado usando o comando `git status`.
- (c) Adicionar o ficheiro `LEIA_ME` ao índice (do inglês *stage* ou *index*) do repositório utilizando o comando `git add LEIA_ME`.
- (d) Verificar novamente o resultado usando o comando `git status`. De notar que o ficheiro `LEIA_ME` está listado na secção *Changes to be committed* enquanto que o ficheiro `LEIA_ME_A_SEGUIR` está listado na secção *Untracked files*.
- (e) Publicar no repositório uma versão deste ficheiro. Utilize o comando `git commit`. Ao publicar alterações pela primeira vez tem de configurar o Git indicando-lhe o nome e endereço de correio eletrónico do utilizador que publica as alterações. Esta informação ficará associada a todas as publicações daqui em diante. Executar os dois comandos pedidos, fornecendo a informação requerida.
- (f) Repetir o comando `git commit`, após a configuração do Git. Vai ser aberto um editor de texto para escrever uma mensagem que descreva um sumário da publicação que está a efetuar. Usar mensagens informativas. O editor a usar depende do sistema operativo e da configuração definida. No caso de ser aberto o editor `vi` deve premir a tecla `i` (de *insert*) para iniciar a inserção de texto. Quando terminar deve premir a tecla `esc` seguida de `ZZ` (duas letras `Z` maiúsculas). A publicação termina com sucesso indicando o ramo onde esta se deu (`master`) e identificador do nó, etc.
- (g) Verificar o efeito da publicação utilizando o comando `git status`. Notar o ficheiro `LEIA_ME` deixou de aparecer listado em qualquer secção. Isto significa que a versão que está no repositório é idêntica à que está no diretório de trabalho.
- (h) Utilizar o comando `git log` para confirmar que a alteração foi registada no repositório.
- (i) Efetuar uma alteração ao ficheiro `LEIA_ME`.
- (j) Confirme, recorrendo ao comando `git status`, que, desta vez, o ficheiro `LEIA_ME` aparece na lista da secção *Changes not staged for commit*. Nesta secção aparecem os ficheiros registados no repositório que estão diferentes dos que estão no diretório de trabalho. O ficheiro `LEIA_ME_A_SEGUIR` continua listado na secção *Untracked files* porque não faz parte do repositório.

- (k) Adicionar ambos os ficheiros ao índice, tal como fez anteriormente para o ficheiro `LEIA_ME`.
 - (l) Verificar que ambos os ficheiros estão listados na secção *Changes to be committed*.
 - (m) Publicar as alterações destes dois ficheiros. É uma boa prática usar a opção `-m` no comando `git commit` para especificar de imediato a mensagem (entre aspas) associada à publicação. Por exemplo, `git commit -m "leia-me outra vez"`.
 - (n) Aceder ao *log* do Git para se assegurarem que as alterações foram publicadas. Utilizem a opção `--name-status` para terem acesso a quais os ficheiros que foram incluídos em cada publicação. A letra **A** significa que o ficheiro foi adicionado pela primeira vez ao repositório nesta publicação. A letra **M** significa que o ficheiro foi modificado nesta publicação.
4. Sincronizar o repositório local com o repositório no *BitBucket*
- (a) Utilizar o comando `git push` para empurrar as publicações locais (duas) para o repositório remoto (no *BitBucket*). Notem as mensagens escritas na consola.
 - (b) Seleccionar a secção *Commits* na interface *Web* do *BitBucket* e conferir que as publicações locais foram enviadas para o repositório remoto.
 - (c) Explorar também a secção *Source* e certificar-se de que os ficheiros adicionados e alterados estão com os mesmos conteúdos do que os que estão no diretório de trabalho.
5. Comparar ficheiros do diretório de trabalho com o repositório e com o índice.
- (a) Para comparar todos os ficheiros do diretório de trabalho com o índice ou com as publicações no repositório utiliza-se o comando `git diff`. Executar este comando e certificar-se que não existem alterações neste momento entre os dois ficheiros no diretório de trabalho e o índice/repositório.
 - (b) Fazer uma alteração no ficheiro `LEIA_ME`, incluindo uma nova linha, alterando uma existente e removendo uma existente.
 - (c) Verificar o estado do diretório de trabalho e certificar-se que o ficheiro `LEIA_ME` se encontra listado na secção esperada.

- (d) Utilizar o comando com `git diff` para averiguar as diferenças entre o diretório de trabalho e o índice/repositório. As linhas marcadas com sinais de menos são as linhas removidas, as marcadas com sinais de mais são as adicionadas. Notem que uma linha alterada é assinalado como tendo sido (i) removida a linha antiga e (ii) adicionada a linha nova.
 - (e) Adicionar o ficheiro `LEIA_ME` ao índice para ser incluído na próxima publicação.
 - (f) Averiguar as diferenças. Não há diferenças? Porque o facto de o ficheiro fazer parte do índice é uma indicação de que já está marcado para ser publicado e por isso não faz sentido mostrar que está diferente.
 - (g) Fazer uma alteração ao ficheiro `LEIA_ME_A_SEGUIR`.
 - (h) Verificar as diferenças. Faz sentido?
 - (i) Alterar o ficheiro `LEIA_ME` outra vez. Sim, ainda antes de publicar as alterações anteriores.
 - (j) Verificar as diferenças. Notar que agora há diferenças entre o ficheiro `LEIA_ME` que está no diretório de trabalho e o que no índice é diferente, bem como o que está no índice e o que está (na cabeça) do repositório.
 - (k) Usar o comando `git diff LEIA_ME` com o nome do ficheiro explícito. Assim só mostra as diferenças deste ficheiro.
 - (l) Usar o comando para apurar as diferenças mas desta vez com a opção `--cached`. Assim, verificam a diferença entre a versão no índice e a versão mais recente (na cabeça) do repositório.
 - (m) Usar o comando `git diff` para apurar as diferenças para outra versão que não a mais recente do ficheiro no repositório. Usar o `git log` para descobrir o nome do nó (dado pelo seu *hashcode*). Incluir este nome como parâmetro do comando para verificar as diferenças. (ver figura)
 - (n) Adicionar o ficheiro `LEIA_ME_A_SEGUIR` ao índice e publicar as alterações. Notar que vão ser publicadas as alterações do ficheiro `LEIA_ME` que está no índice e não as que estão no diretório de trabalho.
6. Recuperar versões dos ficheiros e apagar ficheiros. Atenção, estes comandos podem fazer-vos perder informação no diretório de trabalho, porque irão substituir ou apagar os ficheiros do diretório de trabalho.

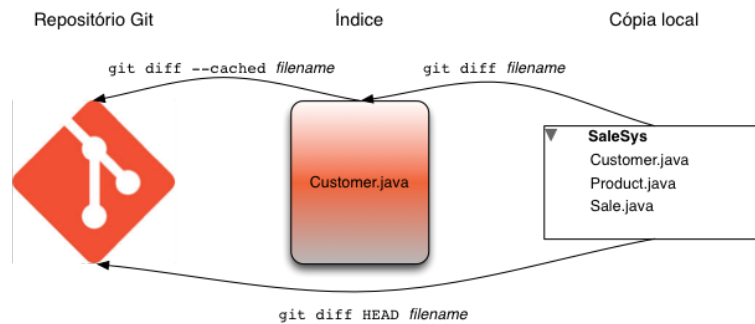


Figura 1: Resumo do comando `git diff`

- (a) Verificar o estado do repositório.
- (b) Substituir o ficheiro `LEIA_ME` no diretório de trabalho pela versão mais recente no repositório. Para tal, devem utilizar o comando `git checkout LEIA_ME`. Dependendo da configuração, o Git irá dar-vos um aviso a indicar que o ficheiro tem alterações locais que não foram publicadas no repositório.
- (c) Apagar o ficheiro do diretório de trabalho e executar o `checkout` de novo. Podem também usar o comando com a opção `-f` que irá ignorar o aviso das alterações locais e que irá substituir o ficheiro localmente.
- (d) Substituir agora o ficheiro `LEIA_ME` pelo ficheiro mais antigo no repositório. Para tal, utilize o comando para verificar o histórico de publicações com a opção que mostre os ficheiros incluídos em cada publicação, determine o nome da publicação e utilize este nome no comando de `checkout`.
- (e) Certificar-se que o conteúdo do ficheiro ficou de acordo com o solicitado.
- (f) Apagar o ficheiro `LEIA_ME_A.SEGUIR` do diretório de trabalho.
- (g) Utilizar o comando `git rm` para eliminar o ficheiro do repositório.
- (h) Publique as alterações.
- (i) Recupere o ficheiro apagado da última versão em que foi publicado no repositório. Uma vez no repositório, no repositório para sempre.

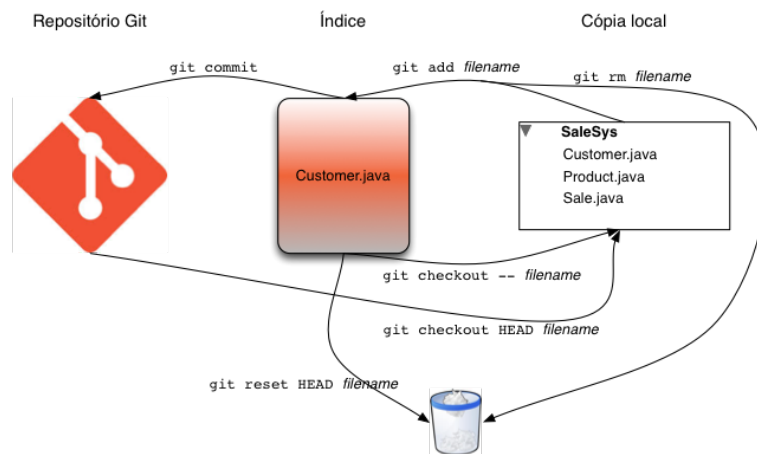


Figura 2: Resumo interação com repositório local