CURSO HANDS-ON

GIT

Do Básico ao Avançado

4 dias | 3 horas por dia | Total: 12 horas

Modalidade: Prático (Hands-On)

Nível: Iniciante a Intermédio

Realizado por:

Jupiter Desenvolvimento Informático

Para:

SETIC - Serviço de Tecnologias de Informação e Comunicação das Finanças Públicas

Instrutor:

Gustavo Caraciolo - Líder Técnico

+244 222 679 500
+244 972 580 299

contacto@jupiter.co.ao

Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

Sumário do Curso

Este curso prático de Git foi desenhado para profissionais que desejam dominar o controlo de versões de forma eficiente. Através de exercícios práticos e cenários reais, os participantes irão desenvolver competências essenciais para trabalhar em equipa e gerir projectos de software.

Objectivos de Aprendizagem

- Compreender os fundamentos do controlo de versões distribuído
- Dominar os comandos essenciais do Git para o dia-a-dia
- Trabalhar com branches e estratégias de branching
- Resolver conflitos de merge de forma eficiente
- Colaborar em projectos usando GitHub/GitLab
- Aplicar boas práticas e workflows profissionais

Pré-requisitos

- Conhecimentos básicos de linha de comandos (terminal/cmd)
- Computador com Git instalado
- Conta no GitHub ou GitLab
- Editor de código (VS Code recomendado)

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

Dia 1: Fundamentos do Git

O que é Controlo de Versões?

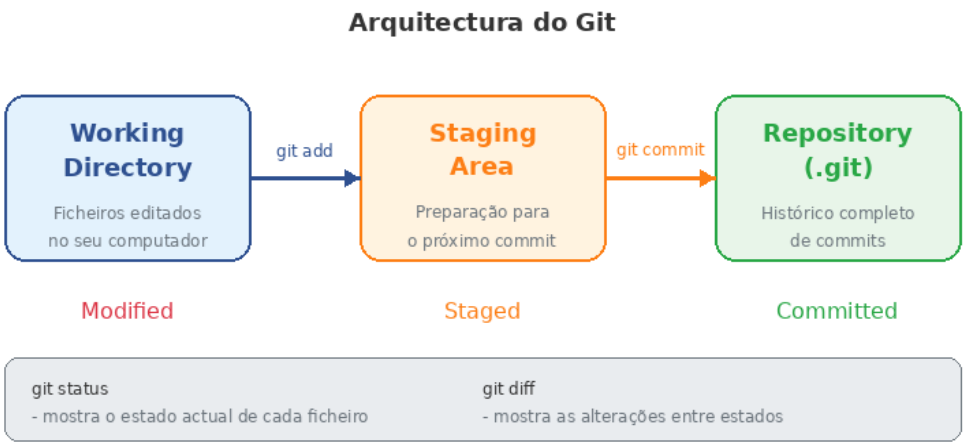
O controlo de versões é um sistema que regista as alterações feitas a ficheiros ao longo do tempo, permitindo recuperar versões específicas posteriormente. É uma ferramenta fundamental no desenvolvimento de software moderno.

Porquê Git?

O Git foi criado em 2005 por Linus Torvalds, o criador do Linux. Ao contrário de sistemas centralizados como SVN, o Git é distribuído - cada programador tem uma cópia completa do repositório, incluindo todo o histórico.

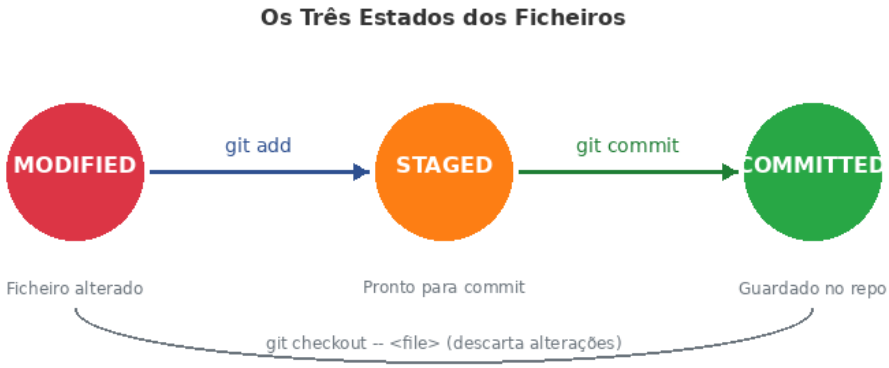
Arquitectura do Git

Para compreender o Git, é essencial entender os três estados principais:



Os Três Estados dos Ficheiros

Um ficheiro no Git pode estar em três estados: modified, staged ou committed.



Prática: Configuração e Primeiro Repositório

Exercício 1: Configuração Inicial

```
git config --global user.name "Seu Nome"
git config --global user.email "seu@email.com"
```

Exercício 2: Criar Primeiro Repositório

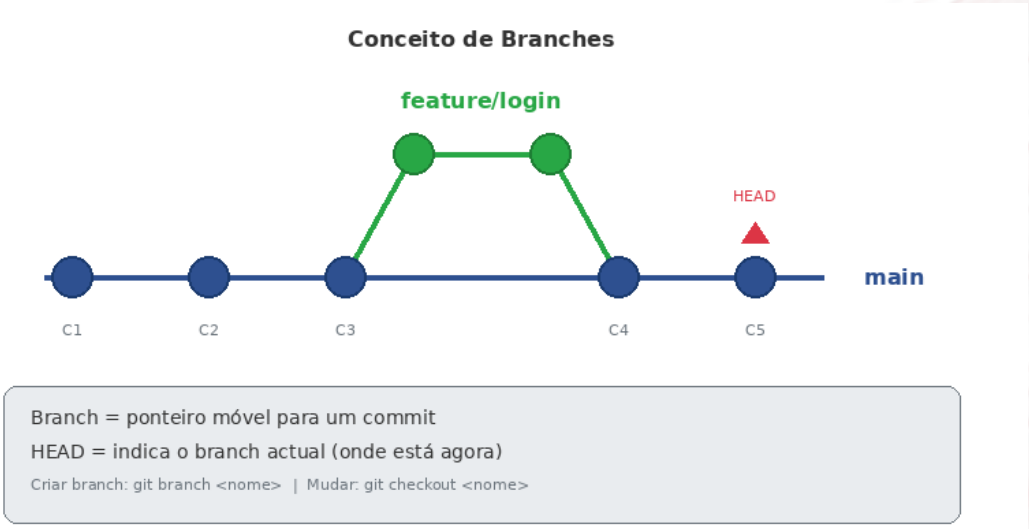
1. Criar pasta e inicializar com git init
2. Criar ficheiros README.md e index.html
3. Adicionar ao staging com git add .
4. Fazer commit com git commit -m "Commit inicial"

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

Dia 2: Branches, Merging e Repositórios Remotos

O Conceito de Branches

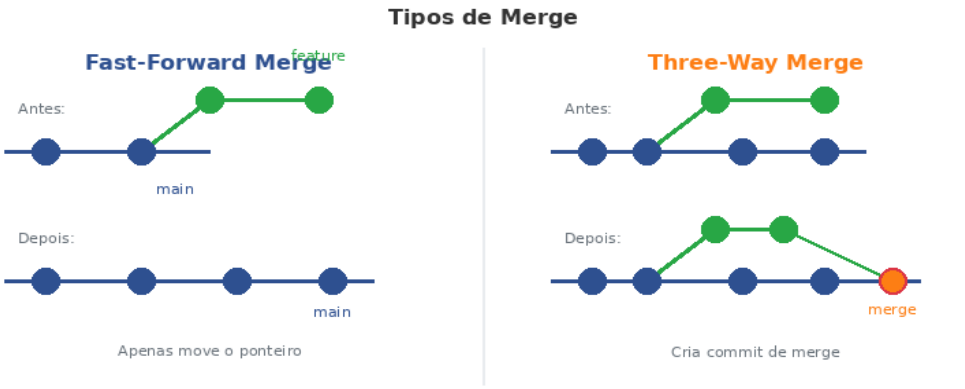
Os branches são uma das funcionalidades mais poderosas do Git. Um branch é essencialmente um ponteiro móvel para um commit.



Tipos de Merge

Quando integra um branch noutro, o Git pode fazer dois tipos de merge:

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola



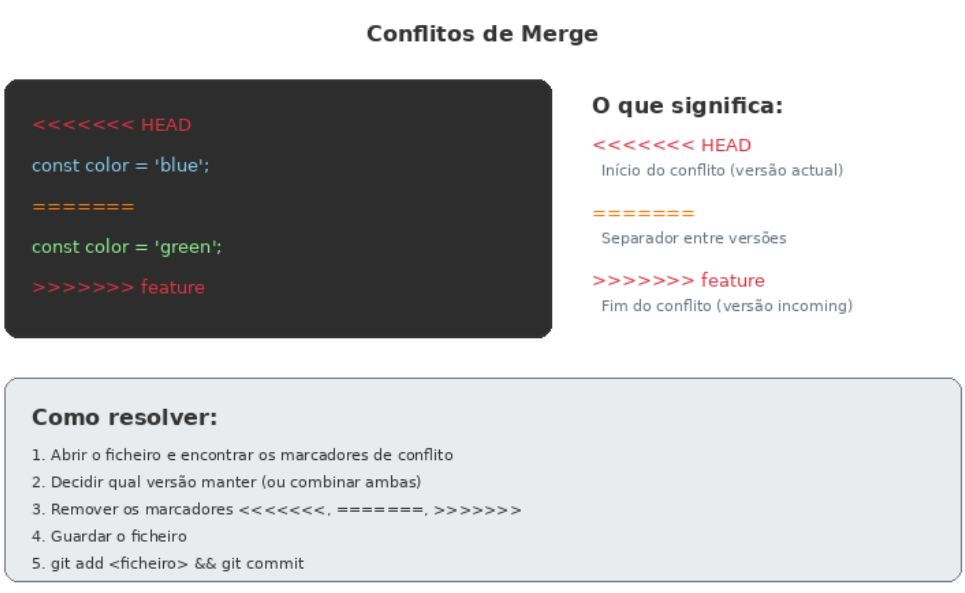
Fast-Forward:
Acontece quando main não tem commits novos

Three-Way:
Acontece quando ambos os branches têm commits novos

Forçar merge commit: git merge --no-ff feature

Conflitos de Merge

Um conflito ocorre quando o Git não consegue determinar como combinar alterações.



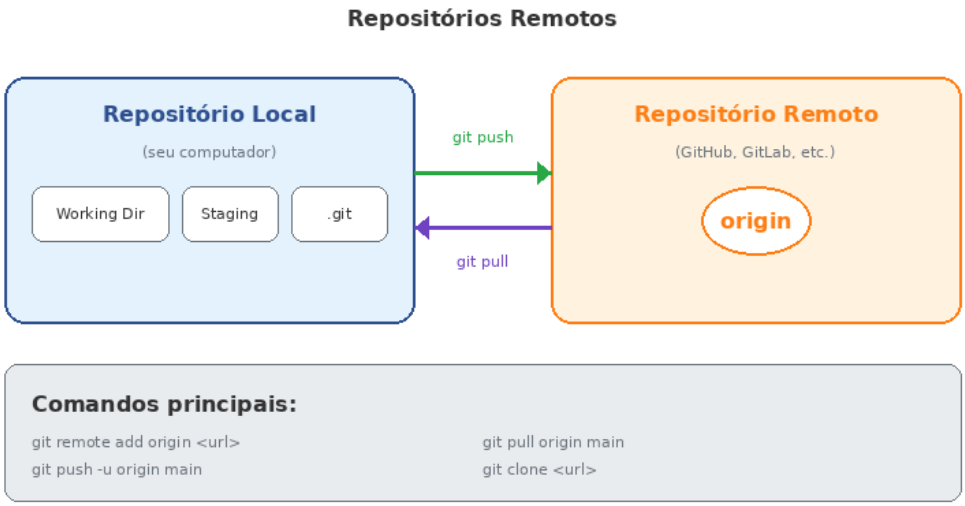
Como resolver:

1. Abrir o ficheiro e encontrar os marcadores de conflito
2. Decidir qual versão manter (ou combinar ambas)
3. Remover os marcadores <<<<<<<, =======, >>>>>>>
4. Guardar o ficheiro
5. git add <ficheiro> && git commit

Repositórios Remotos

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

Um repositório remoto é uma versão do projecto alojada na Internet.



Prática: Branches e Remotos

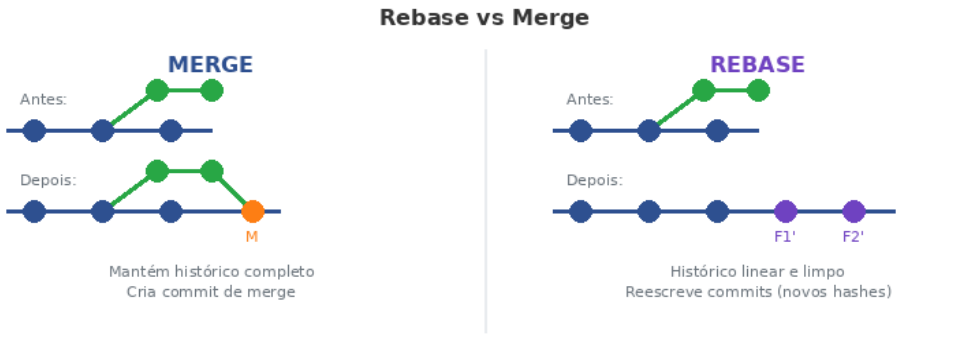
Exercício 1: Branches

1. Criar branch: git branch feature/login
2. Mudar: git checkout feature/login
3. Fazer alterações e commits
4. Merge: git checkout main && git merge feature/login

Dia 3: Técnicas Avançadas

Rebase vs Merge

O rebase é uma alternativa ao merge para integrar alterações, criando um histórico linear.



⚠ **REGRA DE OURO:**

Nunca faça rebase de commits que já foram partilhados (pushed)!

Use rebase apenas para limpar commits locais antes de partilhar.

Merge: `git checkout main && git merge feature`

Rebase: `git checkout feature && git rebase main`

Stash: Trabalho Temporário

O stash permite guardar alterações temporariamente sem fazer commit.

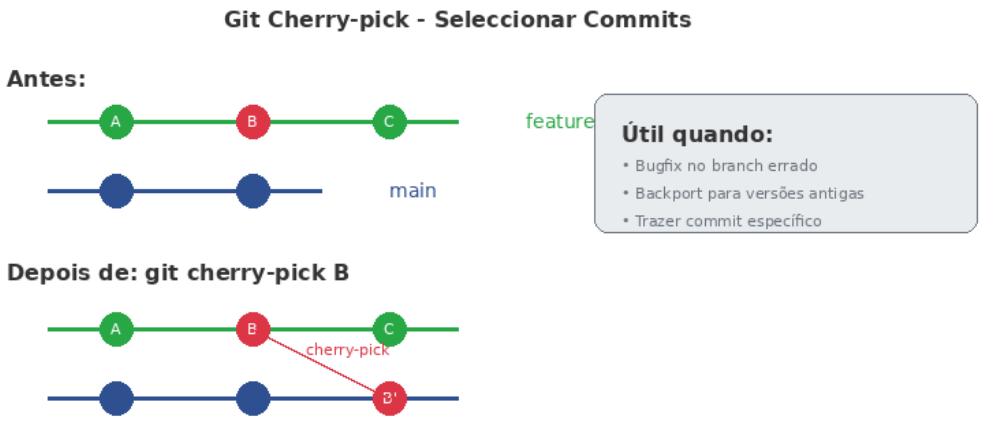
Comandos:

`git stash`	Guarda alterações e limpa working directory
`git stash list`	Lista todos os stashes
`git stash pop`	Recupera o último stash e remove da lista
`git stash apply`	Recupera o stash mas mantém na lista
`git stash drop`	Remove um stash da lista

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

Cherry-pick: Seleccionar Commits

O cherry-pick permite aplicar um commit específico de um branch noutro.



Prática: Técnicas Avançadas

Exercício 1: Rebase

1. Criar branch feature com commits
2. No branch feature: git rebase main

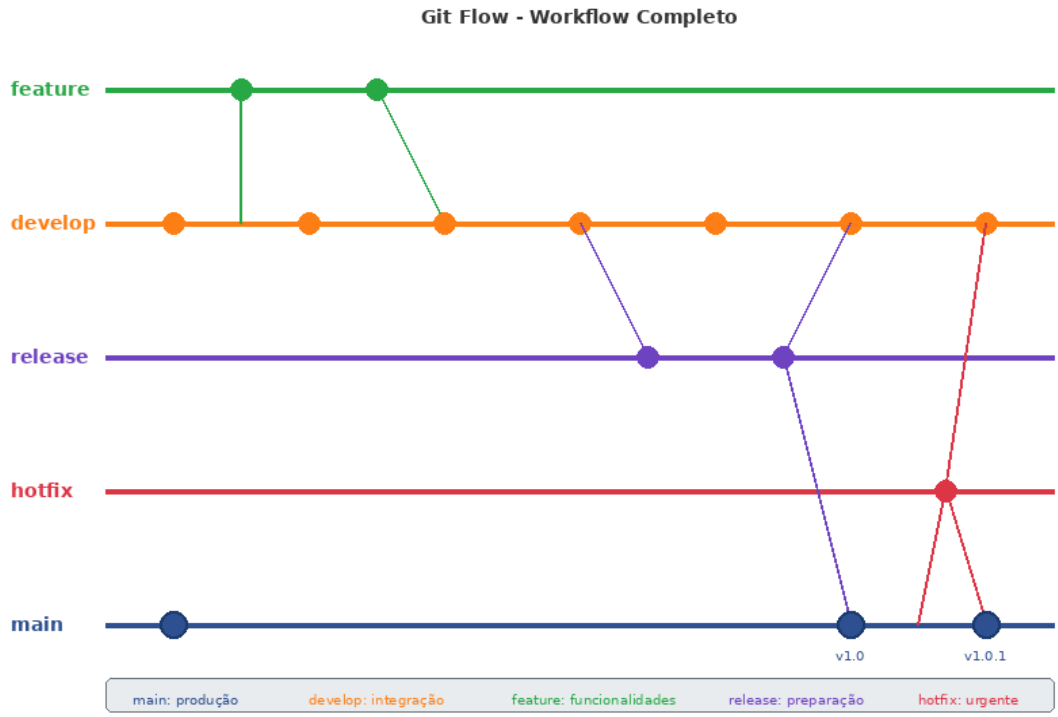
Exercício 2: Stash

1. Fazer alterações sem commit
2. git stash
3. Mudar de branch
4. git stash pop

Dia 4: Workflows e Boas Práticas

Git Flow

O Git Flow define papéis específicos para diferentes branches e como devem interagir.



Boas Práticas

- Commits atómicos: uma alteração lógica por commit
- Mensagens descritivas em modo imperativo
- Pull antes de push
- Nunca reescrever histórico público
- Code review sistemático

Prática: Workflow Completo

Exercício 1: Git Flow

1. Criar estrutura: main e develop
2. Feature branch, desenvolver, merge

+244 222 679 500
+244 946 990 459
contacto@jupiter.co.ao
Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola

3. Release branch e merge em main

Projecto Final

Em equipas de 3-4 pessoas, desenvolver mini-projecto aplicando todos os conceitos aprendidos.

Resumo do Curso

Dia 1 - Fundamentos: init, add, commit, status, log, diff

Dia 2 - Branches e Remotos: branch, checkout, merge, push, pull

Dia 3 - Técnicas Avançadas: rebase, stash, cherry-pick

Dia 4 - Workflows: Git Flow, Boas Práticas, Projecto Final

Recursos Adicionais

- Documentação oficial: https://git-scm.com/doc
- Pro Git Book: https://git-scm.com/book
- Learn Git Branching: https://learngitbranching.js.org

Certificação

Ao completar todas as actividades práticas e o projecto final, o participante receberá um certificado de conclusão do Curso Hands-On de Git.

+244 222 679 500
+244 946 990 459

contacto@jupiter.co.ao

Cond. das Mangueirinhas
Rua dos Generais, s/n
Morro Bento, Luanda
Angola