Git Flow

Objetivo

Este documento descreve o Git Flow do projeto:

- Nomenclatura padronizada das branches;
- Definição do propósito de cada branch e em que momento devem ser utilizadas;
- Regras de funcionamento do processo de merge, incluindo quem pode realizar e os critérios de aprovação;
- Fluxo de Pull Requests (PRs), incluindo quem revisa, critérios de aceitação;
- Padrão de commits, incluindo convenções de escrita;

1.1 Nomenclatura Padronizada das Branches

A nomenclatura das branches é fundamental para manter o histórico organizado e facilitar a rastreabilidade.

As branches seguirão um padrão de nomenclatura:

Sprint X /Tipo de Branch/Descrição da Branch.

Exemplo: Sprint 0 Feature Desenvolvimento e Testes

1.2 Propósito e Uso das Branches

- **main**: Branch principal. Contém o código de produção, sempre estável e pronto para deploy. Somente merges de branches de develop.
- **develop**: Branch de desenvolvimento. Agrega as funcionalidades mais recentes e estáveis. É a base para as branches de feature.
- **feature**: Criada a partir de develop. Nela, o desenvolvedor implementa a nova funcionalidade. Uma vez completa, é mesclada de volta para develop via Pull Request (PR).
- **codigo-juniors**: Criada a partir de feature para isolar o trabalho dos juniores. Apenas os códigos selecionados desta branch são mesclados para a feature principal.

1.3 Regras de Merge e Aprovação

- **Proibido merge direto**: Todas as alterações devem passar por um Pull Request (PR). É estritamente proibido fazer merge diretamente nas branches main, develop e feature.
- **Revisão obrigatória**: Todo PR deve ter a aprovação do Analista de Qualidade. A quantidade de revisores pode variar conforme a complexidade da alteração.

- **Critérios de aprovação**: O PR só pode ser mesclado se todos os testes passarem. Além disso, a feature só receberá o merge dos commits da codigo-juniors do código considerado o melhor entre as propostas dos juniores.
- **Resolução de conflitos**: O Gerente de configurações e mudanças é o único que deve resolver os conflitos de merge.
- **Quem pode mesclar**: Apenas o gerente de configurações e mudanças pode mesclar a branch após a aprovação de todos os critérios.

1.4 Fluxo de Pull Requests (PRs)

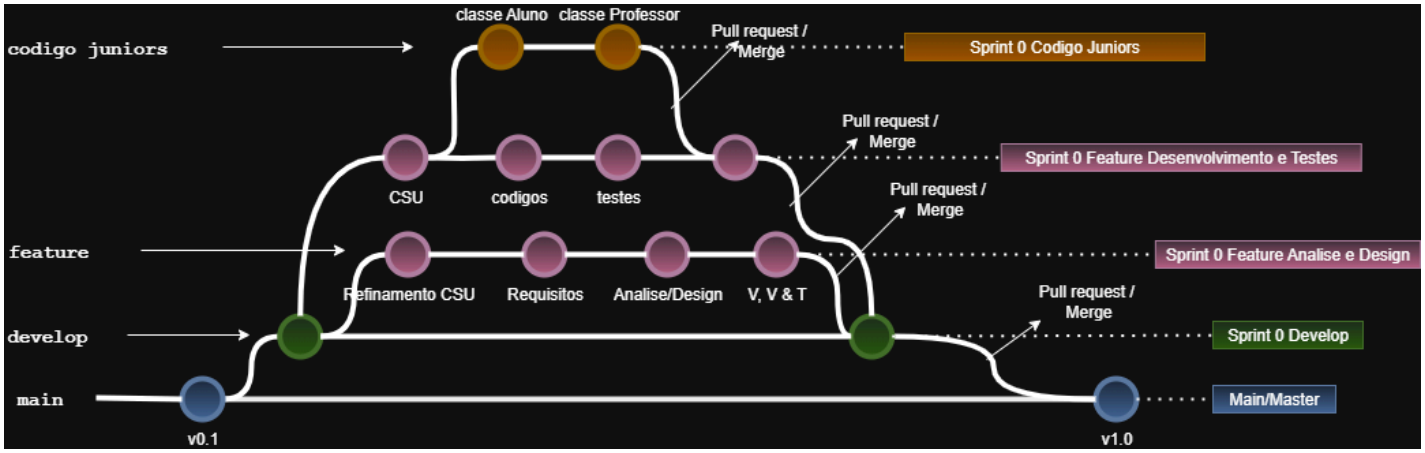
1. **Criação**: O desenvolvedor abre um PR da sua branch (Sprint-X/feature/…) para a branch de destino (main, develop ou feature).
2. **Descrição**: O PR deve ter um título claro e uma descrição detalhada. Deve incluir:
    - O problema que está sendo resolvido ou a funcionalidade implementada.
3. **Revisão**: O(s) Analista(s) designado(s) analisam o código, buscando conformidade com os padrões da equipe, possíveis bugs, e legibilidade.
4. **Aceitação**: O PR é aceito quando todos os revisores o aprovam e todos os testes são aprovados.

1.5 Padrão de Commits

Utilizamos o padrão **Conventional Commits** para padronizar as mensagens. A estrutura é: <tipo>: <descrição>

- **<tipo>**:
    - **feat**: Uma nova funcionalidade.
    - **fix**: Uma correção de bug.
    - **docs**: Alterações na documentação.
    - **style**: Formatação de código (sem alteração lógica).
    - **refactor**: Refatoração do código (sem mudança de funcionalidade).
    - **test**: Adição ou alteração de testes.
    - **chore**: Mudanças de rotina, como atualização de dependências.
- **<descrição>**: Uma breve e clara descrição da mudança.

1.6 Diagrama do Git Flow



ATENÇÃO!!

Somente o gerente de configurações e mudanças tem permissão para fazer alterações no repositório!