

# Engenharia de Software e Arquitetura de Dados – Entrega 1

## **Aplicação dos Métodos Ágeis**

No desenvolvimento do projeto foi adotado o Kanban, utilizando a ferramenta GitHub Projects para organizar as tarefas da equipe.

### ***Justificativa da Escolha***

O Kanban foi escolhido pela sua simplicidade e pela facilidade de visualizar o fluxo de trabalho. A metodologia permite acompanhar o progresso das atividades em colunas como To Do, In Progress e Done. Esse modelo é adequado, pois o projeto está em constante evolução, com novas demandas surgindo conforme as funcionalidades são desenvolvidas.

### ***Organização do Trabalho***

As tarefas foram divididas em histórias de usuário e tarefas técnicas. Cada membro do grupo pode atribuir tarefas a si mesmo e mover no quadro de acordo com o status. Exemplos de tarefas já incluídas no Kanban: configuração do backend Flask e SQLite, desenvolvimento da interface em React, importação de dados simulados, geração de gráficos analíticos e integração com APIs externas.

### ***Resultados Obtidos***

O time passou a ter visibilidade do andamento das entregas, clareza de prioridades e responsabilidade individual. O método permite acompanhar futuras entregas de forma iterativa, garantindo que as metas da disciplina sejam cumpridas.

## **Engenharia de Requisitos**

### ***Requisitos Funcionais (RF)***

RF01: O sistema deve exibir dashboard com gráficos interativos (ticket médio, receita, categorias).  
RF02: O sistema deve permitir importar e processar dados de clientes e transações.  
RF03: O dashboard deve permitir filtrar por período, tipo de cupom e categoria de loja.  
RF04: O sistema deve gerar relatórios analíticos automáticos (PDF).  
RF05: O sistema deve suportar visualização de KPIs (ex.: repasse PicMoney, margem operacional).

### ***Requisitos Não Funcionais (RNF)***

RNF01: O sistema deve ser desenvolvido em React (frontend), Flask (backend) e SQLite (banco de dados).  
RNF02: O tempo de resposta das consultas ao dashboard deve ser inferior a 3 segundos.  
RNF03: O sistema deve ser acessível via navegadores modernos (Chrome, Edge, Firefox).

RNF04: Dados devem ser armazenados de forma segura e rastreável.

RNF05: A aplicação deve ser modular, permitindo futura integração com APIs REST/WhatsApp/AR.

## Requisitos de Domínio (RD)

RD01: O sistema deve utilizar bases simuladas de clientes e transações (até 100k linhas).

RD02: O sistema deve refletir a realidade da Av. Paulista como principal polo de consumo e circulação de cupons.

RD03: O sistema deve considerar os tipos de cupom da PicMoney: Produto, Desconto e Cashback.

RD04: O dashboard deve permitir análises sobre impacto do cashback e fidelização (ponto central do modelo de negócio da PicMoney).

## Conclusão

Com a aplicação do Kanban, a equipe estabeleceu um processo ágil para organizar e monitorar o desenvolvimento. A engenharia de requisitos estruturou as funcionalidades necessárias, os critérios de qualidade e os requisitos de domínio específicos da PicMoney, garantindo que o sistema atenda às necessidades do negócio e aos objetivos acadêmicos.

## Print do Kanban no GitHub Projects

