



FLUXO DE TRABALHO EM GRUPO

- CRIAR SEU REACT APP

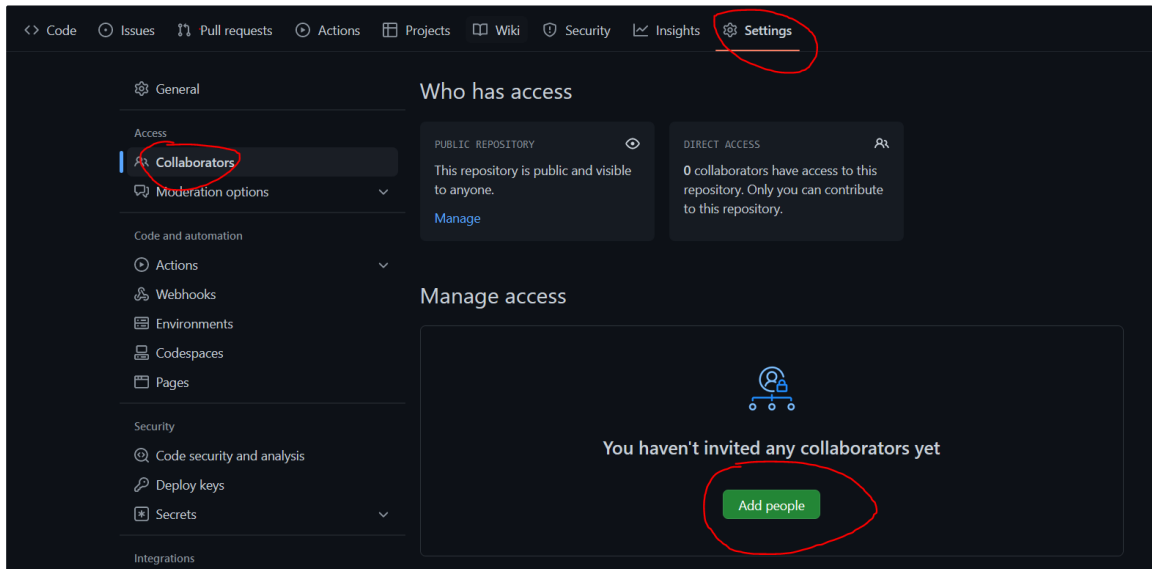
```
npx create-react-app projeto2
```

- INSTALE AS BIBLIOTECAS DE ESCOLHA

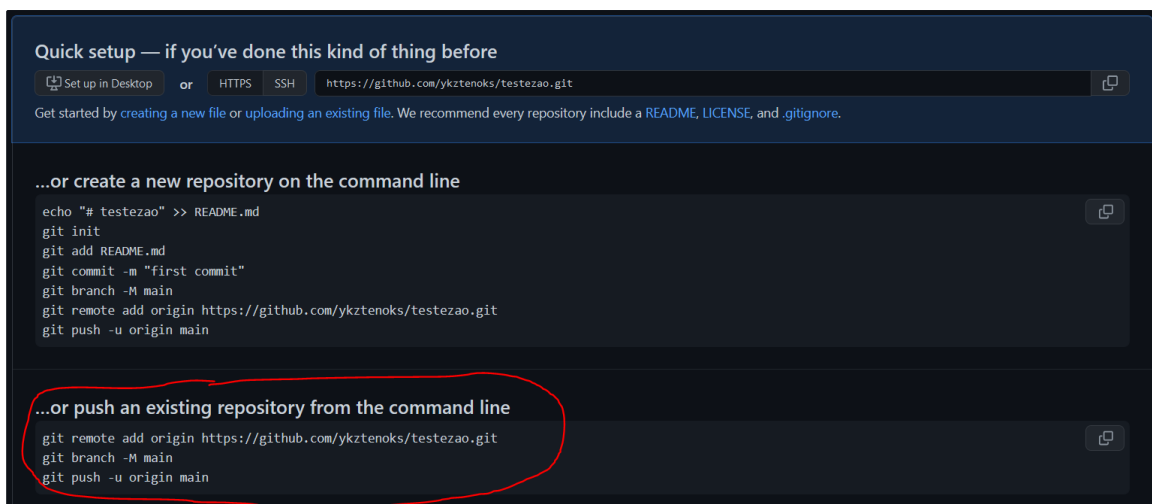
```
//EXEMPLO (podem utilizar outras bibliotecas também) npm install axios  
//obrigatório npm install react-router-dom //obrigatório npm install  
bootstrap react-bootstrap //opcional
```

- Recomendações de bibliotecas (uso opcional):
 - date-fns <https://date-fns.org/docs/Getting-Started>
 - biblioteca para manipulação de DATAS
 - react-hot-toast <https://react-hot-toast.com/>
 - biblioteca de notificações animadas
- ESTRUTURE O SEU PROJETO!
 - Converse com sua dupla e estruture os possíveis componentes e páginas que estarão utilizando.
 - Dividam-se para cada pessoa trabalhar em um arquivo/componente

- SUBA O PROJETO PARA UM NOVO REPOSITÓRIO NO GITHUB!
 - Crie um novo repositório no GitHub para subir o seu projeto já estruturado.
 - Adicione seu colega como colaborador do repositório:

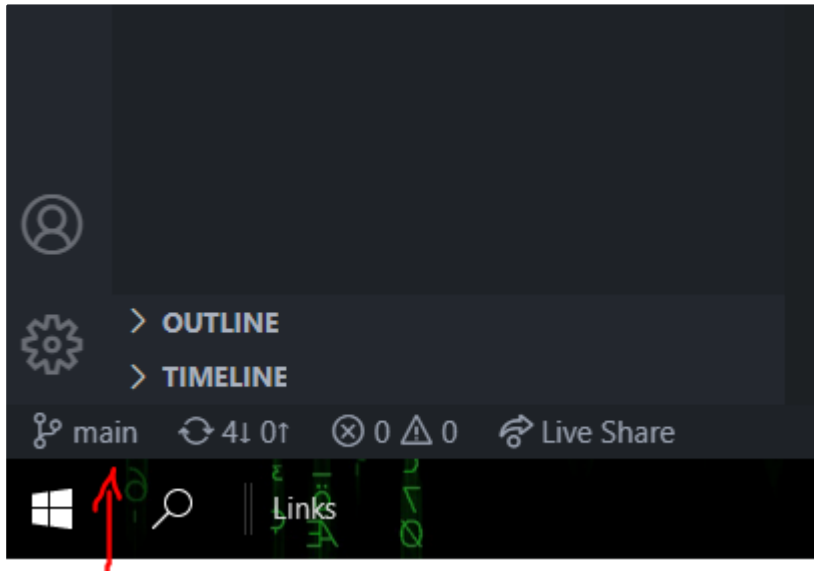


- O convidado irá receber por meio de notificação no GitHub e por **EMAIL** o convite para ser aceito.
- **SIGA O PASSO A PASSO DO GITHUB PARA LINKAR SEU REPOSITÓRIO LOCAL/PROJETO CRIADO**



- o create-react-app já adiciona uma pasta .git no nosso projeto, portanto **não** precisamos fazer o comando git init, sendo assim segue os seguintes comandos:
 - git remote add origin link-aqui
 - git add .
 - git commit -m "mensagem de commit"

- `git push origin main/master` (opcional, por padrão será ou **main** ou **master**, cheque no canto inferior esquerdo do VSCode)



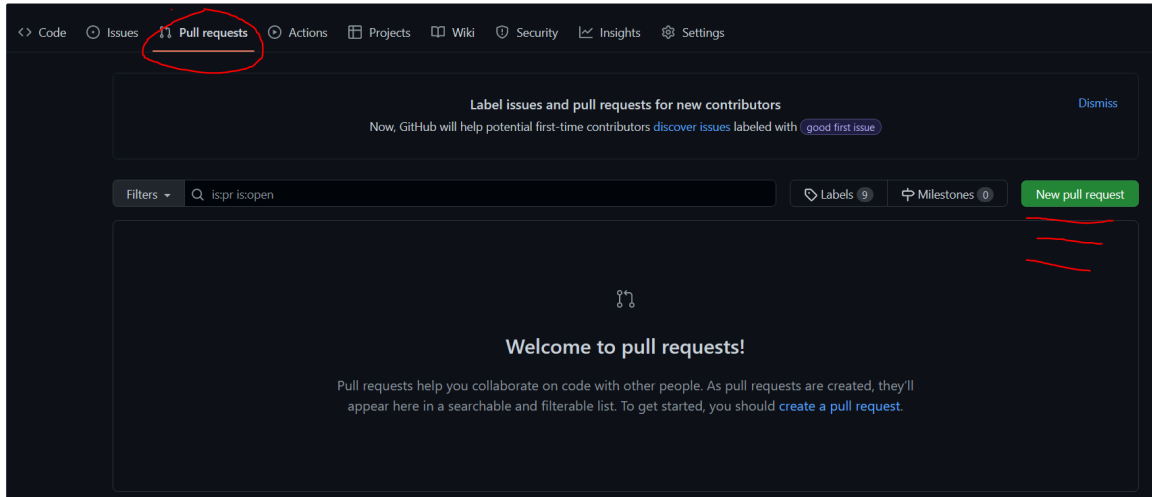
- Após esse passo a passo você tem seu projeto estruturado no computador já linkado com o GitHub!
- **CRIE UMA BRANCH PARA CADA PARTICIPANTE!**
 - Não aconselhamos trabalhar na branch main, trabalhe na sua branch e faça o pull request e merge quando necessário, instruções a seguir.

```
git branch nome-da-branch //cria uma branch git checkout nome-da-branch
//muda de branch OU git checkout -b nome-da-branch //cria e muda de
branch
```

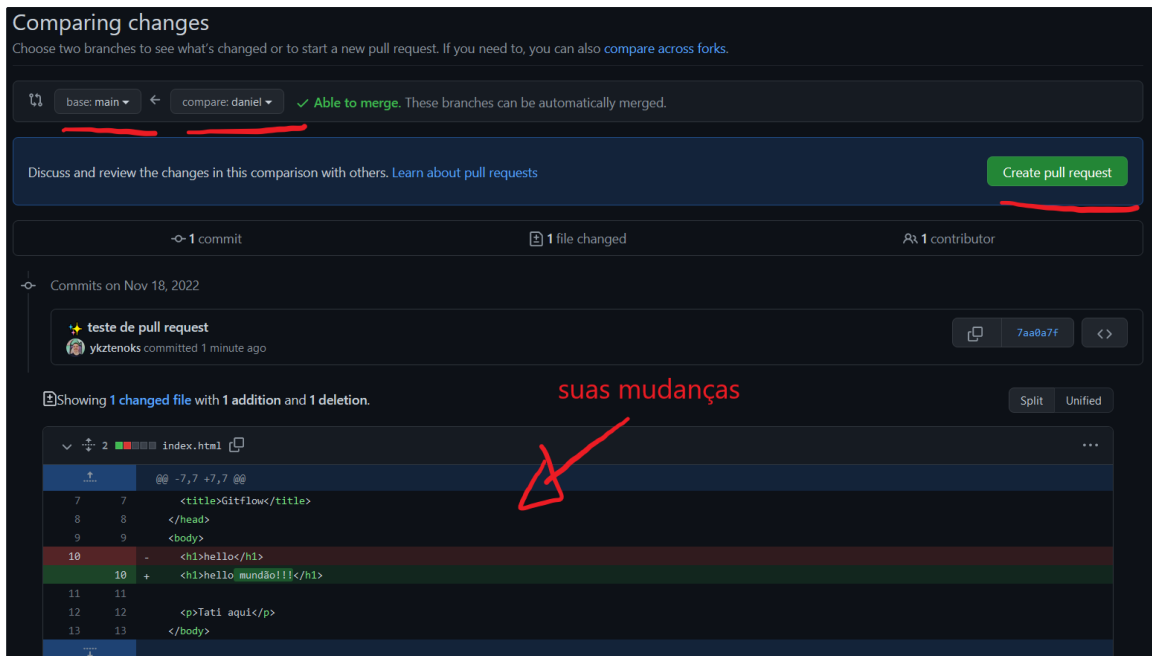
- **COMMIT, COMMIT, COMMIT!!!**
 - A cada mudança importante **faça um commit!**
 - O commit não irá para o GitHub caso não seja feito o push
- **PUSH!!!**
 - Terminou um componente? Converse com seu colega para atualizarem a main e suas branches '*pessoais*'
 - `git push origin minha-branch`
 - Fazendo o push sua branch irá ser atualizada! Segue para o pull request...

- PULL REQUEST!!

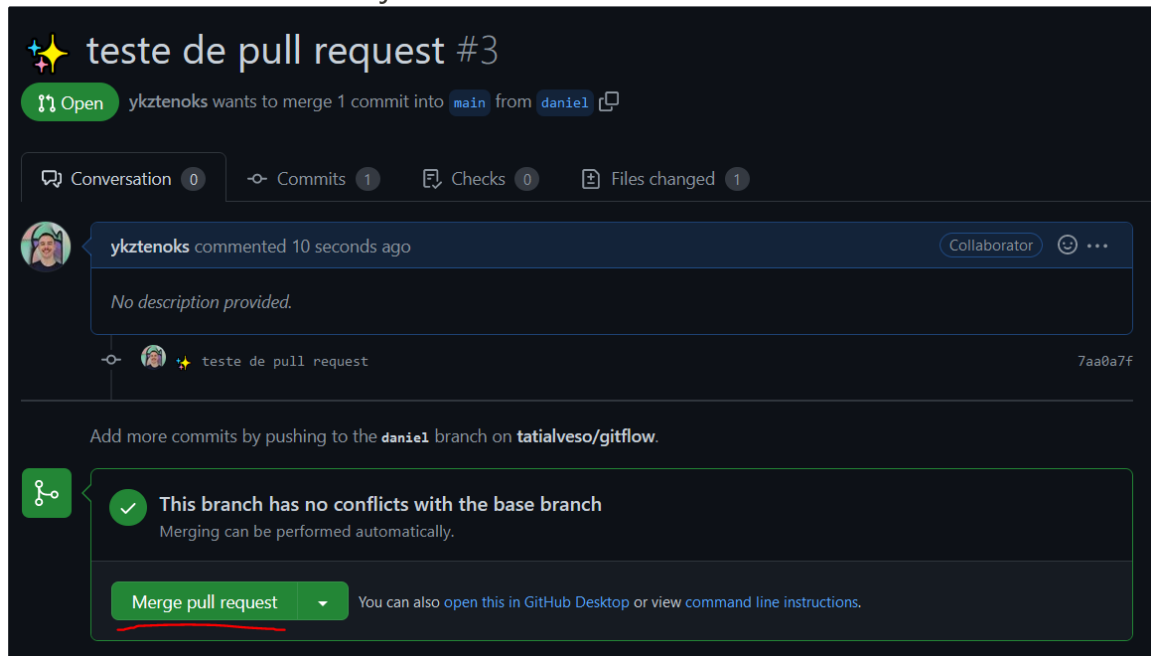
- Após atualizar sua branch é hora de atualizar a sua branch principal! (main/master)
- Vá para o GitHub e crie um pull request



- Selecione a branch que quer comparar com a main! (geralmente está selecionado automaticamente)
- Crie o pull request!



- Seu colega que **deverá aceitar (merge pull request)** o seu pull request, se há conflitos, resolvam juntos.



- Após fazer o merge do seu pull request, a branch principal (**main/master**) será atualizada!
- Nesse momento **AMBOS DEVEM ATUALIZAR SUAS BRANCHS!!!**
 - `git pull origin main` (no terminal do VSCode)
- **COMUNICAÇÃO É A CHAVE!**
 - Se comuniquem sempre sobre essas atualizações e mantenham tanto repositório local (no seu computador) quanto repositório na nuvem (GitHub) atualizados.
 - Sigam esses passos e mantenham o fluxo organizado para que evitem conflitos e possíveis problemas de integração!

