

# Guia de Fluxo de Trabalho e Padrões de Engenharia

Este documento estabelece as regras para o desenvolvimento em nossos projetos. O objetivo é garantir **qualidade de código**, **rastreabilidade** e **paz de espírito** durante os deploys.

---

## 1. A Hierarquia das Branches

Ninguém — absolutamente ninguém — faz commits diretos nas branches principais. Elas são protegidas e servem como nossas "fontes da verdade".

- **main (Produção)**: Estável. Contém apenas código que já foi testado em homologação e aprovado pelo cliente.
  - **hml (Homologação)**: Ambiente de QA/Teste. Onde o cliente valida as entregas antes do deploy final.
  - **dev (Desenvolvimento)**: Nossa branch de integração. Tudo o que estamos construindo se encontra aqui. É a base para todas as novas tarefas.
- 

## 2. Ciclo de Vida de uma Task (Workflow)

Para qualquer card no Monday, o fluxo deve ser rigorosamente este:

1. **Sincronização**: Garanta que sua **dev** local está atualizada:
    - `git checkout dev && git pull origin dev`
  2. **Criação da Branch**: Crie uma branch específica para a tarefa:
    - Padrão: `feature/nome-da-task` ou `fix/nome-do-bug`
  3. **Desenvolvimento**: Codifique sua solução seguindo os padrões do projeto.
  4. **Resolução de Conflitos (Obrigatório)**: Antes de abrir o PR, você deve trazer a **dev** atualizada para a sua branch:
    - `git merge dev` (ou `git rebase dev`)
    - **Atenção**: Resolva os conflitos na sua máquina. Se o Pull Request chegar com conflitos no GitHub, ele será recusado imediatamente.
  5. **Abertura de PR**: Abra o Pull Request apontando da sua branch para a **dev**.
- 

## 3. Padrão de Commits (Conventional Commits)

Nossas mensagens de commit devem ser legíveis por humanos e máquinas. Use o prefixo que melhor descreve a alteração:

Prefixo	Descrição
<b>feat:</b>	Uma nova funcionalidade.
<b>fix:</b>	Correção de um bug.
<b>docs:</b>	Mudanças apenas na documentação.
<b>style:</b>	Mudanças de formatação (espaços, ponto e vírgula, etc).
<b>refactor:</b>	Mudança no código que não corrige bug nem adiciona funcionalidade.
<b>test:</b>	Adição ou correção de testes.

**Exemplo:** feat: adiciona máscara de CNPJ no formulário de cadastro



## 4. Protocolo de Pull Request (PR)

O PR é o momento da revisão por pares. Para que seu código seja aprovado, ele deve:

- **Ter Título Claro:** Referencie o objetivo da tarefa.
- **Link da Task:** Cole o link do card do Monday na descrição.
- **Evidências:** Se houve mudança visual, anexe um print ou um GIF do funcionamento.
- **Limpeza:** Remova `console.log`, comentários de "todo" ou códigos comentados antes de enviar.
- **Checklist de Prontidão:**
  1. O build passa localmente?
  2. O código segue o padrão do projeto?
  3. Conflitos foram resolvidos?



### Dicas para o Time:

- **Commits Atômicos:** Faça commits pequenos e frequentes. É muito mais fácil resolver conflitos em 10 linhas do que em 500.
- **Fale Ceddo:** Se travou em um conflito ou em uma lógica, não espere o fim do dia. Chame no canal!

