

Engenharia de Software e Arquitetura de Dados – Entrega 1

Antônio Petri de Moraes Soares de Moura e Oliveira – 24026144

Daniel Kyoshi Moribe – RA: 24026509

Gabriel Higobassi Paschoal – RA: 21010656

Vitor Kenzo Kanashiro – RA: 24026113

Métodos Ágeis

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).

Resultado: O usuário conseguiu visualizar os gráficos interativos de forma clara e intuitiva, facilitando a análise de desempenho e comparação de métricas.

RF02: O sistema deve permitir importar e processar dados de clientes e transações.

Resultado: O sistema importou corretamente os arquivos CSV, processando as informações de forma automatizada e sem necessidade de ajustes manuais.

RF03: O dashboard deve permitir filtrar por período, tipo de cupom e categoria de loja.

Resultado: Os filtros funcionaram adequadamente, permitindo segmentar as informações de acordo com o tipo de análise desejada.

RF04: O sistema deve gerar relatórios analíticos automáticos (PDF).

Resultado: Os relatórios foram gerados de forma automática e apresentaram gráficos e indicadores consolidados, prontos para uso em apresentações e análises.

RF05: O sistema deve suportar visualização de KPIs (ex.: repasse PicMoney, margem operacional).

Resultado: As métricas principais foram apresentadas em destaque no dashboard, proporcionando rápida interpretação dos resultados.

Requisitos Não Funcionais (RNF)

RNF01: O sistema deve ser desenvolvido em React (frontend), Flask (backend) e SQLite (banco de dados).

Resultado: A aplicação apresentou boa integração entre as tecnologias, com interface responsiva e backend estável.

RNF02: O tempo de resposta das consultas ao dashboard deve ser inferior a 3 segundos.

Resultado: As consultas retornaram resultados em tempo adequado, mantendo a fluidez da navegação.

RNF03: O sistema deve ser acessível via navegadores modernos (Chrome, Edge, Firefox).

Resultado: A aplicação funcionou corretamente nos principais navegadores testados, sem perda de desempenho ou layout.

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

Resultado: As informações foram salvas de maneira consistente, garantindo confiabilidade e rastreabilidade das operações.

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

Resultado: A arquitetura modular facilitou a manutenção e demonstrou compatibilidade com possíveis expansões futuras.

Requisitos de Domínio (RD)

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

Resultado: As bases simuladas foram processadas com eficiência, demonstrando bom desempenho em volume de dados elevado.

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

Resultado: As análises refletiram padrões reais de comportamento de consumo, reforçando a contextualização do cenário proposto.

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

Resultado: Os relatórios e gráficos apresentaram corretamente a segmentação por tipo de cupom, permitindo comparações entre modalidades.

RD04: O dashboard deve permitir análises sobre o impacto do cashback e fidelização.

Resultado: As visualizações possibilitaram observar o aumento de engajamento e fidelização associado ao uso de cupons de cashback.

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

