

Comandos Principais do Git

Curso Hands-On Git | Jupiter Desenvolvimento Informático | SETIC

CONFIGURAÇÃO INICIAL

`git config --global user.name "Nome"` - Define o nome do utilizador
`git config --global user.email "email"` - Define o email do utilizador
`git config --list` - Lista todas as configurações

CRIAR E CLONAR REPOSITÓRIOS

`git init` - Inicializa um novo repositório Git
`git clone <url>` - Clona um repositório remoto

BÁSICOS - STAGING E COMMITS

`git status` - Mostra o estado dos ficheiros
`git add <ficheiro>` - Adiciona ficheiro ao staging
`git add .` - Adiciona todos os ficheiros ao staging
`git commit -m "mensagem"` - Cria um commit com mensagem
`git diff` - Mostra alterações não staged
`git log` - Mostra histórico de commits
`git log --oneline` - Histórico resumido (uma linha por commit)

BRANCHES

`git branch` - Lista todos os branches locais
`git branch <nome>` - Cria um novo branch
`git checkout <branch>` - Muda para o branch indicado
`git checkout -b <nome>` - Cria e muda para novo branch
`git switch <branch>` - Muda para o branch indicado (moderno)
`git merge <branch>` - Integra branch no branch actual
`git branch -d <nome>` - Elimina branch (após merge)

REPOSITÓRIOS REMOTOS

`git remote add origin <url>` - Adiciona repositório remoto
`git remote -v` - Lista repositórios remotos configurados
`git push -u origin main` - Envia commits para o remoto (1ª vez)
`git push` - Envia commits para o remoto
`git pull` - Obtém e integra alterações do remoto
`git fetch` - Obtém alterações sem integrar

TÉCNICAS AVANÇADAS

`git stash` - Guarda alterações temporariamente
`git stash pop` - Recupera alterações do stash
`git stash list` - Lista todos os stashes
`git rebase <branch>` - Reaplica commits sobre outro branch
`git cherry-pick <hash>` - Aplica commit específico noutro branch

DESFAZER ALTERAÇÕES

`git checkout -- <ficheiro>` - Descarta alterações não staged
`git restore <ficheiro>` - Descarta alterações (moderno)
`git reset HEAD <ficheiro>` - Remove ficheiro do staging
`git reset --soft HEAD~1` - Desfaz último commit (mantém ficheiros)
`git reset --hard HEAD~1` - Desfaz último commit (elimina alterações)
`git revert <hash>` - Cria commit que desfaz alterações