

FUNDAÇÃO ESCOLA DE COMÉRCIO ÁLVARES PENTEADO
CAMPUS LIBERDADE
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL CARVALHO MOTA
GUILHERME DE LIMA SIQUEIRA
RODRIGO LUIZ MENEZES DOS REIS
VITORIA LETICIA MACIEL DA SILVA

PICMONEY

ANÁLISE SOBRE APLICAÇÃO DE METODOLOGIAS ÁGEIS NO
DESENVOLVIMENTO DO PROJETO (SCRUM E KANBAN) E ENGENHARIA
DE REQUISITOS

São Paulo, 2025

Sumário

INTRODUÇÃO	3
ENGENHARIA DE REQUISITOS.....	4
Escopo do Projeto	4
Requisitos Funcionais	4
Requisitos Não Funcionais.....	4
Requisitos de Domínio	4
ESTRUTURA	5
Funções.....	5
Sprints e Reuniões	5
Backlog e Kanban	6
CONCLUSÃO	8

INTRODUÇÃO

O objetivo deste documento é apresentar a aplicação dos métodos de engenharia de requisitos no desenvolvimento do projeto e como tem sido a utilização e aplicação de metodologias ágeis como *Scrum* e *Kanban* ao longo do desenvolvimento do projeto, como também avaliar os eventuais benefícios obtidos e principais impactos.

ENGENHARIA DE REQUISITOS

Escopo do Projeto

O presente projeto tem como principal objetivo desenvolver uma análise de dados embasada no modelo de negócio da empresa PicMoney, cuja proposta é desenvolver um aplicativo gamificado de distribuição de cupons em parceria com lojistas e promover a educação financeira entre a parcela mais jovem da população (graduandos, alunos do ensino médio etc.).

De acordo com a proposta de negócio e com uma base de dados disponibilizada a respeito de cupons utilizados por localização, tipo de loja, usuários, potenciais clientes e dispositivos por localização, este projeto se propõe a entregar uma aplicação no estilo *Dashboard* para exibição de gráficos baseados em métricas definidas pela equipe, a fim de auxiliar os diretores (CEO, CTO e CFO) no monitoramento do desempenho da empresa, realizar projeções e ajudar nas tomadas de decisão, exibindo dados financeiros e operacionais.

Com isso, de acordo com uma extensa base de dados e documentação a respeito do modelo de negócio e funcionalidades do software fornecidos. Analisamos e definimos os requisitos necessários para a nossa aplicação cumprir com o seu papel, como descrito abaixo:

Requisitos Funcionais

- Interface gráfica para exibição de Dashboards.
- Chatbot para interação com IA para análise dos dados.
- Gerar gráficos de acordo com as métricas definidas para CEO, CFO e métricas gerais.
- Exportar relatórios em arquivos Excel ou PDF.
- Área para demonstração de mapas de calor para distribuição de cupons e lojas.
- Filtros por período, tipo e localização para cálculo e exibição dos dados.
- Área para projeções de acordo com os KPIs.
- Nível de acesso diferentes de usuário para as páginas (CEO e CFO)

Requisitos Não Funcionais

- Atualização em tempo real do dashboard integrado ao banco de dados.
- Realizar comandos através do Chat.
- Aplicação de padrões de projeto e arquiteturas bem definidas para um código claro, coeso e de fácil manutenibilidade.

Requisitos de Domínio

- O Dashboard deve atender as necessidades da empresa PicMoney para realizar projeções e monitorar resultados a partir dos dados.
- As métricas precisam seguir o modelo de negócio definido da PicMoney.
- A base de dados utilizada deve ser a disponibilizada pela PicMoney.

ESTRUTURA

Funções

Nesta aplicação, decidimos por juntar os papéis de *Scrum Master* e *Product Owner*, visto que a complexidade e amplitude do projeto não exige uma segmentação tão restrita, então adaptamos ao nosso contexto, garantindo a liberdade também de toda a equipe para poder gerir parcialmente o backlog e o andamento das atividades. Neste âmbito, o SM/PO também atuará como parte do Development Team. Com isso posto, dividimos a equipe da seguinte forma:

- Rodrigo Reis: *Scrum Master / Product Owner / Desenvolvedor Frontend*
- Gabriel Mota: *Development Team / Integração de IA (Backend)*
- Vitoria Leticia: *Development Team / Backend*
- Guilherme Siqueira: *Development Team / Backend*

Sprints e Reuniões

Para atualizações do projeto, não foi possível seguir o modelo de reuniões diárias já imposto pelo Scrum (*Daily*) de maneira oficial, visto que o tempo dos integrantes não colaborou para isso, porém, de certa forma, durante as aulas conseguimos discutir sobre o andamento e realizar alguns alinhamentos.

Contornando o problema, temos uma reunião semanal (*Weekly*) planejada para todo final de semana, aos domingos, onde nos reunimos virtualmente e apresentamos o progresso realizado pelos integrantes da equipe, discutimos a respeito de dificuldades e bloqueadores encontrados, assim como sobre possíveis soluções e os próximos passos a seguir.

Cada sprint tem a duração de uma semana, durando entre uma reunião de alinhamento e outra, por isso o volume de atividades neste interim costuma ser alinhado com o tempo e recursos dispostos. Definimos o tamanho e a quantidade de cada Sprint de acordo com o prazo do projeto desde a data de início até a data da última entrega, totalizando 11 Sprints de 1 semana, agrupando-as de acordo com cada entrega, exemplo: *E1/Sprint 1, 2, 3...* e *E2/Sprint 5, 6...*

Backlog e Kanban

A respeito da organização do *Backlog*, utilizamos a metodologia do *Kanban* dentro das funcionalidades de projeto do GitHub, com as seguintes colunas:

- **Backlog:** Atividades identificadas pela equipe a serem realizadas, podendo ser ainda pendentes de refinamento ou atividades já refinadas apenas esperando o início da Sprint correspondente.
- **Sprint:** Atividades da Sprint que devem ser realizadas, mas ainda constam com o início do desenvolvimento pendente.
- **In Progress:** Atividades já em progresso pela equipe.
- **In Review:** Uma vez completa, a atividade é movida para esta coluna, onde deve ser revisada pela equipe e ter o aceite de todos ou do Scrum Master antes de prosseguir para a próxima coluna, caso contrário ela pode voltar para “*In Progress*”.
- **Done:** Atividades já completas após a revisão.

Como mencionado anteriormente, todo o time tem autonomia para mover cards entre as colunas caso necessário e apontar mudanças ou observações a respeito dos cards nos campos de comentário, assim como também poderá criar caso identifique novas atividades de qualquer natureza a serem realizadas (bugs, features, documentação etc.).

Definimos e categorizamos cada atividade de acordo com a Sprint correspondente e o nível de prioridade (também neste contexto, correlacionado ao nível de complexidade e esforço necessário para completar a atividade).

Abaixo estão algumas demonstrações do nosso *board* atualmente:

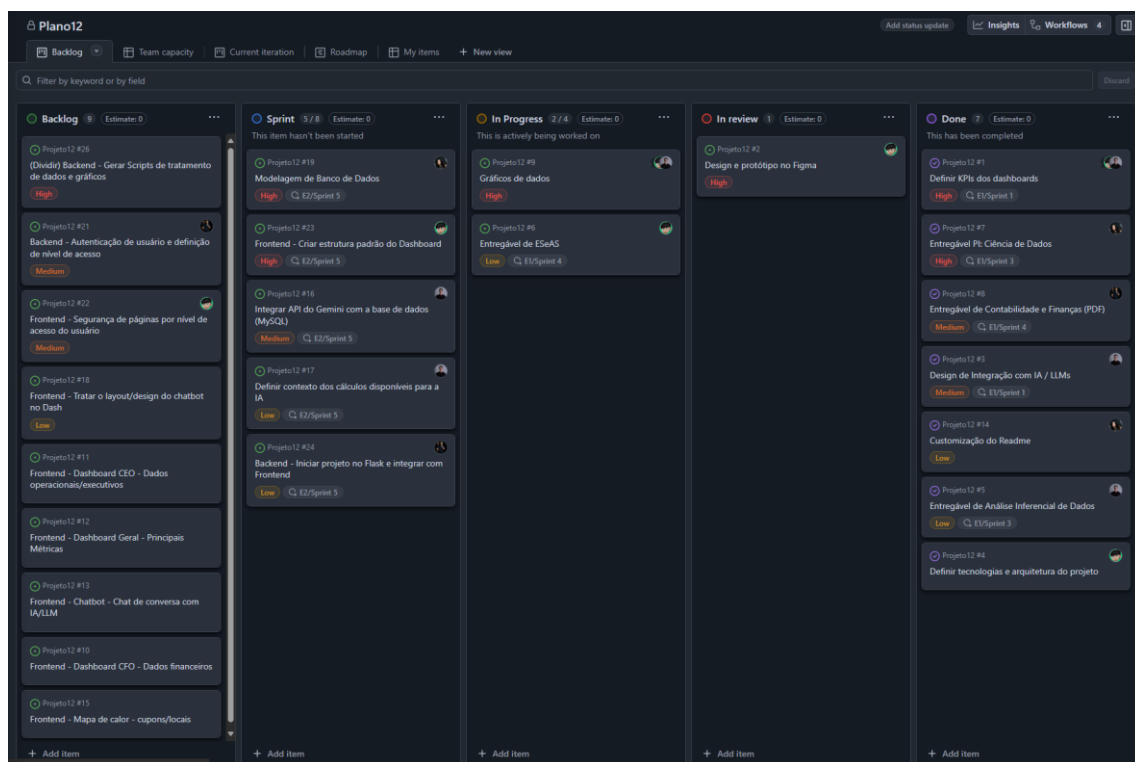


Figura 1 - Board do projeto

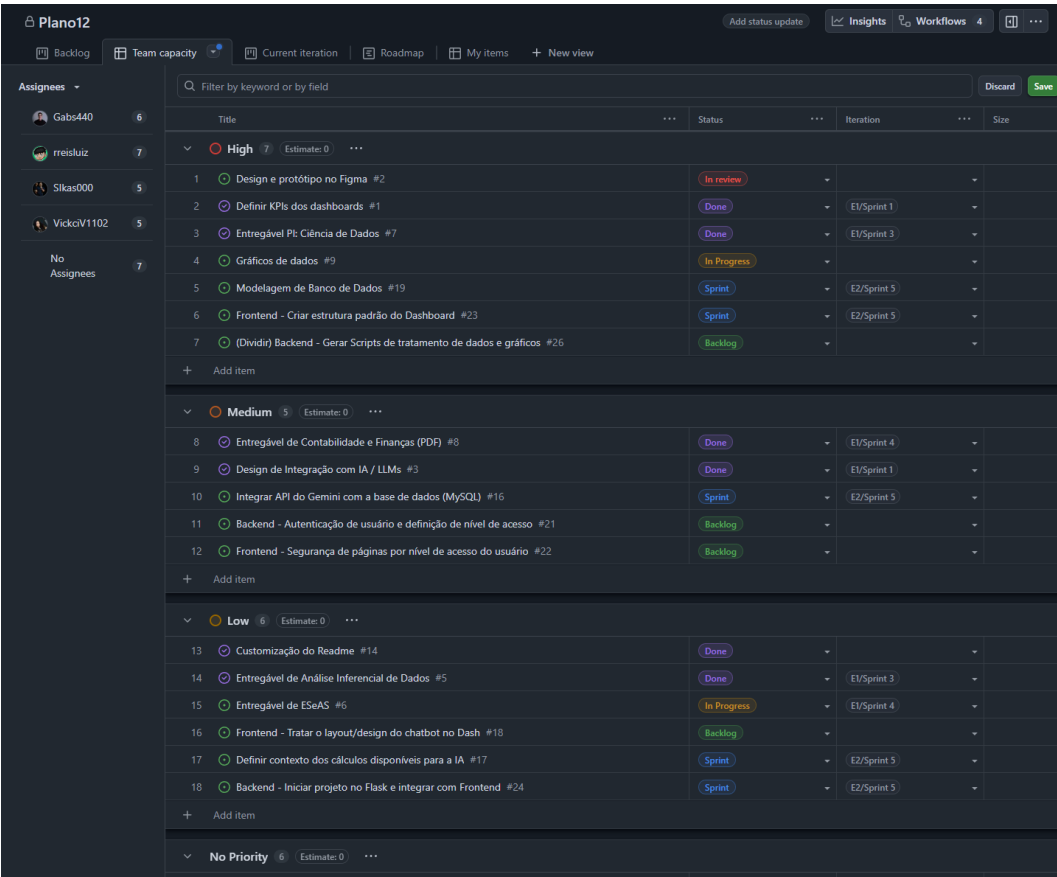


Figura 2 - Capacidade do time (atividades atribuídas)

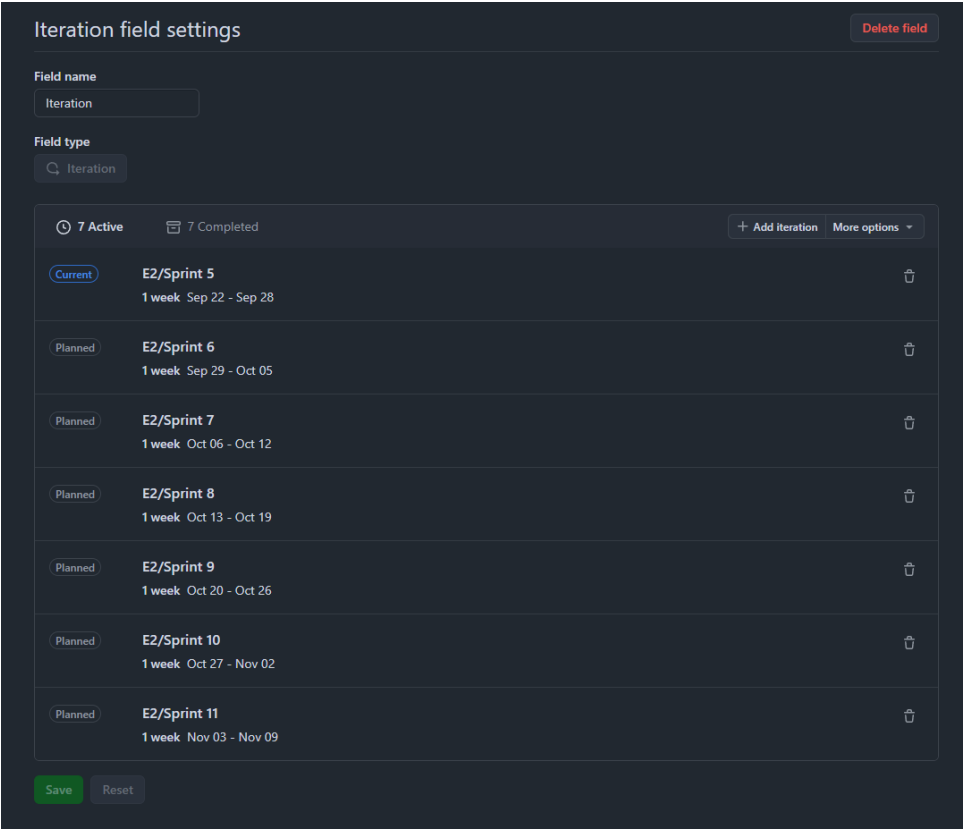


Figura 3 - Sprints planejadas do projeto

CONCLUSÃO

Diferente de semestres passados, a aplicação mais fiel do Scrum e Kanban no desenvolvimento tem sido bastante benéfica para a equipe no geral, garantindo uma maior rastreabilidade do andamento do projeto e tendo uma maior previsibilidade dos prazos de entrega e esforços necessários para cada atividade. Temos de forma mais definida e alinhada a responsabilidade da equipe a respeito de cada módulo do software a ser desenvolvido e de cada função a ser exercida pelos integrantes (Backend, Frontend, dados etc.).

Em destaque temos a realização de reuniões e atualizações regulares a respeito do nosso board e backlog, garantindo um acompanhamento mais próximo das atividades e do trabalho da equipe. Com isso a visão e alinhamento geral de expectativas se torna mais sólido, facilitando a comunicação entre os integrantes, a sinergia e o cumprimento dos prazos dentro do esperado.