

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**

Controle de versão com Git e GitHub

Fluxo de trabalho básico com Git

Aula 1

Código da aula: [DADOS]ANO1C1B3S23A1

Controle de versão
com Git e GitHub

Mapa da Unidade 8 Componente 1

Você está aqui!

Fluxo de trabalho básico
com o Git

semana

23

Branches e Merging

semana

27

semana

20

Fundamentos de
Git

semana

25

Branches no Git

semana

29

GitHub

Controle de versão
com Git e GitHub

Mapa da Unidade 8 Componente 1

Você está aqui!

Fluxo de trabalho básico
com Git

Aula 1

Código da aula: [DADOS]ANO1C1B3S23A1

23





Objetivo da aula

- Vamos revisar o que aprendemos até o momento sobre as ferramentas de desenvolvimento de software, como Git e GitHub, para aprofundar nas aulas seguintes.



Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.



Duração da aula

50 minutos



Competência técnica

- Usar ferramentas de desenvolvimento de software, como Git e GitHub.



Competência socioemocional

- Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns.



Primeiras
ideias

Quando duas pessoas discordam entre si, podemos dizer que há um conflito

Quando falamos de um conflito no Git, o que isso significa?

Como podemos fazer para resolver um conflito entre pessoas?

E como faremos para resolver um conflito no Git?

Ponto de
partida

Revisão – instalação e configuração do Git e GitHub

Assistam ao vídeo a seguir:



EDUCAÇÃO PROFISSIONAL PAULISTA. [DADOS]C1U8S23 – *Configurações iniciais de Git e GitHub*. Disponível em: <https://youtu.be/ccgvKn0cE8k?si=v2KggaF5G4TRITza>. Acesso em: 10 jul. 2024.

Verifiquem se o Git está instalado corretamente nas máquinas que vocês estão utilizando, conforme as configurações necessárias. Não é preciso criar um repositório no GitHub por enquanto; vamos experimentar localmente primeiro.

Construindo
o **conceito**

Fluxo de trabalho básico com Git

- ▶ Em um projeto com Git, a sequência básica de passos a seguir, ao fazer uma feature, é:
 1. Ao começar uma feature, separar uma branch;
 2. Adicionar os arquivos importantes para a feature na área de staging;
 3. Se quiser adicionar todos os arquivos, você pode usar o comando `git add`;
 4. Fazer um commit, explicando o que foi desenvolvido;
 5. Fazer push para o repositório remoto.

Construindo
o **conceito**

Como funciona a área de *staging*?

- ▶ A área de staging, ou staging area, é um conceito-chave no Git, funcionando como uma preparação para as mudanças que serão definitivamente salvas no repositório por meio de um commit.
- ▶ A área de staging permite aos desenvolvedores selecionar e verificar as mudanças que querem incluir em um commit antes de fazer o commit efetivamente. Isso é feito usando o comando `git add`, que adiciona as alterações dos arquivos ao staging.

Construindo
o **conceito**

A área de staging

Repositório com arquivo que não está staged:

```
Inspira@LAPTOP-ESR7KSCB MINGW64 ~/Documents/exemplo_git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    poesia_independente.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Elaborado especialmente para o curso com a ferramenta terminal do PowerShell.

Construindo
o **conceito**

A área de staging

Agora, o arquivo poesia_independente está em staging:

```
Inspira@LAPTOP-ESR7KSCB MINGW64 ~/Documents/exemplo_git (master)
$ git add .

Inspira@LAPTOP-ESR7KSCB MINGW64 ~/Documents/exemplo_git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   poesia_independente.txt
```

Elaborado especialmente para o curso com a ferramenta terminal do PowerShell.

Construindo
o **conceito**

Vantagens da área de staging

- ▶ Usar a área de staging permite uma revisão cuidadosa e garante que apenas as alterações desejadas sejam “comitadas”.
- ▶ Assim, pode-se melhorar o fluxo de trabalho, ao permitir que desenvolvedores agrupem logicamente alterações específicas em commits distintos. Isso facilita a manutenção do histórico de versões limpo e organizado, tornando mais fácil para qualquer membro da equipe seguir as mudanças.
- ▶ Às vezes, não é fácil visualizar o que está acontecendo na área de staging. Por isso, pode ser conveniente instalar uma extensão para facilitar o uso do Git.

Construindo
o **conceito**

Posh-Git: aumentando a produtividade com Git no PowerShell

- ▶ Posh-Git é uma extensão para o PowerShell que integra o Git diretamente na linha de comando do Windows.
- ▶ Ele adiciona informações úteis ao prompt, como o nome da branch atual, o estado dos arquivos (modificados, adicionados à staging area etc.), facilitando o gerenciamento de repositórios Git.

Construindo
o **conceito**

Configurando o Posh-Git para melhorar seu fluxo de trabalho

- ▶ Instalar o Posh-Git é simples e pode ser feito via gerenciadores de pacotes, como Chocolatey, ou diretamente pelo PowerShell.
- ▶ Uma vez instalado, o Posh-Git fornece um feedback visual imediato sobre o estado do repositório, ajudando a evitar erros comuns, como esquecer de adicionar arquivos à staging area ou fazer commits.

Construindo
o **conceito**

Git com Posh-Git

Terminal do PowerShell com Posh-Git indicando as alterações, sem precisarmos usar o git status (usamos só para visualizar):

```
C:\Users\Inspira\Documents\exemplo_git [master +1 ~0 -0 !]> git add .  
C:\Users\Inspira\Documents\exemplo_git [master +1 ~0 -0 ~]> git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   poesia_independente.txt  
  
C:\Users\Inspira\Documents\exemplo_git [master +1 ~0 -0 ~]> |
```

Elaborado especialmente para o curso com a ferramenta terminal do PowerShell.



Colocando
em **prática**

Instalação do Posh-Git

Vamos colocar em prática o que aprendemos hoje? Então, com seu grupo, siga os passos abaixo:

1. Instale o Posh-Git, seguindo as instruções do repositório oficial: GITHUB. *Dahlbyk* – o que é isso? [s. d.]. Disponível em: <https://github.com/dahlbyk/posh-git>. Acesso em: 27 jun. 2024.
2. Configure sua linha de comando para usar o Posh-Git e observe como as informações sobre o repositório são exibidas diretamente no prompt.
3. Como entrega, envie um print do seu prompt com a interface do Posh-Git.



Até o final da aula



Em grupo – até quatro
pessoas



Enviar no AVA



© Getty Images

O que nós
**aprendemos
hoje?**

Então, ficamos assim...

- 1** Entendemos o conceito e a importância da área de staging no Git, onde os desenvolvedores podem preparar e revisar suas alterações antes de “comitar”.
- 2** Conhecemos o Posh-Git, uma ferramenta que integra informações essenciais do Git no prompt do PowerShell, melhorando significativamente a eficiência e a experiência dos usuários ao gerenciar repositórios Git.
- 3** Realizamos a instalação do Posh-Git, seguindo instruções encontradas no repositório oficial da ferramenta.

Saiba mais

Uma das formas de instalar o Posh-Git é utilizar o Chocolatey. Você conhece essa ferramenta?

Esse artigo pode te ajudar a entender mais sobre o assunto!

FERREIRA, F. J. L. Chocolatey - Um poderoso gerenciador de pacotes para Windows. *Medium*, 28 ago. 2019.

Disponível em:

<https://fabiojanio.medium.com/chocolatey-um-poderoso-gerenciador-de-pacotes-para-windows-a87be4372257>. Acesso em: 27 jun. 2024.

Referências da aula

BELL, P.; BEER, B. *Introdução ao GitHub: um guia que não é técnico*. São Paulo: Novatec, 2014.

CHACON, S.; STRAUB, B. *Pro Git. Everything You Need to Know about Git*. EUA: Apress, 2024.

CÓDIGO DO VISUAL STUDIO. *Edição de código redefinido*. [s. d.]. Disponível em: <https://code.visualstudio.com/>.

Acesso em: 27 jun. 2024.

EDUCAÇÃO PROFISSIONAL PAULISTA. *[SIS]C4UIS15 – Configurações iniciais de Git e Git Hub*. Disponível em:

<https://www.youtube.com/watch?v=HBndJYeu2HU>. Acesso em: 27 jun. 2024.

GITHUB. *Vamos construir a partir daqui*. [s. d.]. Disponível em: <https://github.com/>. Acesso em: 27 jun. 2024.

GITHUB. *Dahlbyk – o que é isso?* [s. d.]. Disponível em: <https://github.com/dahlbyk/posh-git>. Acesso em: 27 jun. 2024.

GITHUB DOCS. *Documentação de introdução ao GitHub*. [s. d.]. Disponível em:

<https://docs.github.com/pt/get-started>. Acesso em: 27 jun. 2024.

Identidade visual: imagens © Getty Images.

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**