Curso: Ciência da Computação

Disciplina: Engenharia de Software e Arquitetura de Sistemas

Professora: Lucy Mari

Integrantes:

Otávio Vecchi Zeferino – 24025982

Leonardo Santos da Silva – 24026495

Lucas Silva Maciel – 24025942

Thiago Akira Higa Mitami – 24026254

Lucas de Lima Gutierrez – 24026013

Entrega Engenharia de Software e Arquitetura de Sistemas

Aplicação de Metodologias Ágeis

Foi adotada uma abordagem híbrida: Scrum para organizar o desenvolvimento em Sprints e Kanban para acompanhar visualmente o progresso das tarefas. Essa combinação trouxe clareza no planejamento e flexibilidade na priorização das atividades.

Atividade Prática: Simulação de um Sprint

Cenário do Projeto:

Desenvolver um Dashboard Interativo para a PicMoney que inclua:

- Organização do repositório GitHub e documentação no Readme;
- Início da implementação do frontend (React);
- Definição dos perfis executivos (CEO, CFO, CTO) e seus KPIs;
- Integração das entregas das disciplinas.

Sprint Planning (Semana 1)

Objetivo: criar um protótipo inicial do software.

Tarefas:

- Levantamento de requisitos;
- Definição dos KPIs;
- Início do frontend;
- Diferencial do projeto.

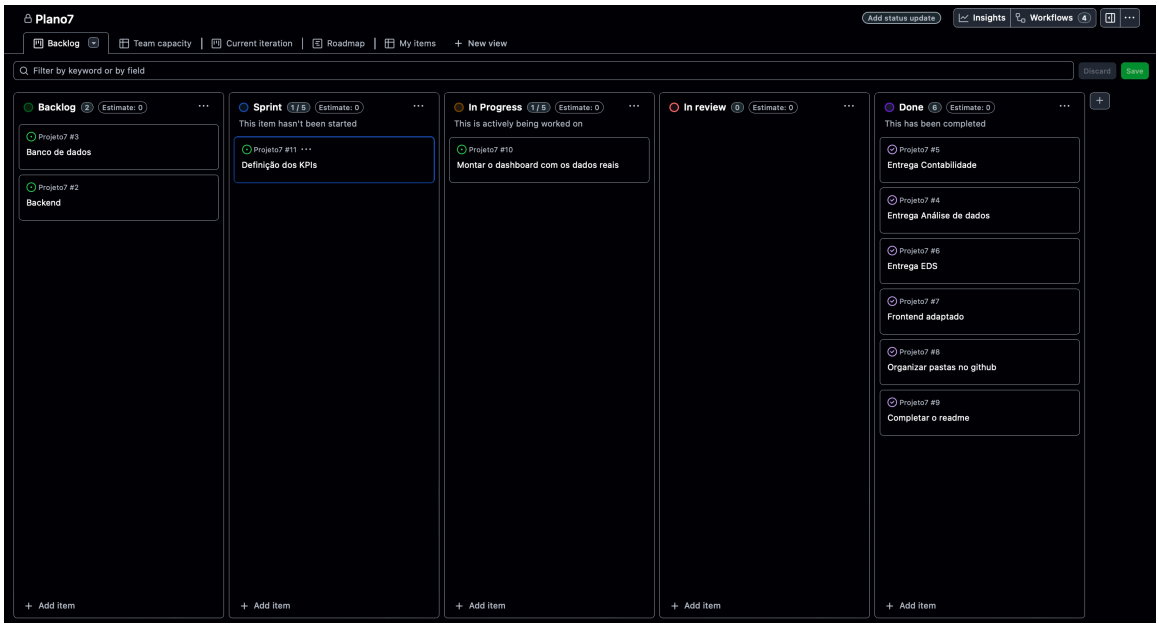
Daily Scrum

Realizado em reuniões de 10 a 15 minutos, abordando:
- O que foi feito no dia anterior;
- O que será feito no próximo dia;
- Principais dificuldades.

Product Backlog Priorizado

Alta prioridade: Login, dashboards, relatórios financeiros, KPIs.
Média prioridade: Design responsivo, prototipação, refatorações.
Baixa prioridade: Customizações estéticas.

Kanban Board



O quadro Kanban foi estruturado com quatro colunas: Backlog, Sprint, In Progress e In Review.

Backlog: Banco de Dados, Backend.
Sprint: Organizar pastas no GitHub, Completar README, Montar dashboard com dados reais, Definição dos KPIs
In Progress: Frontend adaptado, Entrega de Contabilidade, Entrega de Análise de Dados, Entrega de EDS.
In Review: Nenhuma tarefa no momento.

Programação em Par e Testes

O grupo aplicou Programação em Par, alternando papéis de desenvolvedor e revisor. Para garantir qualidade, foi usado TDD (Test-Driven Development). Após a validação, houve refatoração do código.

Discussão em Grupo

Os principais aprendizados foram:
- Maior organização com o uso simultâneo de Scrum e Kanban;
- Necessidade de comunicação constante para superar problemas de integração;
- Benefícios do trabalho colaborativo (pair programming).

Engenharia de Requisitos

Levantamento de Requisitos

Foi realizada uma pesquisa simples com potenciais usuários para entender expectativas em relação ao dashboard interativo.

Requisitos Funcionais

- **RF01:** O sistema deve disponibilizar um painel interativo com visualizações gráficas dos principais indicadores financeiros (ex.: receita, despesas, ticket médio).
- **RF02:** O sistema deve possibilitar a importação e o processamento automático de dados de usuários e transações.
- **RNF03:** O dashboard deve oferecer filtros dinâmicos por período, categoria de transação e perfil executivo (CEO, CFO, CTO).
- **RF04:** O sistema deve gerar relatórios consolidados em diferentes formatos (PDF, Excel).

- **RF05:** O sistema deve apresentar KPIs personalizados de acordo com cada perfil executivo.
- **RF06:** O usuário deve conseguir realizar login seguro e acessar apenas as informações correspondentes ao seu perfil.

Requisitos Não Funcionais

- **RNF01:** A aplicação deve ser desenvolvida utilizando React no frontend, Node.js no backend e SQLite como banco de dados;
- **RNF02:** O tempo de carregamento do dashboard não deve ultrapassar 2 segundos para consultas comuns;
- **RNF03:** O sistema deve ser compatível com os principais navegadores modernos;

- **RNF04:** Todas as transações e dados devem ser armazenados com criptografia e autenticação de acesso.Requisitos Não Funcionais

-**RNF05:** O software deve seguir princípios de modularidade, permitindo integração futura com APIs de terceiros (ex.: serviços financeiros, mensageria).

Requisitos de Domínio

-**RD01:** O sistema deve utilizar bases simuladas contendo registros de clientes e transações em volume considerável (até 50 mil entradas).
- **RD02:** O dashboard deve refletir métricas financeiras aplicáveis ao mercado nacional, considerando regras básicas de contabilidade e finanças.

- **RD03:** O sistema deve contemplar KPIs específicos de gestão executiva, como liquidez, margem de lucro e fluxo de caixa.
- **RD04:** O painel deve possibilitar análises comparativas entre diferentes períodos, permitindo avaliar tendências e apoiar tomadas de decisão.

Especificação Final dos Requisitos

Críticos: Autenticação de usuário, visualização de dashboards, geração de relatórios.
Importantes: Interface responsiva, integração com banco de dados.
Desejáveis: Customização avançada do design.

Critérios de aceitação foram definidos para cada requisito, garantindo clareza sobre quando ele estaria concluído.

Conclusão

A aplicação conjunta das metodologias ágeis e da engenharia de requisitos possibilitou maior clareza e organização no desenvolvimento do projeto. O uso de Scrum e Kanban ajudou a estruturar entregas e visualizar o fluxo de trabalho, enquanto o levantamento e especificação de requisitos asseguraram o alinhamento com as necessidades dos usuários.