* É onde o código será armazenado;
* Na maioria das vezes cada projeto tem um repositório;
* Quando criamos um repositório estamos iniciando um projeto;
* O repositório pode ir para servidores que são especializados em gerenciar repôs, como: GitHub e Bitbucket;
* Cada um dos desenvolvedores do time pode baixar o repositório e criar versões diferentes em sua máquina;
* Para criar um repositório utilizamos o comando: git init;
* Desta maneira o git vai criar os arquivos necessários para inicializá-lo;
* Que estão na pasta oculta .git;
* Após este comando o diretório atual será reconhecido pelo git como um projeto e responderá aos seus demais comandos;

Para acessar o Git através do VSCode, você precisará ir lá em cima em terminal e clicar na opção New Terminal  
Que irá liberar o terminal para você conseguir se comunicar com o git

Para criar o git em alguma pasta crie uma pasta onde você vai salvar o git e depois no terminal digite o caminho onde você vai querer dar o git init:

Exemplo:  
cd C:\Users\pafonso\Desktop\Curso\_prático\git\arquivo

Após o terminal ir até o caminho selecionado, de o git init para que o git seja criado naquela pasta escolhida,

Digitando git status vc ver o status do seu repositório, caso não tenha nenhum repositório criado aparecerá a seguinte mensagem  
Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Para criar um repositório utilizamos o comando git init:  
Texto

O conteúdo gerado por IA pode estar incorreto.  
  
Tanto no Git Bash quanto no vscode, quando damos um git init, o repositório vai ser criado e se vc der um git status, você vera que o “fatal:”, não aparecerá mais.

Provavelmente apareça isso:  
Texto

O conteúdo gerado por IA pode estar incorreto.  
  
essas pastas em vermelho são arquivos e pastas não rastreados pelo git, como você acabou de dar o git init, o git tá indicando tudo o que está no diretório, mas que ainda não foi adicionado ao controle de versão, não é um erro.

* É um serviço para gerenciar repositório, gratuito e amplamente utilizado;
* Podemos enviar nossos projetos para o GitHub e disponibilizá-lo para outros devs;
* O GitHub é gratuito tanto para projetos públicos como privados;
* Podemos facilmente enviar nossos repôs para o GitHub;
* Precisamos criar o projeto no GItHUb, inicializar o git em nossa máquina, sincronizar com o GH e enviar;
* E esta sequência que parece ser complexa é facilmente executada por poucos comandos;
* Vale lembrar que só fazemos uma vez por projeto este fluxo;

PASSA A PASSO

* Primeiro passo clicar, em new repositório e criar um repositório no GitHub, segue exemplo:

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

* Readme é um arquivo que explica o que vem no repositório, de tudo que ele faz, do projeto em si.
* .gitignore é um arquivo que vc adicionar para que o git não monitore arquivos que você não queira monitorar.
* A licença diz o que as pessoas podem ou não fazer com o seu código.
* Após criar o seu repositório no GH ele vai direcionar para você para um local que te mostrará o passo a passo de como seguir para fazer na sua máquina, onde você executará no terminal do seu VSCODE ou Git Bash, segue o print:  
  Interface gráfica do usuário, Texto, Aplicativo, Email

  O conteúdo gerado por IA pode estar incorreto.
* Primeiro passo criar a pasta onde que conterá o arquivo que vc deseja subir para o repositório
* Ao seguir o passo a passo de abrir um novo terminal no VS Code, caso o terminal não esteja aberto na pasta que vc deseja, basta dar o comando cd(+caminho da pasta) que vc deseja, caso ocorra do seu perfil de usuário esteja com o espaço no nome como no exemplo abaixo, você deve colocar o caminho entre aspas, desse jeito:



* Após estar na pasta, dar um git init para criar o git e dar um git status para ver se foi criada corretamente

Texto

O conteúdo gerado por IA pode estar incorreto.

* Lembrando que para subir um repositório para o GH, o repositório não pode estar vazio, então no exemplo acima criamos um arquivo de texto, chamado teste.txt dentro da pasta 1\_primeiro\_repo, ele tá em vermelho com essa informação “untracked” pq ainda não está sendo mapeado pelo git, para resolver isso basta dar um git add (nome e tipo de arquivo) conforme printe abaixo

Texto

O conteúdo gerado por IA pode estar incorreto.

* Após dar o git add, você pode verificar se o arquivo está sendo mapeado corretamente dando um git status e ele estará na cor verde, indicando que agora está sendo mapeado corretamente.
* Próximo passo, você vai dar um git commit -m “atualização”, o git commit é o comando que salva sua alteração dentro do git, para caso futuramente vc queira reutilizar, usei o nome atualização como referência para quando vc querer colocar outro nome, inicialmente colocamos git commit -m “Hello World git”, por causa da superstição dos programadores
* Pode ocorrer de dar o seguinte erro:  
  Texto

  O conteúdo gerado por IA pode estar incorreto.
* Esse erro acima indica que você ainda não configurou sua identidade, ou seja, seu nome e e-mail, que são obrigatórios pra fazer commits.

Para resolver isso é muito simples, abra o terminal e digite os dois comandos abaixo com seus dados:

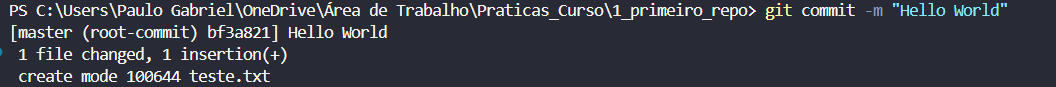
git config --global user.name "Paulo Gabriel"

git config --global user.email “[paulo@email.com](mailto:paulo@email.com)”

Segue exemplo:



Não precisa ser seu e-mail real, mas é bom usar um que você pretende usar no GitHub.

* Depois de configurar você pode dar só um git commit -m “Hello World git”, que irá aparecer da seguinte forma:  
  
* Logo em seguinda basta dar um git status para verificar se está tudo corretamente:  
  Texto

  O conteúdo gerado por IA pode estar incorreto.
* Agora precisaremos criar uma Branch master, uma Branch é uma versão parelela do seu projeto que te permite fazer, alterações, correções ou testes e não interfiram no seu código principal, e a Branch master é a padrão que o Git cria quando você inicializa um repositório, é como a trilha oficial do projeto onde geralmente ficam as versões estáveis e prontas para produção. O comando para criar a Branch é git branch -M master
* Próximo passo é o comando git remote add origin <https://github.com/PauloGAfonso/curso.git_1.git>, ele conecta seu repositório local ao repositório remoto no GitHub, o nome Origin é o apelido padrão para esse repositório remoto.
* E por fim git push-u origin master, que envia seu código (faz o push) seu código local para o GitHub, na Branch master. O -u configura essa Branch para acompanhar a origin/master, ou seja, facilita futuros git push e git pull, como saber se funcionou corretamente?
* Segue comprovação do funcionamento:  
  Interface gráfica do usuário, Texto

  O conteúdo gerado por IA pode estar incorreto.
* Pronto, você subiu um repositório para o github
* As mudanças do projeto podem ser verificadas por: git status
* Este comando é utilizado muito frequentemente;
* Aqui serão mapeadas todas as alterações do projeto;
* Como: arquivos não monitorados e arquivos modificados;
* Podemos também dizer que é a diferença do que já está enviado ao servidor ou salvo no projeto;
* Caso vc adicione novos arquivos ao seu projeto, quando vc der um git status, ele aparecerá como um arquivo não mapeado (untracked), enquanto