



# GitHub: Colaboração Profissional

Bem-vindos! Esta apresentação é o seu guia prático para dominar a colaboração no GitHub, uma habilidade essencial para qualquer desenvolvedor moderno.

Prepare-se para transformar a maneira como você interage com projetos e equipes.

## Agenda da Colaboração

# O Que Vamos Aprender

01

### O Workflow do Pull Request

Dominando o ciclo completo: do fork à fusão.

02

### Boas Práticas de Colaboração

Dicas para uma interação fluida e profissional em projetos.

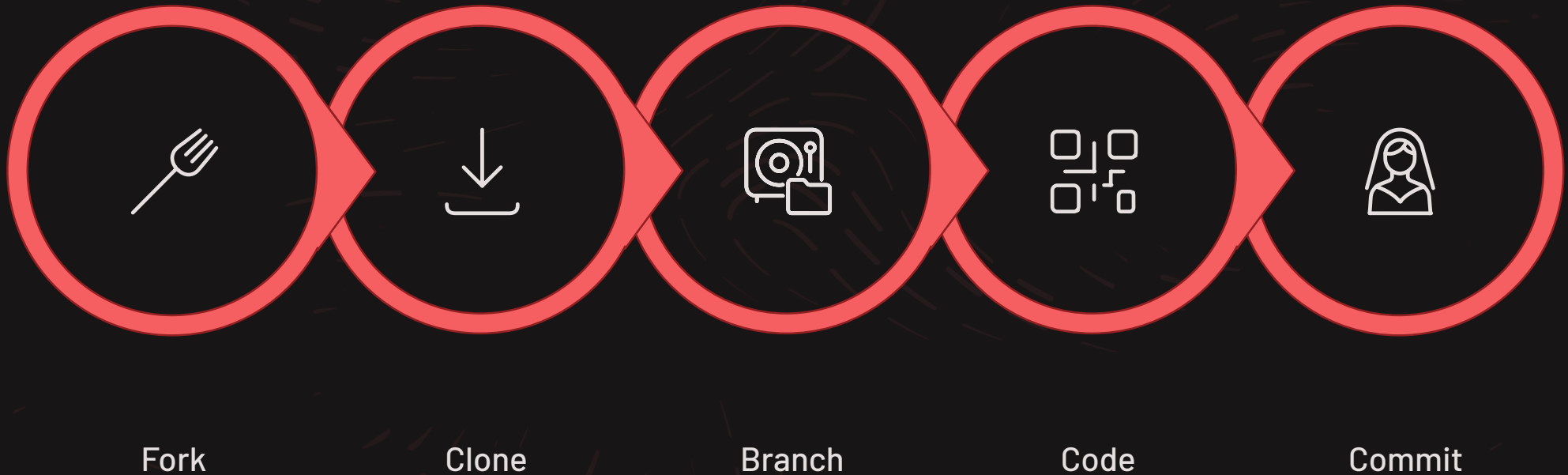
03

### Seu Perfil GitHub de Destaque

Construindo um README de perfil que impressiona.

# Dominando o Pull Request (PR)

O Pull Request é o coração da colaboração no GitHub. Ele permite que você proponha alterações em um projeto, que podem ser revisadas e incorporadas.



# Workflow de um Pull Request Completo



## 1. Fork

Crie uma cópia do repositório para sua conta.

```
git clone [URL do seu fork]
```



## 2. Clone & Branch

Baixe para sua máquina e crie uma nova branch para suas mudanças.

```
git checkout -b minha-nova-feature
```



## 3. Code & Commit

Escreva seu código e salve as alterações localmente.

```
git add .  
git commit -m "feat: minha nova feature"
```



## 4. Push

Envie sua branch para o seu fork no GitHub.

```
git push origin minha-nova-feature
```



## 5. Pull Request

Abra um PR do seu fork para o repositório original no GitHub.

# Template de PR Profissional

Um bom template de Pull Request agiliza o processo de revisão e demonstra profissionalismo. Inclua um checklist e seções claras.

## Exemplo Essencial:

### ### Descrição das Mudanças

Adiciona nova funcionalidade X para resolver Y.

### ### Tipo de Mudança

- ☐ Nova feature
- ☐ Correção de bug
- ☐ Melhoria

### ### Checklist

- ☐ Código testado localmente
- ☐ Documentação atualizada (se necessário)
- ☐ Testes automatizados (se aplicável)
- ☐ Sem erros de lint

### ### Como Testar

1. Clone o projeto.
2. Rode `npm install`.
3. ...



Sempre preencha seu template de PR com detalhes claros e concisos. Isso facilita a vida dos revisores e acelera a fusão do seu código.

# Boas Práticas de Colaboração

Colaborar efetivamente vai além dos comandos: é sobre comunicação e respeito ao fluxo de trabalho do projeto.

## ✓ O Que Fazer

- **Mantenha seu fork atualizado:** Faça `git fetch upstream` e `git merge upstream/main` regularmente.
- **Crie branches específicas:** Uma branch por feature/bug.
- **Commits claros e atômicos:** Cada commit uma mudança lógica.
- **Revise seu próprio código:** Antes de pedir revisão.
- **Seja respeitoso e construtivo:** Ao dar e receber feedback.

## ✗ O Que Evitar

- **Branches longas ou "gigantes":** Dificultam revisões e fusões.
- **Commits genéricos:** "Correções" ou "mudanças".
- **Ignorar templates de PR:** Diminui a eficiência do time.
- **Empurrar diretamente para `main`:** Quebra o fluxo de revisão.
- **Conflitos não resolvidos:** Responsabilidade de quem abre o PR.



# Resolvendo Conflitos Como um Pro

Conflitos de fusão são inevitáveis, mas você pode resolvê-los com confiança.

1

## 1. Puxe as Últimas Mudanças

Antes de iniciar um PR, puxe as últimas alterações da branch principal do repositório original para sua branch local.

```
git pull origin main
```

2

## 2. Identifique os Conflitos

O Git irá marcar os arquivos com conflito com <<<<<<, ===== e >>>>>>.

3

## 3. Edite e Resolva

Manualmente edite o arquivo, escolhendo as linhas que deseja manter de ambos os lados.

4

## 4. Marque como Resolvido

Após editar, use `git add [nome do arquivo]` para marcar o arquivo como resolvido.

5

## 5. Commit e Envie

Faça um commit com a resolução do conflito e envie para sua branch.

```
git commit -m "feat: resolve conflito do PR  
#123"  
git push origin minha-feature
```

# Seu Perfil GitHub de Destaque

Seu perfil GitHub é seu cartão de visitas no mundo do desenvolvimento. O README de perfil é a primeira impressão!



- **README Personalizado:** Crie um repositório com o mesmo nome do seu usuário GitHub. O README.md nesse repositório será seu perfil!
- **Templates:** Use modelos minimalistas, dinâmicos ou profissionais para começar.
- **GitHub Stats:** Mostre suas contribuições e atividades de forma visual e automática.
- **Badges de Tecnologia:** Destaque as linguagens e frameworks que você domina.
- **Contador de Visitas:** Um toque divertido para ver quem visitou seu perfil.
- **Activity Graph:** Uma visão geral das suas contribuições ao longo do tempo.



# Elementos Essenciais para o README de Perfil

Personalize seu README para refletir suas habilidades, projetos e paixões.

```
# Olá, sou [Seu Nome]!
```

# Próximos Passos

## 1. Pratique o Workflow

Crie um fork de um projeto open source e tente abrir seu primeiro Pull Request. Comece pequeno!

## 2. Otimize seu Perfil

Dê um "upgrade" no seu README de perfil. Use as ferramentas e ideias que vimos para deixá-lo profissional.

## 3. Contribua Ativamente

Busque projetos que te interessam no GitHub e comece a contribuir, mesmo que com pequenas correções ou melhorias na documentação.

Obrigado! Mantenham-se conectados e colaborem muito!