

Projeto Tabela FIPE – Consultas

Sumário: • Planning • Modelo de Dados • Modelo de Componentes (TDD resumido)

Alexsandro Pereira
10/02/2026

Sou o responsável por todo o ciclo do projeto e devo atuar com visão de produto e engenharia:
planejar → documentar → prototipar → codar → testar → apresentar.

Critério de avaliação: clareza de arquitetura, organização do trabalho e escopo mínimo
funcionando (não quantidade de funcionalidades).

1) Planning da semana

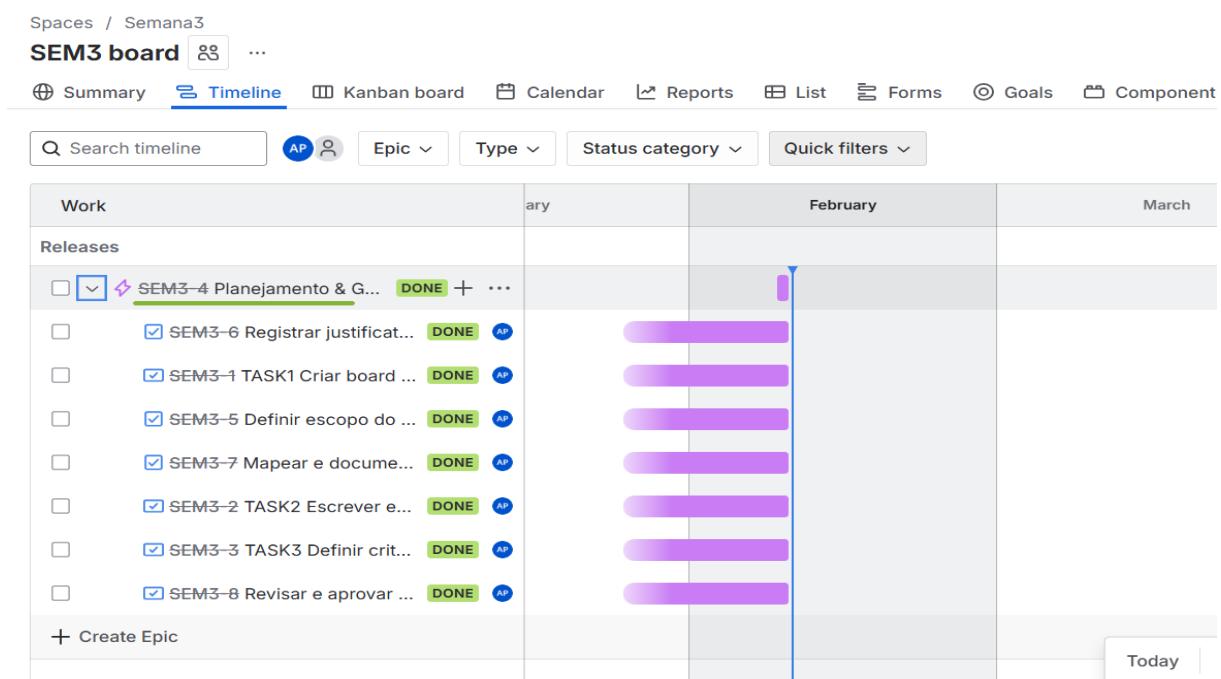
Objetivo do MVP: disponibilizar uma interface de consulta de preços (estilo FIPE) com seleção por **Marca** → **Modelo** → **Ano/Versão**, retornando **preço** e **mês de referência**. Requisito adicional: registrar um log simples de consultas para análise posterior.

Dia	Foco	Entregáveis principais
Segunda	Planning	Backlog, escopo (in/out), critérios de aceite, riscos e plano de execução
Terça	Documentação de projeto	Arquitetura, modelo de dados (ERD) e TDD/Componentes (resumo)
Quarta	Prototipação + início de código	Protótipo da UI, esqueleto do projeto e carga inicial (seed) do banco
Quinta	Codificação + testes	Fluxo de consulta completo, fallback de mês e testes unitários básicos
Sexta	Integração + apresentação	Polimento, revisão final, evidências (prints), roteiro de demo e execução

Board Jira:

<https://ziontrain.atlassian.net/jira/software/c/projects/SEM3/boards/35?issueParent=10043>

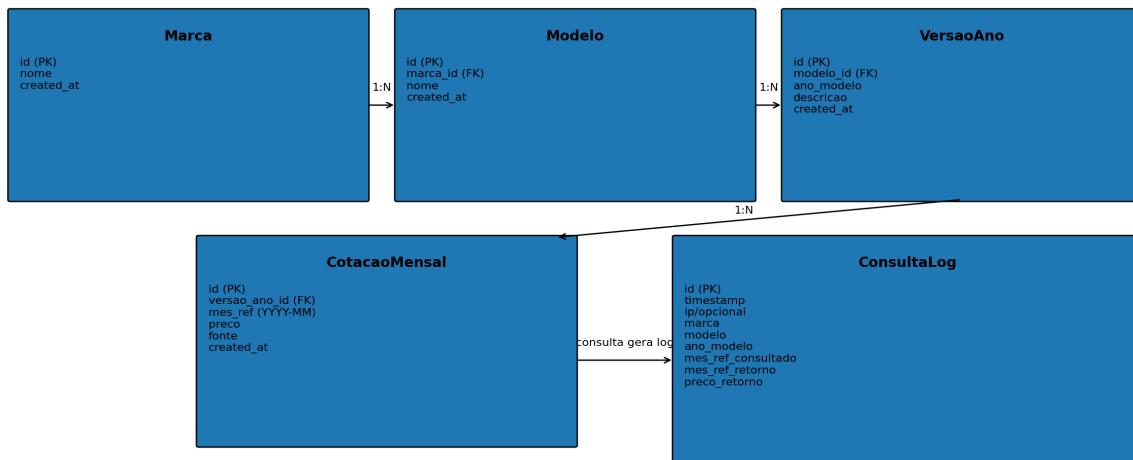
Figura 1 — Visão do board e status dos itens (Jira).



2) Modelo de Dados

O modelo prioriza a consulta do usuário e o registro de evidências. A cadeia **Marca** → **Modelo** → **Versão/Año** representa a navegação na UI. As cotações são armazenadas por mês de referência. O sistema registra cada consulta em *ConsultaLog* para permitir análises posteriores (ex.: volume de consultas por período).

Figura 2 — ERD (entidades principais da consulta).



Regra: se não houver cotação no mês solicitado, retornar a última cotação anterior disponível (fallback).

Regra de fallback do mês: caso não exista cotação para o mês consultado, o serviço retorna a última cotação disponível anterior para a mesma versão/ano, mantendo o mês retornado explícito na resposta.

3) Modelo de Componentes (TDD resumido)

UI (Streamlit): coleta filtros (marca/modelo/ano) e exibe resultado.

Application Service (ConsultaService): aplica regra de fallback, valida entradas e coordena repositórios.

Repositórios (SQLite): acesso a dados de domínio (marcas/modelos/versões) e cotações mensais; persiste logs.

Domínio: entidades e estruturas (Marca, Modelo, VersaoAno, CotacaoMensal, ConsultaLog).

Seed/Bootstrap: carga inicial de dados para viabilizar a consulta no MVP.

Sequência (consulta): Usuário seleciona filtros → UI chama *ConsultaService* → service consulta cotação do mês solicitado → se não existir, busca a última anterior → retorna preço e mês de referência → grava log → UI renderiza resultado.

Figura 3 — Plano semanal (visão de calendário).

Projeto FIPE-like — Semana Final (Solo)
Entregável de hoje (terça): Planning + Modelo de Dados + Modelo de Componentes (TDD resumido).

Contexto e mindset

- Semana 2 (time): módulos divididos (front, regras, BD, API), entregas contínuas em sprints.
- Semana final (solo): você é o time inteiro e precisa agir como empresa: planejar → documentar → prototipar → codar → testar → apresentar.
- Criterio importante: clareza de arquitetura + organização do trabalho + escopo mínimo funcionando (não quantidade de features).

1) Planning da semana (alto nível)

Dia	Foco	Entrega (artefatos)
Terça	Planning + Documentos	Escopo funcional; Modelo de Dados, Modelo de Componentes-TDD
Quarta	TDDs + Protótipos + Início do código	TDD consulta + sequência; protótipo UI; schema SQLite + seed; 1 versão da UI
Quinta	Codar + Testar + Refinar	Código integrado com testes; logs simples; testes básicos; ajustes UX
Sexta	Consolidar + Apresentar	Demo estável + relatório final (decisões, trade-offs, backlog, próximos passos)

Backlog (Kanban mental) e rastreabilidade
Colunas sugeridas: Backlog → Ready → Doing → Review → Done. Exemplos de cards já coerentes com o plano:

Backlog	Ready	Doing	Review	Done
[docs/01_escopo.md]	[docs/02_modelo_dados.md] [schema.sql]	[seed_db.py]	[fallback_ultimo_mes]	[demo + pitch]
	[docs/03_tdd_design.md]	[UI consulta]	[testes CT01-CT03]	

Definition of Done (DoD) — Consulta do Usuário

- Dropdowm dependentes: Marca → Modelo → Versão/Ano-modelo.
- Consulta exibe a média do mês a partir da pesquisa_preco.
- Se não houver dados no mês escolhido, aplica fallback para o último mês anterior disponível e informa mês/ano usado.
- Seed mínimo para demo (dados suficientes para provar o fallback).
- Código organizado por camadas (UI / domínio / repositório).