

# Processo de Rastreabilidade no Desenvolvimento

Este documento descreve o Fluxo de Desenvolvimento da equipe CTRL ALT DEL, um processo de rastreabilidade para o desenvolvimento de software. Ele detalha as etapas, objetivos, responsáveis, insumos, entregáveis e boas práticas desde a definição dos requisitos até a criação de branches, commits e pull requests, garantindo a conexão entre cada fase do desenvolvimento.

## 1. Definição dos Requisitos

**Objetivo:** Especificar as funcionalidades que o sistema precisa ter

**Responsáveis:** Product Owner, Dev Team

**Insumos:**

- Reuniões com equipe
- Reuniões com cliente
- Documentação base fornecida pelo cliente
- Proposta de solução aprovada pelo cliente

**Entregáveis:**

- Lista de Requisitos Funcionais do produto.
- Lista de Requisitos não Funcionais do produto.

**Boas práticas:**

- Documentar de forma objetiva em local de fácil acesso.
- Validar com cliente.

## 2. Backlog do Produto

**Objetivo:** Extrair User Stories dos Requisitos definidos anteriormente, e alocá-los em um Backlog

**Responsáveis:** Product Owner.

**Insumos:**

- Lista de Requisitos validados com o cliente.

**Entregáveis:**

- Backlog com as User Stories iniciais, definidas em prioridade (Alta, Média, Baixa)
- Epics das funcionalidades relacionadas às US.

**Boas práticas:**

- US devem sempre seguir o formato "**Como [ator], quero [funcionalidade] para [finalidade]**".
- Todas as US devem ser referenciadas por pelo menos um Requisito Funcional
- Todas as US devem ter critérios de aceitação associados.
- Todas as US devem estar associadas à uma Epic.

### 3. Sprint Backlog

**Objetivo:** Segmentar os itens do Product Backlog entre sprints de desenvolvimento.

**Responsáveis:** Product Owner, Dev Team

**Insumos:**

- Product Backlog inicial.
- User Stories posteriormente adicionadas ao Backlog inicial (se houver)

**Entregáveis:**

- Sprint Backlog
  - User Story
  - Tasks
  - Metas

**Boas práticas:**

- Garantir que os itens selecionados refletem uma boa entrega de valor antes da sprint.
- Vincular as Tasks às US correspondentes.
- Associar as Tasks às Epics correspondentes.
- Associar as Epics às Metas da Sprint.
- Garantir que as US só podem ser consideradas concluídas após a conclusão de todas as Tasks vinculadas a ela.
- Estimar o esforço das US (story points).

### 4. Criação de Branch e Commit para Rastreabilidade

**Objetivo:** Garantir rastreabilidade entre requisitos, tarefas e código.

**Responsáveis:** Dev Team.

**Insumos:**

- Tasks e Épicas definidos.

**Entregáveis:**

- Branches criadas de forma padronizada.
- Commits rastreáveis.
- Pull Requests padronizados e bem descritos.

**Boas práticas:**

- ID da task trabalhada sempre presente no começo da mensagem de commits e pull requests

- Padrão: GAIA-[ID]
- Tag do tipo de alteração presente logo após ID da task nos commits
  - (feat) Quando da adição de um recurso, uma *\*feature\** (funcionalidade).
  - (fix) Correção de um bug.
  - (docs) Atualização de documentação.
  - (style) Mudança de formatação, sem afetar o código.
  - (refactor) Refatoração do código, sem alterar funcionalidade.
  - (test) Adiciona ou modifica testes.
  - (chore) Atualizações menores que não impactam diretamente a funcionalidade do código.
  - (ci) Alterações nos pipelines de Integração Contínua e Deploy Automático.
- Branch deve ser aberta via task do Jira
- Solicitar revisão de pelo menos 1 code owner do repositório