



Curso PHP

Do XLSX ao CMS (aula 5)

**Aonde você estava
na hora H do dia D?**

Versionamento

É um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo. ([Wikipédia](#))

redis-cli.c

```
static sds cliReadLine(int fd) {
```

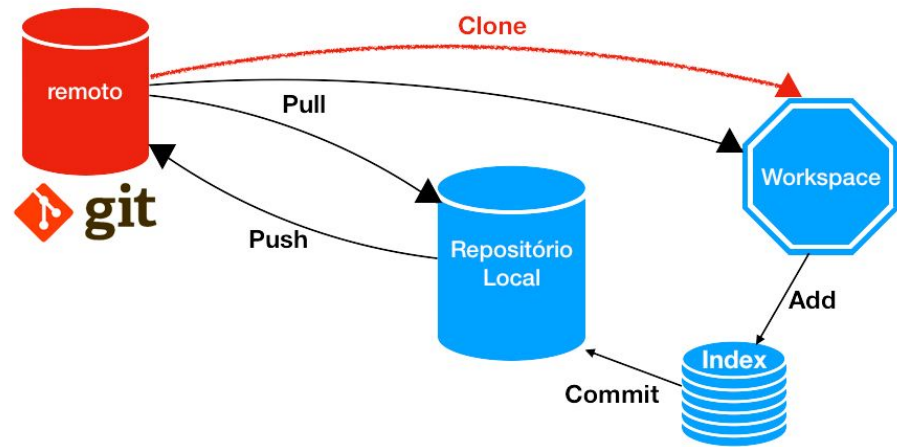
```
135 :135
136 :136     while(1) {
137 :137         char c;
138 :138         ssize_t ret;
139 :
140 :140         if (read(fd,&c,1) == -1) {
141 :141             ret = read(fd,&c,1);
142 :142             if (ret == -1) {
143 :143                 sdsfree(line);
144 :144                 return NULL;
145 :145             } else if (c == '\n') {
146 :146             } else if ((ret == 0) || (c == '\n')) {
147 :147                 break;
148 :148             } else {
149 :149                 line = sdscatlen(line,&c,1);
```

Conceitos

Repositório

Um local central onde todos os arquivos e seu histórico são armazenados.

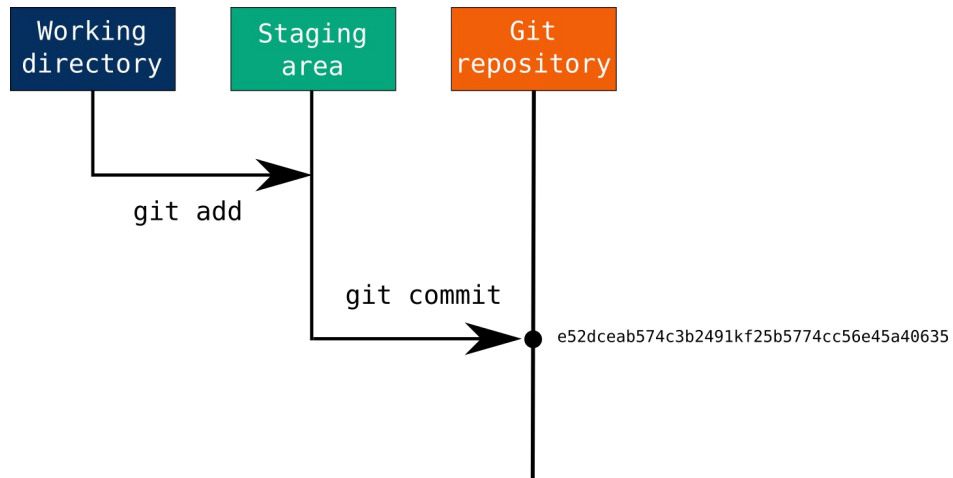
- **Repositório Local:** Uma cópia do repositório em seu próprio computador.
- **Repositório Remoto:** Uma cópia do repositório hospedada em um servidor (como GitHub, GitLab, Bitbucket).



Commit

Um instantâneo de todo o projeto em um determinado momento.

Cada commit possui um identificador único (hash SHA-1).

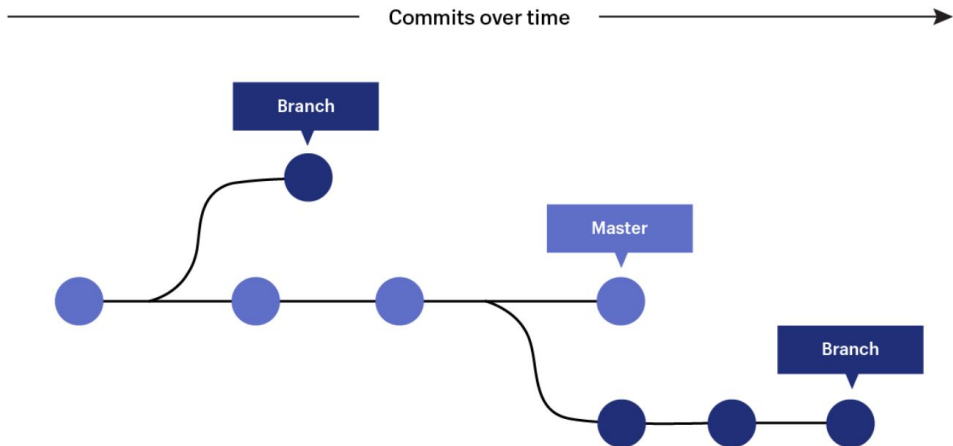




Branch (ramo)

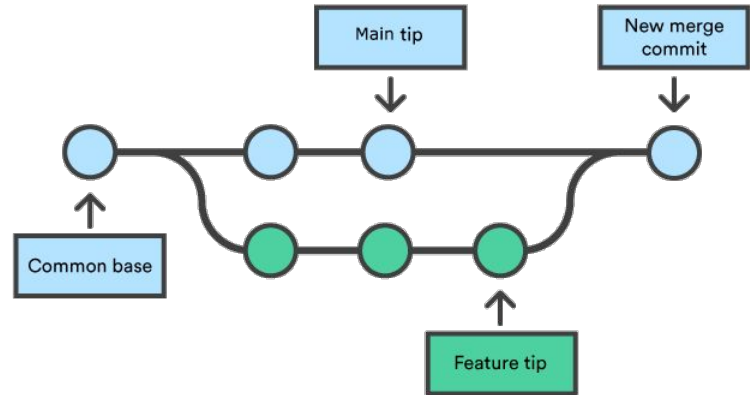
Uma linha de desenvolvimento independente.

Você pode criar branches para trabalhar em novos recursos, experimentar ideias ou corrigir bugs sem afetar o código-base principal.



Merge (mesclar)

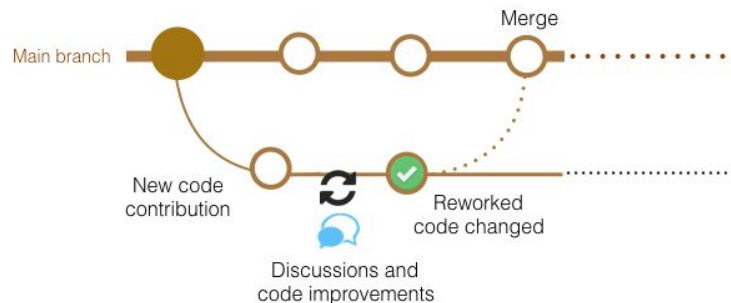
Combinar as alterações de um branch com outro.



Pull Request (PR)

Um mecanismo para propor alterações de um branch para outro (geralmente o branch principal).

Isso permite a revisão de código e discussão antes da mesclagem.



Simplified Pull Request process

Comandos para repositório local

git init

Cria um repositório local

git add ...

Selecciona os arquivos que deseja incluir no próximo commit

git commit

Cria um novo commit

git commit --amend

Para corrigir o último commit

git remote add ...

Adiciona um novo endereço de repositório remoto

git remote -v

Lista os repositórios remotos configurados

git push

Envia seus commits locais para o repositório remoto

git branch ...

Cria um novo branch

git checkout ...

Alternar entre branches

git merge ...

Realiza a mesclagem de branches

git status

Mostra o estado atual do diretório de trabalho e da área de preparação

git log

Visualiza o histórico de commits

git diff

Mostra as diferenças entre arquivos

git reset

Desfaz alterações

git revert

Cria um novo commit que desfaz as alterações de um commit anterior

Comandos para repositório remoto

git clone ...

Cria um repositório local a partir de um repositório (local ou remoto) existente

git fetch

Baixa as informações mais recentes do repositório remoto,
mas não as mescla automaticamente



git pull

Obtém as alterações mais recentes do repositório remoto
e as mescla com o branch local



Problemas comuns



Conflitos de mesclagem

Ocorre quando duas pessoas modificam o mesmo trecho de código simultaneamente, resultando em versões conflitantes.

Para resolver, utilize um editor de texto para revisar as alterações conflitantes e escolher quais partes manter.

Algumas IDEs e ferramentas de linha de comando oferecem interfaces gráficas para ajudar na resolução de conflitos.



Commits desnecessários ou mal feitos

Commits com mensagens vagas, muito grandes ou que incluem arquivos não relacionados podem dificultar a compreensão do histórico de alterações.

Escreva mensagens de commit claras e concisas, descrevendo o objetivo da alteração.

Faça commits atômicos, isto é, cada commit deve representar uma única mudança lógica.

Se precisar corrigir, lembre-se do comando `git commit --amend`.



Problemas com branches

Criar, mesclar e excluir branches incorretamente pode levar a problemas como perda de código ou históricos confusos.

Utilize branches para isolar funcionalidades, evitando trabalhar diretamente na branch principal.

Faça merge com frequência, evitando deixar branches sem merge por muito tempo.

Utilize ferramentas de visualização de branches, pois ferramentas gráficas ajudam a entender melhor a estrutura dos branches.



Problemas com o .gitignore

O arquivo `.gitignore` especifica quais arquivos e diretórios não devem ser rastreados pelo Git. Se configurado incorretamente, pode levar à inclusão de arquivos indesejados ou à exclusão de arquivos importantes.

Teste o arquivo antes de realizar um commit, observando se ele está funcionando como esperado.

Utilize recursos online para gerar o arquivo, além de garantir consistência, o arquivo pode ser gerado com base no tipo de projeto.



Perda de dados

Pode ocorrer devido a operações incorretas como `git reset --hard` ou a exclusão acidental de commits.

Além de backup regulares, mantendo cópias de segurança, utilize o comando `git reflog` para visualizar o histórico de operações.



Falta de conhecimento

A falta de familiaridade com os comandos e conceitos básicos do Git pode levar a erros e dificuldades.

A prática constante e a compreensão dos conceitos fundamentais do Git são essenciais para evitar esses problemas e utilizar o sistema de controle de versão de forma eficiente.

A documentação é uma excelente fonte de informações.

**Utilize o Git em projetos pessoais
para ganhar experiência.**

Ei! Quer teclar?