

---

# PROJETO PIX ANALYSER

---

## Arquitetura de Branches

Para boas práticas serão utilizados **três níveis de branches**: *main*, *homolog* e *development*. Cada uma dessas branches possui um propósito específico dentro do fluxo de desenvolvimento, garantindo organização, estabilidade e um processo estruturado de deploy.

- Na **Branch main**, só será upado assim que for validado pela equipe e garantido que não ocorra problemas.
  - Apenas código **testado e aprovado** será integrado.
  - O código presente na main deve estar **livre de erros críticos**, garantindo que a aplicação esteja pronta para uso em produção.
  - Atualizações nesta branch só ocorrem após a aprovação e testes rigorosos na homolog(hom).
- Na **Branch homolog(hom)**, será upado quando tiver funções completas, mas ainda não testadas para subir na main.
  - Serve como ambiente intermediário entre development(dev) e main.
  - Nesta branch, serão upadas funcionalidades que já foram **finalizadas**, mas que ainda precisam passar por testes antes de serem promovidas para a main.
  - Equipes de revisores podem utilizar essa branch para validar as novas implementações.
  - Após os testes e aprovações, as alterações podem ser **mescladas na main**.
- Na **Branch development(dev)**, será postado qualquer código que esteja em desenvolvimento.
  - Esta branch é utilizada para **desenvolvimento ativo**.
  - Qualquer código em progresso será postado aqui, incluindo novas funcionalidades, correções de bugs e experimentações.
  - Pode conter código instável, inacabado e sujeito a mudanças frequentes.
  - O objetivo é que cada desenvolvedor trabalhe em suas próprias branches de feature e depois faça merge para developer.

- Na **Branch feature/documentacao**, serão desenvolvidas e aprimoradas todas as melhorias do projeto.
  - Esta branch é utilizada exclusivamente para **criação, atualização e manutenção de materiais descritivos e estruturais** do sistema.
  - Qualquer alteração relacionada a esse contexto será postada aqui, incluindo:
    - **Novos conteúdos e informações essenciais** para os usuários e equipe.
    - **Melhoria na organização e estruturação** dos conteúdos existentes.
    - **Atualização de informações** conforme novas implementações forem realizadas.
    - **Correção de erros e ajustes técnicos** para garantir clareza e objetividade.
  - Pode conter **informações em andamento**, rascunhos e materiais sujeitos a revisão antes da integração com development(dev).
  - O objetivo é permitir que a equipe trabalhe nessas informações de forma independente, garantindo que, ao serem mescladas na development(dev), estejam **alinhadas com as novas funcionalidades e melhorias do projeto**.



Esse fluxo garante que a **branch main permaneça estável**, enquanto **development(dev)** serve para experimentação e **homolog(hom)** para validação.

Esse processo permite um desenvolvimento mais estruturado, reduzindo riscos e garantindo qualidade no código antes da liberação final.