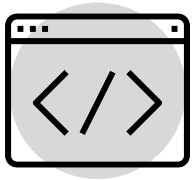
escola britânica de artes criativas & tecnologia

DE OLHO NO CÓDIGO

Fundamentos do Git



- Conheça o Git
- Crie repositórios no GIt
- Registre e altere arquivos
- Mescle arquivos

Conheça o Git

- O Git é uma ferramenta essencial para quem trabalha com análise de dados, pois facilita o controle de versões e colaboração em projetos. Ele permite que você acompanhe as alterações em seu código, dados e documentos, além de colaborar de forma eficiente com outras pessoas em um projeto.

- **Mas atenção**: evite adicionar senhas, chaves de API, informações pessoais ou outros dados sensíveis ao controle de versão. Use arquivos **'.gitignore'** para especificar quais arquivos não devem ser rastreados pelo Git.

Crie repositórios no Git

- Com o Git instalado e o repositório criado, você pode começar a rastrear arquivos específicos ao controle versão utilizando o comando '**git add**'. Por exemplo:

  git add arquivo1.csv

  Isso preparará o arquivo 'arquivo1.csv' para ser incluído no próximo commit.

- Um "commit" representa uma unidade de alteração que é registrada no histórico de um repositório de controle de versão, como um repositório Git. O commit é uma espécie de "instantâneo" das alterações feitas em um projeto em um determinado momento no tempo.

  Você pode verificar o histórico de commits no seu repositório Git usando o comando '**git log**'. Isso exibirá a lista de commits com informações como o autor, data e mensagem descritiva.

Crie repositórios no Git

- Você também pode utilizar uma *tag* para marcar pontos específicos na história do seu repositório, como versões de lançamento. As tags facilitam a identificação de versões específicas do seu projeto e são úteis para análise de dados quando você deseja reproduzir resultados ou rastrear mudanças em um ambiente controlado.

- Para criar uma tag, use o comando ‘**git tag** nomedatag’. Você também pode adicionar uma anotação mais descritiva com o comando ‘**git tag -a** nomedatag **-m** "Descrição da tag” ‘.

  Para compartilhar as tags com um repositório remoto, você deve usar o comando ‘**git push origin** nomedatag’

Registre e altere arquivos

- Evite criar um grande número de ramificações, o que pode tornar o repositório desorganizado. Siga um sistema lógico para nomear e gerenciar suas ramificações.

- Também não se esqueça de revisar seu código! Ignorar essa etapa pode levar a erros não detectados, problemas de qualidade e dificuldades de colaboração. Sempre revise as alterações de código antes de fazer o commit e incentive a revisão de código por outros membros da equipe.

Registre e altere arquivos

Veja outros comandos importantes em Linux para análise de dados:

- **git status**

Durante o processo de análise de dados, é útil verificar o estado atual do repositório Git para ver quais arquivos foram modificados, adicionados ou removidos. O comando **git status** fornece essa informação, ajudando a manter o controle sobre o progresso do projeto.

- **git checkout e git reset**

Em análises de dados, às vezes é necessário voltar a uma versão anterior dos dados ou do código. Os comandos **git checkout** e **git reset** podem ser úteis para isso, permitindo que você restaure os arquivos para um estado anterior, se necessário.

Mescle arquivos

- Depois de desenvolver uma funcionalidade em uma ramificação separada, você pode mesclar as alterações de volta para a ramificação principal usando:

  git checkout *nomedaramificação*
  git merge *nomedaramificaçãodestino*

- As alterações feitas em uma ramificação podem ser mescladas de volta ao ramo principal (ou a outra ramificação) quando o trabalho estiver concluído e revisado.

Mescle arquivos

- Antes de confirmar as alterações que você tiver realizado, verifique as diferenças entre os arquivos usando o comando baixo. Isso ajuda a minimizar erros.

  git diff script_analise_dados.py

- Você também pode resolver conflitos de mesclagem de forma mais visual e interativa usando o comando abaixo. Esse comando abre uma ferramenta com interface gráfica que permite visualizar as diferenças entre as versões do arquivo e selecionar quais alterações devem ser mantidas na versão final.

  git mergetool

escola britânica de artes criativas & tecnologia

Bons estudos!