

# Capacitação Git e GitHub

Adriely Lima Novais



# Ferramentas:





# O que veremos?

- Conceitos.
- Como iniciar um projeto?
- Como clonar um projeto?
- O que são branches?
- Comandos básicos.
- O que é o .gitignore?
- O que é o merge e quem fica responsável por ele.
- O que são tags?

# Conceitos:

## ❖ O que é o git ?

- O git é uma ferramenta open source, que é usada para fazer o controle de versão.

## ❖ Qual a diferença entre GitHub e Git?

- GitHub é o local onde fica o repositório do código fonte dos projetos e o git gerencia as versões do projeto.

## ❖ O que é um repositório?

- Um repositório é onde o código fonte fica armazenado. Os repositórios podem ser locais ou remotos, hospedado em serviços como GitHub.

# Conceitos:

- ❖ **Como é criado um novo repositório no Git?**
  - Para criar um novo repositório Git em sua máquina usamos o comando **git init**. Isso inicializa um repositório vazio e permite que você comece a rastrear as alterações em seus arquivos.
- ❖ **Qual o comando para adicionar arquivos ao repositório local?**
  - Usamos o comando **git add nome\_do\_arquivo** para adicionar arquivo ao seu repositório local, preparando-o para ser incluído no próximo commit.

# Conceitos:

## ❖ Como commitar mudanças no Git?

- Para commitar suas mudanças no Git, usamos o comando **git commit -m "mensagem do commit"**, que a mensagem é uma breve descrição das mudanças que você está commitando. Essas mensagens ajudam a documentar o histórico de alterações do projeto.

## ❖ O que é uma branch (ramo) no Git e por que você usaria um?

- Uma branch é uma ramificação isolada do código-fonte, que permite o desenvolvimento de recursos, correções de bugs ou experimentos sem afetar diretamente a versão principal do projeto.

# Conceitos:

- ❖ **Como você cria e muda para um novo branch no Git?**
  - Para criar um novo branch, você pode usar o comando **git branch nome\_da\_branch**. No entanto, para mudar para um novo branch, você pode usar o comando **git checkout nome\_da\_branch\_destino**.
- ❖ **O que é um conflito de merge no Git e como você resolve isso?**
  - Um conflito de merge ocorre quando duas branches têm alterações conflitantes no mesmo arquivo. Para resolver um conflito de merge, você precisa analisar as alterações conflitantes, decidir como deseja mesclá-las manualmente e, em seguida, commitar as alterações resolvidas.

# Conceitos:

- ❖ **Como você envia suas alterações para um repositório remoto no Git?**
  - Para enviar suas alterações para um repositório remoto no Git, você utiliza o comando **git push**. Esse comando atualiza o repositório remoto com as mudanças que você fez em seu repositório local.



# Como iniciar um projeto?

- Criar um diretório.
- `git init`
- `git add`
- `git commit -m "mensagem do commit"`
- `git branch -M master`
- `git remote add origin`  
`"https://github.com/ADRIELYNOLVAIS/CapacitacaoGit.git"`
- `git push -u origin main`

# O que são branches?

- Branches (ramificações) no Git são cópias isoladas do código-fonte de um projeto. Eles permitem que diferentes desenvolvedores trabalhem em paralelo em diferentes recursos ou correções de bugs sem interferir no trabalho um do outro. As branches são usadas para desenvolvimento separado e, quando estão prontas, podem ser mescladas (merged) de volta à branch principal do projeto, como a branch "master" (principal). Isso facilita o gerenciamento de código em equipe e o desenvolvimento de recursos de forma organizada.

# Comandos Básicos

- `git clone` ou `git init`
- `git pull`
- `git branch` `nome_da_branch`
- `git add` `nome_do_arquivo` ou `git add .` (Que irá adicionar todos os arquivos que foram alterados)
- `git commit -m "mensagem"`
- `git push`
- `git push --set-upstream` `origin` `nome_da_branch`
- `git checkout` `master`
- `git branch -d` `nome_da_branch`

# O que é o gitignore?

- O arquivo `.gitignore` é usado no Git para especificar quais arquivos ou diretórios devem ser ignorados, ou seja, não devem ser rastreados ou versionados pelo Git. Isso é útil para evitar que arquivos temporários, arquivos gerados automaticamente, senhas, chaves de API e outros dados desnecessários sejam incluídos no controle de versão.

# O que é o merge e quem fica responsável por ele?

- O "merge" é o processo de combinar alterações feitas em uma branch (ramificação) de desenvolvimento de volta à branch principal, e a pessoa responsável por ele é geralmente o desenvolvedor que trabalhou na branch de desenvolvimento ou um membro da equipe encarregado de revisar e mesclar as alterações.

# O que são tags?

- As "tags" no Git são marcadores utilizados para identificar e marcar pontos específicos na história do repositório, geralmente associados a versões estáveis ou marcos importantes no desenvolvimento do código. Elas são usadas para referenciar facilmente versões específicas do código-fonte.

# Vamos prática?

