

## Exercício 1: Análise do Kanban

### Problemas por coluna

1. **Backlog:** Na coluna de backlog possuímos uma mistura de bugs e tarefas, sem priorização clara. Isso pode gerar sobrecarga nos itens que ainda não foram analisados.
2. **Análise:** Não há item nessa coluna, pode indicar que a análise está sendo ignorada, ou não está acontecendo.
3. **Desenvolvimento:** Há poucos itens em desenvolvimento, mas há bugs misturados com tarefas. Pode haver um possível impedimento na entrada de novas tarefas.
4. **Testes:** Há muitos itens, pode indicar uma sobrecarga, ou que o time de QA não está conseguindo acompanhar o fluxo.
5. **Feito:** Há apenas um item, indica que o fluxo está lento, possui poucas entregas concluídas. Pode indicar problemas de impedimento nas fases anteriores.

### Possíveis soluções

- Tentar criar uma priorização clara entre bugs e tarefas no backlog;
- Fazer questão que cada item passe pela análise antes de ir para desenvolvimento;
- Definir o limite de quantidade de itens em desenvolvimento e em testes para evitar gargalos;
- Separar bugs e tarefas nas colunas para evitar confusão e facilitar acompanhamento;
- Revisar e ajustar o fluxo para que a coluna “Feito” receba entregas contínuas;
- Realizar dailys eficientes para identificar bloqueios rapidamente.

## Exercício 2: Papel do QA

1. **Planning:** O QA, assim como o todo o DEVteam deve ajudar a identificar os possíveis problemas da sprint, além disso, estimar o esforço do time para cada história de usuário. Também, pode sugerir critérios de teste para histórias.

2. **Daily:** O QA vai informar ao time o progresso dos testes, bloqueios ou bugs encontrados. Desta forma, vai ajudar o time a reagir rapidamente aos problemas de qualidade.
3. **Review:** O QA pode validar se as histórias entregues estão atendendo aos critérios de aceitação. Fora isso, tirar dúvidas do PO durante a reunião.
4. **Retrospectiva:** O QA pode contribuir pontuando falhas no fluxo de desenvolvimento, testes ou comunicação entre a equipe. Pode sugerir melhorias para prevenir bugs futuros e para melhorar o relacionamento do time.
5. **Refinamento:** O QA vai ajudar a avaliar riscos e a complexidade das histórias antes de entrarem em desenvolvimento. Além de também ajudar a detalhar requisitos e identificar cenários de teste de forma antecipada.