

Téc em Desenvolvimento
de Sistemas Bilíngue

Git e Github

Pull Requests

Pull Request (PR) é uma ferramenta usada para notificar os mantenedores de um repositório sobre mudanças que você fez em seu branch.

É uma forma de pedir que essas mudanças sejam revisadas e potencialmente integradas ao branch principal do projeto.

Pull Requests

Quando você cria um Pull Request, está basicamente dizendo: "Eu fiz essas mudanças no código e acho que elas devem ser incluídas no projeto principal. Alguém pode revisar e aprovar?"

Pull Requests

Colaboração: Facilita a colaboração entre desenvolvedores, permitindo que várias pessoas trabalhem simultaneamente no mesmo projeto sem conflitos.

Pull Requests

Revisão de Código: Permite que o código seja revisado antes de ser mesclado ao projeto principal, melhorando a qualidade do código.

Pull Requests

Histórico de Mudanças: Mantém um histórico claro e auditável das mudanças propostas e implementadas, permitindo rastrear o que foi alterado, por quem e por quê.

Téc em Desenvolvimento
de Sistemas Bilíngue

Criando Pull Requests

Pull Requests

Etapas de um Pull Request

1 - Criação da branch

Desenvolvedores criam um branch para trabalhar em uma nova funcionalidade ou correção de bug.

Exemplo: `git checkout -b minha-feature`.

Lembrando que branch é uma "ramificação" separada do projeto principal onde você pode fazer alterações sem afetar o código original.

Pull Requests

Implementação e Commit:

Realizam mudanças no código e fazem commit dessas mudanças.

Exemplo: `git add .` e `git commit -m "Adiciona nova funcionalidade X"`.

Lembrando que commit é como salvar o estado atual do seu trabalho com uma mensagem explicativa.

Pull Requests

Push para o Repositório Remoto:

Enviam o branch para o repositório remoto.

Exemplo: `git push origin minha-feature`.

Isso torna suas mudanças acessíveis a outros colaboradores através do repositório hospedado no GitHub.

Pull Requests

Criação do Pull Request:

No GitHub, navegam até o repositório e criam um novo Pull Request a partir do branch recém-criado.

Aqui você descreve as mudanças, explica o que foi feito e por quê, e pede para que outros revisem o seu código.

Pull Requests

Revisão e Comentários:

Outros desenvolvedores revisam o Pull Request, adicionam comentários e sugerem mudanças, se necessário.

Isso ajuda a garantir que o código adicionado é de alta qualidade e não introduz problemas.

Pull Requests

Aprovação e Fusão:

Após aprovação, o Pull Request é mesclado ao branch principal (geralmente main ou master).

Exemplo: `git merge minha-feature` no repositório remoto.

Fusão é o processo de integrar as mudanças do seu branch ao branch principal, consolidando as contribuições.

Pull Requests

Estrutura e Clareza:

- **Título Descritivo:** Use títulos claros e descritivos.
- **Descrição Detalhada:** Forneça uma descrição detalhada das mudanças, incluindo o motivo e o impacto.

Comunicação Efetiva:

- **Comentários Construtivos:** Ao revisar, seja construtivo e específico.
- **Responda aos Comentários:** Mantenha a comunicação aberta e responda aos comentários feitos no Pull Request.

Téc em Desenvolvimento
de Sistemas Bilingue

Fork

Fork

Fork é uma cópia de um repositório existente no GitHub.

Quando você faz um fork de um repositório, você cria uma cópia independente desse repositório em sua própria conta.

O propósito principal do fork é permitir que você trabalhe em um projeto sem afetar o repositório original.

Isso é particularmente útil para contribuir para projetos de código aberto.

Fork

Desenvolvimento Isolado: Permite que você experimente e faça mudanças sem impactar o repositório original.

Fork

Contribuição para Projetos de Código Aberto: Facilita a contribuição para projetos de outros desenvolvedores. Você pode fazer mudanças no seu fork e, depois, propor essas mudanças para o repositório original através de um Pull Request.

Fork

Colaboração: Simplifica a colaboração em projetos de grandes equipes ou projetos públicos, permitindo que múltiplas pessoas trabalhem simultaneamente em diferentes partes do projeto.

Téc em Desenvolvimento
de Sistemas Bilingue

Etapas do Fork

Fork

Fazer o Fork de um Repositório:

- No GitHub, navegue até o repositório que você deseja forkar.
- Clique no botão "Fork" no canto superior direito da página.
- Isso criará uma cópia do repositório em sua própria conta GitHub.

Fork

Clonar o Fork para sua Máquina Local:

- `git clone https://github.com/seu-usuario/seu-fork.git`
- Isso cria uma cópia local do repositório forkado para que você possa trabalhar nele.

Fork

Fazer Mudanças no Código:

- Crie um novo branch para suas mudanças: `git checkout -b minha-feature`
- Faça as alterações desejadas no código.

Fork

Commit e Push:

- `git add .`
- `git commit -m "Adiciona nova funcionalidade"`
- `git push origin minha-feature`

Fork

Criar um Pull Request:

- No GitHub, navegue até o seu fork e clique em "Pull Requests".
- Clique em "New Pull Request" e siga as instruções para submeter suas mudanças para o repositório original.

Fork

Diferença entre Fork e Clone:

- Fork: Cria uma cópia do repositório no GitHub sob sua própria conta. É uma operação feita na plataforma GitHub.
- Clone: Cria uma cópia do repositório em sua máquina local. É uma operação feita via Git, e você pode clonar tanto repositórios originais quanto forks.

Fork

Manter o Fork Sincronizado:

- Adicionar Repositório Upstream: `git remote add upstream https://github.com/original-usuario/original-repositorio.git`
- Buscar Mudanças do Upstream: `git fetch upstream`
- Mesclar Mudanças no Fork: `git merge upstream/main`

Fork

Contribuição Efetiva:

- **Comunicação:** Seja claro na descrição das mudanças feitas ao criar um Pull Request.
- **Pequenas Mudanças:** Tente fazer Pull Requests com mudanças pequenas e concisas para facilitar a revisão.
- **Teste:** Teste suas mudanças antes de submeter um Pull Request para garantir que elas não quebrem o código existente.

Fork

Upstream refere-se ao repositório original do qual você fez um fork.

Quando você faz um fork de um repositório, seu repositório forkado não recebe automaticamente as atualizações feitas no repositório original.

Configurar um upstream permite que você sincronize seu fork com as mudanças mais recentes do repositório original.

É essencial para manter seu fork atualizado com o trabalho contínuo no projeto principal.

Fork

Importância e Benefícios:

- **Manter Atualizado:** Ajuda a garantir que seu fork esteja atualizado com as últimas mudanças e melhorias do repositório original.
- **Facilitar Contribuições:** Reduz a probabilidade de conflitos ao enviar Pull Requests, pois seu fork estará alinhado com o repositório principal.
- **Colaboração Eficiente:** Permite que você acompanhe o progresso do projeto e integre mudanças relevantes em seu próprio trabalho.

Fork

Etapas para Configurar Upstream:

1. Adicionar o Repositório Upstream:

- Vá até o diretório do seu repositório forkado.
- Adicione o repositório original como um repositório remoto chamado upstream:

```
git remote add upstream https://github.com/usuario-origem/repositorio-origem.git
```


Fork

Etapas para Configurar Upstream:

Adicionar o Repositório Upstream:

- Vá até o diretório do seu repositório forkado.
- Adicione o repositório original como um repositório remoto chamado upstream:

```
git remote add upstream https://github.com/usuario-origem/repositorio-origem.git
```

- Verifique se o novo remoto foi adicionado corretamente:

```
git remote -v
```


Fork

Buscar Atualizações do Upstream:

- Busque as mudanças do repositório upstream:

`git fetch upstream`

- Mesclar Mudanças no Branch Principal:

`git checkout main`

`git merge upstream/main`