Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Controle de versão com Git e GitHub

Fluxo de trabalho básico com Git

Aula 1

Código da aula: [DADOS]ANO1C1B3S23A1

Mapa da Unidade 8 Componente 1

Controle de versão com Git e GitHub

semana 20

Fundamentos de Git

semana 23

Você está aqui!

Fluxo de trabalho básico com o Git

semana 25

Branches no Git

semana 27

Branches e Merging

semana 29

GitHub

Controle de versão com Git e GitHub

Mapa da Unidade 8 Componente 1

Você está aqui!

23

Fluxo de trabalho básico com Git

Aula 1

Código da aula: [DADOS]ANO1C1B3S23A1

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

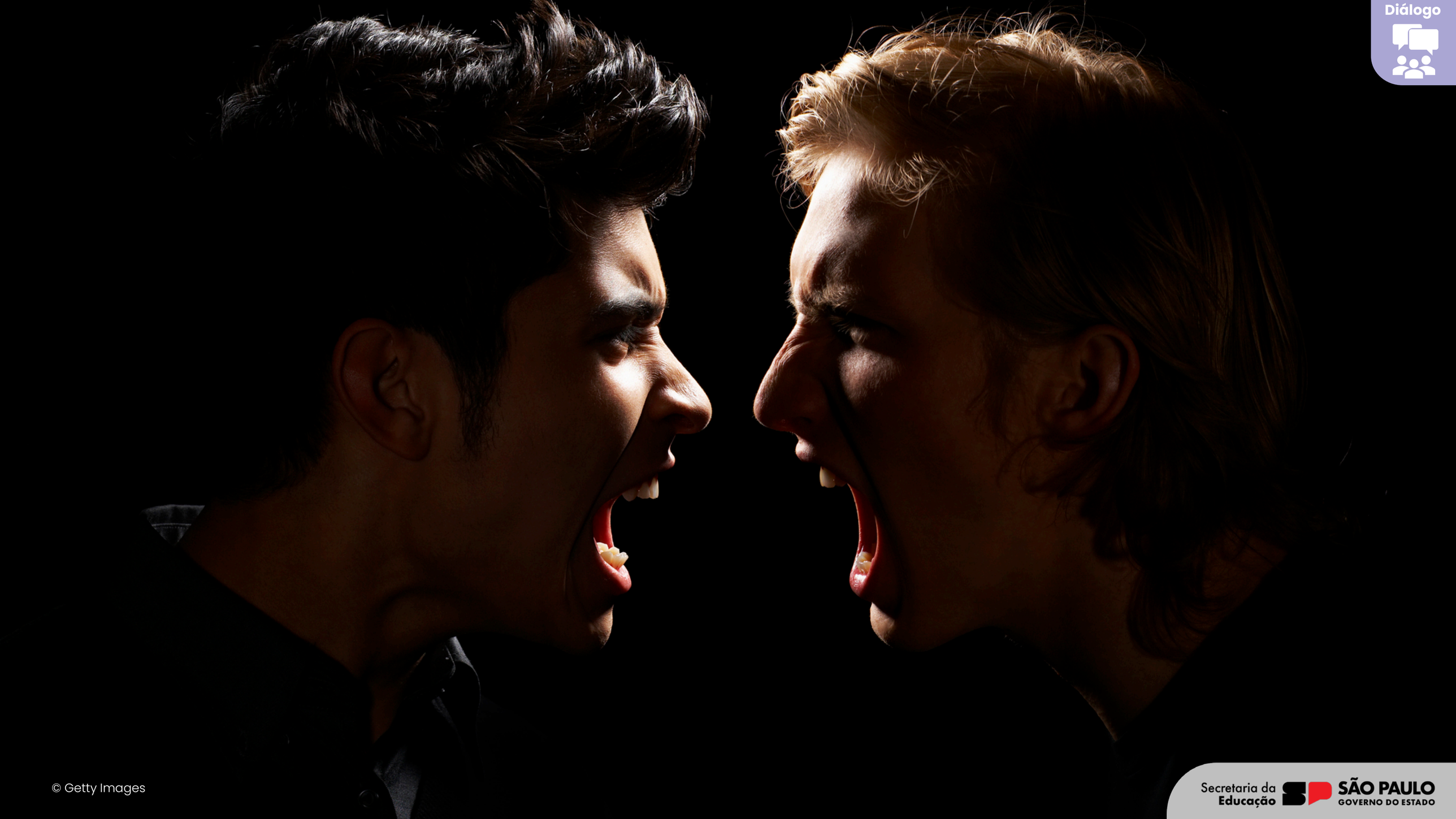
50 minutos

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.



Diálogo

Primeiras
ideias

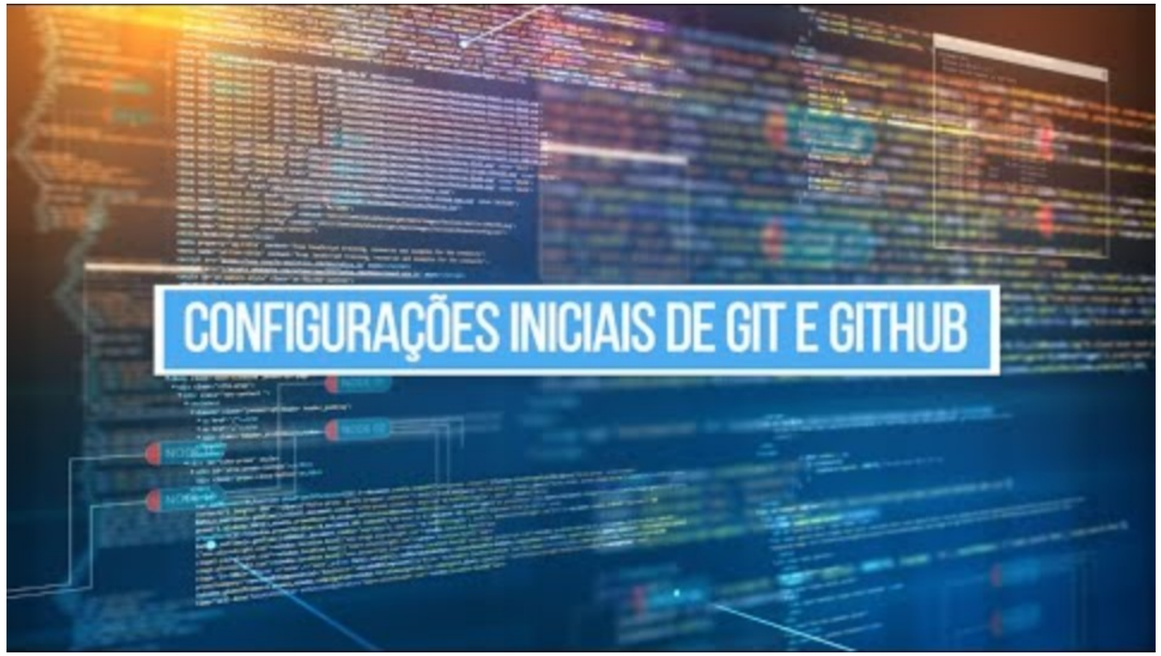
Quando duas pessoas discordam entre si, podemos dizer que há um conflito

- 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

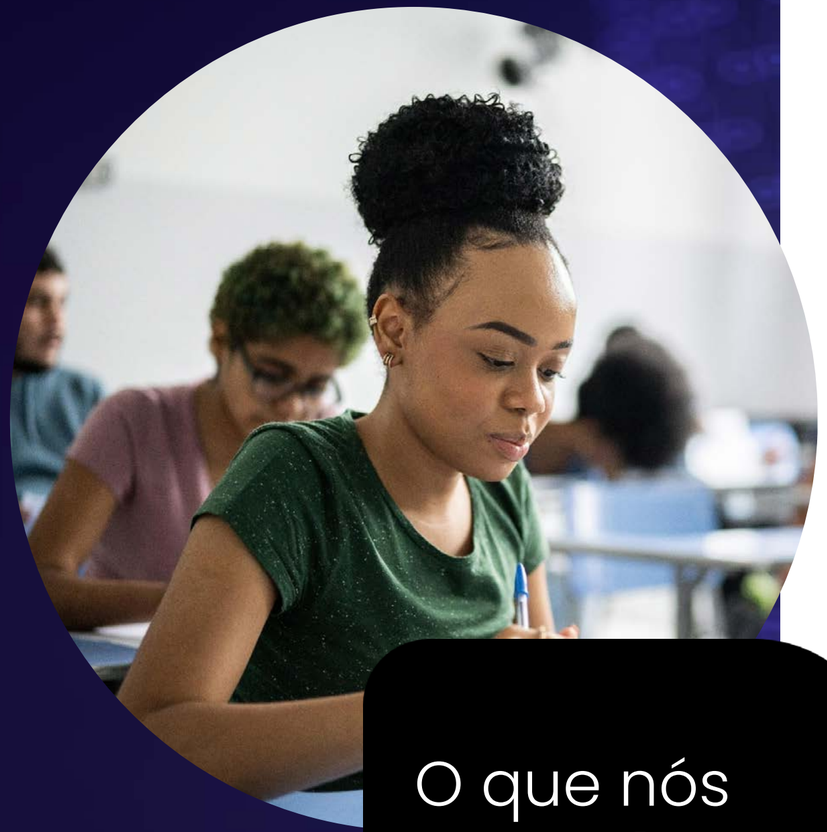
Vamos colocar em prática o que aprendemos hoje? Então, com seu grupo, siga os passos abaixo:

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]C4U1S15 – 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**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Controle de versão com Git e GitHub

Fluxo de trabalho básico com Git

Aula 2

Código da aula: [DADOS]ANO1C1B3S23A2

Mapa da Unidade 8 Componente 1

Controle de versão com *Git* e *GitHub*

semana 20

Fundamentos de Git

semana 23

Você está aqui!

Fluxo de trabalho básico com o Git

semana 25

Branches no Git

semana 27

Branches e Merging

semana 29

GitHub

Controle de versão com *Git* e *GitHub*

Mapa da Unidade 8 Componente 1

Você está aqui!

23

Fluxo de trabalho básico com Git

Aula 2

Código da aula: [DADOS]ANO1C1B3S23A2

Objetivo da aula

- Aprender o que são conflitos de merge e como resolvê-los.

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

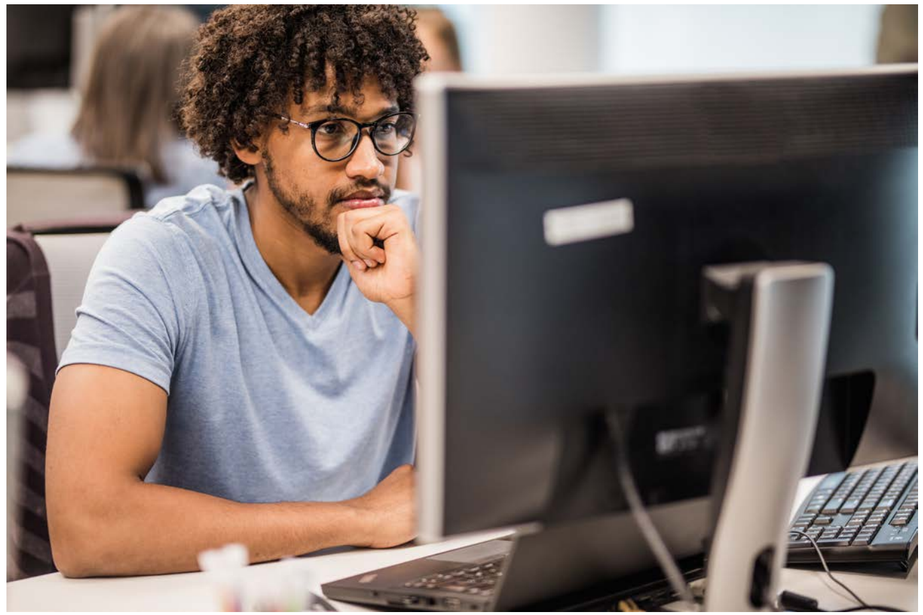
- Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns.

Construindo o **conceito**

Conflitos no Git

Na aula passada, falamos sobre como conflitos ocorrem no Git e como resolvê-los.

Hoje, vamos nos aprofundar nesse tema, considerando os conflitos de merge – o que são – e como podemos resolvê-los.



Construindo
o **conceito**

O que são conflitos de merge?

- **Conflitos de merge** ocorrem no Git quando duas alterações independentes entram em colisão, geralmente quando duas branches modificam a mesma linha de um arquivo ou quando um arquivo é modificado em uma branch e deletado em outra.
- Para resolver conflitos de merge, é preciso fazer uma **intervenção manual.**

Construindo o **conceito**

Entendendo as causas de conflitos

- Conflitos de merge são mais comuns em equipes grandes ou em projetos com muitas branches ativas.
- Eles ocorrem, pois alterações simultâneas nos mesmos arquivos, especialmente em áreas de código modificadas com frequência, causam confusão no Git sobre qual das mudanças deveria ser permanente, podendo ser uma, outra ou ambas.

Construindo
o **conceito**

Prevenindo conflitos de merge

- O ideal é manter uma comunicação clara entre membros da equipe e realizar merges pequenos e frequentes, evitando que haja necessidade de resolver conflitos significativos.
- Utilizar ferramentas de revisão de código e incorporar práticas de integração contínua também são estratégias eficazes contra conflitos, pois reduzem a quantidade de alterações em um mesmo bloco de código que entram de uma única vez e deixam a equipe mais ciente do caminho que o desenvolvimento está tomando.

Construindo o **conceito**

Exemplo de conflito de merge

Temos o texto da poesia escrito no repositório – pense que funcionará igual para código:

poesia_independente.txt

```
1   Estrelas no manto da noite a brilhar,
2   Segredos sussurram na névoa a dançar,
3   Sombra e luz, numa dança sem par,
4   No silêncio da lua, sonhos a criar.
5
6   Nas asas do vento, desliza um desejo,
7   Ecoando memórias de tempos distantes,
8   Fantasmas de riso, de toque e de beijo,
9   Fazendo do agora momentos vibrantes.
10
11  Entre os véus do horizonte dourado,
12  Amanhã espreita, novo e sorrateiro,
13  Na cadência dos dias, segue ao lado,
14  De um coração valente, livre e inteiro.
```

Elaborado especialmente para o curso com a ferramenta *VS Code*.

Construindo o **conceito**

Exemplo de conflito de merge

Fizemos uma pequena alteração no último verso e vamos “comitá-la” em outra branch:

poesia_independente.txt

You, 1 second ago | 1 author (You)

```
1  Estrelas no manto da noite a brilhar,
2  Segredos sussurram na névoa a dançar,
3  Sombra e luz, numa dança sem par,
4  No silêncio da lua, sonhos a criar.
5
6  Nas asas do vento, desliza um desejo,
7  Ecoando memórias de tempos distantes,
8  Fantasmas de riso, de toque e de beijo,
9  Fazendo do agora momentos vibrantes.
10
11 Entre os véus do horizonte dourado,
12 Amanhã espreita, novo e sorrateiro,
13 Na cadência dos dias, segue ao lado,
14 De um coração valente, livre e faceiro.
```

Elaborado especialmente para o curso com a ferramenta VS Code.

```
C:\Users\Inspira\Documents\exemplo_git [alteracao]> git add .
C:\Users\Inspira\Documents\exemplo_git [alteracao +0 ~1 -0 ~]> git commit -m 'Mudei a poesia'
[alteracao be971c0] Mudei a poesia
 1 file changed, 1 insertion(+), 1 deletion(-)
C:\Users\Inspira\Documents\exemplo_git [alteracao]>
```

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

Construindo o **conceito**

Exemplo de conflito de merge

Fizemos uma outra alteração no último verso e vamos “comitá-la” na master:

poesia_independente.txt

You, 8 seconds ago | 1 author (You)

```
1  Estrelas no manto da noite a brilhar,
2  Segredos sussurram na névoa a dançar,
3  Sombra e luz, numa dança sem par,
4  No silêncio da lua, sonhos a criar.
5
6  Nas asas do vento, desliza um desejo,
7  Ecoando memórias de tempos distantes,
8  Fantasmas de riso, de toque e de beijo,
9  Fazendo do agora momentos vibrantes.
10
11 Entre os véus do horizonte dourado,
12 Amanhã espreita, novo e sorrateiro,
13 Na cadência dos dias, segue ao lado,
14 De um coração valente, solto e inteiro.
```

Elaborado especialmente para o curso com a ferramenta VS Code.

```
C:\Users\Inspira\Documents\exemplo_git [master +0 ~1 -0 !]> git add .
C:\Users\Inspira\Documents\exemplo_git [master +0 ~1 -0 ~]> git commit -m 'Alteracao 2'
[master e95df41] Alteracao 2
 1 file changed, 1 insertion(+), 1 deletion(-)
C:\Users\Inspira\Documents\exemplo_git [master]>
```

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

Construindo o **conceito**

Exemplo de conflito de merge

Ao fazer o merge, temos o conflito aparecendo. Se abrirmos em um editor de texto especializado, esse é o resultado:

```
Entre os véus do horizonte dourado,
Amanhã espreita, novo e sorrateiro,
Na cadência dos dias, segue ao lado,
<<<<<<< HEAD
De um coração valente, solto e inteiro.
=======
De um coração valente, livre e faceiro.
>>>>>>> alteracao
```

```
C:\Users\Inspira\Documents\exemplo_git [master]> git merge alteracao
Auto-merging poesia_independente.txt
CONFLICT (content): Merge conflict in poesia_independente.txt
Automatic merge failed; fix conflicts and then commit the result.
C:\Users\Inspira\Documents\exemplo_git [master +0 ~0 -0 !1 | +0 ~0 -0 !1 !]>
```

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

Construindo o **conceito**

Como resolver conflitos de merge

- Para resolver um conflito de merge, primeiro, é necessário entender as mudanças conflitantes. Use comandos, como git status, para identificar arquivos conflitantes e, então, abra-os no editor.
- Para fazer as correções, escolha manualmente as alterações que deseja manter e salve o arquivo corrigido. Em seguida, faça um novo commit com a resolução do conflito.

Construindo
o **conceito**

Ferramentas de merge para auxiliar na resolução

- Resolver conflitos não costuma ser uma tarefa trivial e exige análise cuidadosa dos blocos de código e comunicação interna com a equipe.
- Ferramentas com interface gráfica, como a do Visual Studio Code (VS Code), podem ser utilizadas para simplificar o processo.

Construindo
o **conceito**

Exemplo de resolução de conflito

No VS Code, o conflito aparece assim:

Construindo
o **conceito**

Exemplo de resolução de conflito

Vou aceitar as duas mudanças e ajustar manualmente.

poesia_independente.txt
You, 4 minutes ago | 1 author (You)

```
1  Estrelas no manto da noite a brilhar,
2  Segredos sussurram na névoa a dançar,
3  Sombra e luz, numa dança sem par,
4  No silêncio da lua, sonhos a criar.
5
6  Nas asas do vento, desliza um desejo,
7  Ecoando memórias de tempos distantes,
8  Fantasmas de riso, de toque e de beijo,
9  Fazendo do agora momentos vibrantes.
10
11 Entre os véus do horizonte dourado,
12 Amanhã espreita, novo e sorrateiro,
13 Na cadência dos dias, segue ao lado,
14 De um coração valente, solto e inteiro.
15 De um coração valente, livre e faceiro.
```

poesia_independente.txt
You, 32 seconds ago | 1 author (You)

```
1  Estrelas no manto da noite a brilhar,
2  Segredos sussurram na névoa a dançar,
3  Sombra e luz, numa dança sem par,
4  No silêncio da lua, sonhos a criar.
5
6  Nas asas do vento, desliza um desejo,
7  Ecoando memórias de tempos distantes,
8  Fantasmas de riso, de toque e de beijo,
9  Fazendo do agora momentos vibrantes.
10
11 Entre os véus do horizonte dourado,
12 Amanhã espreita, novo e sorrateiro,
13 Na cadência dos dias, segue ao lado,
14 De um coração valente, solto e faceiro.
```

Elaborado especialmente para o curso com a ferramenta *VS Code*.

Construindo
o **conceito**

Exemplo de resolução de conflito

Agora, basta “comitarmos” a mudança salva e o conflito resolvido.

```
C:\Users\Inspira\Documents\exemplo_git [master +0 ~0 -0 !1 | +0 ~0 -0 !1 !]> git add .
C:\Users\Inspira\Documents\exemplo_git [master +0 ~1 -0 ~]> git commit -m 'resolve conflito'
[master aafb178] resolve conflito
C:\Users\Inspira\Documents\exemplo_git [master]>
```

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

Vamos fazer um **quiz**

O que causa um conflito de merge no Git?

- Falha na conexão com o repositório remoto.
- Mudanças nas mesmas linhas de arquivo em branches diferentes.
- Comitar arquivos muito grandes.
- Não atualizar o repositório local frequentemente.

Vamos fazer um **quiz**

Qual é a melhor prática para prevenir conflitos de merge significativos?

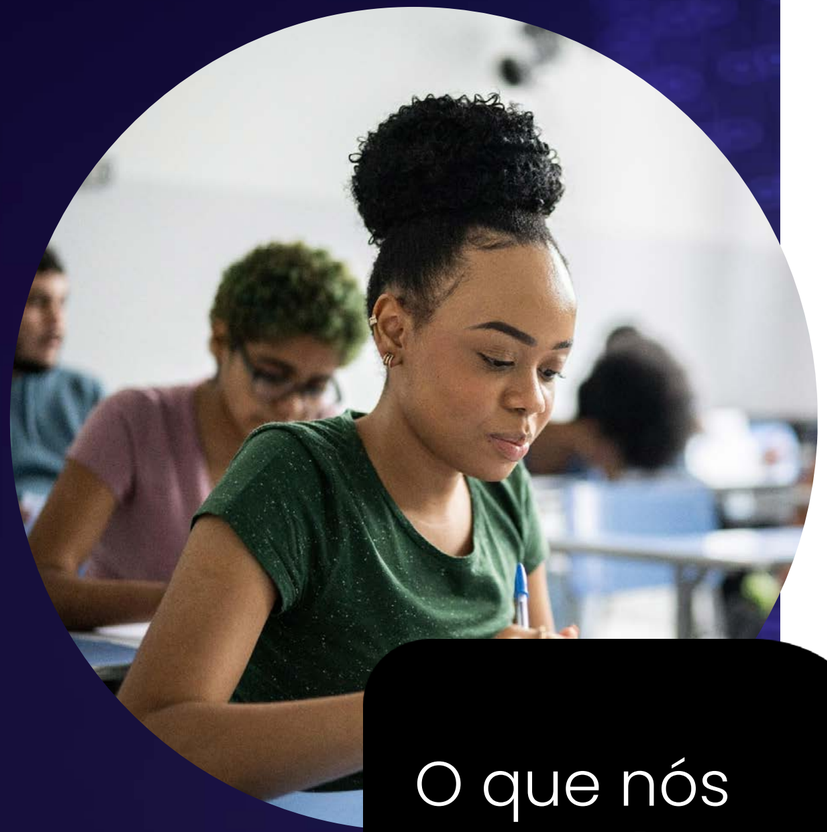
- Merges grandes e infrequentes.
- Limitar o número de branches no projeto.
- Merges pequenos e frequentes.
- Usar apenas uma branch para todos os desenvolvedores.

Registro

Vamos fazer um **quiz**

Qual ferramenta pode ser utilizada para facilitar a visualização e a resolução de conflitos de merge?

- **Git status.**
- **Git log.**
- **Git branch.**
- **VS Code.**



© Getty Images

O que nós **aprendemos hoje?**

Então, ficamos assim...

1. Explicamos o que são conflitos de merge, destacando que ocorrem quando alterações conflitantes são feitas nas mesmas linhas de um arquivo em branches diferentes.

2. Discutimos estratégias para prevenir conflitos de merge, como manter comunicação eficaz entre a equipe e realizar merges pequenos e frequentes.

3. Abordamos o uso de ferramentas de merge gráficas, como o VS Code, para facilitar a visualização e a resolução de conflitos, permitindo escolhas interativas sobre qual conteúdo manter.

Saiba mais

Quer saber mais sobre como usar o VS Code para resolver conflitos de merge? Neste artigo da Microsoft, podemos ver claramente quais as melhores formas de lidar com os conflitos:

MICROSOFT. *Como resolver conflitos de mesclagem no Visual Studio*, 22 abr. 2024. Disponível em: https://learn.microsoft.com/pt-br/visualstudio/version-control/git-resolve-conflicts?view=vs-2022. 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.

GITHUB DOCS. *Documentação de introdução ao GitHub*. [s. d.]. Disponível em: https://docs.github.com/pt/get-started. Acesso em: 27 jun. 2024.

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Controle de versão com Git e GitHub

Fluxo de trabalho básico com Git

Aula 3

Código da aula: [DADOS]ANO1C1B3S23A3

Mapa da Unidade 8 Componente 1

Controle de versão com Git e GitHub

semana 20

Fundamentos de Git

semana 23

Você está aqui!

Fluxo de trabalho básico com o Git

semana 25

Branches no Git

semana 27

Branches e Merging

semana 29

GitHub

Controle de versão com Git e GitHub

Mapa da Unidade 8 Componente 1

Você está aqui!

23

Fluxo de trabalho básico com Git

Aula 3

Código da aula: [DADOS]ANO1C1B3S23A3

Objetivo da aula

- Praticar o fluxo de trabalho com o Git e resolver conflitos.

Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.
- Máquinas com VS Code instalado.

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.

Colocando em **prática**

Prática de Git com conflitos de merge

Reúnam-se em grupos de quatro pessoas e iniciem um repositório Git em uma pasta. Para a realização da atividade, sigam o roteiro abaixo:

1. Abram a pasta usando o VS Code e escrevam um código Python para receber três notas, calcular a média aritmética e determinar se o aluno passou de ano.
2. Em seguida, dividam entre si as seguintes atividades, cada uma em uma branch:
    a. Implementar a média como média geométrica (caso não tenha conhecimento sobre como calcular, o grupo poderá consultar o Google).
    b. Implementar uma função que determine se a pessoa passou de ano considerando um input de nota (a nota deve ser maior que 5).
    c. Implementar uma função que determine se a pessoa passou de ano, com o mesmo nome do item b, considerando um input de presença (a presença deve ser superior a 75%).

Até o final da aula

Grupo (quatro pessoas)

Colocando em **prática**

Prática de Git com conflitos de merge

3. Ao juntar as informações na branch original, verifique se há conflitos – e resolva-os –, fazendo, assim, com que a função de passar de ano considere a média e a presença do aluno.

Ao finalizar a atividade com seu grupo, envie a pasta .zip, contendo o código implementado após o merge, como registro para o AVA (Ambiente Virtual de Aprendizagem).

Até o final da aula

Grupo (quatro pessoas)

Ser
sempre +

Situação

Em grupos de até quatro alunos, e utilizando os aprendizados dessa semana, discuta sobre o problema a seguir:

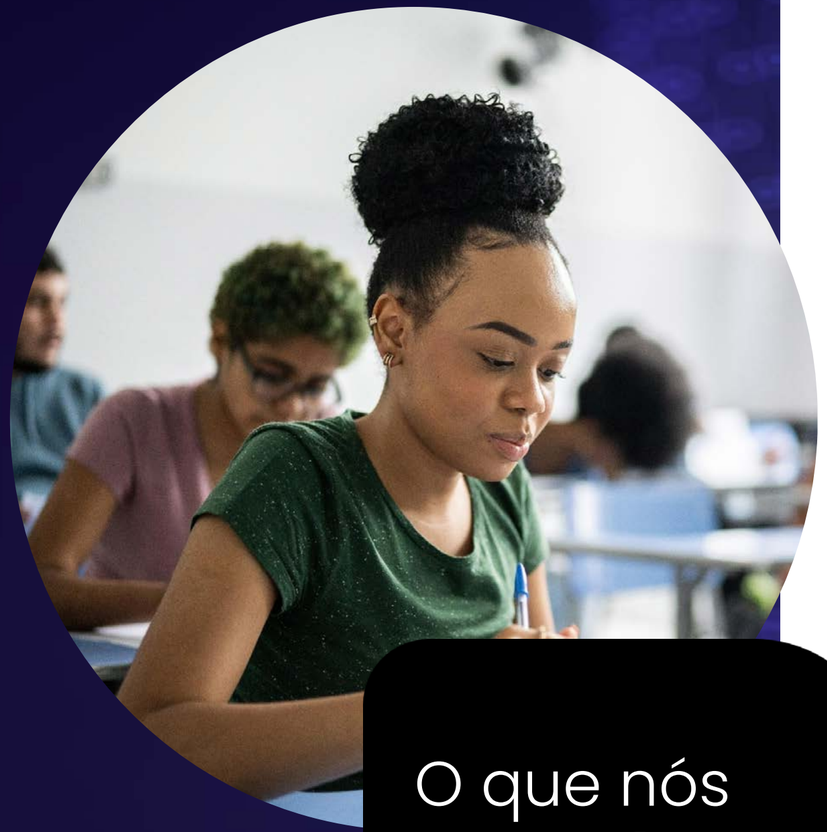
Trabalhando em um sistema como técnico em Ciência de Dados, você faz um commit com uma nova feature.

Após uma semana, sua feature sumiu do código: um colega de equipe fez outra feature para a mesma função e acabou resolvendo o conflito, descartando suas mudanças.

Essa situação te causa bastante chateação.

O que você faria nessa situação? Discuta com o seu grupo e, em seguida, com a turma.

Situação fictícia elaborada especialmente para o curso.



© Getty Images

O que nós **aprendemos hoje?**

Então, ficamos assim...

1 Vimos de forma prática, e em equipe, como funciona o fluxo de trabalho com o Git.

2 Identificamos como trabalhar com problemas de merge – e resolvê-los – no Git.

3 Exercitamos nossas competências ao lidar com frustrações no trabalho e dificuldades do trabalho em equipe.

Saiba mais

Para lidar melhor com o trabalho colaborativo, é importante estabelecer um conjunto de regras para trabalhar com o Git da melhor forma em uma equipe. Esses são chamados Git Workflows.

Quer saber mais sobre eles? Leia esse artigo abaixo:

MONTEIRO, A. *Git Workflow*: o que é e seus principais tipos. Medium, 22 jul. 2023. Disponível em: https://desenvbr.medium.com/git-workflow-o-que-%C3%A9-e-seus-principais-tipos-c51f1b7e4575. 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.

GITHUB DOCS. *Documentação de introdução ao GitHub.* [s. d.]. Disponível em: https://docs.github.com/pt/get-started. Acesso em: 27 jun. 2024.

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO