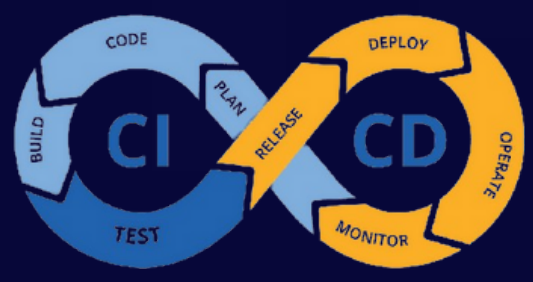
o mínimo que vc precisa saber sobre versionamento

git flow

conventional commits



pipeline

Isaac Gomes

Introdução

quando trabalhamos na construção de **software** padrões são necessários incluindo para o **versionamento**

padrões como:

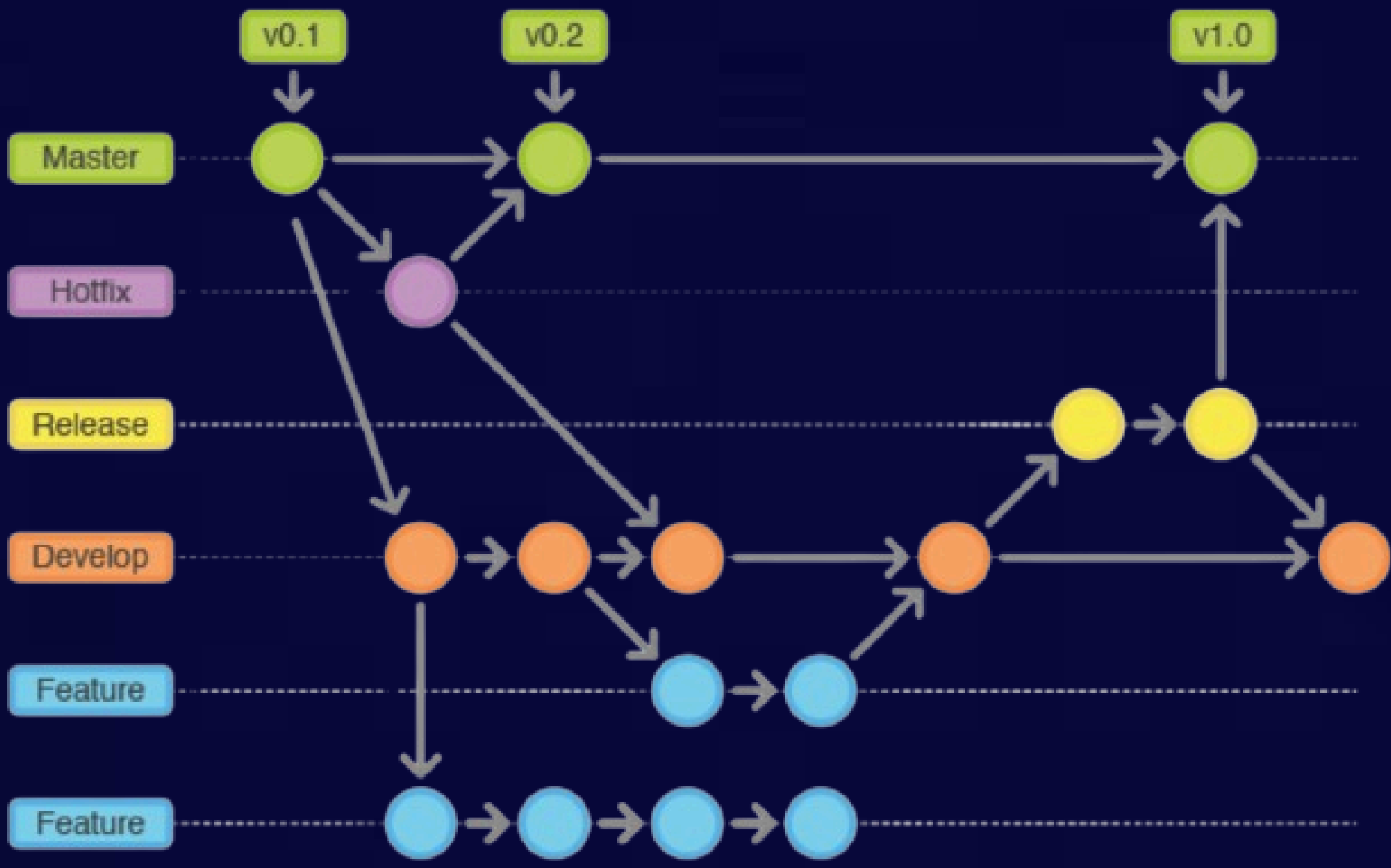
git flow: padrão de controle de versão

conventional commits: commit semântico

Isaac Gomes

git flow

usando git flow o controle de versão fica:



cada branch sendo responsável por:

Isaac Gomes

git flow

MASTER

Esta é a branch principal do repositório. É onde o código em produção é mantido. Quando uma nova versão do software é considerada estável e pronta para implantação, ela é mesclada na branch master.

Isaac Gomes

git flow

DEVELOP

Esta é outra branch principal, onde o código em desenvolvimento é integrado continuamente. É a partir desta branch que novas funcionalidades são desenvolvidas. É uma branch mais instável que a master, já que é onde as novas funcionalidades são introduzidas e testadas.

Isaac Gomes

git flow

FEATURE

Cada **nova funcionalidade** ou conjunto de alterações é desenvolvido em sua própria **branch de feature**. Isso mantém o trabalho isolado e permite que **múltiplas funcionalidades** sejam desenvolvidas ao mesmo tempo sem interferências. Após a conclusão do desenvolvimento da funcionalidade, ela é mesclada de volta na branch **develop**

Isaac Gomes

git flow

RELEASE

Quando o código na branch develop atinge um ponto onde está pronto para ser lançado, uma branch de release é criada a partir dela. Isso permite preparar o código para o lançamento final, como realizar testes finais, ajustes de última hora e preparação da documentação. Uma vez que a release esteja pronta, ela é mesclada tanto na branch master quanto na develop, marcada com uma tag de versão e, opcionalmente, excluída.

Isaac Gomes

git flow

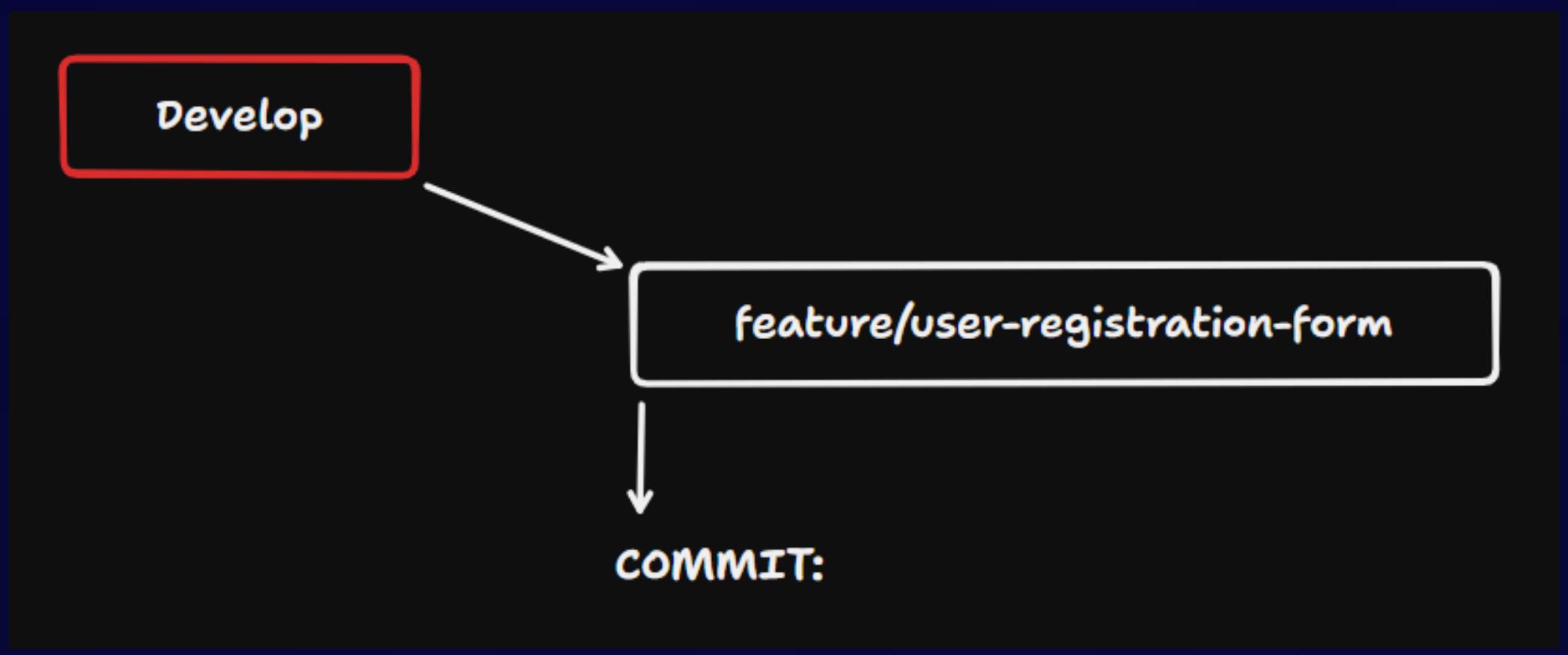
HOTFIX

Se surgirem problemas críticos no código em produção, uma branch de hotfix é criada a partir da branch master. Isso permite corrigir os problemas rapidamente e implantá-los sem afetar o desenvolvimento contínuo na branch develop. Após a correção do problema, a branch de hotfix é mesclada tanto na branch master quanto na develop, para garantir que as correções também estejam presentes no próximo lançamento.

Isaac Gomes

conventional commits

agora que ja entendemos o fluxo de versão... vamos supor que vamos realizar a criação de um form de "registro de usuário" como podemos introduzir nossas alterações?



faremos a introdução das modificações com base no conventional commits sendo ele:

Isaac Gomes

conventional commits

PADRÃO

<tipo>[âmbito opcional]: <descrição>

Isaac Gomes

conventional commits - tipos

tipo: **feat**

Usado para indicar a adição de uma **nova funcionalidade** ao código.

tipo: **fix**

Utilizado para indicar a **correção** de um **bug** ou problema existente no código.

Isaac Gomes

conventional commits - tipos

tipo: docs

Indica alterações na documentação, como atualizações de README, documentação de código ou comentários.

tipo: style

Utilizado para alterações que não afetam o comportamento do código, como formatação, organização de código, espaçamento, entre outros.

Isaac Gomes

conventional commits - tipos

tipo: refactor

Indica uma modificação no código que não corrige um bug nem adiciona uma nova funcionalidade, mas melhora a estrutura ou o desempenho do código.

tipo: test

Utilizado para adicionar novos testes ou melhorar os testes existentes.

Isaac Gomes

conventional commits - tipos

tipo: chore

Indica alterações relacionadas a tarefas de manutenção, build, configuração ou outras tarefas não relacionadas diretamente ao código em si.

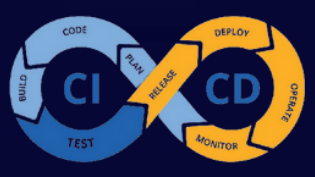
tipo: perf

Utilizado para indicar alterações de código que melhoram o desempenho do sistema.

tipo: revert

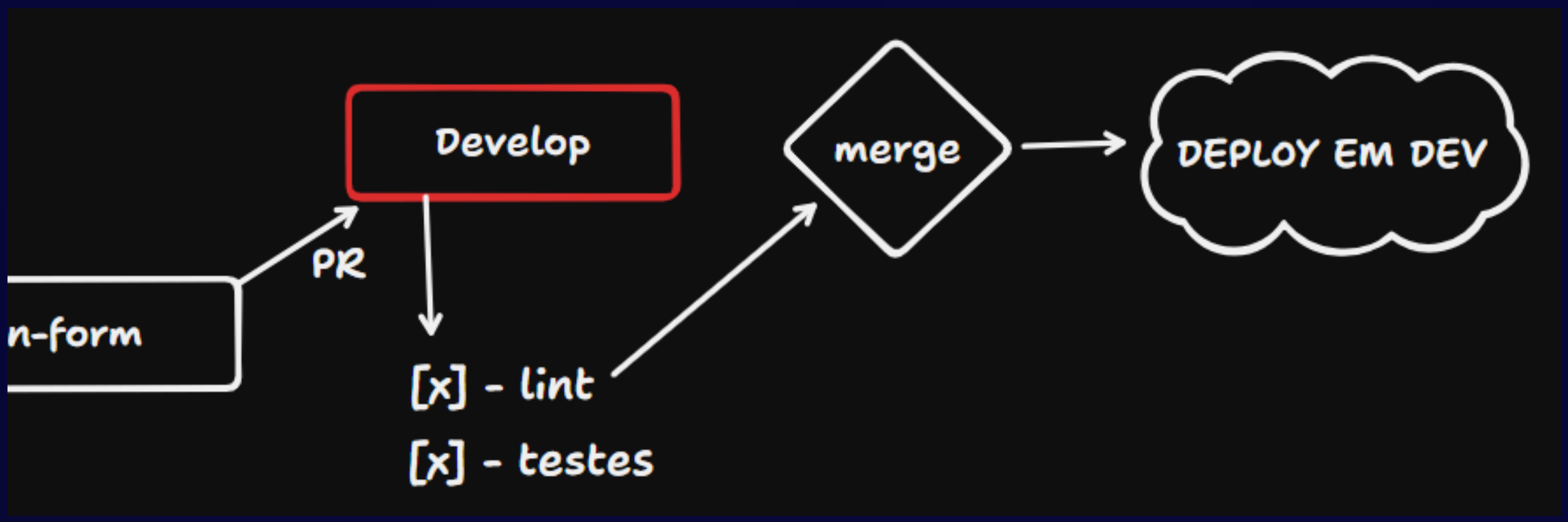
Indica que o commit está revertendo uma alteração anterior.

Isaac Gomes



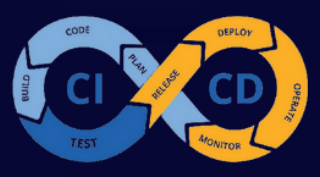
CI/CD - pipeline

agora que ja criamos nossa branch e ja adicionamos nossos commits...
podemos abrir o nosso PR



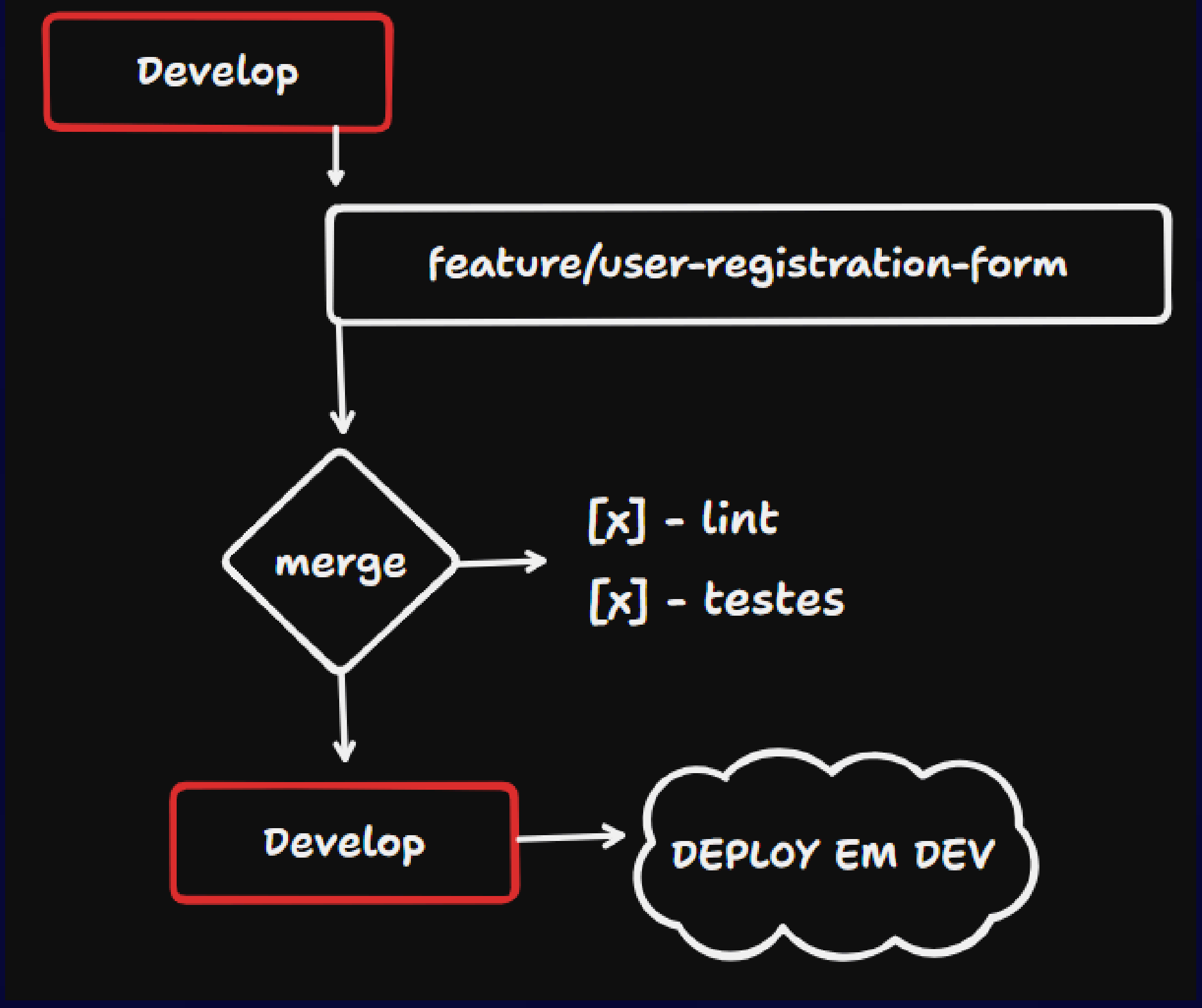
agora ao mesclar nossas alterações ira verificar o lint, testes e passando realizar o deploy para o ambiente dev

Isaac Gomes

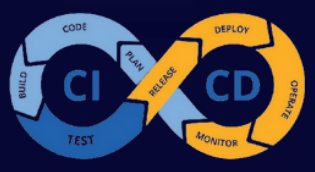


CI/CD - pipeline

veja com isso temos uma noção boa de como seria o fluxo de criação de uma feature... mesmo que de maneira simplificada



Isaac Gomes



CI/CD - pipeline

com isso estabelecemos:

- um padrão de controle de versão
- um padrão de commit
- entrega automatizada em dev

Isaac Gomes

git flow

lembrando:

git flow é um dos fluxos possíveis...
existem outros como githubFlow,
GitLab Flow, eature Branch
Workflow, Forking Workflow...

Isaac Gomes

conventional commits

lembrando:

para aplicar o conventional commits podemos aplicar ferramentas que verifiquem se seu commit segue o padrão duas delas são:

Husky e lefthook

Isaac Gomes

conventional commits

ambos servem para a mesma função, porem, o lefthook tem um diferencial que é não depender do JS, permitindo usar ele em projetos de outras techs

exemplo

With **Go** (>= 1.21):

```
go install github.com/evilmartians/lefthook@latest
```

With **NPM**:

```
npm install lefthook --save-dev
```

With **Ruby**:

```
gem install lefthook
```

Isaac Gomes

Gostou?

Curta

Compartilhe

Salve

Isaac Gomes