



Introdução ao Git e Github



Instrutores:

Anna Giulia Gomes Miranda

Mário Roque Conceição dos Anjos Junior

O que é Git?

- **História**

- Criado em 2005 por Linus Torvalds em conjunto com a comunidade como uma alternativa a um software pago
- Inicialmente utilizado para alterações dentro do desenvolvimento do linux

- **Controle de versão**

- **Utilização**

- **Projetos**
 - Programação - C, Python, R, C++, etc.
- **Alterações em documentos**
 - TCC, arquivos em LateX, etc.

Por que fazer controle de versão?

- **Organiza e registra todas as alterações do projeto**
- **Permite voltar para versões anteriores (evitar perdas)**
- **Facilita corrigir erros rapidamente**
- **Ajuda no trabalho em equipe (vários contribuidores)**
- **Torna decisões reversíveis**
- **Facilita o compartilhamento e distribuição do projeto**

Instalação
+
Configurações

- **Materiais e aula**

- donthpad.com/minicurso_git_github

- **Instalação**

- **Windows**

- <https://git-scm.com/downloads>

- **Linux**

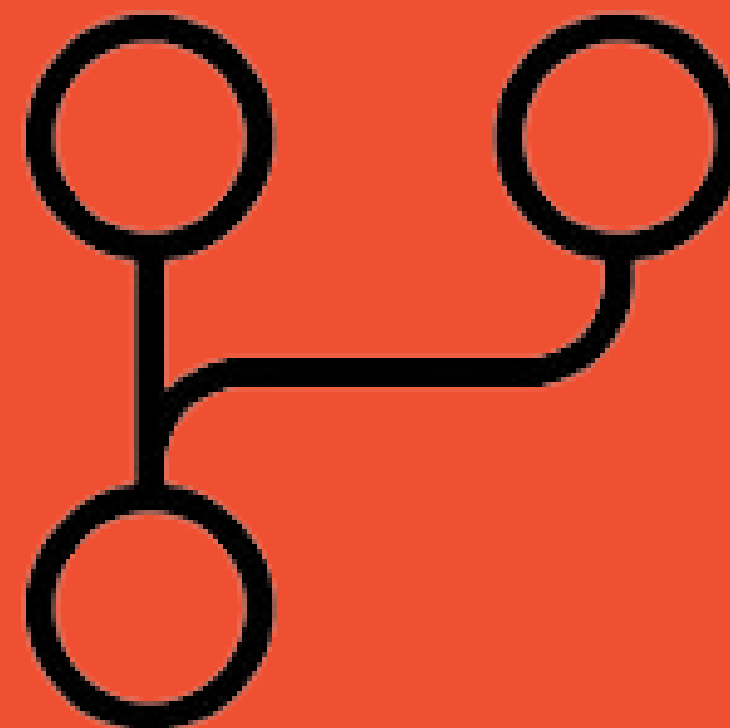
- **Debian/Ubuntu: sudo apt install git**
 - **Arch-based: sudo pacman -S git**
 - **Fedora: sudo dnf install git**

- **MacOS**

- **/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"**
 - **brew install git**



Git



Conceitos Importantes

- **Repositório**
- **Staging Area**
- **Commit**
- **Histórico**
- **Branch**

Comandos bash:

- **Comandos de navegação**

- **ls** → Lista todos os arquivos e diretórios do local atual
- **pwd** → Mostra em que diretório estou
- **cd <pasta/diretório>** → Transitar entre um diretório e outro

- **Comandos de visualização, edição e criação pastas**

- **mkdir <novo_diretorio>** → Cria um diretório
- **cat <nome_arquivo>** → Mostra tudo que tem dentro de um arquivo
- **touch <novo_arquivo>** → Cria um arquivo vazio
- **nano <novo_arquivo>** → Abre um editor de texto para criar ou editar um arquivo

Repositório

- **Diretório onde está hospedado o seu projeto e onde o git está operando**
- **Primeiros comandos:**
 - **git init .** → **Cria um repositório git dentro do diretório**
 - **git config --global user.name “seu nome”** → **Configura seu nome para o commit**
 - **git config --global user.email “seu.email@email.com”** → **Configura seu email para o commit**
 - **git status** → **Checa em que estado está o arquivo além do que possa ter mudado**
- **Documentação**
 - **git --help nome_comando** → **Documentação explicativa do comando solicitado**
 - **git nome_comando --help** → **Mesma função do comando anterior**

Staging Area

- **Área onde as alterações estão prontas para ocorrerem, também conhecido como 'estágio de preparação pré-commit'**
- **Comandos:**
 - **git add → Adiciona um arquivo para a staging area**
 - **git add nome_arquivo → Adiciona um arquivos específico**
 - **git add . → Adiciona todos os arquivos do diretório**

Commit

- **Registro e assinatura de uma das versões do projeto**
- **Comandos:**
 - **git commit -m “Mensagem Explicativa de Commit” → Salva as versões feitas dentro desse repositório, assinando-as com o nome e email de quem as submeteu**

Histórico

- **Visualização dos registros de alterações entre os commits**
- **Comandos:**
 - **git log** → **Retorna o histórico de commits com todas as alterações feitas**
 - **git restore** → **Restaura versões anteriores**
 - **git restore nome_arquivo** → **Restaura um arquivo modificado antes de ser adicionado**
 - **git restore --staged nome_arquivo** → **Restaura um arquivo adicionado**
 - **git restore --source=<endereço_hash> nome_arquivo** → **Restaura um arquivo já commitado**

EXERCÍCIO

Crie um repositório local e salve suas primeiras versões de um arquivo

Branches

- **Copia do repositório criada para não fazer alterações diretamente no projeto principal**
- **Comandos:**
 - **git branch**
 - **git branch → Checa quantas**
 - **git branch nova_branch → Cria uma nova branch, mas permanece na original**
 - **git branch -d nome_branch → Deleta uma branch**
 - **git switch**
 - **git switch nova_branch → Troca de uma branch para outra**
 - **git switch -c nova_branch → Cria uma nova branch e te move direto pra ela**

Checkout

- **git checkout - “Faz tudo” ou “Canivete Suíço”**
 - **Comando antigo que acumulava muitas funções:**
 - **Trocar de branch**
 - **git checkout nome_branch**
 - **Criar uma nova branch e se mover direto para ela**
 - **git checkout -b nova_branch**
 - **Restaurar arquivos modificados antes de serem adicionados**
 - **git checkout --nome_arquivo**
 - **Restaurar um arquivo já commitado**
 - **git checkout <hash> nome_arquivo**

Checkout

X

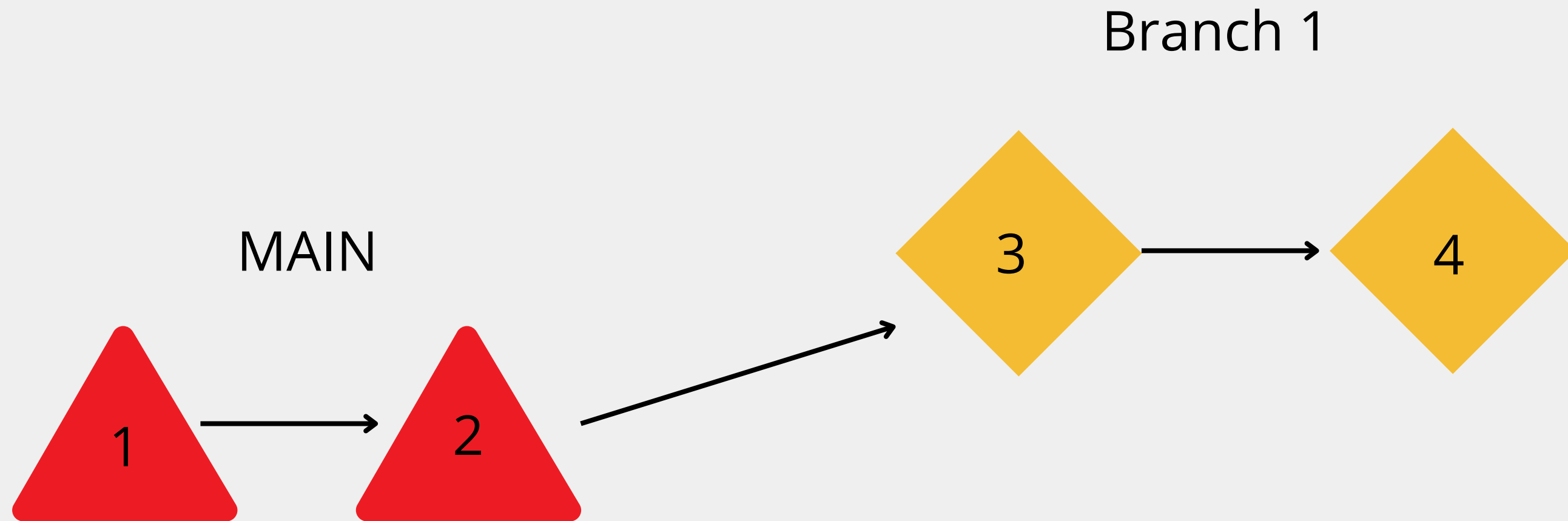
Switch + Restore

Checkout **vs** Switch

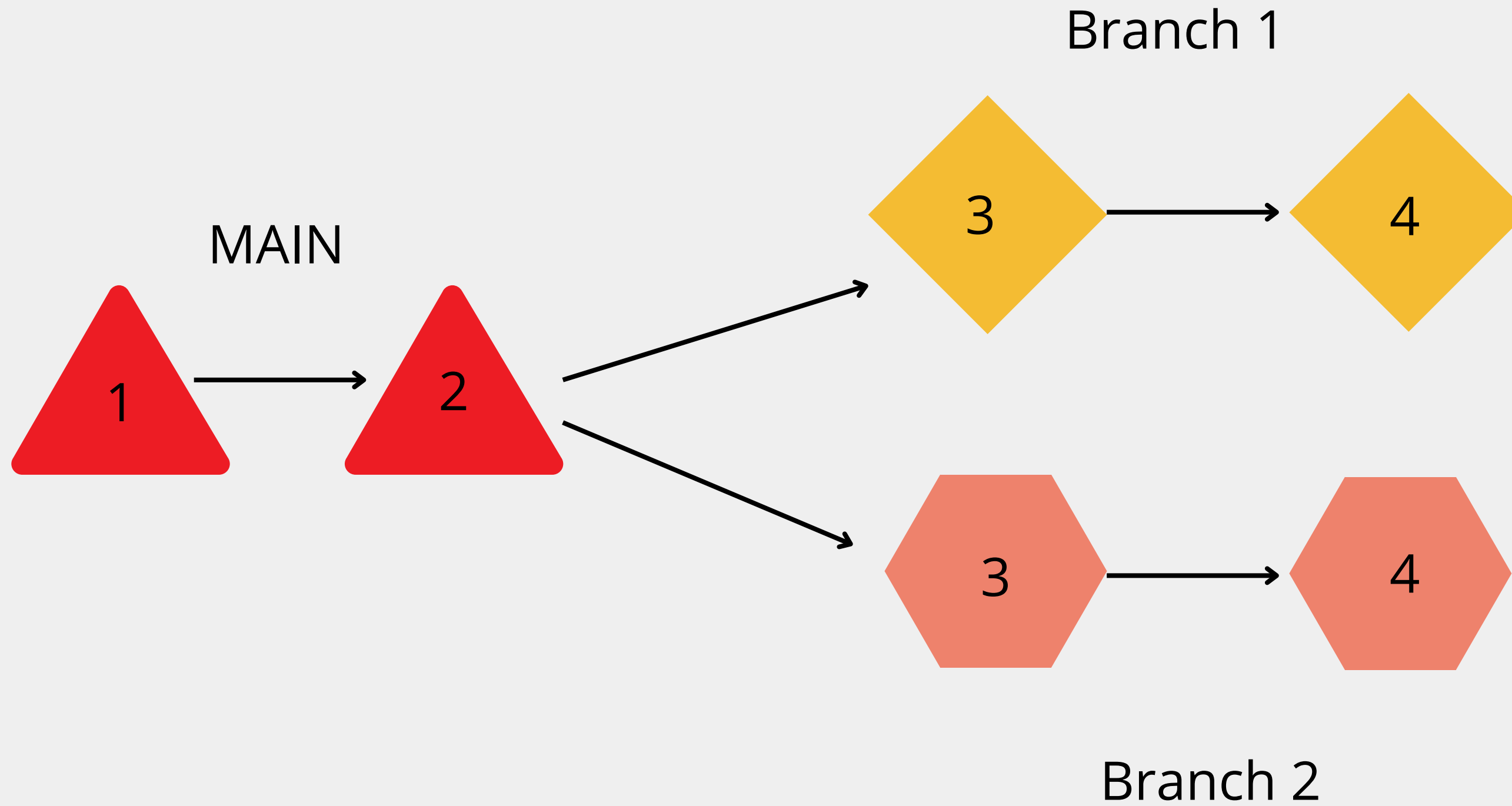
Antigo	Modernos
<ul style="list-style-type: none">○ git checkout<ul style="list-style-type: none">▪ git checkout nome_branch▪ git checkout -b nova_branch	<ul style="list-style-type: none">○ git switch<ul style="list-style-type: none">▪ git switch nome_branch▪ git switch -c nova_branch
<ul style="list-style-type: none">○ git checkout<ul style="list-style-type: none">▪ git checkout -- nome_arquivo▪ git checkout <hash> nome_arquivo	<ul style="list-style-type: none">○ git restore<ul style="list-style-type: none">▪ git restore nome_arquivo▪ git restore --source=<hash> nome_arquivo

Workflow

Branches



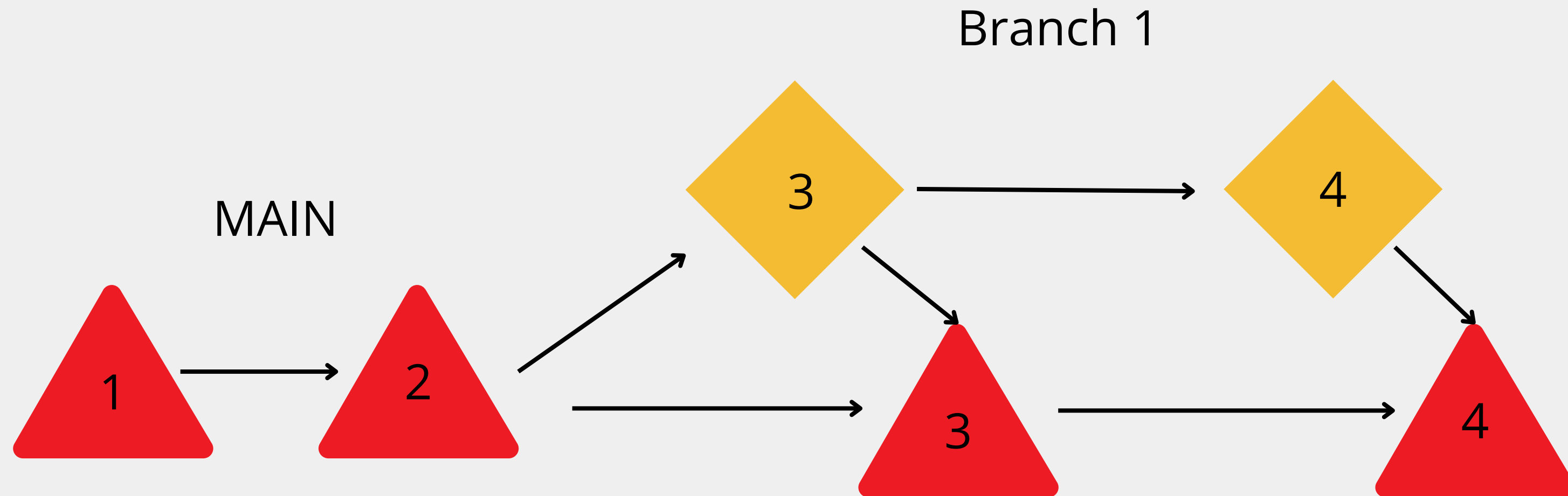
Branches



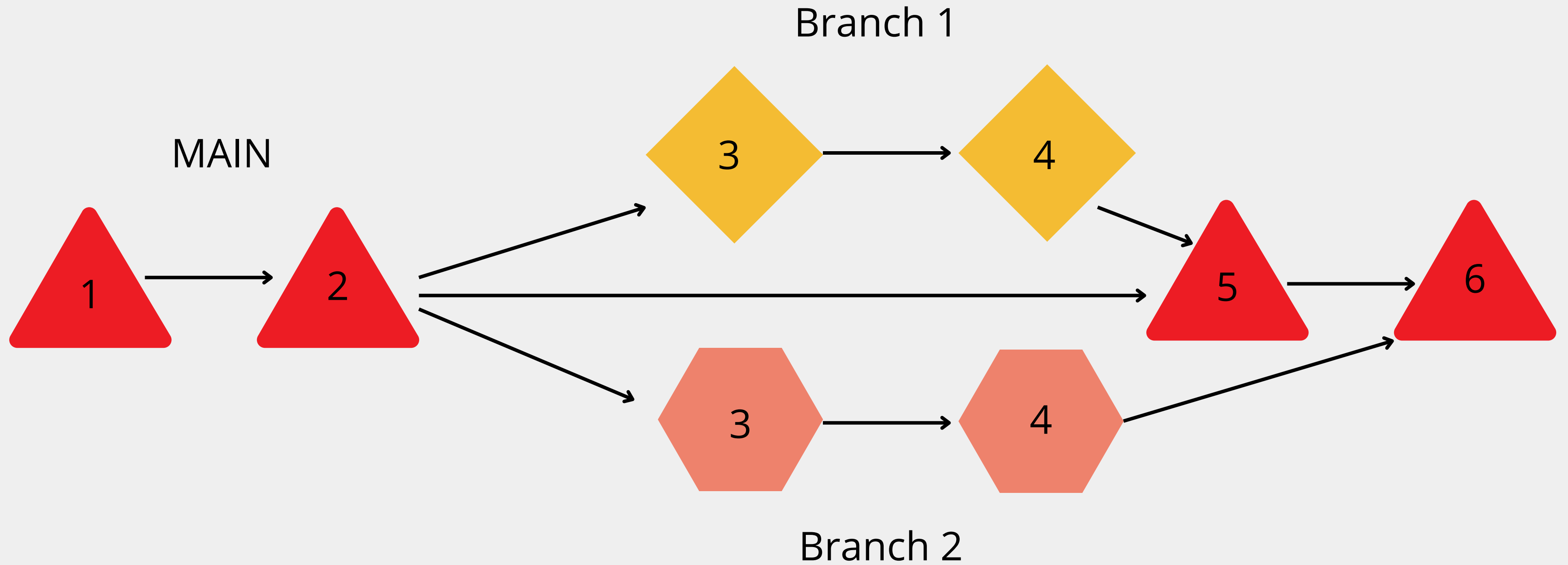
Merge

- **Mesclar as alterações de uma branch secundária para a branch principal**
- **Comandos:**
 - **git merge nome_branch → Mescla as alterações de uma branch para uma outra branch**

Branches

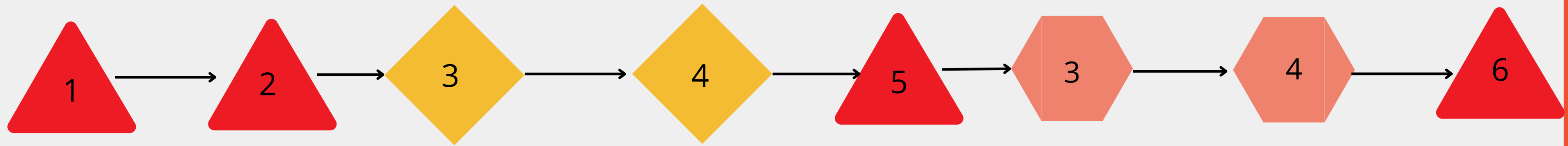


Branches



Branches

MAIN



EXERCÍCIO

Crie uma branch

Faça um commit e depois um merge para a branch principal



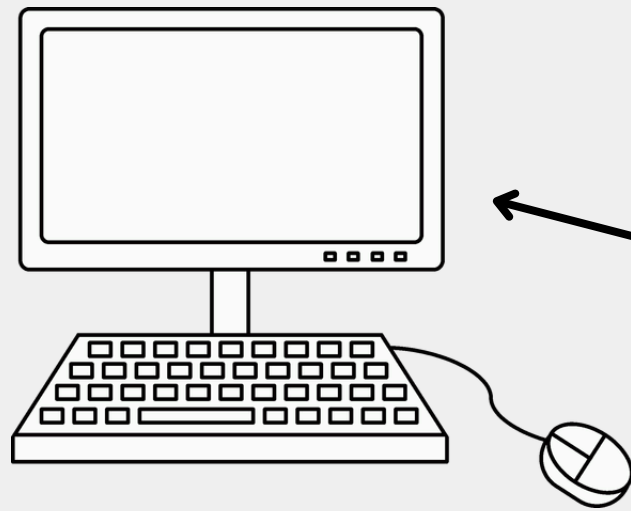
GitHub



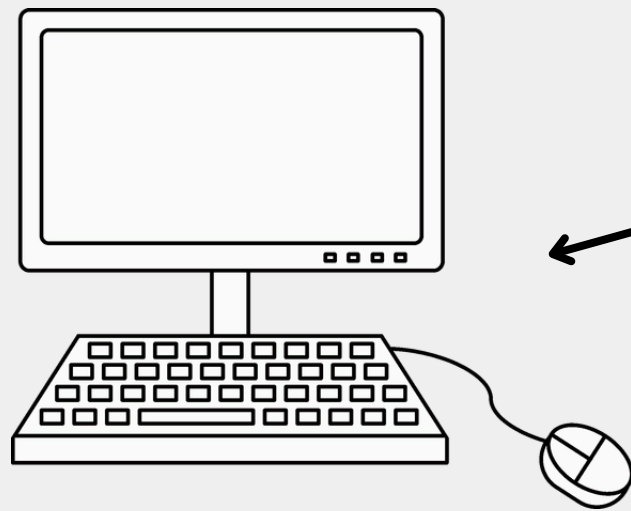
O que é Github?

- **Plataforma de hospedagem de código**
- **Permite armazenar e compartilhar projetos**
- **Facilita o trabalho em equipe**

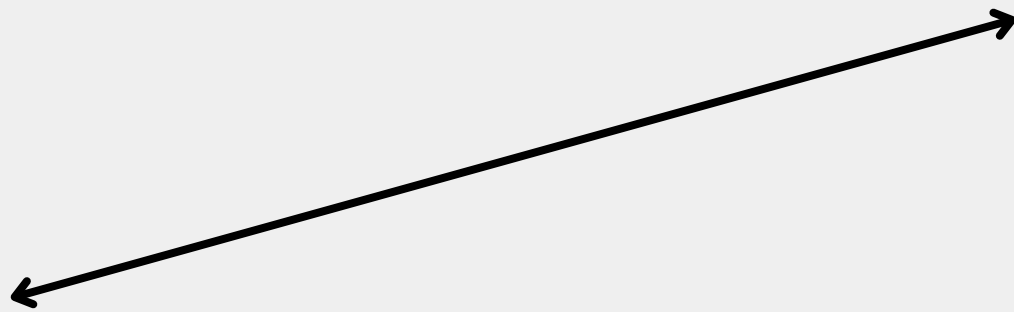
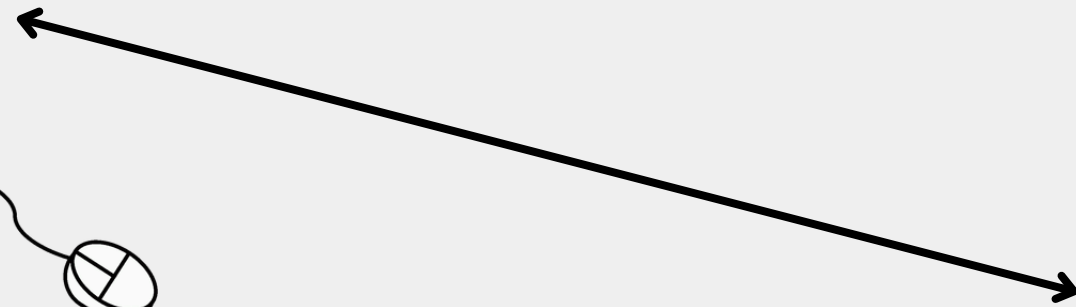
LOCAL



LOCAL



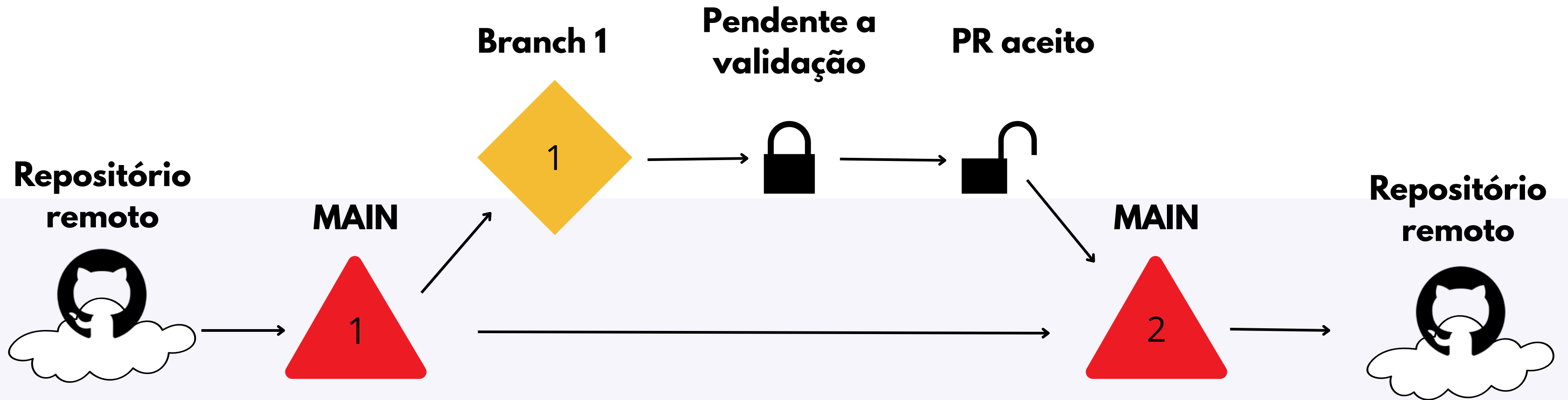
REMOTO

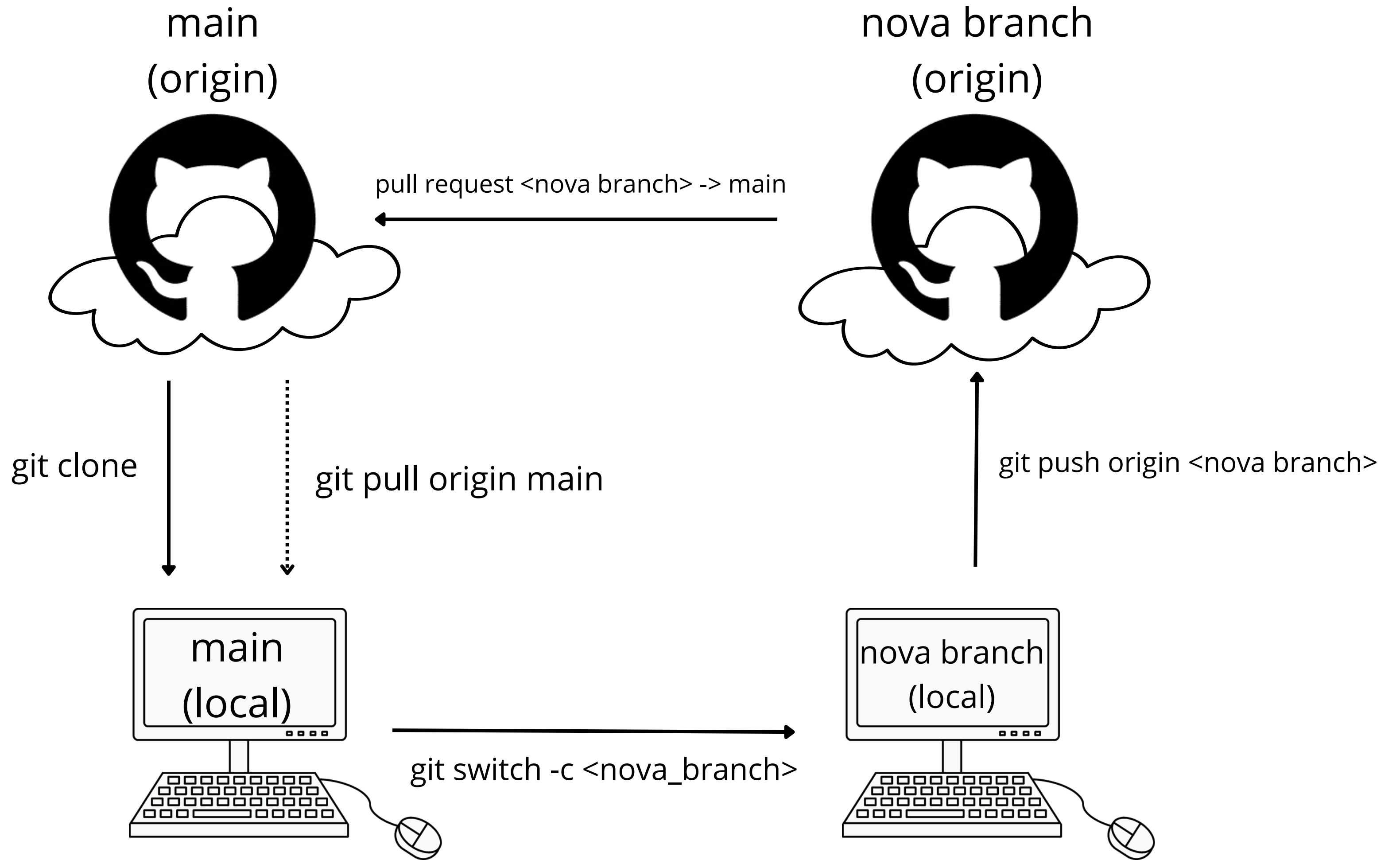


Pull Request

- **Pedido de análise e aceitação de alterações numa branch secundária para a versão principal do projeto num ambiente remoto**
- **Comandos:**
 - **git clone** → **Cria uma cópia local de um repositório que está hospedado remotamente**
 - **git remote add** → **Vincula o seu repositório local a um endereço de repositório remoto**
 - **git push** → **Envia as suas alterações confirmadas (commits) do repositório local para o servidor remoto**
 - **git pull** → **Busca as novidades do repositório remoto e as mescla automaticamente na sua branch atual**

Pull Request





EXERCÍCIO

Subir repositório local para o GitHub

Fazer push e pull das alterações.

Fork

Fork

O QUE É UM FORK?

- **É uma cópia exata de um repositório de outra pessoa para a sua conta do GitHub.**
- **Criação de um ambiente seguro para modificações**
- **Diferença principal: O clone copia o projeto para o seu computador (local). O fork copia o projeto de um servidor para outro servidor (entre contas do GitHub)**

Fork

POR QUE FAZER UM FORK?

- **Open Source (Código Aberto):** É a forma oficial de contribuir para projetos públicos onde você não tem permissão de edição direta
- **Ponto de partida:** Usar o projeto ou template de alguém como base para criar o seu próprio projeto independente.
- **Ambiente Seguro (Sandbox):** Experimentar e quebrar o código à vontade, com a garantia de que o projeto original ficará intacto.

Fork

FLUXO

- **Fork:** Você clica no botão "Fork" no GitHub original.
- **Clone:** Você baixa o seu Fork para o seu computador (git clone).
- **Modificação:** Você cria uma branch, faz as alterações e commita (git commit).
- **Push:** Você envia as alterações para o seu Fork no GitHub (git push).
- **Pull Request (PR):** Você abre um pedido para o dono do repositório original avaliar e, se aprovar, unir o seu código ao dele.

Projeto Final

EXERCÍCIO

Façam um fork do repositório de exemplo

Crie uma branch com seu nome

Submeta um Pull Request para o repositório principal

MATERIAL COMPLEMENTAR

- **Documentação:** git-scm.com/docs/git/pt_BR
- **Curso de Git e Github:** https://youtube.com/playlist?list=PLvIkVRRKOYFQyKmdrassLNxkzSMM6tcSL&si=-ezoQC06vzTp_jdD
- **Livro:** git-scm.com/book/pt-br/v2
- **Site:** githowto.com/pt-BR

**Obrigado por ter
participado**