



WORKSHOP: GIT & GITHUB

● IGOR FALUHELYI

Aluno de mestrado

Cientista de dados na Bancorbrás

Email para contato

igor.faluhelyi@aluno.unb.br



TABLE OF CONTENT

WORKSHOP: GIT & GITHUB



01 Introdução

02 Preparação

06 Exercício prático

04 Conceitos básicos do Git

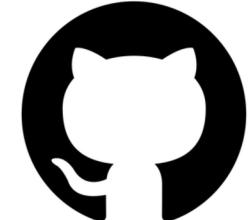
05 Introdução ao GitHub



INTRODUÇÃO

-  **git** 2005 - Linus Torvalds (Engenheiro de Software e criador do Linux)

Sistema de controle de versão para gerenciar mudanças em códigos ou projetos

-  **GitHub** 2008, Califórnia (Estados Unidos);
2018 - Microsoft (criadora do Windows)
Plataforma para hospedar repositórios e colaborar com outras pessoas

- Benefícios das tecnologias: Rastrear histórico, trabalho colaborativo, integração com outras ferramentas (VS Code, Amazon, etc...)



PREPARAÇÃO

● Pré-requisitos:

Instale o Git: [link oficial](#)

Crie uma conta no GitHub: [link oficial](#)

● Configuração inicial do Git:

```
git config --global user.name "Seu Nome"
```

```
git config --global user.email "seuemail@example.com"
```



CONCEITOS BÁSICOS DO GIT

- **Repositório:**
 - Pasta especial** que organiza e armazena os arquivos de um projeto.
 - Rastreia** as mudanças feitas nos arquivos (o que foi alterado, adicionado ou removido).
 - Permite que você volte para versões anteriores dos arquivos caso algo dê errado.
 - Facilita a colaboração, já que várias pessoas podem contribuir no mesmo projeto.



CONCEITOS BÁSICOS DO GIT

● **Criação de um Repositório:**

Escolha uma pasta. Dentro dela, via terminal, **execute:** `git init`

● **Adicionar arquivos ao controle de versão:**

Crie um arquivo dentro de um Repositório (**pasta especial**)

Adicione e faça o **commit**

`git add README.md`

`git commit -m "Adiciona README
inicial"`



CONCEITOS BÁSICOS DO GIT

Histórico de commits

Para verificar os **commits** anteriores e suas mensagens git log

Rastrear mudanças

Após editar os arquivos dentro do **repositório** git status
git diff



CONCEITOS BÁSICOS DO GIT

● **Integração Git-VScode**

Utilização mais visual do Git e suas funcionalidades.



CONCEITOS BÁSICOS DO GIT

● Trabalho em equipe com Git

Jargão técnico: **Branch** == ramificação == trabalho em paralelo

Trabalhar em novas funcionalidades sem interromper o funcionamento
do projeto principal

Testar alterações antes de mesclá-las (**merge**) com o código principal

Trabalhar com outras pessoas: Cada colaborador pode trabalhar em
uma **branch** separada



CONCEITOS BÁSICOS DO GIT



Trabalho em equipe com Git

Branch principal criada por padrão no Git: Master ou Main

Ver as **branches** existentes git branch

Criar nova **branch** == Cópia exata da **branch** que você está no momento
git branch nome-da-branch

Alternar entre **branches**

git checkout nome-da-branch



INTRODUÇÃO AO GITHUB



Repositório

Local: Fica no seu computador e é gerenciado pelo Git.

Remoto:

Está hospedado em uma plataforma como GitHub.

Conecta-se ao **repositório local** para sincronizar as mudanças.





INTRODUÇÃO AO GITHUB

Utilização do GitHub - link entre repositórios Locais e Remotos

Repositório local para Repositório remoto:

Crie um **Repositório remoto** no GitHub (usando o site mesmo)

Configure o **Repositório Remoto**

```
git remote add origin <URL-do-repositorio>
git branch -M main
```

Envie os Arquivos para o **Repositório Remoto**

```
git push -u origin main
```



INTRODUÇÃO AO GITHUB

Utilização do GitHub - link entre repositórios Locais e Remotos

Repositório remoto para **Repositório local**:

Basta clonar o **repositório remoto** hospedado no GitHub

`git clone <URL-do-repositorio>`



INTRODUÇÃO AO GITHUB



Colaboração via GitHub + Git

Encontre o **repositório remoto** em que você deseja colaborar.

Faça um **fork** desse **repositório** (pelo site do GitHub mesmo)

Siga a estrutura **Repositório remoto** para **Repositório local**:

```
git clone <URL-do-repositorio>
```

Faça as mudanças (colaboração) em uma nova **branch**

Siga o passo a passo do [Slide 10](#)



INTRODUÇÃO AO GITHUB

Colaboração via GitHub + Git

Jargão técnico: **Issue** == tarefa == discussão

Pode ser usada para relatar bugs, sugerir melhorias ou
planejar novas funcionalidades

Jargão técnico: **Pull Request** == formalizar um proposta de
mudanças em um repositório

Ele permite que você envie suas alterações (de uma
branch ou fork) para revisão antes que sejam
incorporadas ao projeto principal.



EXERCÍCIO PRÁTICO

- Façam um **Fork** do **repositório remoto** que foi criado nesse workshop.
Depois, clonem o **fork** na máquina de vocês.
- Criem uma nova **branch**, e, nela, criem um arquivo .py com código python (pode ser um simples `print('hello')`). Esse arquivo .py deve possuir nome do arquivo sendo seu respectivo nome e matrícula.
- Façam um Pull Request para o **repositório remoto** que foi criado nesse workshop
- A presença de vocês será medida com a conclusão desse exercício. Data de entrega: 03/02/2025



EXERCÍCIO BÔNUS

- Façam um **merge** de todas as **branches** do repositório CEE2 que vocês criaram para essa disciplina e veja o que acontecerá





THANK YOU

