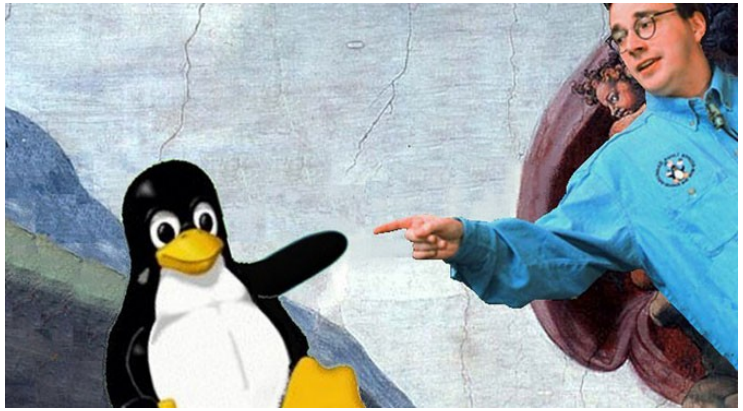
Introdução ao Git

Usando o GitHub

O que é o GIT

- Uma ferramenta de versionamento de código
- Pode ser usada via terminal ou via interface gráfica
- Compatível com Linux, Mac e Windows
- Criar e gerenciar repositórios
- Criado por Linus Torvalds



O GitHub

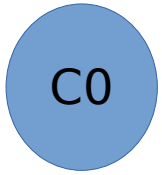
- **Um repositório online para hospedar projetos**
- **Uma ‘rede social’**
- **Plataforma mais importante para qualquer desenvolvedor**
  - Aprender;
  - Compartilhar;
  - Portfólios profissionais;
- **Problema: em inglês...**
- **github.com → crie sua conta já!**

A ideia principal

- **Problema:**
  - Códigos em constante alteração
  - Gerenciar arquivos por nomes?
  - Por cópias?
  - Uma cópia local e outra remota?
- **Solução**
  - Ferramentas de versionamento
  - Uso de estados.
  - Alterações salvas, não cada cópia.
  - **Commit !?**

Commit

- **Conjunto de alterações feitas em um determinado momento**
  - Início do programa

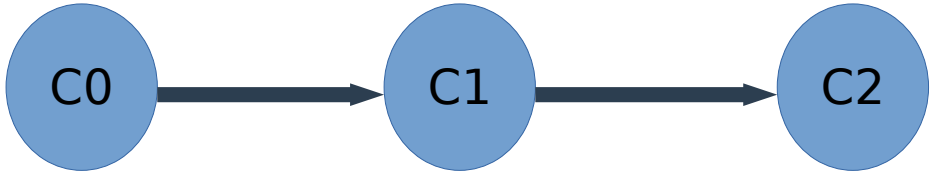


Commit

- **Conjunto de alterações feitas em um determinado momento**
  - Criou um método novo.

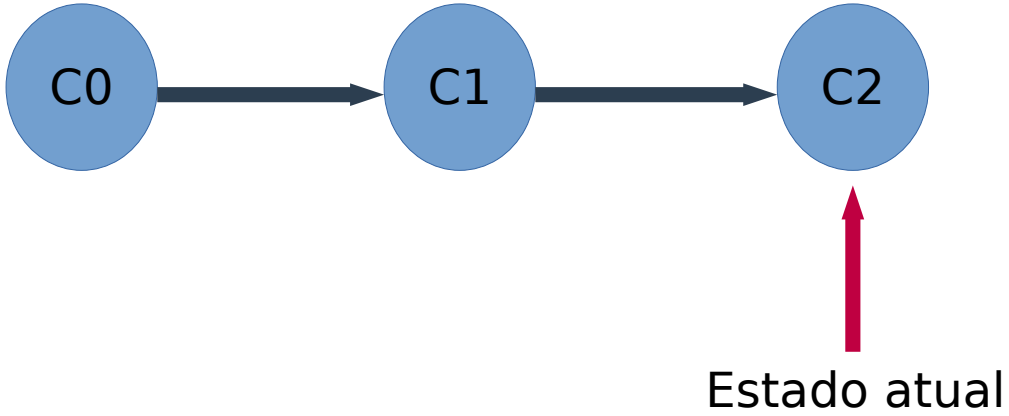
Commit

- **Conjunto de alterações feitas em um determinado momento**
  - Criou uma classe nova.



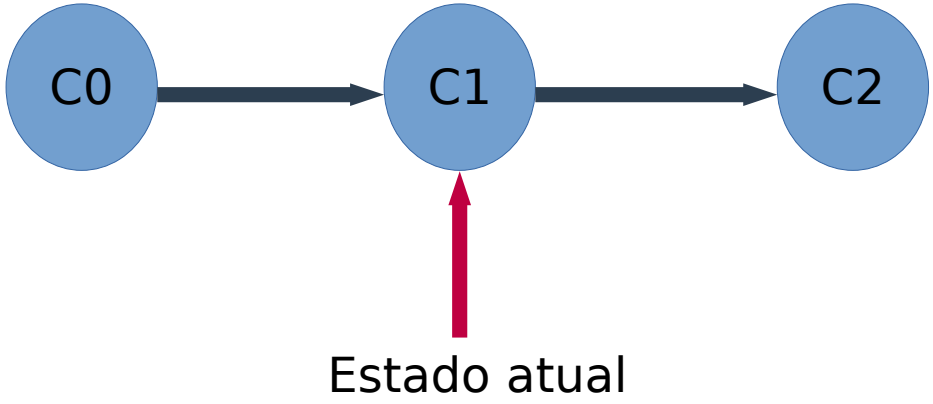
Commit

- **Conjunto de alterações feitas em um determinado momento**
  - Estado atual do projeto está em C3



Commit

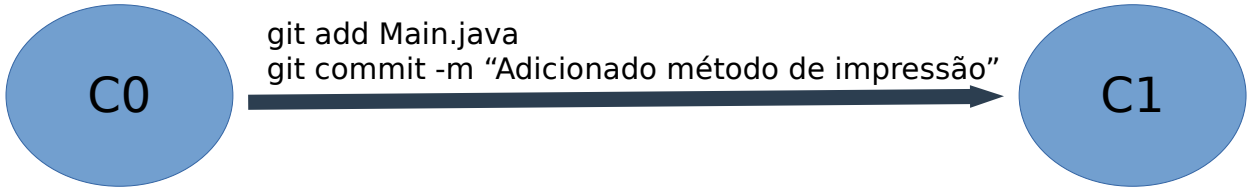
- **Conjunto de alterações feitas em um determinado momento**
  - Mas é possível retornar a outro estado...



Commit

- **Como ‘*commitar*’:**
  - Informar os arquivos que foram alterados:
    - **git add .** (O ponto . informa todos os arquivos de uma vez)
  - Fazer o commit (cria-se um estado novo no projeto)
    - **git commit -m “Criado o método para impressão”**

Commit




```
import java.util.Scanner;

public class Main {

    static Scanner LER = new Scanner(System.in);

    public static void main(String[] args) {

        System.out.println("Digite dois números inteiros:
");

        int a = LER.nextInt();

        int b = LER.nextInt();

    }

}
```

```
import java.util.Scanner;

public class Main {

    static Scanner LER = new Scanner(System.in);


    public static void main(String[] args) {

        System.out.println("Digite dois números
inteiros: ");

        int a = LER.nextInt();

        int b = LER.nextInt();

    }


    public static void imprimeNumeros(int n1, int
n2){

        System.out.println(n1 + ", " +n2);

    }

}
```

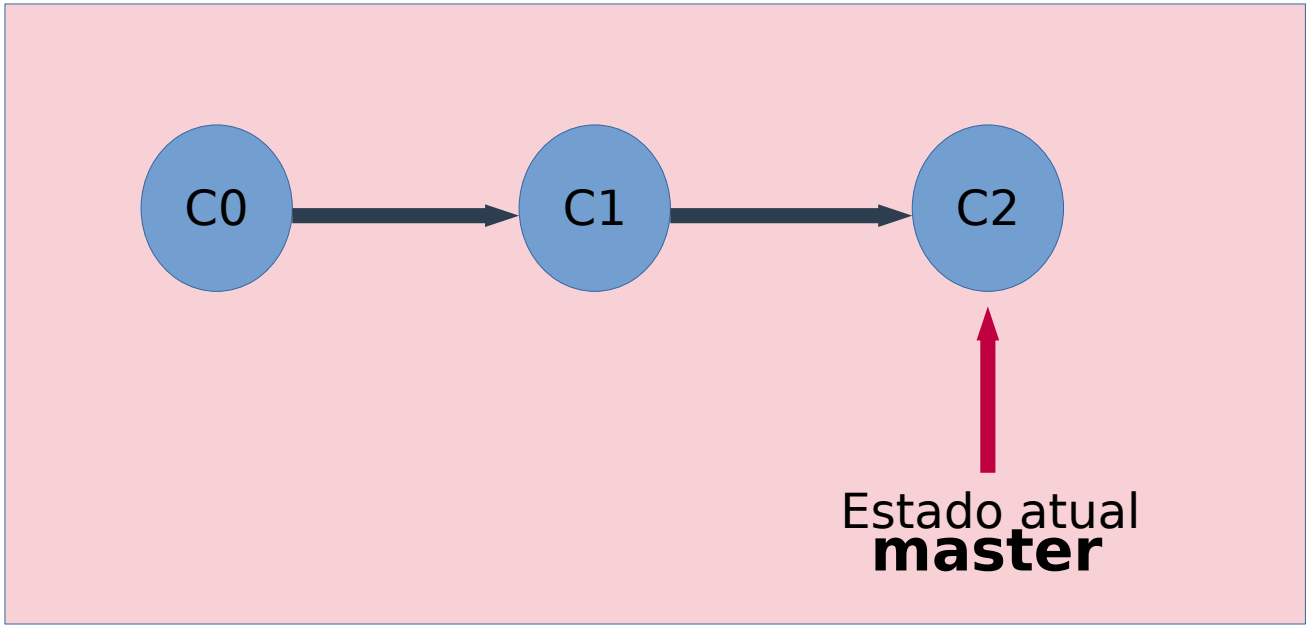
Sincronizar com o repositório online

- **push** (vem de *empurrar, enviar*)
- Mas antes é preciso sincronizar
  - git branch -M main (alterar o nome do ramo)
  - git remote add origin url_do_repositório
  - git **push** -u origin main

- **Mas o que aconteceu?**

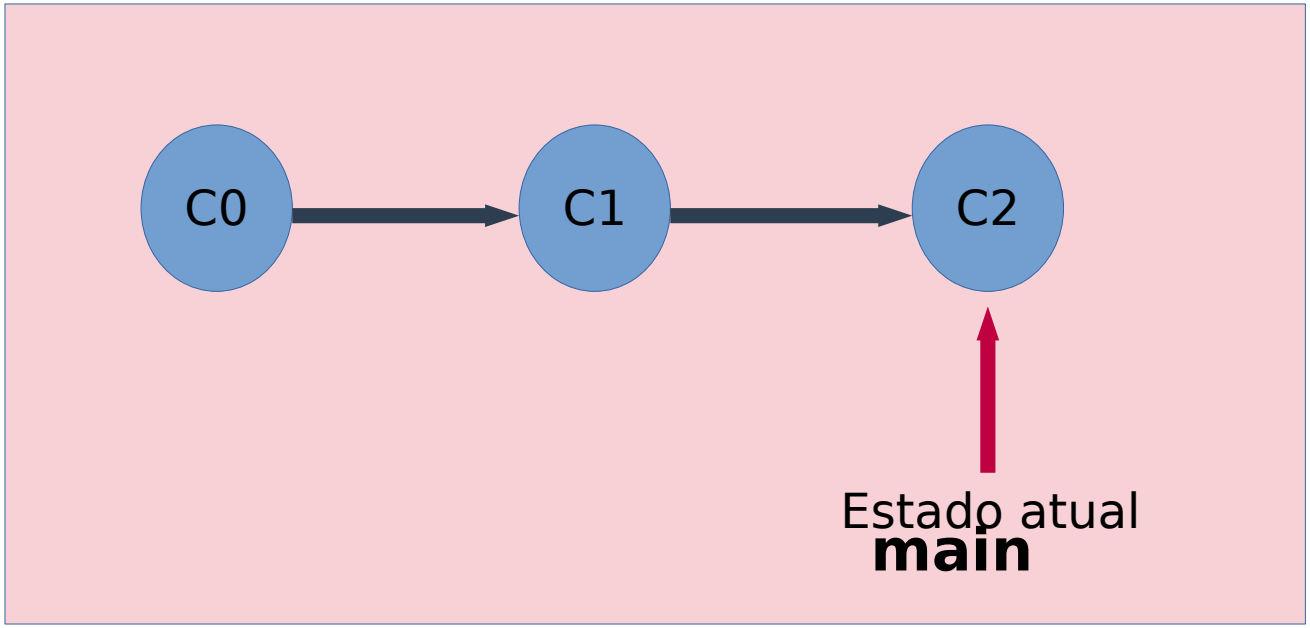
Branch (ramo)

- Branch **master**
  - é default (padrão)

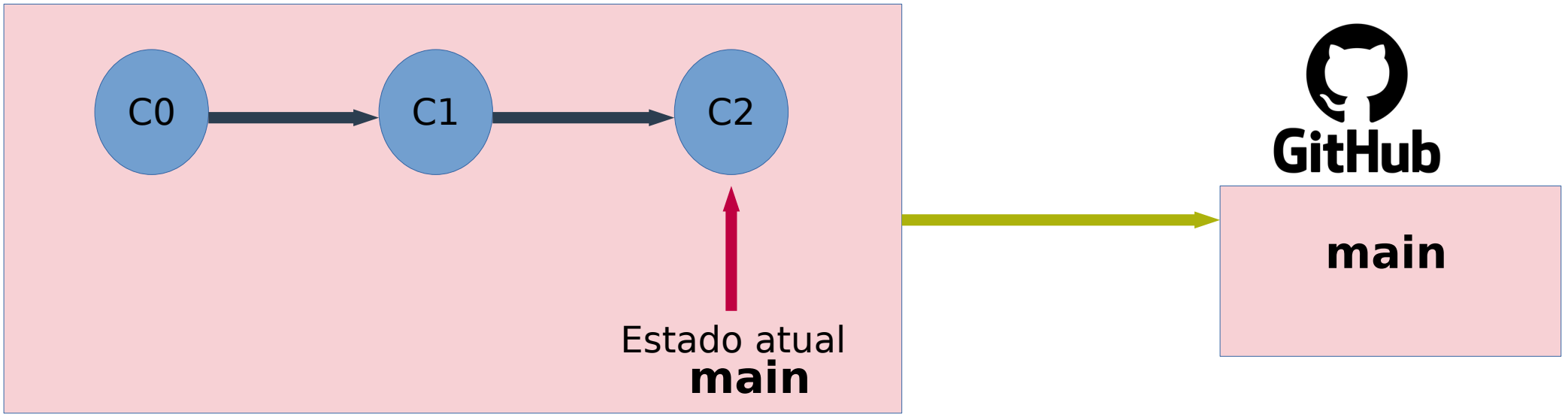


Branch (ramo)

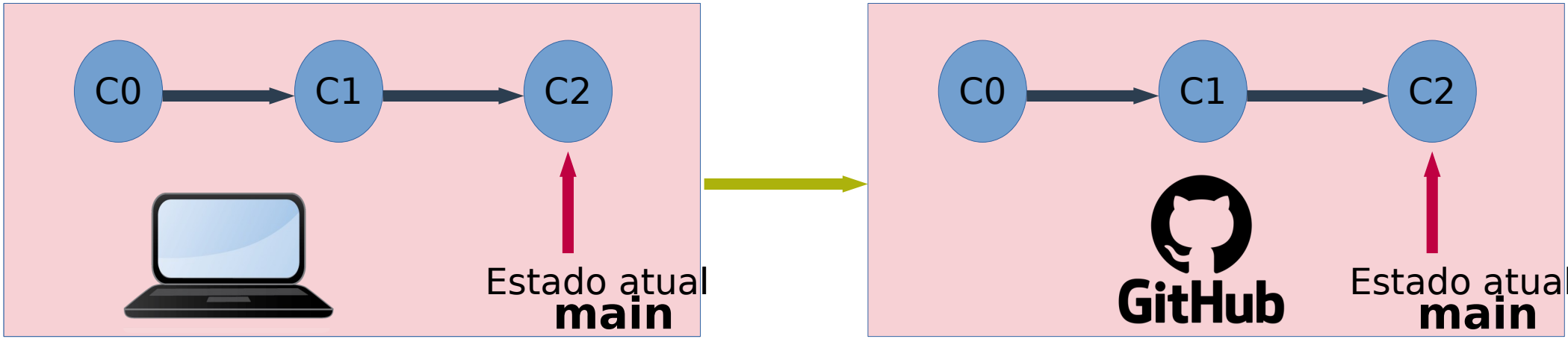
- O comando ***git branch -M main***
  - *troca o nome para main*



- O comando **git remote add origin url_do_repositório**
  - **Sintoniza o repositório local com o remoto**



- O comando ***git push -u origin main***
  - **Envia as mudanças e inserções do repositório local para o remoto**
  - **Agora ambos estão sincronizados!**



Agora...

- **Qualquer usuário pode fazer um *clone***
- **O projeto está em um lugar seguro**
- **Pode ser compartilhado**
- **Pode ser modificado de qualquer lugar com acesso à internet**

Primeiro exercício – clonar um projeto

- **Download do Git e do VSCode**
- **Testar o git:**
  - git –version
- **Criar um diretório/pasta “Projetos”**
- **Abrir esta pasta no VSCode**
- **Abrir o terminal pelo VSCode**
- **Fazer o clone do projeto pelo link passado:**
  - *git clone link*
- **Entrar na pasta e verificar os materiais**

Segundo exercício – Criar um projeto

- **Abrir a pasta Projetos no VSCode**
- **Criar uma pasta com o nome do projeto MinhaArte no VSCode**
- **Entrar na pasta pelo terminal:** cd MinhaArte
- **Iniciar o git:** git init
- **Configurar usuário e e-mail:**
  - git config user.email “seuemail”
  - git config user.name “seunome”
- **Criar um arquivo chamado** README.md
  - Arquivo com a descrição do projeto (aparecerá na página do projeto no github)
- **Verificar o status atual dos arquivos:**
  - git status
- **Adicionar o arquivo:**
  - git add . (ou git add README.md)
- git status
- **Fazer o commit (commitar)**
  - git commit -m “Arquivo README adicionado”
  - git status

Segundo exercício – Criar um projeto

- **Neste momento o repositório local está OK, apenas aguardando para enviar para o remoto**
- **Criar o repositório remoto pelo site do GitHub**
- **Sintonizar:**
  - git remote add origin LINK
- **Renomear a branch**
  - git branch -M main
- **Verificar a branch:**
  - git branch
- **Enviar os arquivos para o github:**
  - git push -u origin main
- **Se tudo deu certo, abrir o GitHub para verificar**
- **Seguir os próximos passos**

Terceiro exercício – Contribuir com um projeto

- Clonar o projeto especificado
- Criar um arquivo html com suas informações
- O arquivo deve ter o seguinte formato no nome: ultimonome.html
- Dar o push
- Esperar próximas instruções

Trabalhando com Branchs

- **Impede besteiras no ramo principal (produção)**
- **Criar um ramo a partir do principal:**
  - git branch novoRamo
- **Mudar para o ramo novo:**
  - git checkou novoRamo
- **Todos os commits serão feitos neste novo ramo;**
- **Para adicionar este branch ao repositório remoto:**
  - git remote add origin novoRamo
  - git push
- **Ao fim, para juntar o trabalho no ramo principal, basta mudar para o ramo principal:**
  - git checkout main
- **E fazer o merge (a união dos ramos):**
  - git merge novoRamo

Prática em equipe

- **Dividir em equipes de 5 membros.**
- **Definir um tema para um site com 3 páginas (principal e mais duas)**
- **Definir um líder**
- **Todos devem criar e trabalhar em um branch chamado *dev***
- **Dividir as tarefas como a seguir:**

Prática em equipe

- **Membro 1: Criar o repositório e adicionar os demais (incluindo o professor). Criar o README.md do repositório com a descrição.**
- **Membro 2: Criar a página inicial (grupoX.html) com os links para as demais paginas.**
- **Membro 3: Criar uma das páginas.**
- **Membro 4: Criar uma das páginas.**
- **Membro 5: Criar o arquivo estilo.css.**

Projeto em equipe

- **Todos os membros devem fazer o *push* para o branch *dev*.**
- **Após o membro 1 confirmar, deve fazer o *merge* para o branch *main*, todos devem fazer o *pull* e abrir o site.**
- **SUCESSO!**