

Git e GitHub: compartilhando e colaborando em projetos

57%

03. Utilizando Git na IDE

ATIVIDADES

01. Git no VSCode

02. Simulando um conflito

03. Resolvendo conflitos

04. Resolvendo conflitos com Visual Studio Code Merge Editor

05. Conflitos em Git

06. Faça como eu fiz: utilizando o Git na IDE

07. Desafios: hora da prática

08. O que aprendemos?

OUTROS LINKS

Discord Alura

Fórum do curso

Voltar para Dashboard

MODO CLARO

Nicolas Sartor Parisotto

11.9k xp

04. Resolvendo conflitos com Visual Studio Code Merge Editor

PRÓXIMA ATIVIDADE

Quando você está trabalhando em um projeto grande, principalmente se muitas pessoas estão envolvidas, é comum aparecerem os chamados "conflitos". Mas o que são esses conflitos?

Imagine que o projeto é um grande quebra-cabeça e cada pessoa está trabalhando em uma parte dele. Se duas pessoas tentarem encaixar peças diferentes no mesmo lugar, surge um conflito. No mundo do desenvolvimento de software, isso ocorre quando duas pessoas editam o mesmo pedaço de código de formas diferentes.

Há várias formas de resolver conflitos, podemos utilizar a linha de comando ou o próprio Visual Studio Code. O editor de código fornece mais de uma forma para resolver conflitos de mesclagem entre o código local (o que está na sua máquina) e o remoto(o que está no github, por exemplo).

Uma possibilidade é utilizar a ferramenta “Merge Editor”. Vamos conferir seu funcionamento?

Resolvendo conflitos no Merge Editor

No exemplo a seguir, nós temos duas versões de um código na branch `main`: uma no repositório do github e outra modificação diferente no ambiente local. Ao realizarmos o `git -push` para a branch `main`, ocorreu um **conflito** e precisamos resolvê-lo para que a atualização suba para o repositório no github corretamente, como a imagem nos apresenta:

Para solucionarmos o problema, clicamos na opção “Resolve in Merge Editor”, como no print abaixo:

Após o clique, somos redirecionados para outra aba que apresenta as modificações no arquivo, vamos entender o que cada opção significa:

A tela do Visual Studio Code está dividida em três partes:

- Incoming (remoto):** modificações que chegam do repositório remoto.
- Current (local):** modificações locais.
- Result (resultado):** resultado do `merge`, ou seja, a resolução dos conflitos de mesclagem. É o estado atual do arquivo.

Os quadrados na cor amarela em volta do código no campo “Incoming” e “Current” são **marcadores de conflito**: exibem o conteúdo que apresenta conflito no arquivo.

Campo Incoming

-> No campo “Incoming”, acima da linha de código dos marcadores de conflito no campo há outras opções que resultam na alteração do código atual:

- Accept Incoming:** aceita modificações oriundas do **remoto**

- Accept Combination Incoming First:** realiza a combinação com as linhas do código do repositório remoto no topo.

- Ignore:** ignora as modificações.

Campo Current

-> O campo “Current” trabalha com as modificações locais do documento.

- Accept Current:** Aceita a modificação local no resultado do documento
- Accept combination Current First:** Aceita a combinação local + remoto. Nos resultados a linha de código com a tag `<h2>` fica antes de `<h1>`, comprovando que o código local é inserido primeiro que o remoto.

- Ignore:** ignora as modificações no resultado no final.

Após selecionarmos a opção com o resultado desejado, devemos:

- Salvar o arquivo
- Clicar no botão “complete Merge”
- Realizar o commit das modificações
- Sincronizar as modificações para realizar o push.

Para saber mais:

- [Como o Visual Studio facilita o controle de versão com o Git](#)
- [Como resolver conflitos de mesclagem no Visual Studio](#)
- [Um guia muito útil para mesclar conflitos - em Inglês](#)

DISCUTIR NO FÓRUM

PRÓXIMA ATIVIDADE