



# Front-End



# GLOSSÁRIO



# Versionamento de código usando o Git



Dica: para encontrar rapidamente a palavra que procura aperte o comando CTRL+F e digite o termo que deseja achar.

- **Aprenda a usar o terminal e conheça o Git Stage**
- **Salve as alterações (Commits)**
- **Liste alterações (Logs)**
- **Consulte o histórico**
- **Crie ramificações no código (Branches)**
- **Controle a versão (Github)**



# Aprenda a usar o terminal e conheça o Git Stage



# Aprenda a usar o terminal e conheça o Git Stage

## Glossário

### **Stage (área de preparação)**

Também conhecida como "staging area" ou "index" é onde as modificações nos arquivos são selecionadas para serem incluídas no próximo commit.



# Salve as alterações (Commits)



# Salve as alterações (Commits)

## Glossário

### ● Commit (Confirmar)

É uma operação que registra as alterações realizadas nos arquivos incluídos na área de preparação. Ele cria um ponto de referência na história do repositório.

### ● Markdown “md” (Marcação de texto)

É uma linguagem de marcação simples e leve amplamente utilizada para formatar e estruturar texto de forma rápida e fácil, com o objetivo de criar documentos legíveis tanto em formato de texto simples como em HTML.

### ● Staged (Preparado)

Significa que as modificações nos arquivos foram selecionadas e adicionadas à área de preparação, prontas para serem incluídas no próximo commit. As alterações staged serão registradas no histórico do repositório após o commit.

### ● Unstaged (Não preparado)

Refere-se a modificações em arquivos que foram alterados no diretório de trabalho, mas ainda não foram adicionados à área de preparação. Essas modificações não serão incluídas no próximo commit.



# Liste alterações (Logs)



# Liste alterações (Logs)

## Glossário

### Logs (Registros)

Referem-se aos registros de eventos ou informações relevantes que são registrados por um sistema ou aplicativo. Esses registros são geralmente armazenados em arquivos de log, que registram atividades específicas ocorridas durante a execução de um programa, sistema operacional, serviço da web ou qualquer outro sistema de software.



# Consulte o histórico



# Consulte o histórico

## Glossário

### Git restore

É um comando do Git que permite desfazer modificações em arquivos do repositório. Ele permite reverter alterações em arquivos modificados ou até mesmo restaurar arquivos excluídos.

### Head (Referência principal)

É uma referência especial que aponta para o commit mais recente em um ramo específico. Ele representa a posição atual em que você está no histórico do repositório. Basicamente, o "HEAD" indica a versão do código-fonte que você está visualizando ou trabalhando.



# Crie ramificações no código (Branches)



# Crie ramificações no código (Branches)

## Glossário

### ● **Master (Branch principal)**

Aos poucos a palavra master está sendo retirada do mundo da programação, por fazer referência a escravidão, termos como master e slave estão deixando de ser utilizados. No Github, a branch master deu lugar a branch main, apesar disso ao iniciar o Git no computador local a branch (por enquanto) se chamará master.



# Controle a versão (Github)



# Controle a versão (Github)

## Glossário

### Git push

Comando usado no Git para enviar as alterações locais do seu repositório para um repositório remoto. Ele sincroniza o seu repositório local com o repositório remoto, enviando os commits e ramificações que foram feitos.

### Git remote

Comando usado para gerenciar repositórios remotos. Ele permite visualizar, adicionar, renomear e remover repositórios remotos associados ao seu repositório local.

### Read me

Quando um repositório é hospedado no GitHub, o arquivo README.md é exibido de forma proeminente na página inicial do repositório. É uma prática comum fornecer informações importantes sobre o projeto nesse arquivo, como descrição, instruções de instalação, exemplos de uso, documentação e *links* úteis.



# Controle a versão (Github)

## Glossário

### Git pull

É o oposto do git push, o push envia as alterações para o Github (ou outro serviço) já o git pull irá baixar esses alterações (caso tenham sido feitas em outro computador ou por outro usuário), ambos atualizam o repositório, com o push atualizamos o repositório remoto com o pull o repositório local.



# Bons estudos!

