

Documento de Entregáveis

Projeto: Plataforma de Auditoria e Mensuração de Transporte por Telemetria – Atvos

Sprint 1: Levantamento de Requisitos e Modelagem Inicial

Período: Semanas 1–2

Entregáveis:

- **Requisitos Funcionais e Não Funcionais**

Documento descrevendo as funcionalidades mínimas do MVP: ingestão de telemetria via **API** (com fallback para upload de CSV em caso de indisponibilidade), cálculo automático do km variável, consolidação de custos fixos e variáveis, geração de relatórios D-1 (diários) e mensais. Inclui também requisitos não funcionais: usabilidade da interface web, governança dos dados, segurança de acesso (perfis Operação/Financeiro/Admin), desempenho mínimo (processar dados D-1 até D+1) e rastreabilidade (logs e trilha de auditoria).

- **Mapeamento de Processos e Jornada do Usuário**

Levantamento dos atores principais: Gestor de Transporte (acompanhar viagens e validar divergências), Financeiro (fechamento mensal e auditoria), Operacional (monitoramento diário) e Transportadora (fornecimento de dados). Definição das jornadas dentro da plataforma (ex.: gestor visualiza viagens no D-1, filtra divergências >10%, exporta CSV; financeiro gera relatório mensal consolidado).

- **Dicionário de Dados e Tabelas de Apoio**

Inventário das fontes de dados (API de telemetria, boletim do motorista, mensagem da transportadora) e definição das tabelas auxiliares necessárias: veículos (placa, tipo, unidade), tarifas por km (por rota/área/tipo), custos fixos (mensal por veículo/rota), Polis (usina, frente, garagem, alojamento com geofencing), turnos (janelas de horário), dicionário de rotas (palavras-chave → rota canônica).

- **Arquitetura Inicial da Solução**

Documento técnico descrevendo a arquitetura básica: frontend em React ou Streamlit, backend em Python (Flask ou FastAPI) para orquestrar ingestão e processamento, banco de dados relacional (PostgreSQL no MVP, com opção SQLite para protótipos), camadas de dados (raw, stg, fact, agg). Definição de fluxo de ingestão via API, estratégia de deploy (local/servidor simples) e logs.

Tecnologias: Google Docs (documentação), Miro/Figma (personas e jornadas), PostgreSQL/SQLite, Python.

Sprint 2: Tratamento de Dados e Arquitetura Técnica

Período: Semanas 3–4

Entregáveis:

- **Pipeline de API e Tratamento de Dados - marcos e joao**
Desenvolvimento de rotina em Python para ingestão e pré-processamento dos dados de telemetria. O módulo permite leitura via CSV (simulando a API) e, futuramente, consumo direto da API oficial. Inclui validação de schema, normalização básica dos dados (odômetro, velocidade), integração com o banco de dados e geração de datasets confiáveis para os cálculos de custo e auditoria.
- **Base de Dados Estruturada - murilo e icaro**
Implementação de banco de dados PostgreSQL para centralizar as informações (hospedado no Supabase). Criação de camadas: raw (dados crus), stg (dados normalizados), fact (viagens, auditorias, consolidados) e agg (diário/mensal). Inclusão das tabelas auxiliares (veículos, tarifas, custos fixos, pontos de interesse e turnos). Configuração de views consolidadas (D-1 e mensal), rotinas de carga incremental, políticas de retenção (raw 6–12 meses, consolidados 24+ meses) e perfis de acesso (Operação, Financeiro, Admin).
- **Wireframe - isa**
Criação de protótipo navegável em Figma para representar a experiência inicial da plataforma. Inclui telas de login, lista de viagens D-1, filtros de pesquisa (data, unidade, placa), visualização detalhada de viagem e exportação de relatórios. O wireframe serve como guia visual para o desenvolvimento da interface web na Sprint 3, garantindo alinhamento entre requisitos funcionais, jornadas de usuário e dados disponíveis no banco.

Tecnologias: Python (pandas, geopandas), PostgreSQL/SQLite, SQLAlchemy, APIs de telemetria.

Sprint 3: Plataforma Web (MVP de Operação)

Período: Semanas 5–6

Entregáveis:

- **Interface Web Inicial - Murilo e Isabelle**
Protótipo funcional com login simples e perfis básicos. Tela principal com lista de viagens do dia, km medido pela telemetria, divergências com motorista/transportadora e filtros por unidade, placa, área e data.
- **Mapa Interativo - João**
Visualização geográfica das rotas percorridas (origem/destino marcados por Polos), com destaque AGR x IND e rotas divergentes em cores diferentes.
- **Cadastro e Upload de Tabelas de Apoio - Icaro**
Módulo de cadastro via interface (veículos, tarifas, custos fixos, Polos e turnos). Possibilidade de upload de CSV apenas para atualização em lote.

- **Exportação CSV D-1 - Marcos**

Geração de relatório diário consolidado com viagens, km telemetria, km declarado, divergências em %, flags e custo variável.

Tecnologias: React/Streamlit, Python (Flask ou FastAPI), Plotly/Leaflet, PostgreSQL.

Sprint 4: Relatórios e Outputs de Fechamento

Período: Semanas 7–8

Entregáveis:

- **Relatório Mensal Consolidado**

Geração de relatório agregado por veículo, rota e unidade com custo fixo, km variável, custo variável e total. Inclui divergências entre telemetria e transportadora/motorista, bem como score de auditoria por viagem e por frota.

- **Pacote de Medição no Layout Atvos**

Exportação em Excel/CSV com colunas no formato já utilizado pela Atvos no boletim de medição mensal, garantindo integração direta com processos atuais.

- **Painel Financeiro Mensal**

Dashboard com KPIs: custo/km, proporção fixo x variável, top rotas/unidades em divergência, ranking de transportadoras por diferença acumulada.

- **Alertas de Divergência (opcional MVP)**

Alertas automáticos para viagens com diferenças superiores a 10% em relação ao declarado.

Tecnologias: React/Streamlit, Python (pandas, openpyxl/xlsxwriter), PostgreSQL.

Sprint 5: Testagem, Validação e Refinamento

Período: Semanas 9–10

Entregáveis:

- **Testes de Usabilidade e Validação Funcional**

Execução de roteiro de testes com gestores de transporte e financeiro, validando acurácia do cálculo, clareza da interface e aderência aos relatórios. Ajustes aplicados com base no feedback.

- **Refinamento da Interface e Relatórios**

Melhorias em UX (filtros mais intuitivos, legendas visuais) e clareza dos relatórios

exportados (nomes de colunas, formatação compatível com sistemas internos).

- **Treinamento e Documentação**

Manual de uso detalhado (PDF), guia rápido (1 página) e sessão de treinamento remoto/presencial com a equipe Atvos.

- **Roadmap Pós-MVP**

Documento com propostas de evolução: integração com ERP financeiro, módulo mobile para motoristas registrarem ocorrências, auditoria avançada com machine learning para detectar padrões de fraude.

Tecnologias: Python (pytest/unittest), React/Streamlit, Exportação em PDF/Excel (ReportLab/openpyxl).

Documento de Entregáveis 2

Projeto: Plataforma de Auditoria e Mensuração de Transporte por Telemetria – Atvos

Sprint 1 PART1 – Backend e Ingestão de Dados

Período: Semanas 1–2

Entregáveis

1. Módulo de Ingestão via API (Django)

Implementação do backend em Django responsável por consumir os dados de telemetria.
Inclui:

- Conexão com API da Atvos (ou simulação com JSON ou via CSV local).
- Estruturação de endpoints e rotinas de coleta diária (D-1).
- Validação e padronização de campos (placa, rota, timestamps, odômetro).
- Registro de logs e fallback em caso de falha de comunicação.

2. Armazenamento Bruto no Cloud Storage

Configuração do pipeline inicial para gravação de dados crus (raw data) no **Google Cloud Storage**, garantindo versionamento e rastreabilidade.

3. Estrutura Base de Dados no BigQuery

Criação do projeto e datasets no **BigQuery** com camadas:

- **raw**: dados brutos da telemetria.
- **stg**: dados limpos e normalizados.
- **fact**: viagens consolidadas com distâncias e tempos de percurso.

4. Setup do Ambiente e Deploy Inicial

Hospedagem do backend no **Cloud Run** e integração com **GCP IAM**, definindo papéis e permissões de serviço.

Tecnologias: Django, Python, Google Cloud Storage, BigQuery, Vercel, pandas.

Sprint 1 PART2 – Pipeline Analítico e Cálculo de Custos

Período: Semanas 3–4

Entregáveis

1. ETL Automático (Django → Cloud Storage → BigQuery)

Desenvolvimento das rotinas de transformação e carga de dados:

- Limpeza de registros duplicados, tratamento de gaps de GPS e normalização de odômetro.
- Cálculo automático do **km variável** e consolidação de custos fixos.
- Criação de tabelas auxiliares: veículos, tarifas, custos fixos, Pols e turnos.

2. Módulo de Auditoria Automática

Motor de verificação com **flags de divergência (>10%)**, cálculo de score por viagem e geração de tabela de inconsistências no BigQuery.

3. Exportação Técnica de Dados Consolidados

Geração de views analíticas (D-1 e mensal) para consumo direto pelo Power BI e exportação em CSV / Excel para testes.

Tecnologias: Django, BigQuery SQL, pandas, Airflow (opcional), Cloud Scheduler.

Sprint 2 – Dashboards, Relatórios e Refinamento Final

Período: Semanas 5–6

Entregáveis

1. Painéis Power BI Conectados ao BigQuery

- Dashboard **operacional D-1**: viagens realizadas, km telemetria vs. declarado, divergências e custo variável.
- Dashboard **financeiro mensal**: consolidação AGR × IND, ranking de transportadoras, custo total e score de auditoria.
- Filtros por unidade, período, placa e tipo de rota.

2. Exportação de Relatórios Padronizados (Layout Atvos)

Criação de templates no Power BI / Excel com as colunas exigidas pela Atvos para fechamento mensal e boletim de medição.

3. Monitoramento e Observabilidade

Integração com **Prometheus + Grafana** para métricas de ingestão, processamento e falhas de pipeline.

4. Refinamento MVP + Documentação Técnica

- Ajustes de performance e UX nos dashboards.
- Documentação de endpoints, ETL jobs e permissões GCP.
- Guia rápido de uso e entrega do MVP validado com a Atvos.

Tecnologias: Power BI, BigQuery, Django, Grafana, Prometheus, Google Cloud Storage.

Resumo Geral

Sprint	Foco Principal	Entregáveis-chave	Duração
1	Backend + Ingestão	API Django + Cloud Storage + BigQuery (raw/stg)	2 semanas
2	Pipeline Analítico	ETL + Cálculo km variável + Auditoria	2 semanas
3	Dashboards e Relatórios	Power BI + Relatórios Atvos + Monitoramento + Docs	2 semanas