

Relatório de Design de Software e UML: Dashboard Executivo PicMoney

Alunos:

Flávia da Costa Rodrigues Faria - 20021548

Guilherme Muniz Gomes - 24026572

Lucas Moreira Godoy - 24026298

Maria Eduarda Cabral Foloni – 24026502

Aplicação do Design de Software no Projeto

Este documento detalha as decisões de design e a arquitetura de software aplicadas no desenvolvimento do Dashboard Executivo PicMoney, um aplicativo web interativo construído com Streamlit e Pandas.

Modelo de Design: Arquitetura em Camadas

Seguindo os modelos de design apresentados, o projeto foi estruturado utilizando uma **Arquitetura em Camadas**. Embora seja um único script (dashboard.py), o código é logicamente dividido em três camadas distintas que se comunicam de forma organizada:

1. Camada de Persistência de Dados (Data Layer):

Responsabilidade: Armazenar e fornecer os dados brutos.

Implementação no Projeto: Esta camada é representada pela pasta data/ e seus arquivos CSV (base_de_transacoes_limpa.csv, etc.). A função @st.cache_data def load_data() atua como a "ponte" oficial para esta camada, sendo a única parte do software responsável por ler e preparar os dados.

2. Camada de Lógica de Negócio (Business Logic Layer):

Responsabilidade: Aplicar as regras de negócio e realizar os cálculos.

Implementação no Projeto: Esta camada é implementada inteiramente pela biblioteca **Pandas**. Toda a lógica de filtragem (ex: df_transacoes_filtrado = ...), cálculos de KPIs (DAU, Sessões, Deltas) e agrupamentos para os gráficos (ex: df_plot.groupby(...)) são executados aqui.

3. Camada de Apresentação (Presentation Layer):

Responsabilidade: Exibir a interface para o usuário e capturar suas interações.

Implementação no Projeto: Esta camada é implementada pelo framework **Streamlit** (ex: st.title, st.metric), pelas bibliotecas de visualização (st.plotly_chart) e pelo componente de navegação (streamlit-

option-menu). Esta camada não sabe *como* os cálculos são feitos, ela apenas solicita e exibe os resultados da camada de lógica.

Aplicação dos Princípios Fundamentais de Design

O design do projeto seguiu princípios fundamentais para garantir a manutenibilidade e a qualidade, conforme descrito nos slides.

- **Coesão:**

Conceito: Cada módulo deve ter uma única responsabilidade bem definida.

Aplicação no Projeto: O projeto busca alta coesão em suas funções. O exemplo mais claro é a função `load_data()`. Sua única responsabilidade é carregar e limpar os dados. Ela não se mistura com a lógica de exibição (Streamlit) nem com os cálculos de KPIs (Pandas). Da mesma forma, cada seção `if pagina_selecionada == ...` agrupa de forma coesa toda a lógica referente àquela tela específica.

- **Acoplamento:**

Conceito: Reduzir as dependências entre os módulos.

Aplicação no Projeto: O projeto mantém um **baixo acoplamento** entre as camadas. A Camada de Apresentação (`st.metric`) não depende diretamente do Pandas. Ela apenas exibe uma variável (ex: `delta_dau_str`). Se decidíssemos trocar o Pandas por outra biblioteca de cálculo (como Polars ou Spark), a camada de apresentação permaneceria intacta, demonstrando baixo acoplamento.

Diagramas UML Aplicados ao Projeto

Para documentar visualmente o sistema, foram aplicados dois dos principais diagramas da UML, conforme solicitado.

Diagrama 1: Diagrama de Casos de Uso

Este diagrama representa as interações entre os usuários (Atores) e as principais funcionalidades do sistema (Casos de Uso).

- **Atores:**

CEO: Perfil executivo principal.

CFO: Perfil financeiro.

- **Casos de Uso:**

Visualizar Visão Geral (CEO): Acessa os KPIs de engajamento (DAU, Sessões) e gráficos operacionais.

Visualizar Finanças (CFO): Acessa os KPIs financeiros (Receita, Margem, Ticket Médio).

Ver Alertas: Acessa a página de monitoramento de anomalias (disponível para ambos).

Filtrar Dados: Funcionalidade central que afeta todos os relatórios, permitindo filtrar por período, região e parceiro.

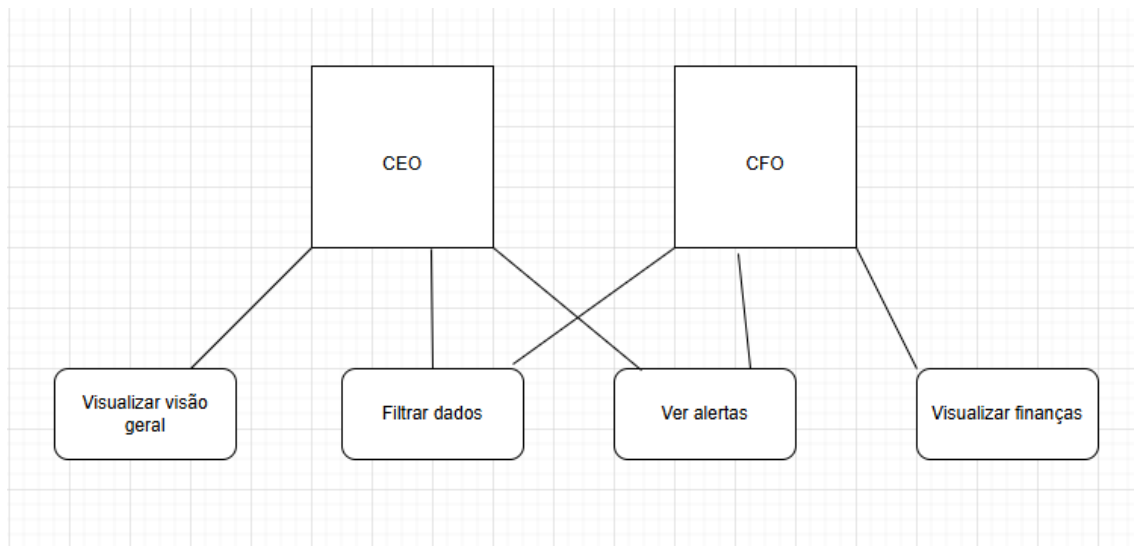


Diagrama 2: Diagrama de Atividade

Este diagrama representa o fluxo de trabalho (workflow) de uma funcionalidade chave do sistema. O fluxo abaixo detalha a principal atividade do dashboard: **a filtragem de dados**.

1. O fluxo se inicia quando o usuário (CEO ou CFO) acessa uma das páginas de visualização.
2. O usuário **seleciona os filtros** (Data, Região, Parceiro) e clica em "Aplicar".
3. O sistema (Camada de Lógica) **filtra o dataframe principal** do Pandas com base nas seleções.
4. O sistema **recalcula todos os KPIs** (DAU, Total de Transações, etc.) e os deltas de comparação com o período anterior.
5. O sistema **redefine os dados dos gráficos** (Plotly).
6. A interface (Camada de Apresentação) é **atualizada** e exibe os novos números e gráficos para o usuário.
7. O fluxo se encerra, aguardando uma nova interação.

Resultados no Projeto

A aplicação (mesmo que parcial) de um design estruturado em camadas e a busca por alta coesão e baixo acoplamento permitiram que o dashboard fosse desenvolvido de forma modular.

O resultado é um sistema funcional onde a lógica de dados (Pandas) está separada da lógica de visualização (Streamlit). Isso não apenas facilitou a manutenção (como corrigir os gráficos de pizza), mas também permitiu adicionar novas funcionalidades complexas (como os KPIs de *delta* e o gráfico de DAU/WAU/MAU) sem a necessidade de reescrever o código de carregamento de dados ou a interface da sidebar.