

PRECIFICAÇÃO DE VEÍCULOS

Plataforma de inteligência de mercado automotivo, focada em precificação de veículos para empresas do setor.

DRIVE INTEL

MELISSA ROCHA

Sobre o produto

O DriveIntel é uma plataforma especializada em análise de dados do mercado automotivo. Nosso foco é transformar informações em inteligência acionável, auxiliando equipes na tomada de decisões estratégicas com mais precisão e confiança. Com insights aprofundados, relatórios detalhados e visualizações claras, ajudamos a antecipar tendências, otimizar precificação e fortalecer estratégias de mercado.

Tip: Plataforma de inteligência de mercado automotivo

PRINCIPAL



Lojistas multimarcas

Público-alvo



Investidores no
mercado
automotivo



Instituições
financeiras



Leiloeiros

Requisito

No mercado de veículos usados, a precificação é uma etapa essencial para quem atua com revenda, leilões ou avaliação de ativos. Lojistas multimarcas, revendedores independentes e instituições financeiras dependem de valores precisos para garantir competitividade, liquidez e segurança nas negociações. A definição assertiva do preço não é apenas uma prática comercial — é um elemento estratégico que sustenta decisões, protege margens e assegura a confiança em transações cada vez mais orientadas por dados.

Empecilho

Dados

Falta de acesso a dados relevantes em quantidade suficiente.

Volatilidade

Dificuldade em obter tais dados atualizados devido a volatilidade do mercado automotivo.

Previsibilidade

Ausência de meios confiáveis de previsibilidade de preços.

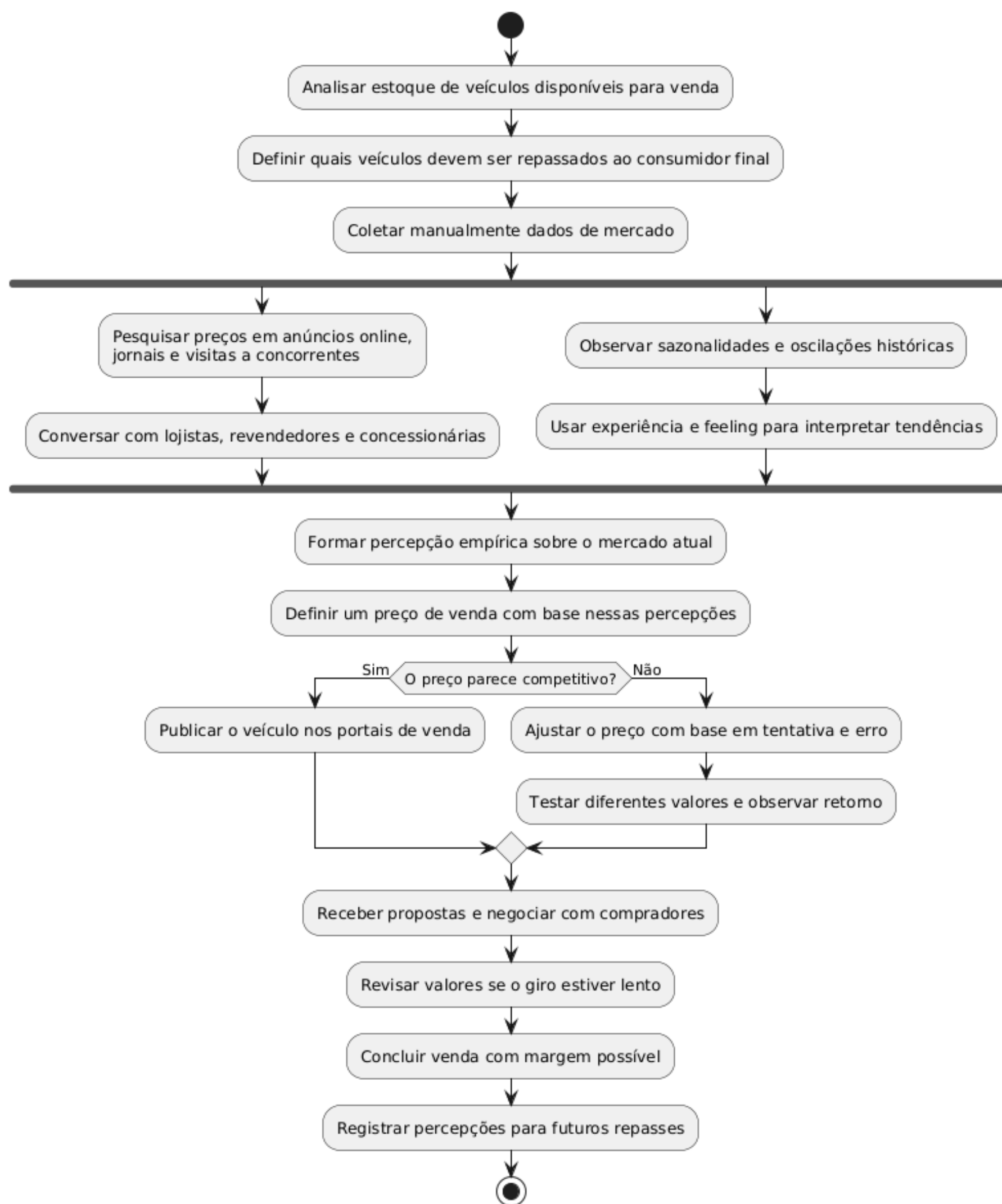


Diagrama Atividades do Negócio (SEM SISTEMA)

Objetivos

OBJETIVO GERAL

Desenvolver uma ferramenta de inteligência de mercado para o setor automotivo, capaz de coletar, analisar e interpretar dados de precificação de veículos por meio de automação, estatística e inteligência artificial, auxiliando empresas na tomada de decisões estratégicas sobre compra, venda e precificação.

Dados	Estatística	Insighsts
Coletar e organizar grandes volumes de dados provenientes de anúncios de veículos por meio de web scraping, consolidando informações relevantes para a análise de preços e tendências do mercado automotivo.	Aplicar modelos estatísticos e técnicas de inteligência artificial para identificar padrões de precificação e permitir simulações baseadas em diferentes cenários, considerando variáveis como modelo, ano e características do veículo.	Fornecer insights estratégicos para empresas do setor automotivo, permitindo que concessionárias, locadoras e gestores tomem decisões baseadas em dados, otimizando investimentos e estratégias comerciais.

OBJETIVOS ESPECÍFICOS

Technologias Usadas

Backend



DRF



Django



Fast API

WebScraping

BeautifulSoup

BeautifloulSoup



Selenium

Testes Automatizados



PyTest

Infra



Docker



Ngnix

Database



Postgres



Redis

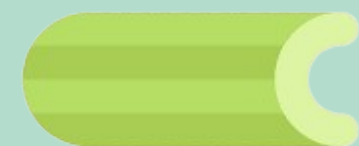
Outros



ScktiLearn

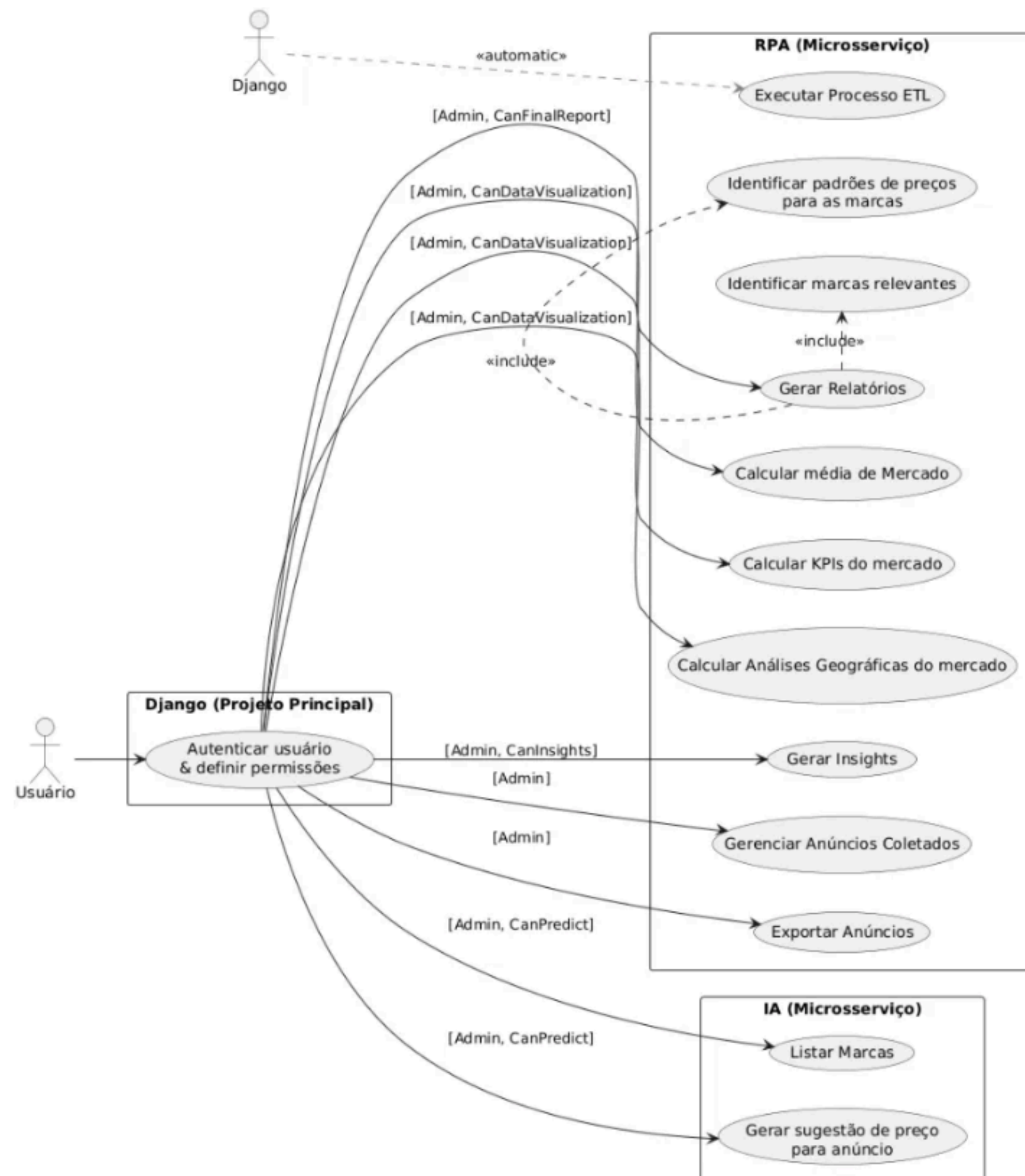


Swagger



Celery

Diagrama de Caso de Uso Geral





Engenharia de requisitos

MICROSSERVIÇOS RPA

Requisitos Funcionais

Código	Requisito	Descrição
RF001	Executar Processo ETL	O sistema deve realizar o processo de web scraping nos sites pré-definidos, coletando as URLs dos anúncios de veículos disponíveis. Os dados extraídos devem passar por um processo de sanitização para remover inconsistências e garantir sua integridade. Em seguida, devem ser validados conforme as regras de negócio e armazenados no banco de dados para uso posterior.
RF002	Cálculo média de Mercado	O sistema deve calcular a média de preço dos veículos com base nos anúncios coletados, considerando critérios de agrupamento definidos nas regras de negócio.
RF003	Gerar Relatórios	A aplicação deve gerar relatórios contendo insights e visualizações de dados baseados nos anúncios coletados em um determinado período. Os relatórios devem incluir métricas estatísticas relevantes para análise de preços e tendências do mercado de veículos.
RF004	Gerenciar Anúncios Coletados	O sistema deve permitir operações de CRUD nos dados coletados.
RF005	Exportar Anúncios	O sistema deve permitir o download dos anúncios coletados em formato XLSX, proporcionando aos usuários a possibilidade de analisar os dados em ferramentas externas, como planilhas eletrônicas.

Requisitos Funcionais

Código	Requisito	Descrição
RF006	Cálcular KPIs do mercado	O sistema deve fornecer indicadores-chave de desempenho (KPIs) do mercado automotivo com base nos dados coletados.
RF007	Cálcular Análises Geográficas do mercado	O sistema deve calcular a distribuição geográfica dos preços de veículos, apresentando a média de preços por estado e cidade. Além disso, deve medir a correlação entre preço e quantidade de anúncios em cada região, permitindo a visualização da concentração de ofertas e variações de preço no mercado.
RF008	Gerar Insights	O sistema deve consumir os dados brutos coletados e transformar os mesmos em informações acionáveis que apoiem o usuário na tomada de decisão estratégica para precificação de veículos.
RF009	Identificar marcas relevantes	O sistema deve com base em cálculos estatísticos identificar marcas relevantes para o setor.
RF010	Identificar padrões de preços para as marcas	O sistema deve identificar padrões de preços para as marcas, permitindo o entendimento de média de preços, preços acima da média bem como preços abaixo da média para a marca.

Requisitos Não Funcionais

Código	Requisito	Descrição
RNF001	Escalabilidade	O sistema deve ser escalável, permitindo o aumento da capacidade de processamento conforme a demanda de novos sites e maior volume de dados.
RNF002	Modularidade	A arquitetura deve ser estruturada de forma que seja fácil adicionar novos sites ao processo de web scraping sem impactar os sites já existentes.
RNF003	Tolerância a Falhas	O processo de automação (RPA) deve ser robusto e implementar tratativas de erro para lidar com falhas transitórias, como perda de conexão com a internet, timeout de resposta dos sites e páginas inacessíveis.
RNF004	Processamento Assíncrono	O sistema deve permitir a execução assíncrona das tarefas para garantir melhor aproveitamento dos recursos e evitar bloqueios durante a coleta e processamento dos dados.
RNF005	Baixo Acoplamento	Os módulos do sistema devem ser desenvolvidos de forma desacoplada, permitindo que funcionalidades possam ser alteradas ou substituídas sem impactar outras partes do sistema.

Requisitos Não Funcionais

Código	Requisito	Descrição
RNF006	Observabilidade e Monitoramento	O sistema deve possuir logs detalhados das execuções do robô e cálculos estatísticos, possibilitando auditoria e rastreabilidade dos processos.
RNF007	Performance	O sistema deve ser capaz de executar tarefas que envolvem operações de entrada e saída (I/O-bound), como chamadas a APIs e acesso a banco de dados, sem impedir a execução eficiente de tarefas que exigem alto uso de CPU (CPU-bound), como cálculos estatísticos e processamento de grandes volumes de dados.
RNF007	Documentação da API	O sistema deve possuir uma documentação automática e atualizada no Swagger, refletindo todas as rotas disponíveis e seus respectivos parâmetros. Além disso, o acesso à documentação deve ser protegido por um token de autenticação.
RNF008	Testes Automatizados	O sistema deve possuir uma suíte de testes automatizados para validar as funcionalidades críticas, garantindo a confiabilidade do código antes da implantação.
RNF009	Segurança das Credenciais	O sistema deve armazenar credenciais de acesso a sites e serviços externos de forma segura, utilizando variáveis de ambiente ou um gerenciador de segredos, evitando exposição em código-fonte.

Regras de Negócio + Código

Código	RF	Regra	Descrição
RN001	RF001	Descarte de Anúncios Inválidos	Anúncios sem informações essenciais (preço, modelo, ano modelo) são descartados.

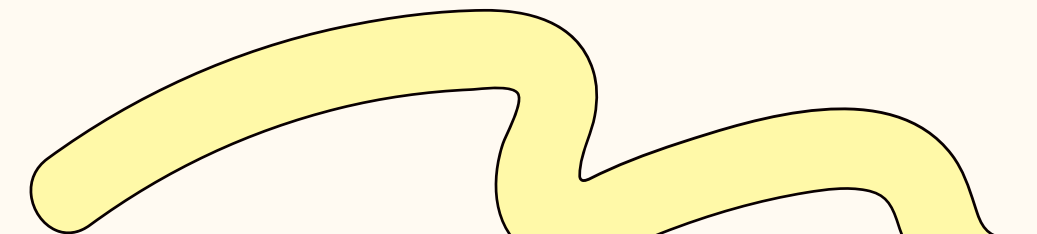
```
# Apply should_discard rule
if BusinessRules.should_discard(
    normalized_data["price"],
    normalized_data["model"],
    normalized_data["year_model"],
):
    causes_of_disposal.append("missing required fields")
```




Regras de Negócio + Código

Código	RF	Regra	Descrição
RN002	RF001	Corte de Preço	Veículos com preço abaixo de R\$5.000 são descartados.

```
@staticmethod 3 usages 🧑 Mel-Rocha
def price_cutoff_value(price):
    if price < Decimal("5000"):
        return None
    return price
```



Regras de Negócio + Código

Código	RF	Regra	Descrição
RN003	RF001	Inferir dados essenciais ausentes	Antes de descartar o anúncio que possua algum dado essencial faltante o sistema deve tentar inferir o valor do mesmo buscando o em alguma parte do anúncio, tal qual o título descrição etc.

```
> @staticmethod 1 usage  👤 Mel-Rocha
> def infer_missing_year(year_model, title, description):...

> @staticmethod 1 usage  👤 Mel-Rocha
> def infer_missing_price(price, description):...

> @staticmethod 3 usages  👤 Mel-Rocha
> def infer_missing_model(model, title):...
```

Regras de Negócio + Código

Código	RF	Regra	Descrição
RN004	RF001	Padronização de Atributos do Veículo	O sistema deve padronizar alguns atributos do veículo, garantindo que os valores estejam alinhados às nomenclaturas consideradas válidas pelo sistema.

```
@staticmethod 1 usage  🧑 Mel-Rocha
def standardize_fuel(fuel: str) -> str or None:...

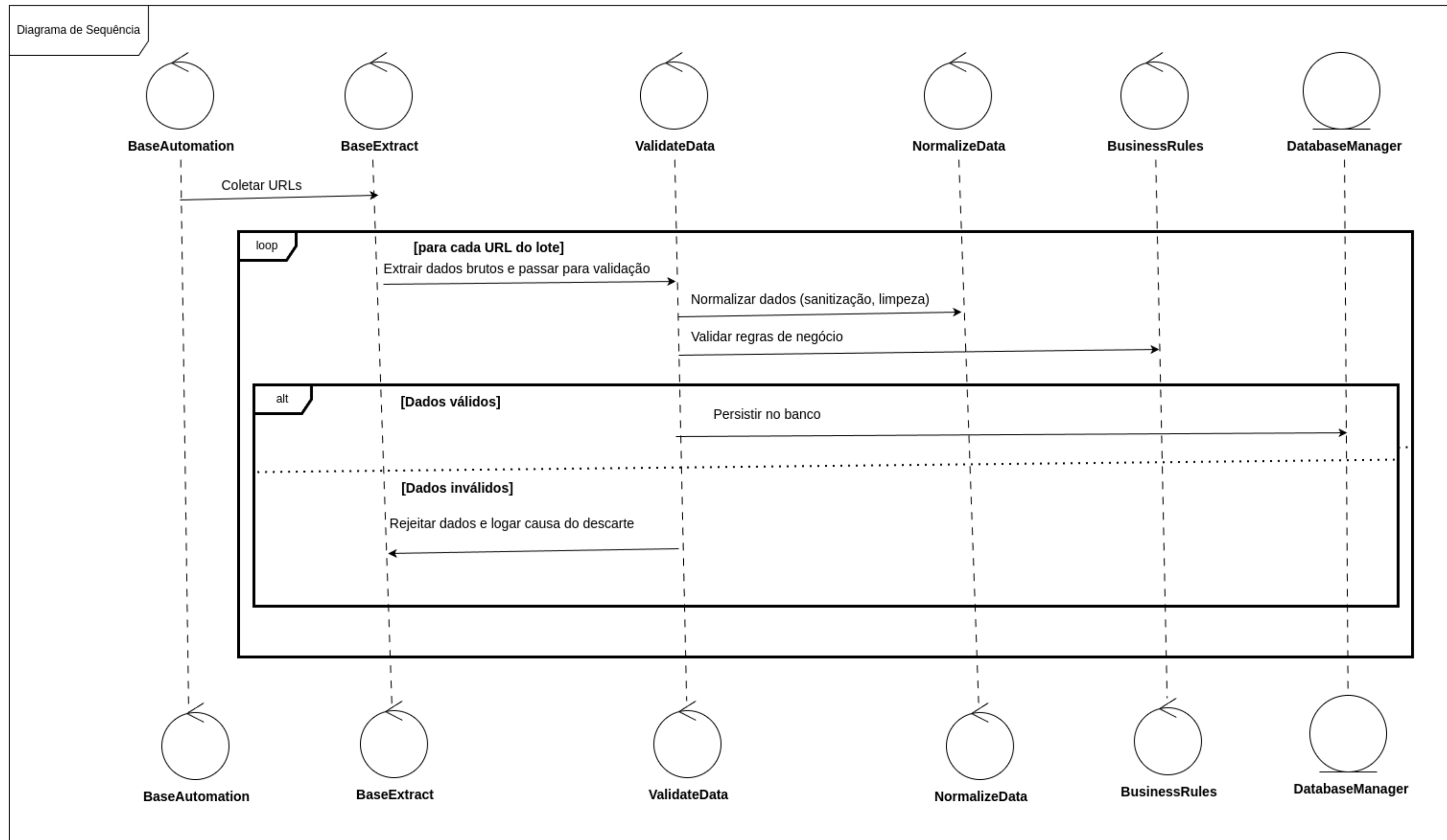
@staticmethod 1 usage  🧑 Mel-Rocha
def standardize_gear(gear: str) -> str or None:...

@staticmethod 1 usage  🧑 Mel-Rocha
def standardize_bodywork(bodywork: str) -> str or None:...
```

Matriz Rastreabilidad

[illegible]

Diagrama de Sequência



Código

```
@shared_task(bind=True, max_retries=3, default_retry_delay=60) 3 usages  🧑 Mel-Rocha *
def process_crawler(self, machine, site):
    try:
        start_time = datetime.now()
        all_results = []
        logging.info(f"Starting processing for site: {site}")

        # Obtendo parâmetros específicos do site
        params = SITE_PARAMS.get(site)
        unique_page = params["unique_page"]
        page_name_param = params["page_name_param"]
        page_number = params["page_number"] if params["page_number"] else 0

        # Executando a automação e coletando os dados
        result = automation_task(
            site, machine, unique_page, page_name_param, page_number
        )
        collected_urls = result["collected_urls"]
        batches = [
            collected_urls[i : i + BATCH_SIZE]
            for i in range(0, len(collected_urls), BATCH_SIZE)
        ]

        # Configura o grupo de tarefas (task_group) para extrair e salvar dados
        task_group = group(
            chain(
                *tasks: extract_task.s(
                    {"collected_urls": batch, "metrics": result["metrics"]},
                    site,
                    machine,
                ),
                save_data_task.s(machine), # Passando a máquina como argumento
            )
            for batch in batches
        )

        # Executa o grupo de tarefas
        task_group_result = task_group.apply_async()
```

Sequência

1. Coletar todas urls de um site, via tarefa automation_task;
2. Dividir as urls em lotes (batches);
3. Para cada lote, executar duas tarefas encadeadas (chain):
 - extract_task
 - save_data_task
4. Rodar tudo em paralelo usando group
5. loga informações de execução

chain: é usado quando uma tarefa depende da anterior — executa tarefas em ordem.

group: é usado quando as tarefas são independentes — executa todas ao mesmo tempo (paralelo).

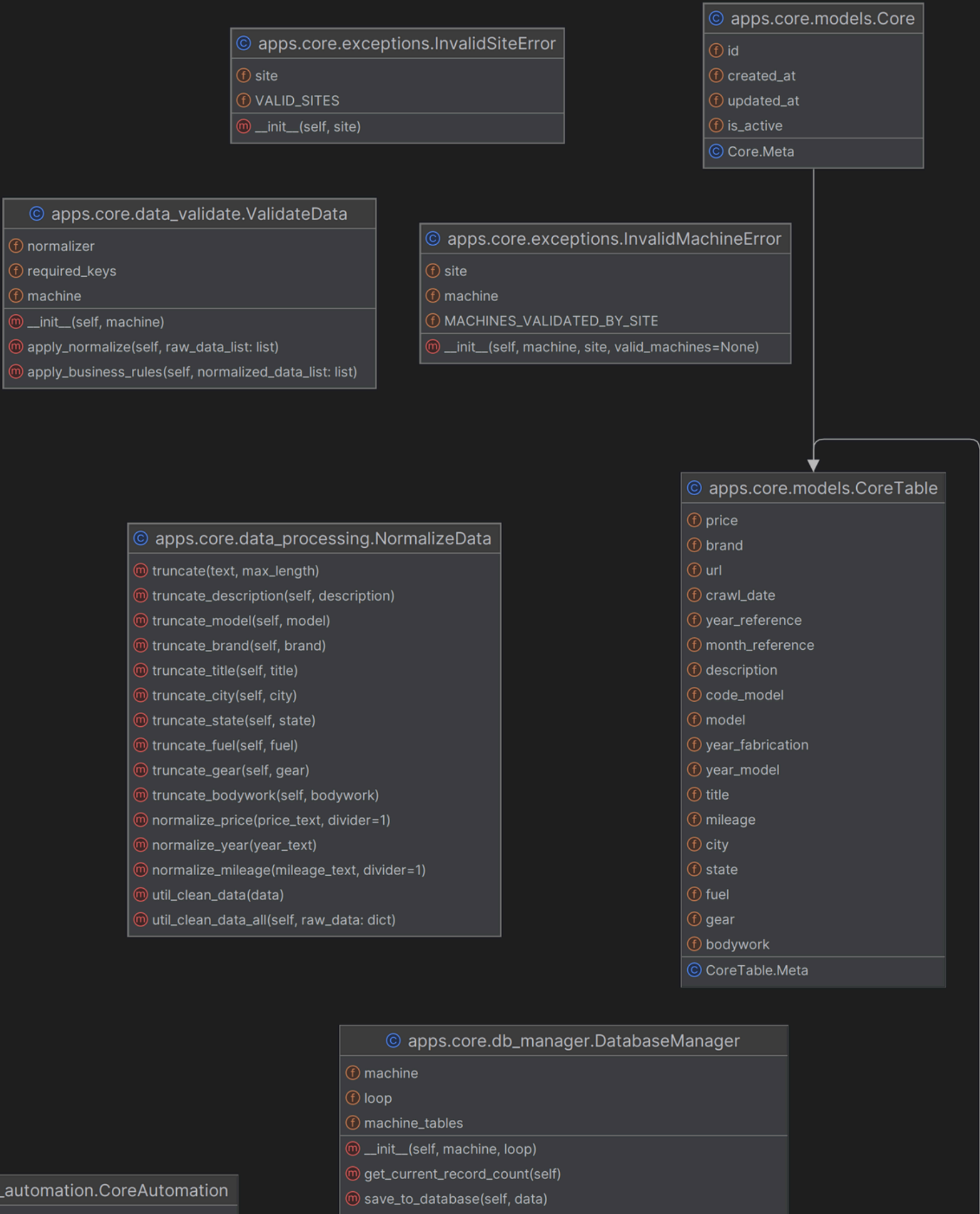


Diagrama de Classes

mer

machinetableaverage	
id	[UUID] NOT NULL
ad_count	[INTEGER] NOT NULL
average	[NUMERIC(18, 2)] NOT NULL
code_model	[VARCHAR(50)] NOT NULL
created_at	[TIMESTAMP] NOT NULL
is_active	[BOOLEAN] NOT NULL
reference_month	[INTEGER] NOT NULL
reference_year	[INTEGER] NOT NULL
updated_at	[TIMESTAMP] NOT NULL
year_model	[INTEGER] NOT NULL

machinetable	
id	[UUID] NOT NULL
bodywork	[VARCHAR(50)]
brand	[VARCHAR(255)]
city	[VARCHAR(50)]
code_model	[VARCHAR(50)]
crawl_date	[TIMESTAMP] NOT NULL
created_at	[TIMESTAMP] NOT NULL
description	[VARCHAR(600)]
fuel	[VARCHAR(50)]
gear	[VARCHAR(50)]
is_active	[BOOLEAN] NOT NULL
mileage	[NUMERIC(18, 3)]
model	[VARCHAR(255)]
month_reference	[INTEGER]
price	[NUMERIC(18, 2)]
state	[VARCHAR(6)]
title	[VARCHAR(255)]
updated_at	[TIMESTAMP] NOT NULL
url	[VARCHAR(255)] NOT NULL
year_fabrication	[INTEGER]
year_model	[INTEGER]
year_reference	[INTEGER]

aerich	
id	[INTEGER] NOT NULL
app	[VARCHAR(100)] NOT NULL
content	[JSONB] NOT NULL
version	[VARCHAR(255)] NOT NULL

1--0..N

1--0..N

machine_table_average_ad_link	
id	[UUID] NOT NULL
core_table_average_id	[UUID] NOT NULL
core_table_id	[UUID] NOT NULL
created_at	[TIMESTAMP] NOT NULL
is_active	[BOOLEAN] NOT NULL
updated_at	[TIMESTAMP] NOT NULL

Caso de uso

Será apresentado um diagrama de caso de uso, junto à especificação funcional e à interface correspondente, ilustrando uma funcionalidade central do sistema.

Diagrama

Especificação

Sistema/Tela

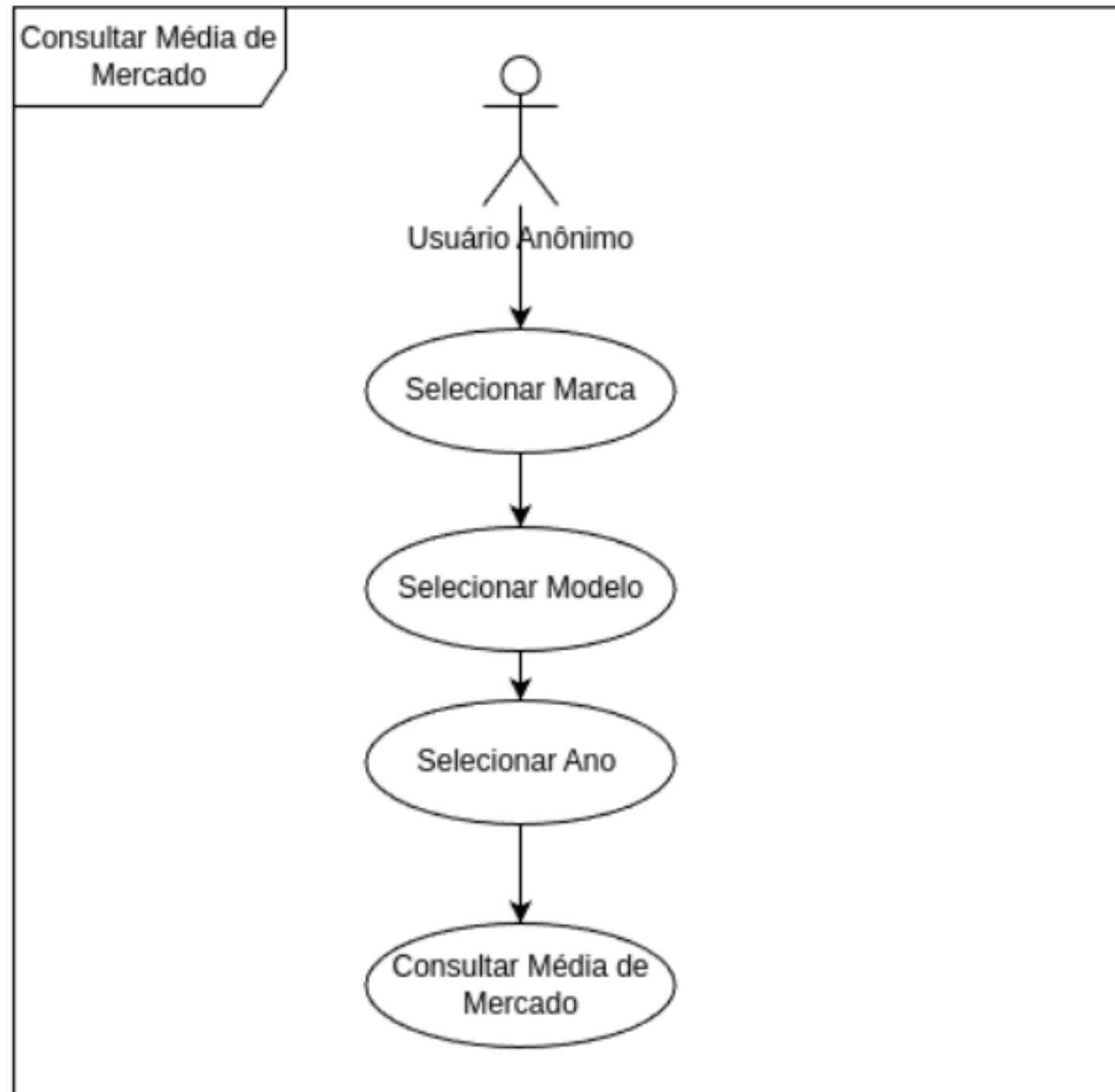


Diagrama
de Caso de
Uso

Especificação

Nome do Caso de Uso	UC01 - Consultar Médias de Mercado
Caso de Uso Geral	Permite que o usuário consulte as médias de preços de veículos a partir da seleção de uma marca, modelo e ano.
Ator Principal	Usuário anônimo
Atores Secundários	
Resumo	
Pré-condições	1. O sistema deve ter uma base de dados preenchida com marcas, modelos e anos. Portanto o processo ETL deve ter ocorrido previamente. Ou o backup do banco deve ter sido restaurado.
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
O Usuário acessa a interface da aplicação, na funcionalidade de consulta de médias.	
	O sistema apresenta uma lista de marcas disponíveis.
O Usuário seleciona uma marca.	
	O sistema apresenta uma lista de modelos associados à marca selecionada.
O Usuário seleciona um modelo.	
	O sistema apresenta uma lista de anos associados ao modelo selecionado.
O Usuário seleciona um ano.	
	O sistema retorna a média de preços para o veículo com os critérios escolhidos.
Restrições Validações	
Cenário Alternativo	
Ações do Ator	Ações do Sistema

Sistema



[Início](#) [Consultar](#) [Sobre o Produto](#) [Público Alvo e Aplicações](#) [Autenticação](#)

Consulta de carros

Obtenha a média de mercado de veículos com base em dados atualizados. Selecione a marca, o modelo e o ano do veículo para acessar informações precisas sobre os preços praticados, auxiliando na sua análise e tomada de decisão.

Pesquisa Comum



Marca:

HYUNDAI



Modelo:

HB20



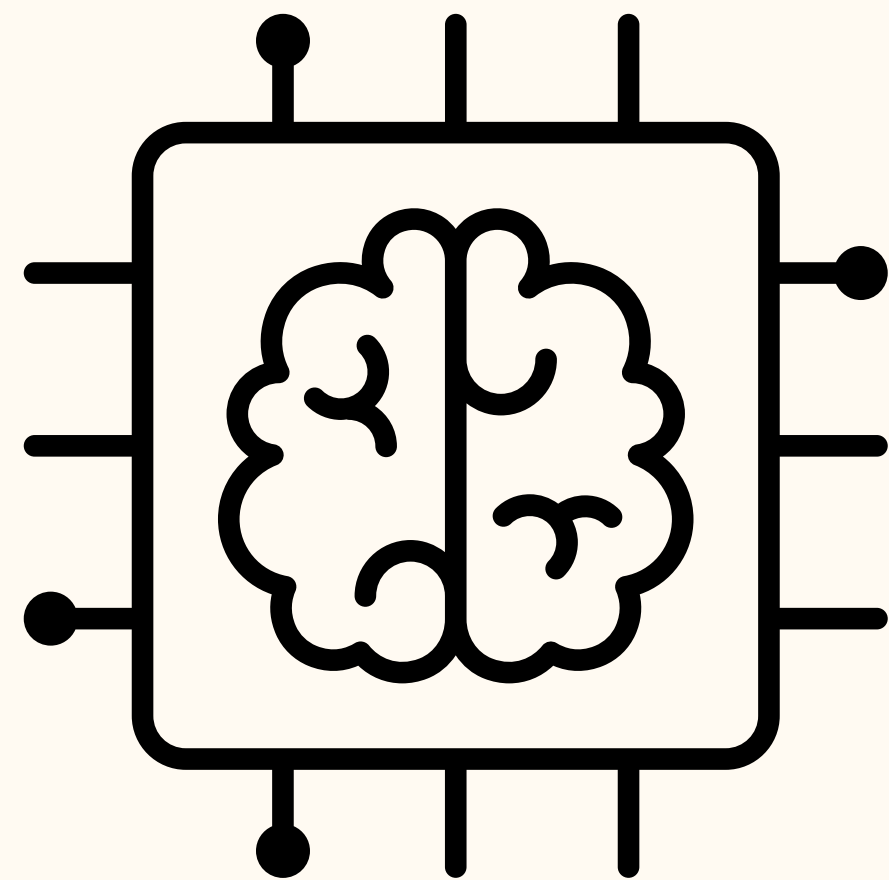
Ano:

2024



Pesquisar

Ano de referência:	2025
Mês de referência:	1
Código modelo:	MC00093
Ano modelo:	2024
Marca:	HYUNDAI
Modelo:	HB20
Média de mercado:	R\$ 84900



Inteligência Artificial


- QUAL FUNCIONALIDADE FOI IMPLEMENTADA?
- COMO O CÁLCULO É FEITO?
- COMO O MODELO DE MACHINE LEARNING TREINADO É USADO?



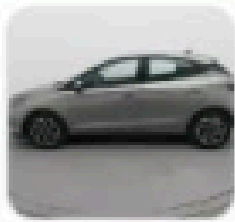

Conhecendo os dados

Hyundai HB20

1.0 COMFORT PLUS TURBO Á/G AUTOMÁTICO

Feature = Variável independente





ANO	QUILOMETRAGEM	COMBUSTÍVEL	CÂMBIO
2023/2024	48.443km	Á/G	Automático
CARROCERIA	CÓDIGO	COR	CIDADE
Hatch	5230274	Prata	Betim - MG

OPCIONAIS

Alarme, Ar Condicionado, Freio ABS, Limp e/ou Desemb, Trava Elétrica, Direção Elétrica, Controle eletrônico de estabilidade

Favoritos

Compartilhar

R\$ 89.290,00

Ver parcelas

Nome

Target = Variável Alvo, dependente

E-mail

Telefone

Chamar no Whatsapp

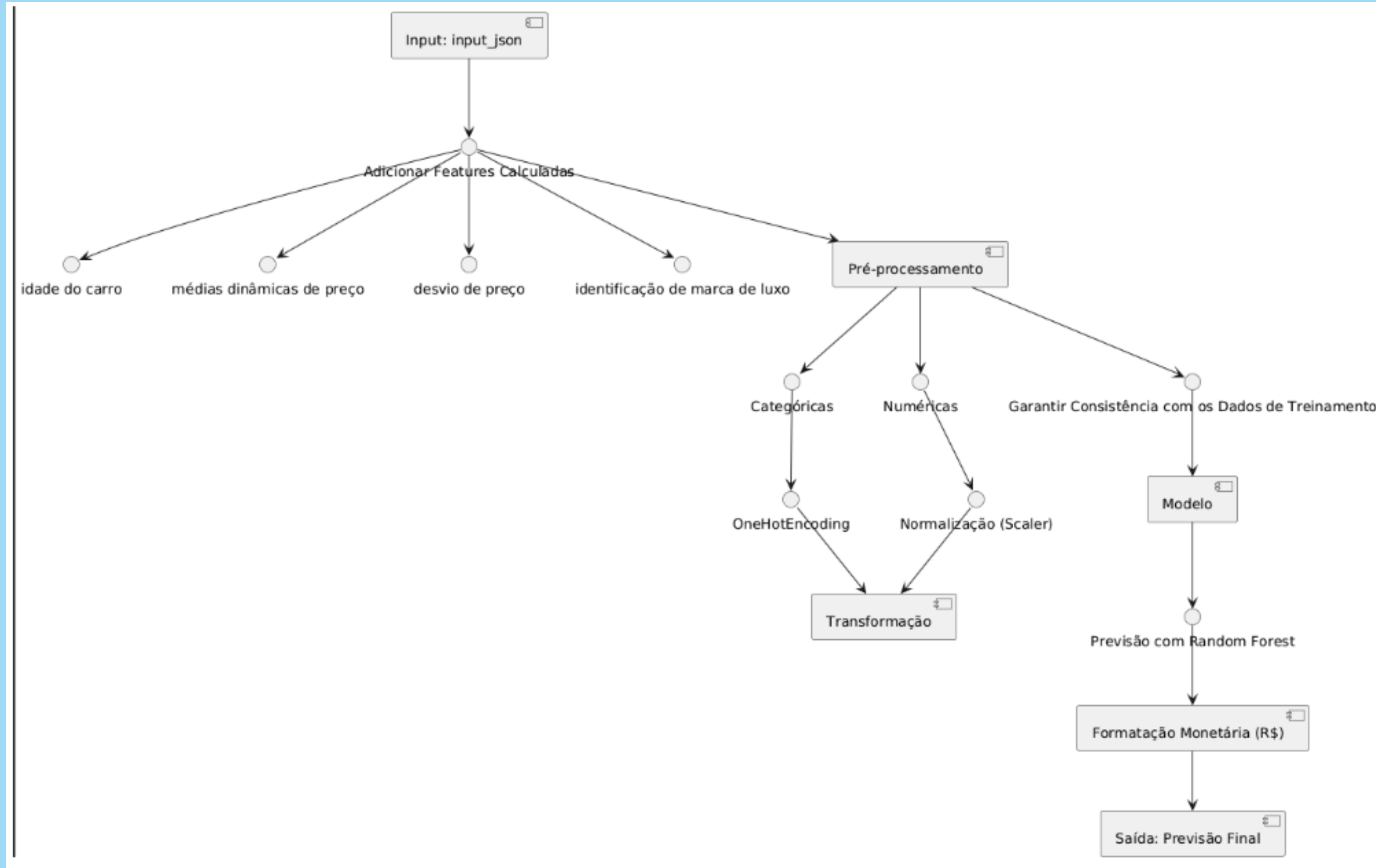
Enviar mensagem

Features = Variáveis independentes

Random Forest Regressor

BOOTSTRAP AGGREGATION (BAGGING)	<ul style="list-style-type: none">• O modelo cria vários subconjuntos aleatórios do dataset de treino (com reposição).• Cada subconjunto treina uma árvore de decisão diferente.
ÁRVORES INDEPENDENTES	<ul style="list-style-type: none">• Cada árvore é treinada com uma amostra diferente de dados e features (subamostragem de colunas).• Isso reduz o overfitting comum em árvores individuais.
PREVISÃO FINAL	<ul style="list-style-type: none">• No momento da predição, o modelo consulta todas as árvores.• A saída final é a média das predições de todas as árvores.

Pipeline de dados



Conclusão

Objetivos

Desenvolvimento

Análise

O desenvolvimento do DriveIntel atingiu plenamente seu **objetivo geral** de criar uma ferramenta de inteligência de mercado voltada ao setor automotivo, capaz de coletar, analisar e interpretar dados de precificação de veículos por meio de automação, estatística e inteligência artificial.

Foram alcançados todos os **objetivos específicos**:

- A coleta e organização dos dados foi viabilizada por técnicas de web scraping, estruturando informações relevantes extraídas de anúncios online.
- A aplicação de modelos estatísticos e de IA permitiu identificar padrões de precificação e simular cenários com base em variáveis como modelo, ano e características do veículo.
- A geração de insights estratégicos possibilitou às empresas do setor maior precisão na tomada de decisões sobre compra, venda e definição de preços.

Como complemento à Tabela FIPE, o DriveIntel se estabelece como uma solução robusta e dinâmica, ampliando a capacidade analítica das empresas diante das constantes oscilações do mercado automotivo.