**Exercício 1: Análise do Kanban**

Gráfico, Gráfico de cascata

Descrição gerada automaticamente

1. Quais os problemas você enxerga neste quadro? Liste pelo menos um **problema por coluna (backlog, análise, desenvolvimento, testes, feito)**

**Backlog:** Não deveriam ter bugs, pois bugs são problemas provenientes de desenvolvimento ou de itens em produção  
  
**Análise:** Sem tarefas, muito provável porque o kaban esteja com acúmulo de tarefas em outras colunas

**Desenvolvimento:** com baixo fluxo de tarefas (Mas, precisaria avaliar a quantidade de desenvolvedor e reajustar o fluxo como um todo visto a coluna de testes.  
  
**Testes:** QA totalmente sobrecarregado, muitas tarefas e muitos Bugs para reteste.  
 **Feito**: Pelo tamanho do backlog e pelo quantidade de acumulo de testes, o PO não está fazendo as quebras de trabalho da forma mais adequada.

**2. Quais as possíveis soluções? Escreva pelo menos 5 soluções.**

a. Bugs tem que entrar no quadro direto em uma coluna de impedimentos, serem tratados com SLA

b. Os desenvolvedores poderiam tocar determinada quantidade de tarefas.  
c. QA precisa de apoio para concluir os testes de tarefas e bugs, pois conforme quadro o fluxo de trabalho está totalmente travado nos testes. A equipe poderia fazer uma força tarefa para ajudar, caso seja um único QA

d. O product Owner deverá avaliar tarefas mais complexas do backlog e alinhar com os desenvolvedores para irem adiantando trabalhos, visando o tempo de liberação da coluna de QA

e. Caso exista a possibilidade de automatizar os testes, ajudaria com a coluna Testes. Se são todos testes manuais, precisamos pensar se estamos com pouco QA.

f. O quadro precisa ser totalmente alinhado, será necessário repensar em quantos cards a equipe poderá ter em cada coluna, ajudando assim a ser um kaban mais fluído.

**Exercício 2: Papel do QA**

**Qual o papel do QA em cada evento ágil (Planning, Daily, Review, Retrospectiva, Refinamento)? Descreva como o QA pode ajudar em cada evento deste.**

**Planning**: Garantir clareza nos critérios de aceitação, estimar o esforço de testes e definir estratégias de testes.

**Daily:** Atualizar o status dos testes, identificar impedimentos, citar sobre automações em execução e as que possam vir a automatizar.

**Review**: Validar que os critérios de aceitação foram atendidos e apresentar os resultados dos testes.

**Retrospectiva:** Citar o que foi bom durante um projeto, o que foi ruim e propor melhorias na qualidade

**Refinamento**: Identificar requisitos para atuar em um teste mais criterioso, validar especificações técnicas junto aos desenvolvedores, discutir sobre a necessidade de testes de performance.