

Turma: 3ADJT – GP13

POSI<sup>TECH</sup>

DOUGLAS HENRIQUE CAMPOS BARBOSA

FABIO SHOITI HAMADA

JEFFERSON LUAN FERREIRA

JUAN PAULO PINTO RODRIGUEZ

RM352745 - PEDRO LUIS HERVENCIO DA SILVA

ARQUITETURA E  
DESENVOLVIMENTO JAVA

FASE 1 - PRIMEIROS PASSOS COM SPRING

**TECH CHALLENGE**

## Sumário

Empresa LocACar.....	4
Contexto.....	4
Estratégia.....	4
Organograma.....	5
Processos.....	5
Design Estratégico .....	7
Domínios e Subdomínios.....	7
Domain Storytelling.....	8
Aquisição de um Veículo .....	8
Venda de um Veículo.....	9
Reserva de um Veículo .....	10
Locação de um Veículo.....	11
Devolução de um Veículo .....	12
Plano de Manutenção Preventiva .....	13
Planejamento Mensal de Manutenção .....	14
Manutenção Preventiva de um Veículo .....	15
Event Storming .....	16
Aquisição e Vendas.....	16
Reserva e Locação .....	19
Manutenção e Reparo .....	21
Linguagem Ubíqua.....	23
Contexto Delimitado .....	25
Contextos.....	25
Mapa de Contexto.....	25
Design Tático .....	26
Contexto Delimitado: Aquisição e Venda .....	26
Arquitetura Aplicação Cotação.....	26
Agregados, Entidade e Objetos de Valor .....	27
Arquitetura Aplicação Compra .....	28
Agregados, Entidade e Objetos de Valor .....	29
Arquitetura Aplicação Venda .....	30
Agregados, Entidade e Objetos de Valor .....	31
Contexto Delimitado: Plano de Manutenção Preventiva .....	32
Arquitetura Aplicação PlanoManutenção .....	32
Agregados, Entidade e Objetos de Valor .....	33

Contexto Delimitado: Manutenção Preventiva .....	34
Arquitetura Aplicação Manutenção .....	34
Agregados, Entidade e Objetos de Valor .....	35
Contexto Delimitado: Planejamento Mensal .....	36
Arquitetura Aplicação Planejamento Mensal.....	36
Agregados, Entidade e Objetos de Valor .....	37
Contexto Delimitado: Reserva.....	38
Arquitetura Aplicação Reserva .....	38
Agregados, Entidade e Objetos de Valor .....	39
Contexto Delimitado: Locação .....	40
Arquitetura Aplicação Locação.....	40
Agregados, Entidade e Objetos de Valor .....	41
Implementação .....	42
Configurações do Hibernate.....	42
Considerações Finais .....	43
Técnicas Utilizadas.....	43
Design Patterns Utilizados.....	43
Reflexões sobre o Processo de Levantamento de Requisitos.....	43
Benefícios Esperados.....	43
Ferramentas Utilizadas.....	45
Execução de Fluxos.....	46
Aquisição e Venda .....	46
Aplicação Cotação .....	46
Aplicação Compra.....	49
Aplicação Venda .....	58

## Empresa LocACar

### Contexto

A locadora LocACar é uma pequena empresa familiar de locação de automóveis, atualmente seus processos são manuais e registrados em livros.

Atende a zona oeste de São Paulo e sua frota é de 30 automóveis.

A família prevê tempos difíceis e para enfrentar esta situação pretende:

- A) Mecanizar e informatizar processos para reduzir custos e melhorar a gestão e
- B) expandir seus negócios para fora do país.

### Missão:

Ajudar pessoas e famílias em seus deslocamentos com liberdade, segurança e praticidade.

### Visão:

Ser a primeira opção para cliente que conheçam seus serviços.

### Valores:

- Integridade e honestidade;
- Qualidade é melhor do que quantidade
- Flexibilidade e agilidade

### Estratégia

A estratégia para alcançar estes objetivos é evoluir gradualmente, mas com a visão de longo prazo em que cada passo prepara o próximo e o mais importante, não inviabiliza o objetivo final.

1. Informatizar os processos internos;
2. Portar o sistema para os idiomas inglês e espanhol e as moedas Dólar e Euro;
3. Implementar o índice Bic Mac para permitir comparar a rentabilidade das filiais e
4. criação de um Website e App para clientes

## Organograma



### Direção

Avalia e define os rumos da empresa. Aprova a contratação de pessoal, compra e venda de veículos e contratação de serviços vultuosos como seguro da frota.

### Front Office (Atendimento e Vendas)

É o pessoal que atende, interage e conhece o cliente. Recebe e cadastra o cliente, aluga o veículo, também acompanha a devolução e realiza a cobrança. Este é o setor responsável por elaborar novas campanhas e prospectar novos cliente.

### BackOffice (Administrativo, Contabilidade, Gestão de Pessoas)

É o pessoal que realiza a contabilidade, prepara os relatórios de performance da empresa, cuida de assuntos relacionados ao fisco. Também prepara os processos para aquisições de novos veículos, venda dos antigos, processos com a seguradora e oficina de reparos (externas) e eventuais processos judiciais.

### Oficina

É o pessoal que inspeciona o carro em uma devolução (se acionado pelo atendente). Realiza pequenos reparos: troca de setas e lanternas, polimento e até pequenos retoques. Faz a lavagem e limpeza do veículo. Responsável pela manutenção como troca de óleo, filtros, rodízio de pneus, mesmo que seja realizada em uma oficina parceira.

## Processos

- Reserva de um veículo;
- Cancelamento da reserva de um veículo;
- Locação de um veículo;
- Devolução de um veículo;
- Aquisição de um veículo;
- Venda de um veículo;
- Reparo de um veículo;
- Planejamento mensal de manutenção;
- Manutenção preventiva de um veículo;

- Credenciamento de um parceiro;
- Descredenciamento de um parceiro;
- Acidente com veículo durante a locação;
- Falha mecânica com veículo durante a locação;

## Design Estratégico

### Domínios e Subdomínios

**Domínio da Empresa:** Locação de Veículos

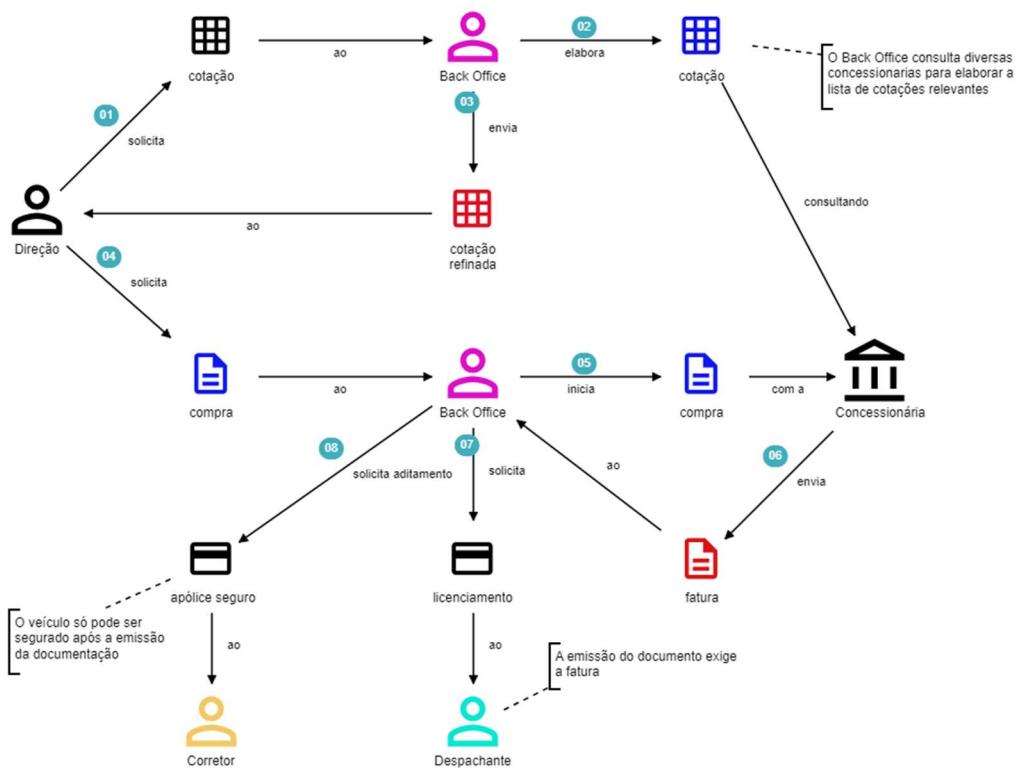
<b>Tipo do Subdomínio</b>	<b>Subdomínio</b>
Subdomínios Principais	Reserva de Veículos Locação de Veículos Aquisição e Venda de Veículos Manutenção e Reparo de Veículos
Subdomínio de Suporte	Integração com sistema de Contabilidade Credenciamento de Parceiros
Subdomínios Genéricos	Autenticação e Permissionamento de Usuários Gestão de Pessoas (RH) Contabilidade

## Domain Storytelling

### Aquisição de um Veículo

Periodicamente a Direção avalia diversos fatores como desgaste da frota, mercado de veículos novos, expectativa da demanda por locação e decide adquirir veículos.

A Direção define quantos veículos e quais tipos serão adquiridos e solicita ao Backoffice uma cotação. O Backoffice consulta as diversas concessionárias e revendas para montar a lista com as cotações, descarta as mais discrepantes e devolve à direção. Com estas informações a Direção escolhe o fornecedor e pede ao Backoffice que inicie o processo de compra. Quando o veículo estiver faturado o Backoffice solicita ao despachante que elabore a documentação. Quando a documentação estiver pronta o backoffice solicita ao corretor de seguros o aditamento da apólice de seguros com o novo veículo. Isto finaliza o processo de aquisição e libera o veículo para locação. O Backoffice também sinaliza a Oficina para que elabore o Plano de Manutenção deste veículo.



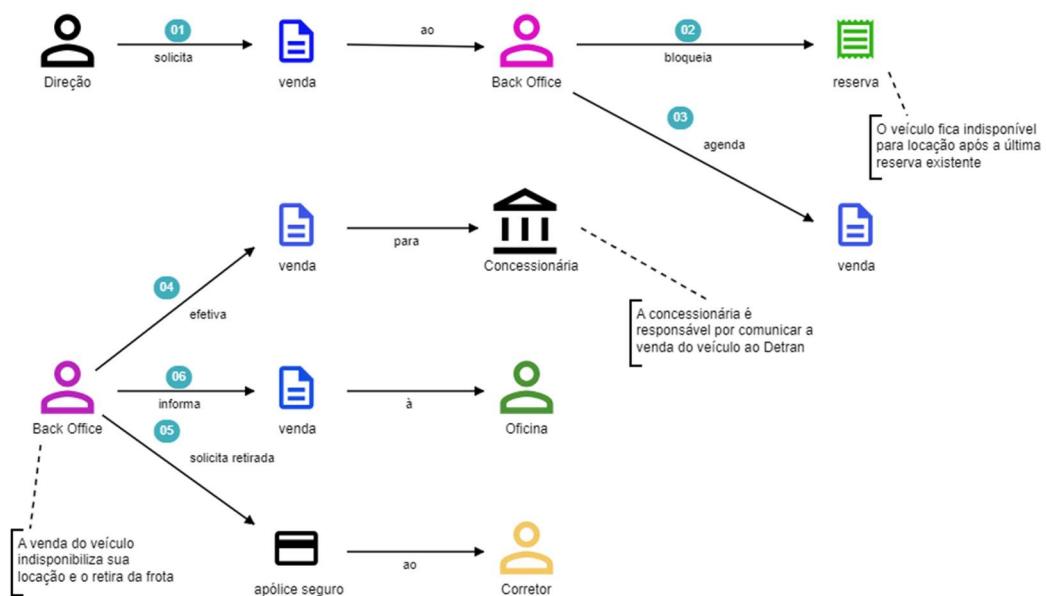
## Venda de um Veículo

Periodicamente a Direção avalia diversos fatores como desgaste da frota, mercado de veículos usados, expectativa da demanda por locação e decide vender veículos.

A Direção define quais veículos serão vendidos e solicita a venda ao Back Office.

O Backoffice verifica as reservas de locação para aquele veículo, bloqueia sua disponibilidade para locações posteriores e programa o início do processo de venda para fim da última locação.

Na data agendada o Back Office realiza a venda do veículo, isto indisponibiliza definitivamente sua locação e a Oficina é informada para cancelar quaisquer agendamentos de manutenção.

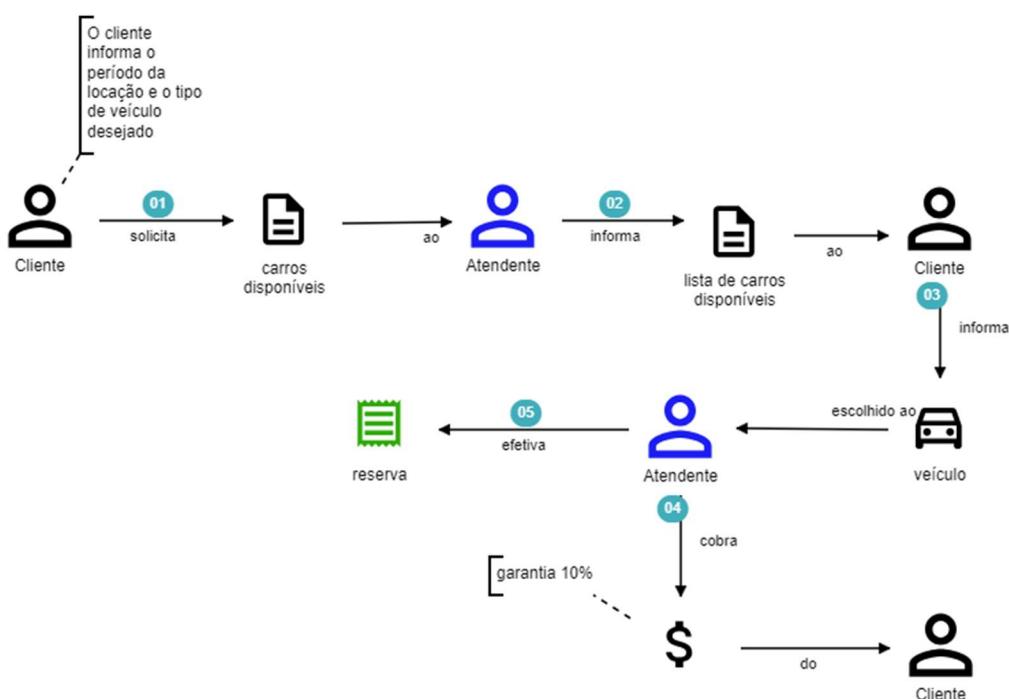


## Reserva de um Veículo

Um cliente liga para a loja da empresa para fazer uma reserva, deve informar o período da locação e o tipo do veículo (P/M/G ou Utilitário). O atendente verifica quais veículos estão disponíveis neste período e informa a lista ao cliente. É comum que não haja disponibilidade ou que o cliente não goste das opções e novas pesquisas podem ser feitas.

Com a escolha do veículo é calculado o valor da locação e é solicitado uma garantia de 10% da locação. O mesmo processo pode ser realizado presencialmente no balcão.

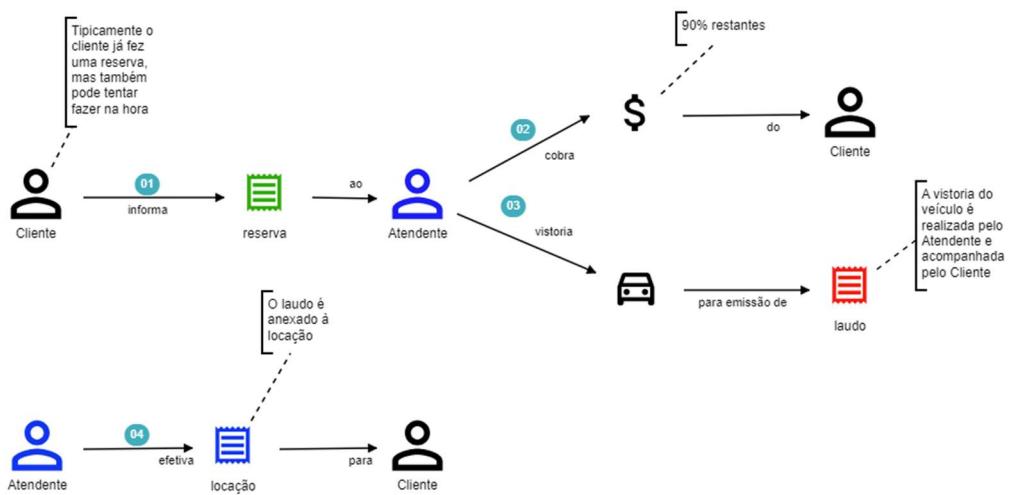
É alertado ao cliente que o veículo só será entregue a um condutor devidamente habilitado.



## Locação de um Veículo

Um cliente se desloca fisicamente para a loja da empresa para alugar o veículo, tipicamente já há uma reserva em seu nome, mas há casos em que a locação é realizada diretamente.

O atendente localiza a reserva, registra os dados do condutor do veículo e cobra os 90% restantes. Antes de entregar o carro o atendente vistoria junto com o cliente o veículo para identificar e registrar eventuais riscos, pequenas avarias, quilometragem e quantidade de combustível. Também são verificados itens e acessórios como macaco, estepe e triangulo. Esta vistoria gera um laudo.



## Devolução de um Veículo

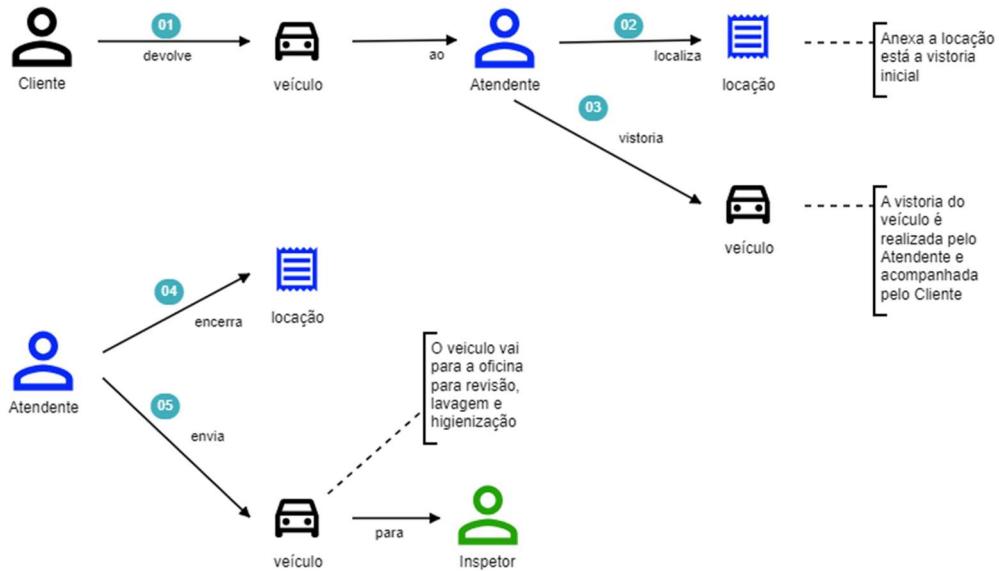
Um cliente se desloca fisicamente para a loja da empresa para devolver o veículo.

O atendente localiza a locação e o laudo e junto com o cliente vistoria novamente o veículo para identificar novos riscos, avarias e se estão faltando itens e acessórios.

Caso novos riscos ou avarias sejam encontradas serão cobradas do seguro ou do cliente pelo setor de Back Office. Este setor também realizará a cobrança de eventuais itens faltantes.

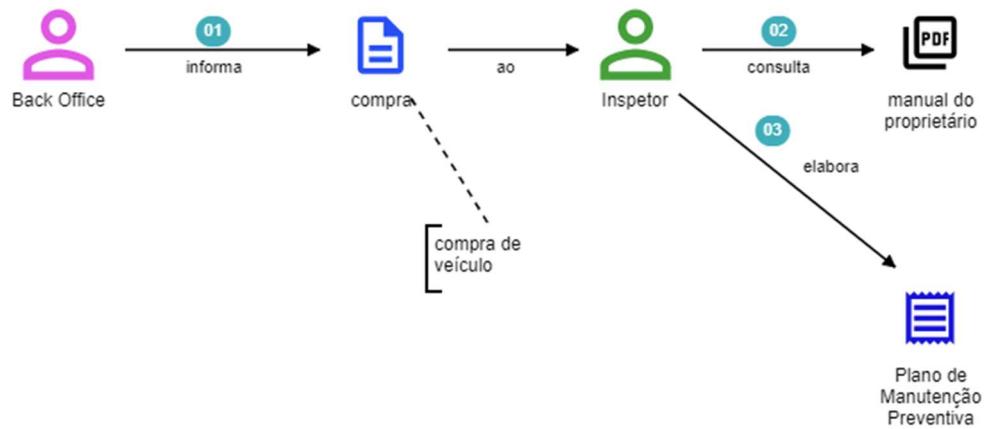
Caso o veículo seja devolvido com menos combustível que entregue serão cobrados neste momento os “quartos” faltantes.

Após a devolução o veículo sempre volta para a oficina para uma revisão mais detalhada, lavagem e higienização, eventuais reparos como riscos que saem com polimento, verificar a parte mecânica, calibrar pneus, completar reservatório de água do para-brisa etc. Tipicamente o veículo fica indisponível por 03 horas.



## Plano de Manutenção Preventiva

Quando um veículo é adquirido (pelo Back Office) a Oficina monta o seu Plano de Manutenção Preventiva de acordo com o Manual do Proprietário. Este plano acompanha toda a vida útil do veículo e orienta e registra as diversas manutenções e reparos que o veículo passa. Serve de base para o Planejamento Mensal de Manutenção.



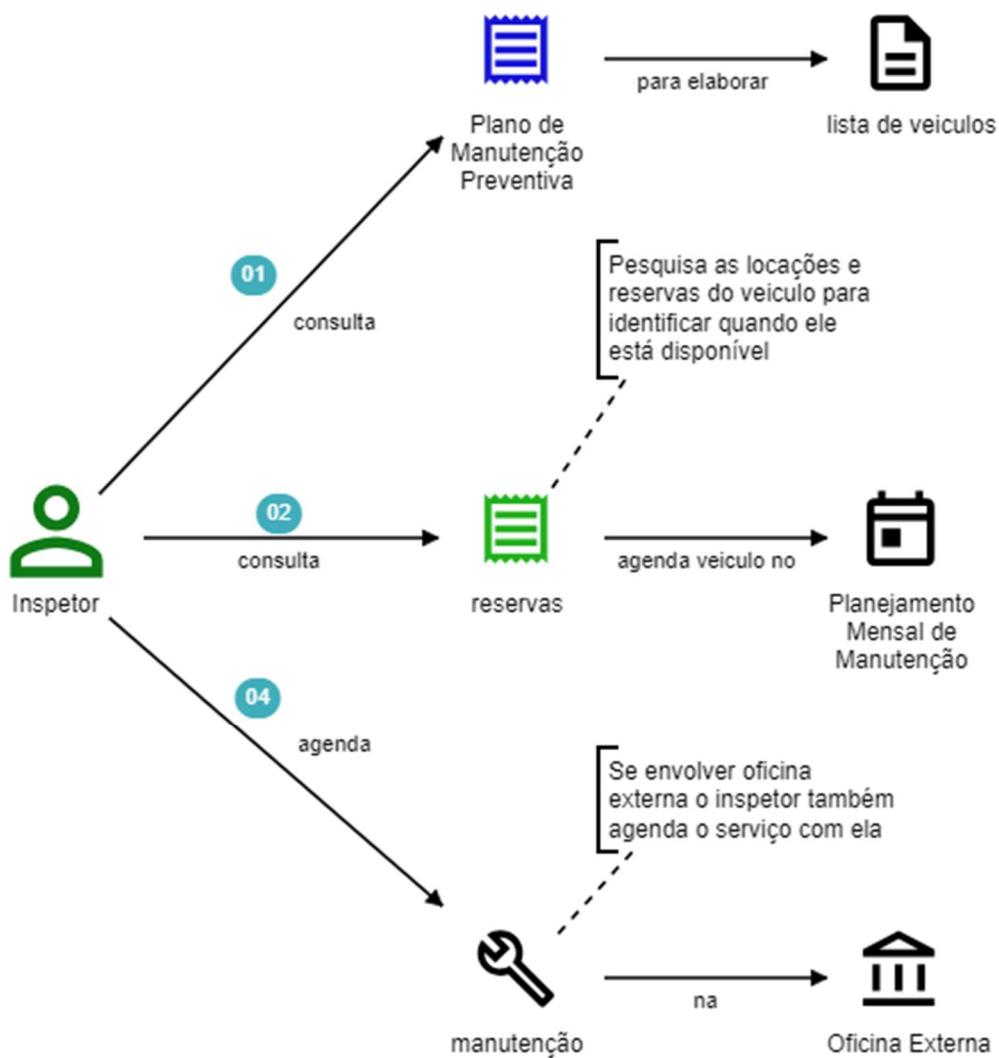
## Planejamento Mensal de Manutenção

Mensalmente o inspetor da oficina elabora uma lista dos veículos que devem passar pela manutenção preventiva do próximo mês baseando-se no Plano de Manutenção Preventiva de cada veículo. São considerados tanto a quilometragem como o tempo dos itens, por exemplo troca de óleo, 10.000 Km ou 1 ano, o que vencer primeiro.

Com esta lista ele pesquisa quando o veículo não está agendado para algum cliente e reserva a data (período) para manutenção.

Se a manutenção for realizada por uma oficina mecânica externa ele também agenda este serviço com ela, sempre conciliando com a disponibilidade do veículo.

Há serviços rápidos como troca de óleo que dispensam o agendamento prévio.



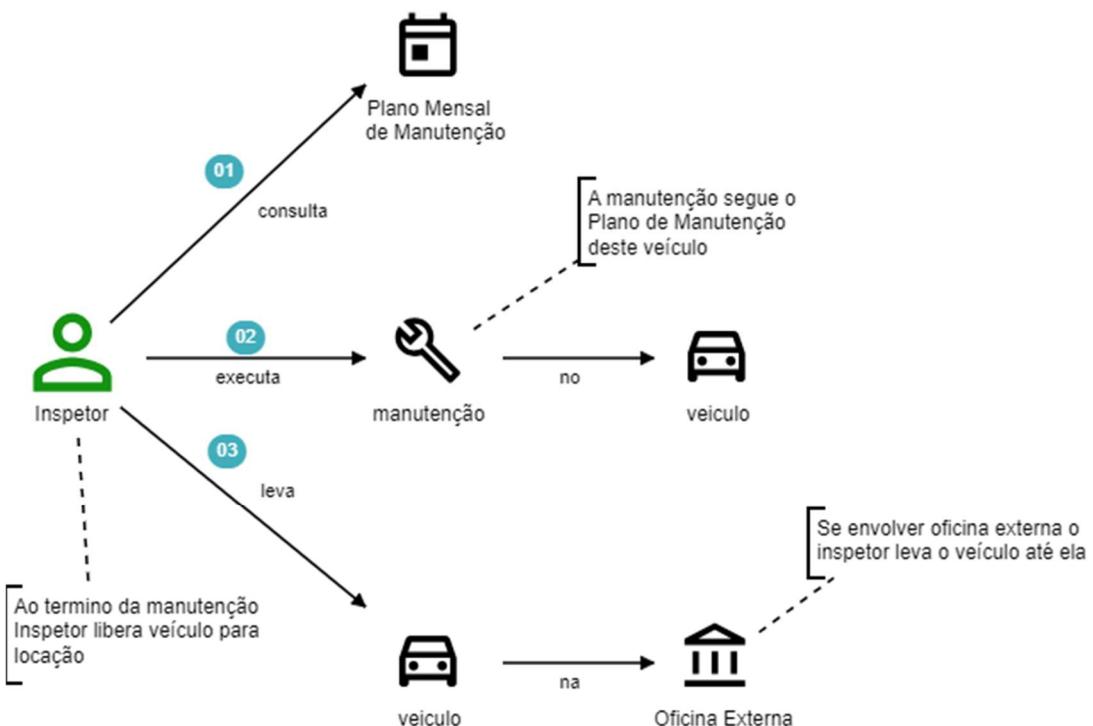
## Manutenção Preventiva de um Veículo

A Manutenção Preventiva foi agendada previamente no Planejamento de Manutenção Mensal.

Quando chega a data agendada o Inspetor faz a manutenção do veículo obedecendo o Plano de Manutenção daquele veículo. Se necessário ele leva o veículo até uma oficina externa conforme agendamento realizado no Planejamento de Manutenção Mensal.

Há serviços rápidos como troca de óleo que dispensam o agendamento prévio.

Ao término da manutenção o Inspetor libera o veículo para locação.



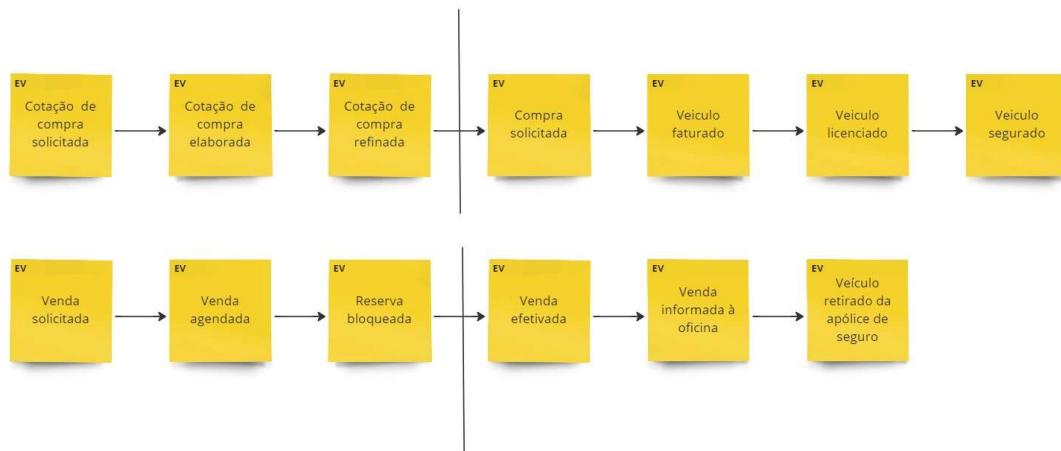
## Event Storming

## Aquisição e Vendas

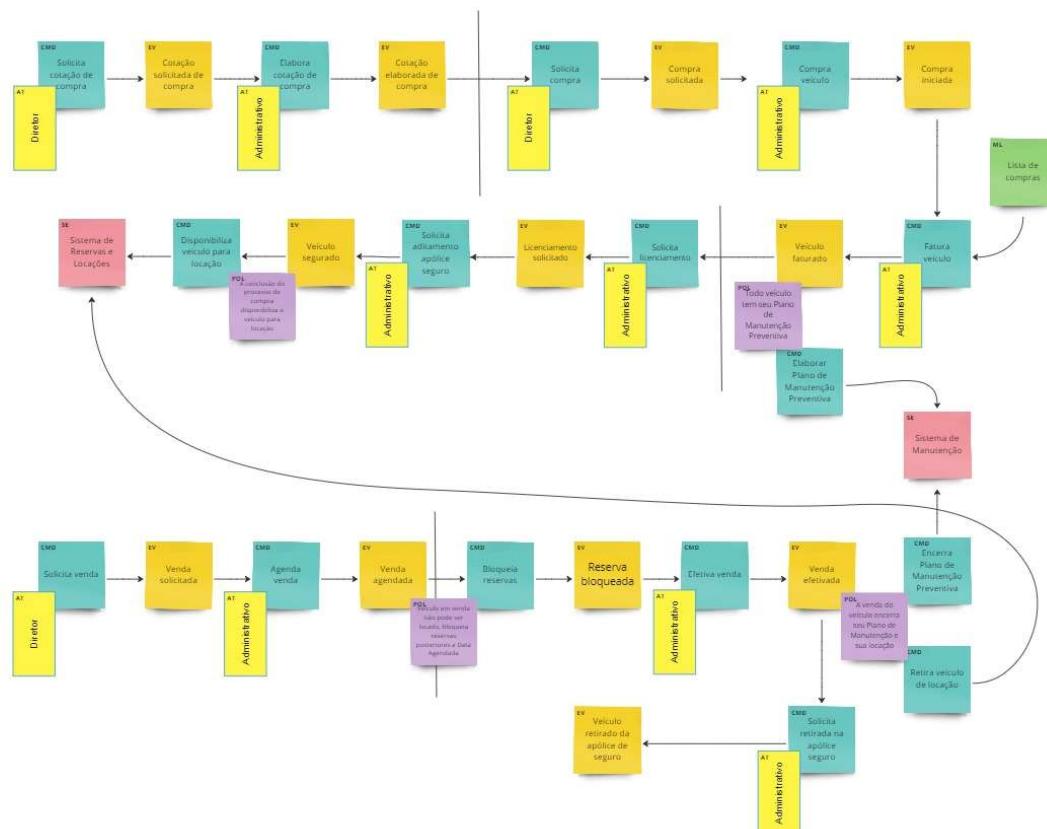
**Brainstorming**

<b>EV</b> Cotação de compra solicitada	<b>EV</b> Cotação de compra elaborada	<b>EV</b> Cotação de compra refinada	<b>EV</b> Compra solicitada
<b>EV</b> Veiculo Faturado	<b>EV</b> Veiculo Licenciado	<b>EV</b> Veiculo Segurado	
<b>EV</b> Venda solicitada	<b>EV</b> Reserva bloqueada	<b>EV</b> Venda agendada	<b>EV</b> Oficina informada
<b>EV</b> Veiculo vendido	<b>EV</b> Veículo retirado da apólice de seguro		

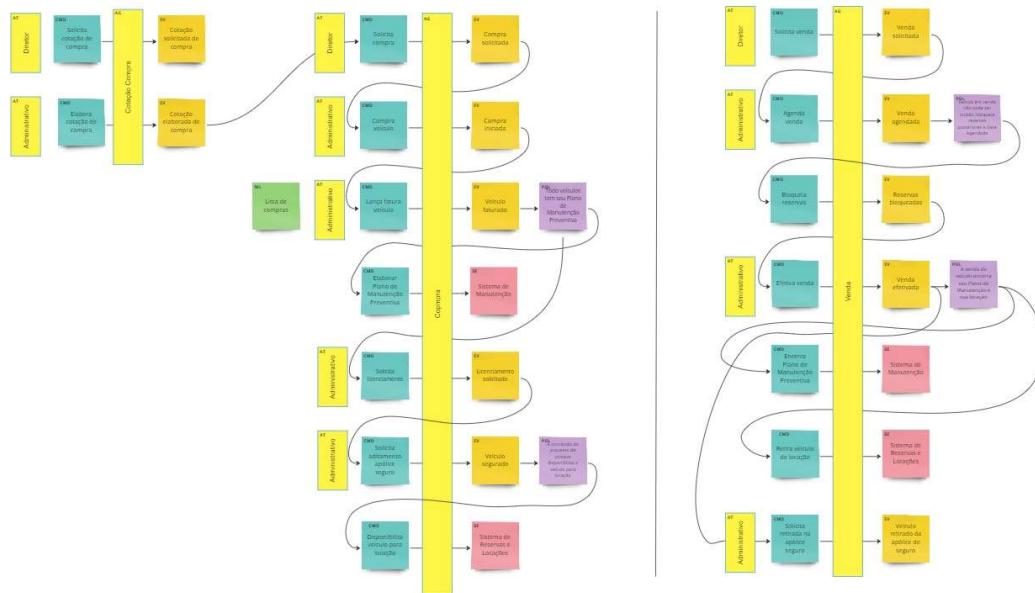
## Ordenação



## Comandos e Modelos de Leitura



## Aggregados



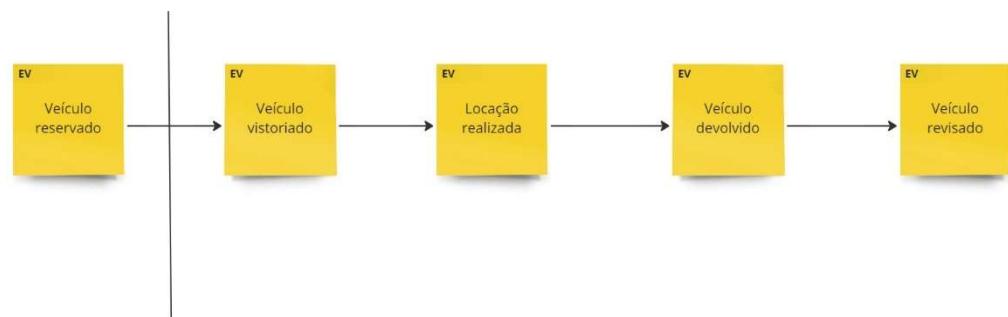
## Reserva e Locação

## Brainstorming



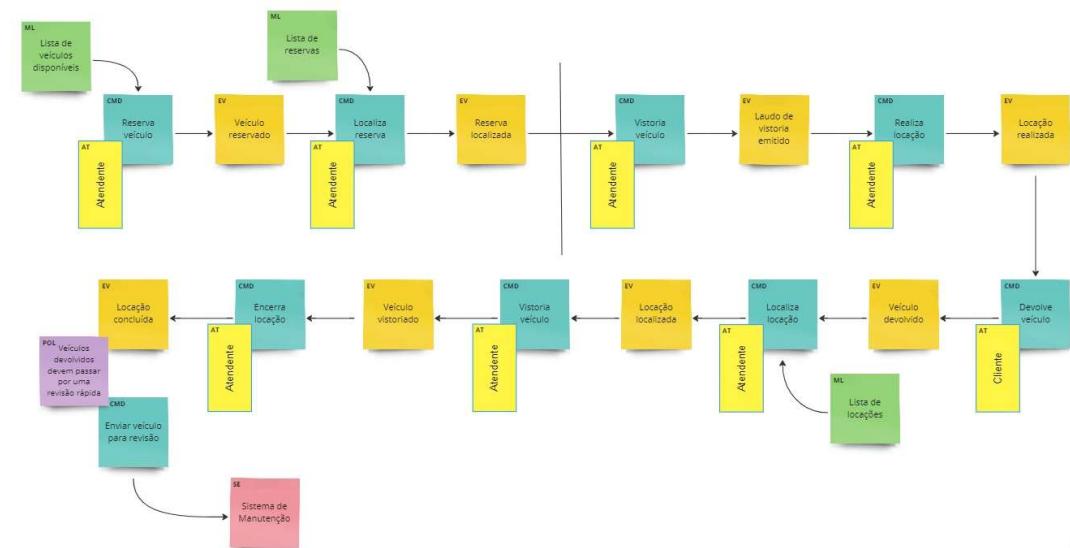
miro

## Ordenação



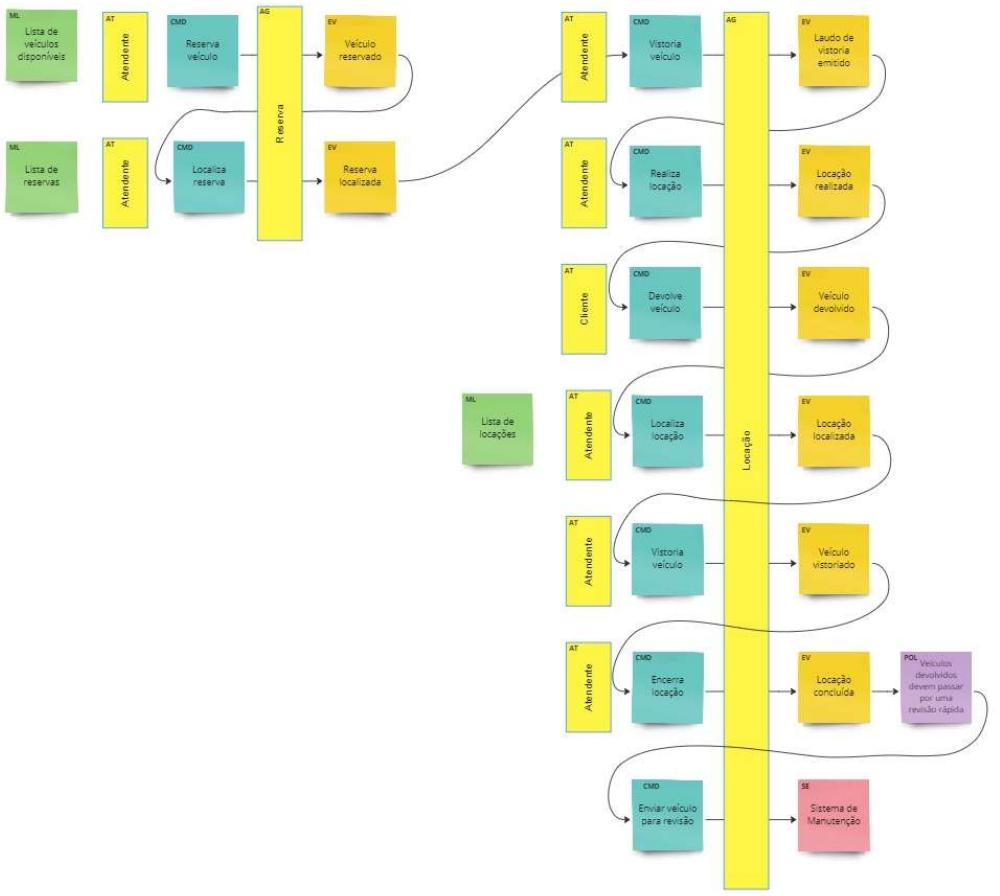
miro

## Comandos e Modelos de Leitura



miro

## Aggregados



miro

## Manutenção e Reparo

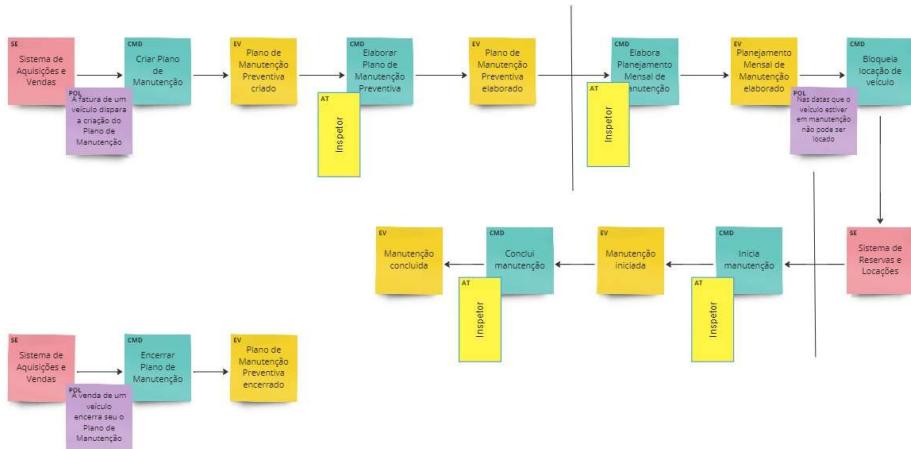
### Brainstorming



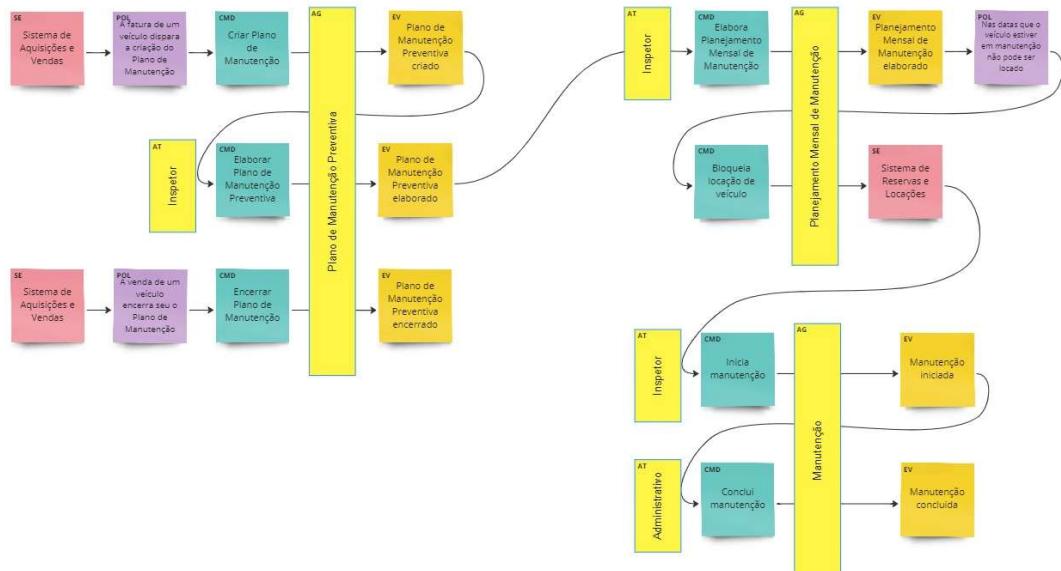
### Ordenação



## Comandos e Modelos de Leitura



## Aggregados



## Linguagem Ubíqua

**Aditamento:** Aditamento em uma apólice de seguros, é a inclusão de um novo veículo na apólice de seguros da empresa.

**Administrativo:** É o profissional que trabalha no Back Office, responsável por processos internos da empresa.

**Agendamento:** É a marcação de um evento em determinada data

**Apólice de Seguro:** É o seguro da empresa, ela segura toda a frota de veículos, quando carros são comprados ela é aditada, quando carros são vendidos são retirados da apólice.

**Atendente:** É o profissional que trabalha no Front Office e atende aos clientes.

**Categoria:** É a classificação do veículo segundo seu tamanho e utilidade. Veículos são classificados em Pequenos, Médios, Grandes ou Utilitários. É utilizada para cálculo do valor da diária.

**Concessionária:** São as empresas vendem veículos e ou fornecem serviços de manutenção.

**Devolução:** É o retorno de um veículo que foi locado.

**Diária:** É a unidade de cobrança da locação de veículos, corresponde ao período de 24 horas entre a retirada e devolução do veículo

**Fatura:** É o documento emitido pela concessionária na compra de um veículo.

**Fornecedor:** É o fornecedor de veículos no processo de aquisição

**Frota:** É o conjunto dos veículos que a empresa possui.

**Garantia:** É o valor correspondente a 10% da Locação do veículo cobrado no momento da Reserva.

**Inspetor:** É o profissional que trabalha na Oficina responsável pela revisão e manutenção dos veículos

**Intervalo Km:** É a quantidade de quilômetros em que cada componente (peça) precisa ser revista, reparada ou trocada.

**Laudo:** É o resultado de uma vistoria de um veículo antes da entrega ao cliente e registra eventuais riscos, danos, ruídos. Permite identificar se o cliente foi ou não responsável por eventuais danos.

**Locação:** É o aluguel do veículo, o único tipo de locação disponível é Diária com Km Livre. As lojas funcionam das 06h00 às 20h00, período disponível para retirada e devolução do veículo.

**Manutenção Preventiva:** É a verificação e ou reparo que cada veículo passa regularmente seguindo o seu Plano de Manutenção Preventiva montado pelo pessoal da oficina.

**Marca:** Corresponde aos fabricantes de veículos, como Honda, Toyota, Nissan, Ford.

**Modelo:** É o tipo de veículo que as marcas comercializam, como Civic e Fit.

**Operação:** É a ação a ser tomada em uma manutenção preventiva sobre determinado componente, como limpar filtro de ar, trocar filtro de óleo, verificar fluido de freios.

**Planejamento Mensal de Manutenção:** É o agendamento das manutenções que ocorreram em determinado mês, em sua elaboração são consideradas: disponibilidade do veículo, agendamento com as concessionárias (oficinas externas) e o plano de manutenção de cada veículo.

**Plano de Manutenção Preventiva:** É o plano elaborado seguindo as instruções do fabricante (Manual do Proprietário), para revisões, trocas e reparos de componentes em determinado intervalo de quilometragem. Inclui itens como rodízio de pneus, troca de óleo, filtros, bateria, correias dentadas.

**Proposta:** É a oferta de venda de um veículo que uma concessionária (fornecedor) faz à empresa.

**Reserva:** É o agendamento de uma locação garantindo a disponibilidade do veículo naquele período. Neste momento é cobrada uma garantia de 10% da locação.

**Tipo do Veículo:** É a classificação dos veículos em Pequenos, Médios, Grandes ou Utilitários para cálculo do valor da diária.

**Veículo:** - A empresa trabalha apenas com a locação de veículos de passeio e caminhonetes leves (pick-ups).

**Vistoria:** É a verificação que Atendente e Cliente fazem no veículo antes de sua entrega para identificar eventuais riscos, amassados danos e ruídos. Neste momento também são anotadas a quilometragem e quantidade de combustível. Este processo gera o laudo que será anexado à Locação.

**Nota:** Apesar da Linguagem Ubíqua considerar cada domínio, dado o tamanho da empresa não foram encontradas divergências entre as diversas áreas, resultando em um dicionário único.

## Contexto Delimitado

### Contextos

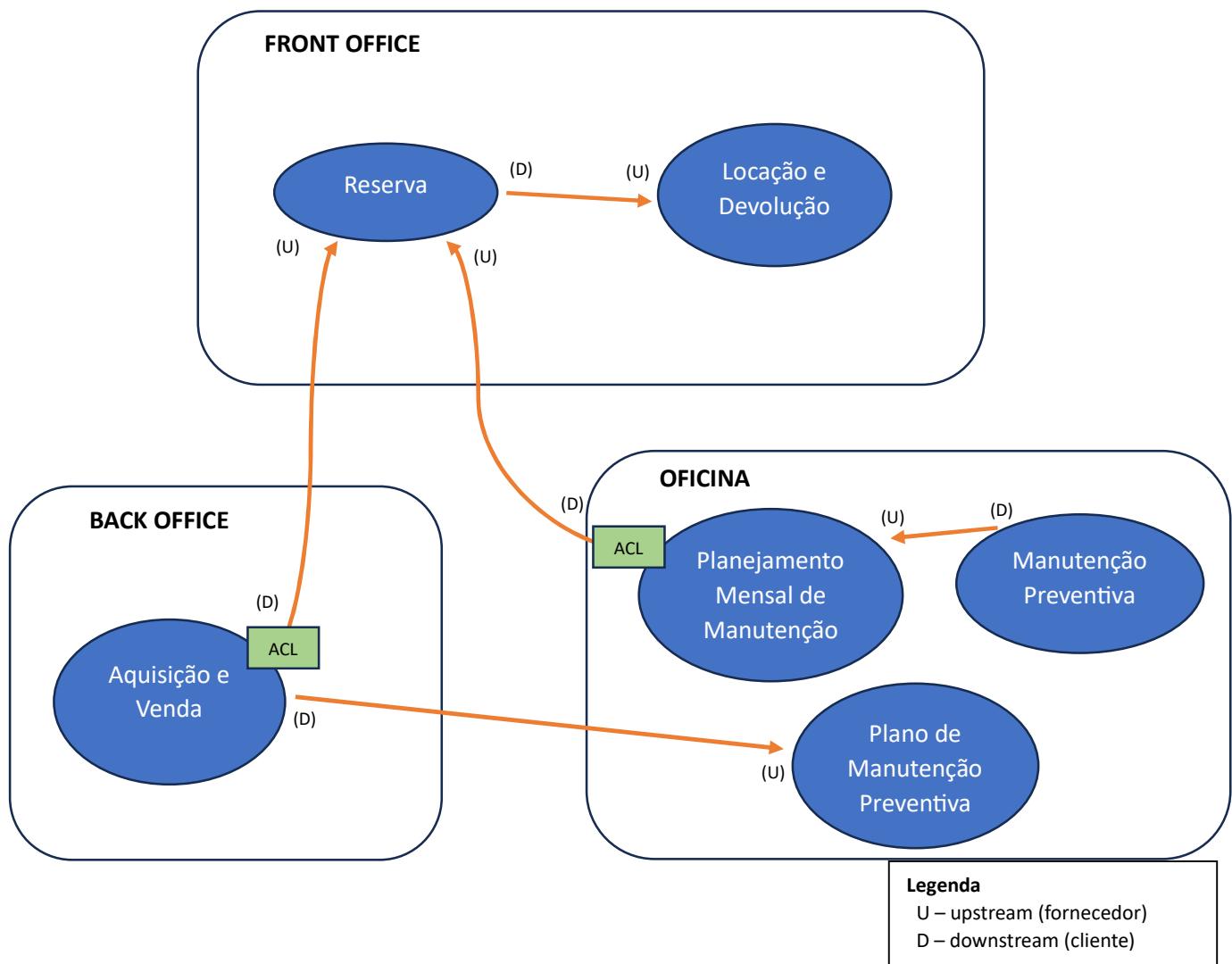
Refinando o entendimento e baseado na Linguagem Ubíqua mapeamos os seguintes Contextos Delimitados:

#### Domínio da Empresa: Locação de Veículos

Áreas	Contexto Delimitado
Front Office	Reserva de Veículos
	Locação e Devolução de Veículos
Back Office	Aquisição e Venda de Veículos
Oficina	Plano de Manutenção Preventiva
	Planejamento Mensal de Manutenção
	Manutenção Preventiva de um Veículo

### Mapa de Contexto

#### Mapa de Contexto da Empresa: Locação de Veículos



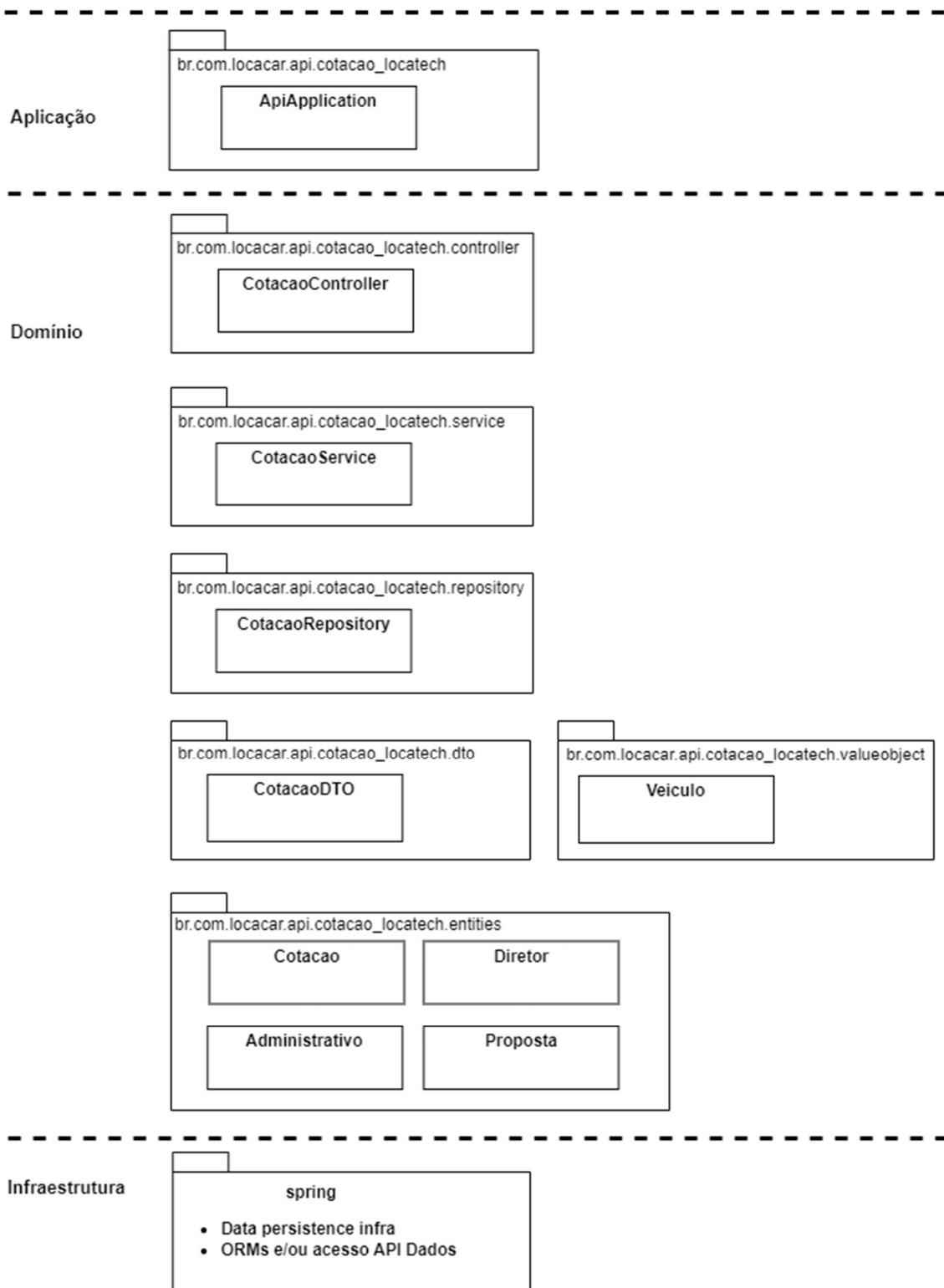
## Design Tático

Contexto Delimitado: Aquisição e Venda

Arquitetura Aplicação Cotação

*Aplicação Cotação*

Interface do Usuário



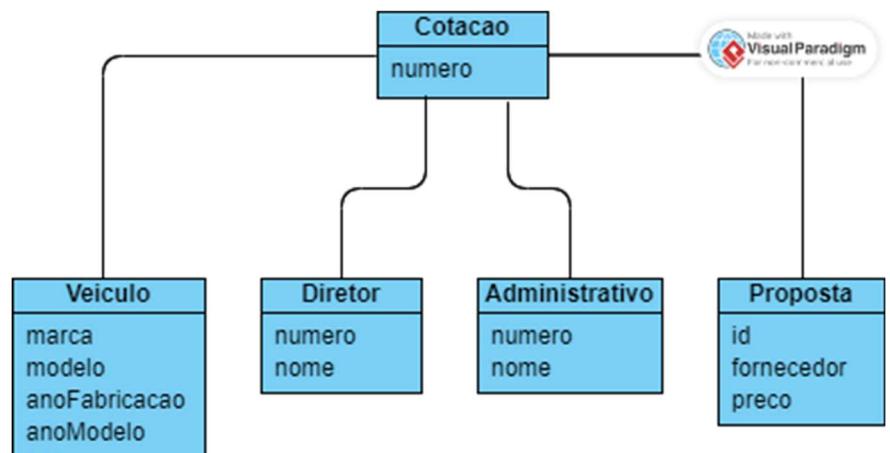
## Agregados, Entidade e Objetos de Valor

*Agregado Cotação*Entidades

- Cotação
- Diretor
- Administrativo
- Proposta

Objetos de Valor

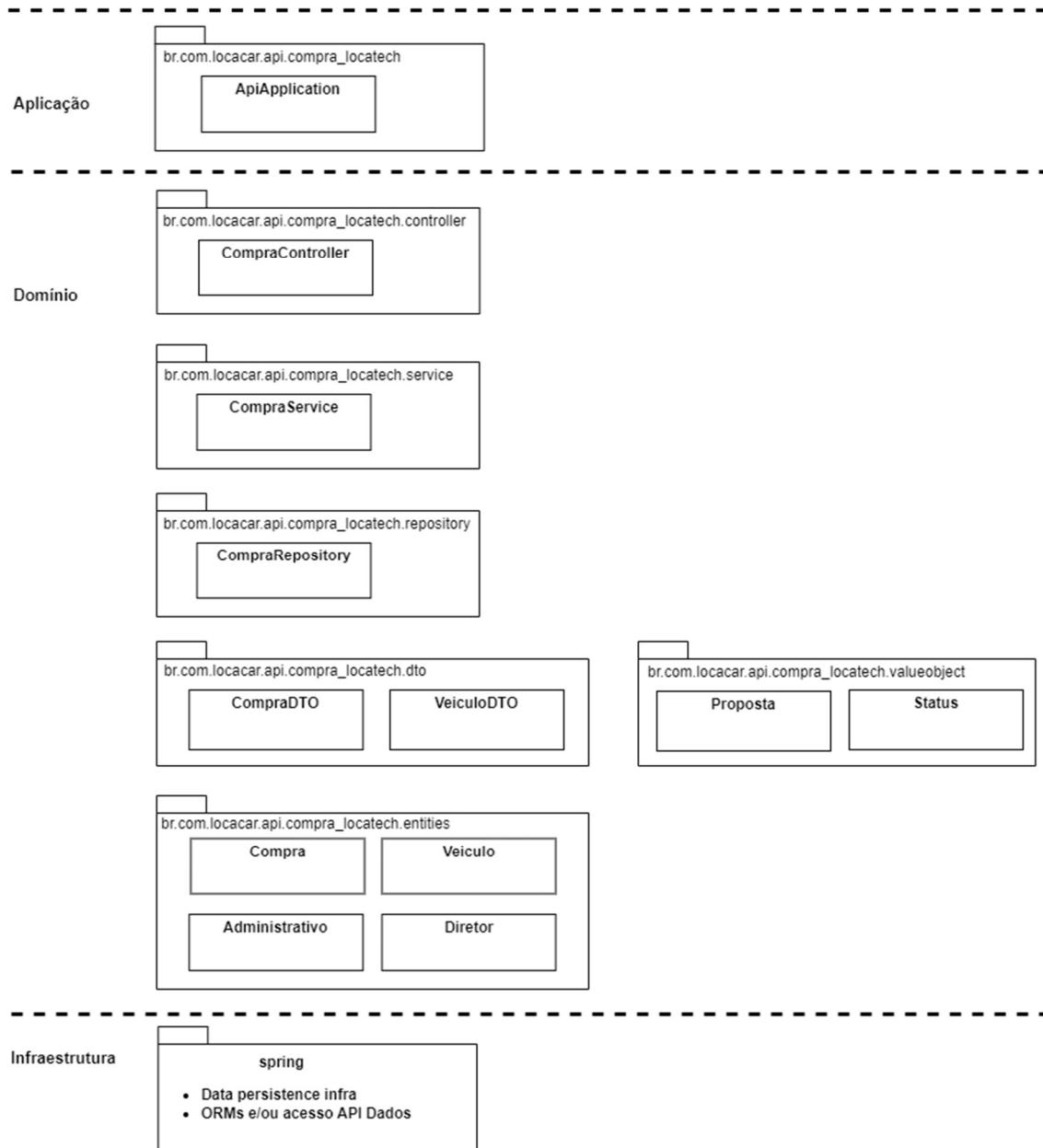
- Número
- Código Funcional
- Nome
- Veículo
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor
- Fornecedor
- Preço



## Arquitetura Aplicação Compra

*Aplicação Compra*

Interface do Usuário



Agregados, Entidade e Objetos de Valor

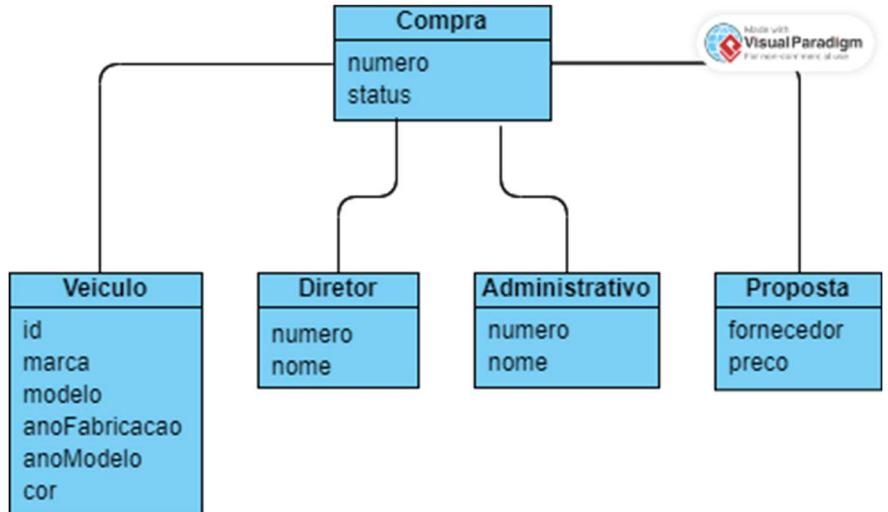
### *Aggregado Compra*

#### Entidades

- Compra
- Diretor
- Administrativo
- Veículo

#### Objetos de Valor

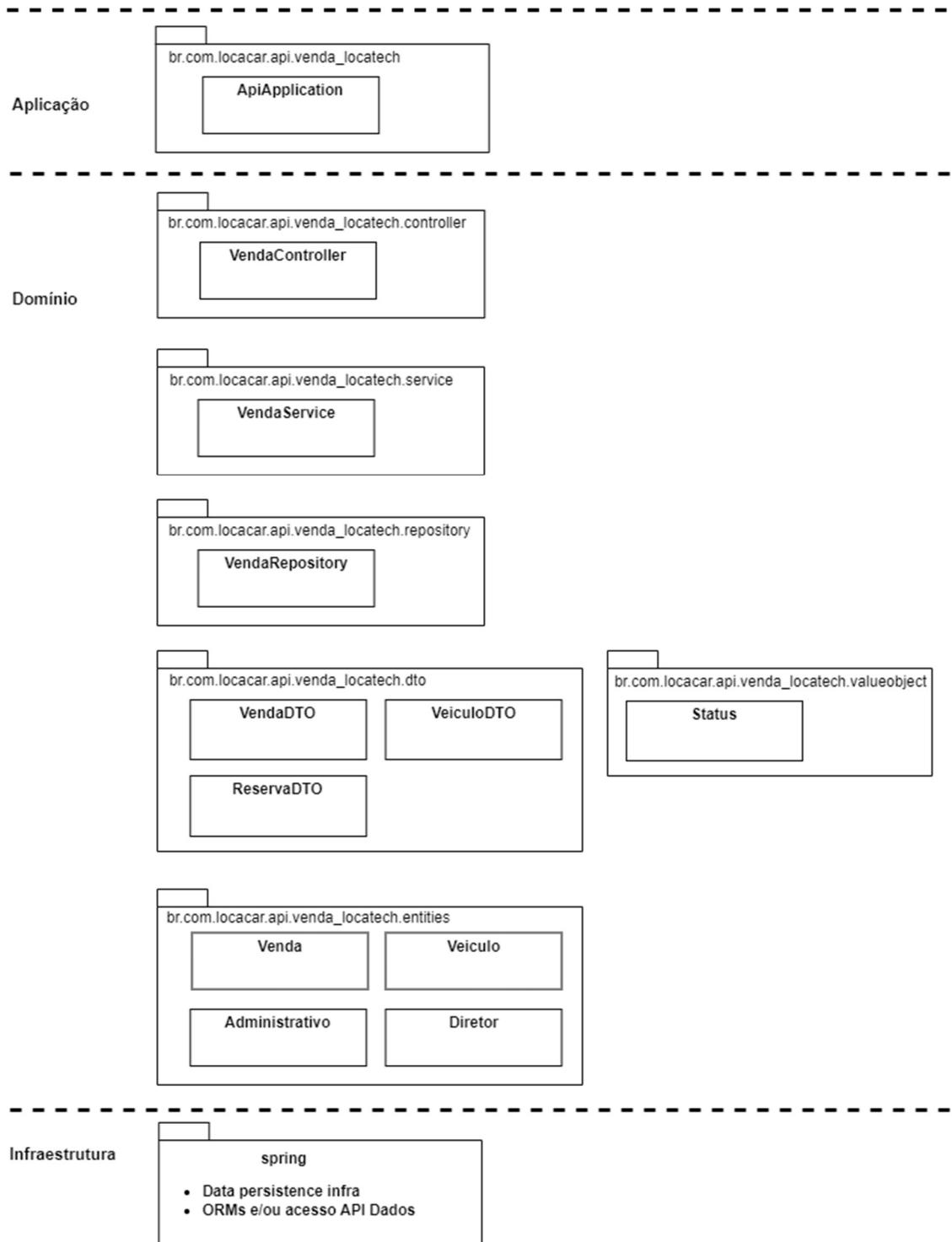
- Número
- Nome
- Id
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor
- Proposta
- Fornecedor
- Preço
- Status



## Arquitetura Aplicação Venda

*Aplicação Venda*

Interface do Usuário



Agregados, Entidade e Objetos de Valor

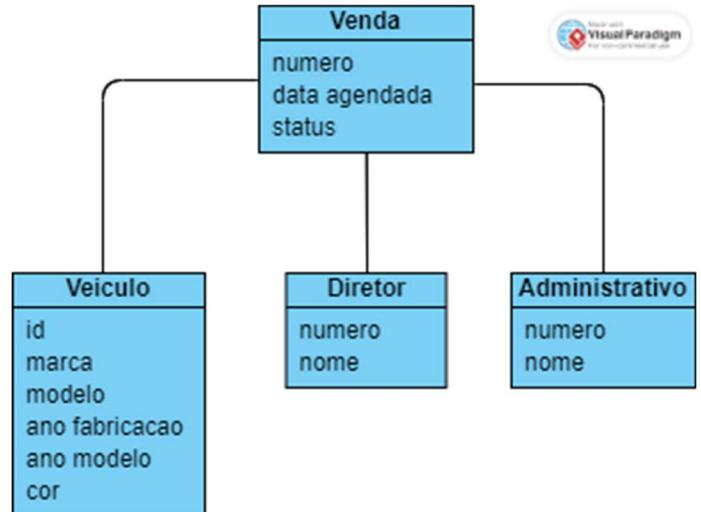
### Agregado Venda

#### Entidades

- Venda
- Diretor
- Administrativo
- Veículo

#### Objetos de Valor

- Id
- Número
- Data agendada
- Nome
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor
- Status

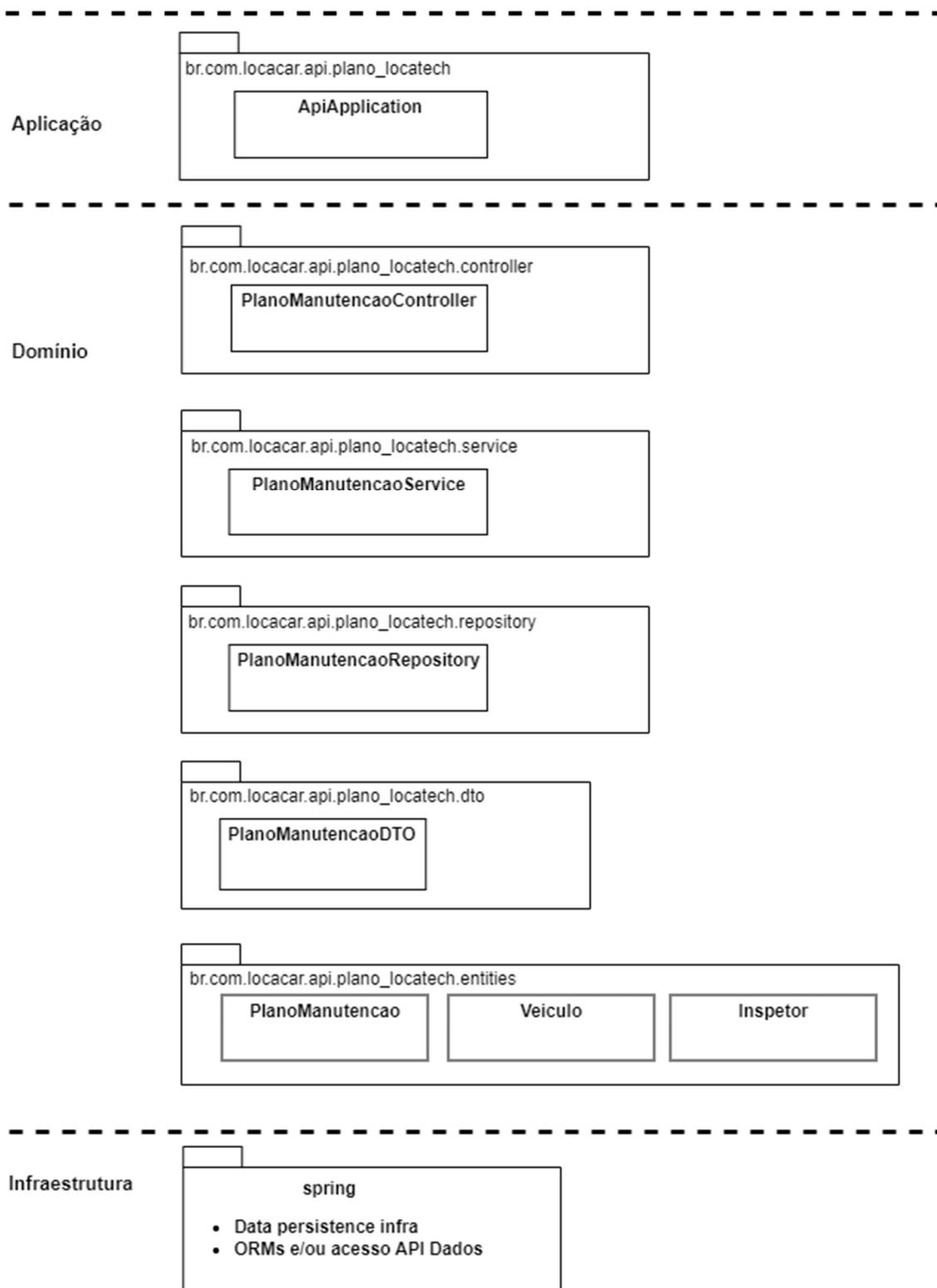


Contexto Delimitado: Plano de Manutenção Preventiva

Arquitetura Aplicação PlanoManutenção

### *Aplicação PlanoManutenção*

Interface do Usuário



Agregados, Entidade e Objetos de Valor

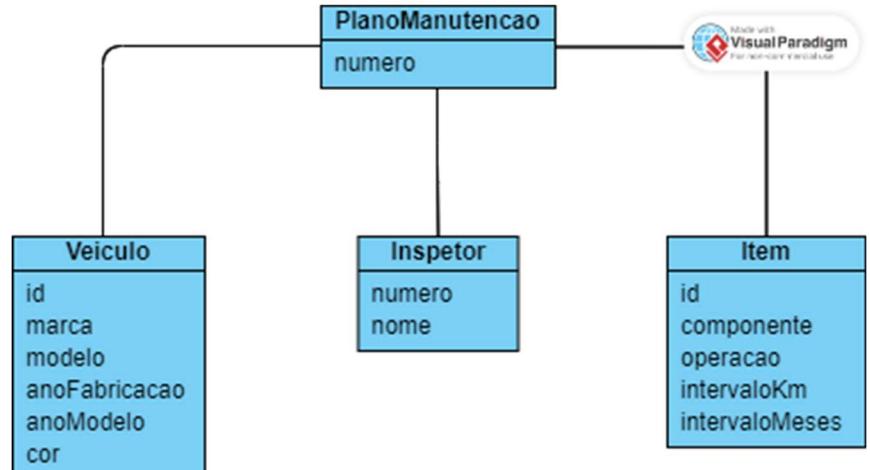
### Agregado Plano de Manutenção

#### Entidades

- PlanoManutencao
- Inspetor
- Veículo
- Item

#### Objetos de Valor

- Id
- Número
- Nome
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor
- Componente
- Operação
- Intervalo Km
- Intervalo Meses

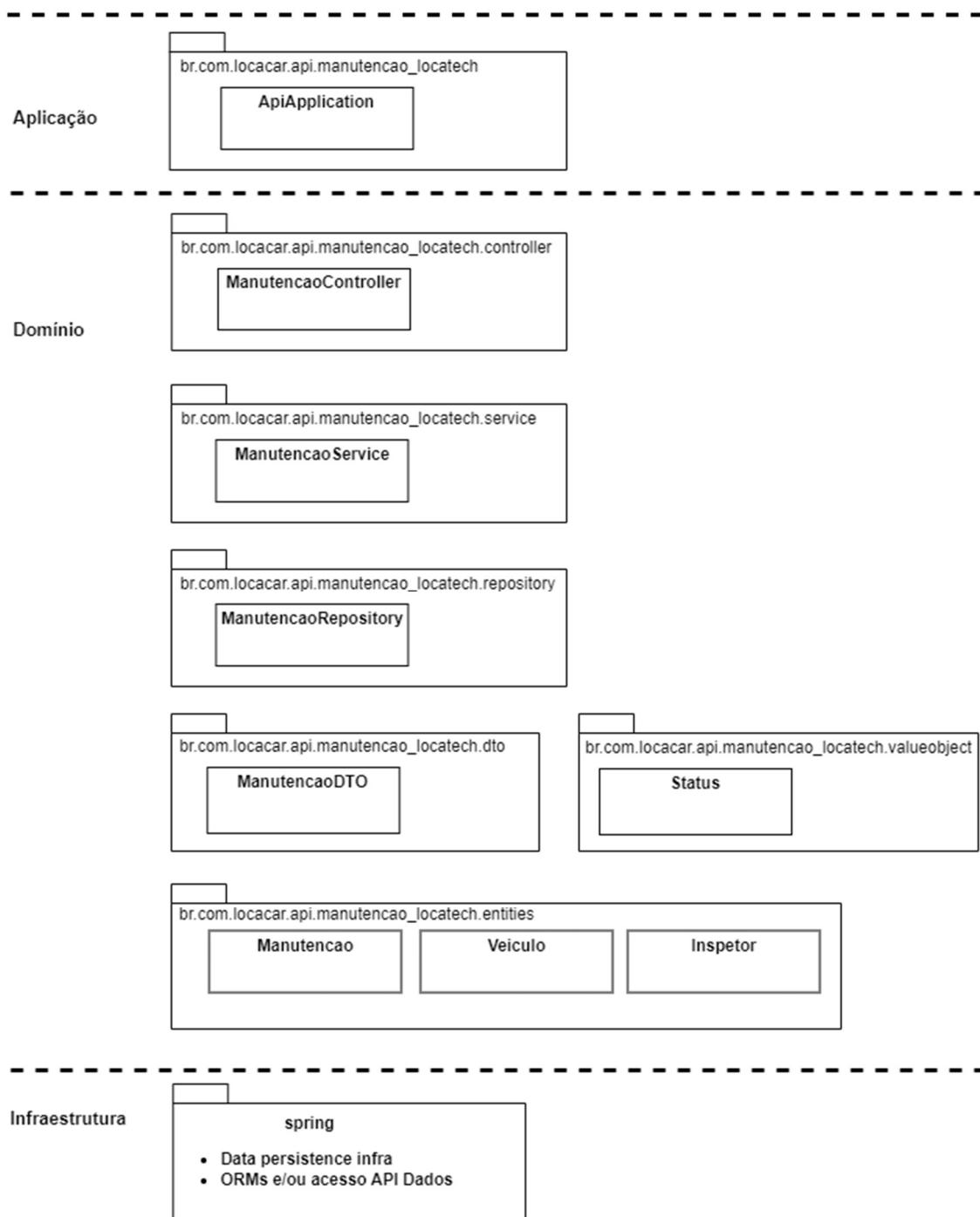


## Contexto Delimitado: Manutenção Preventiva

Arquitetura Aplicação Manutenção

**Aplicação Manutenção**

Interface do Usuário



Agregados, Entidade e Objetos de Valor

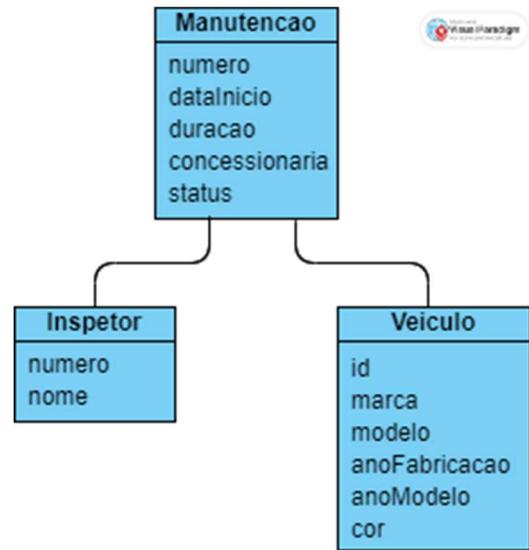
### *Aggregado Manutenção*

#### Entidades

- Manutenção
- Inspetor
- Veículo

#### Objetos de Valor

- Número
- Data Início
- Duração
- Concessionária
- Status
- Nome
- Id
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor

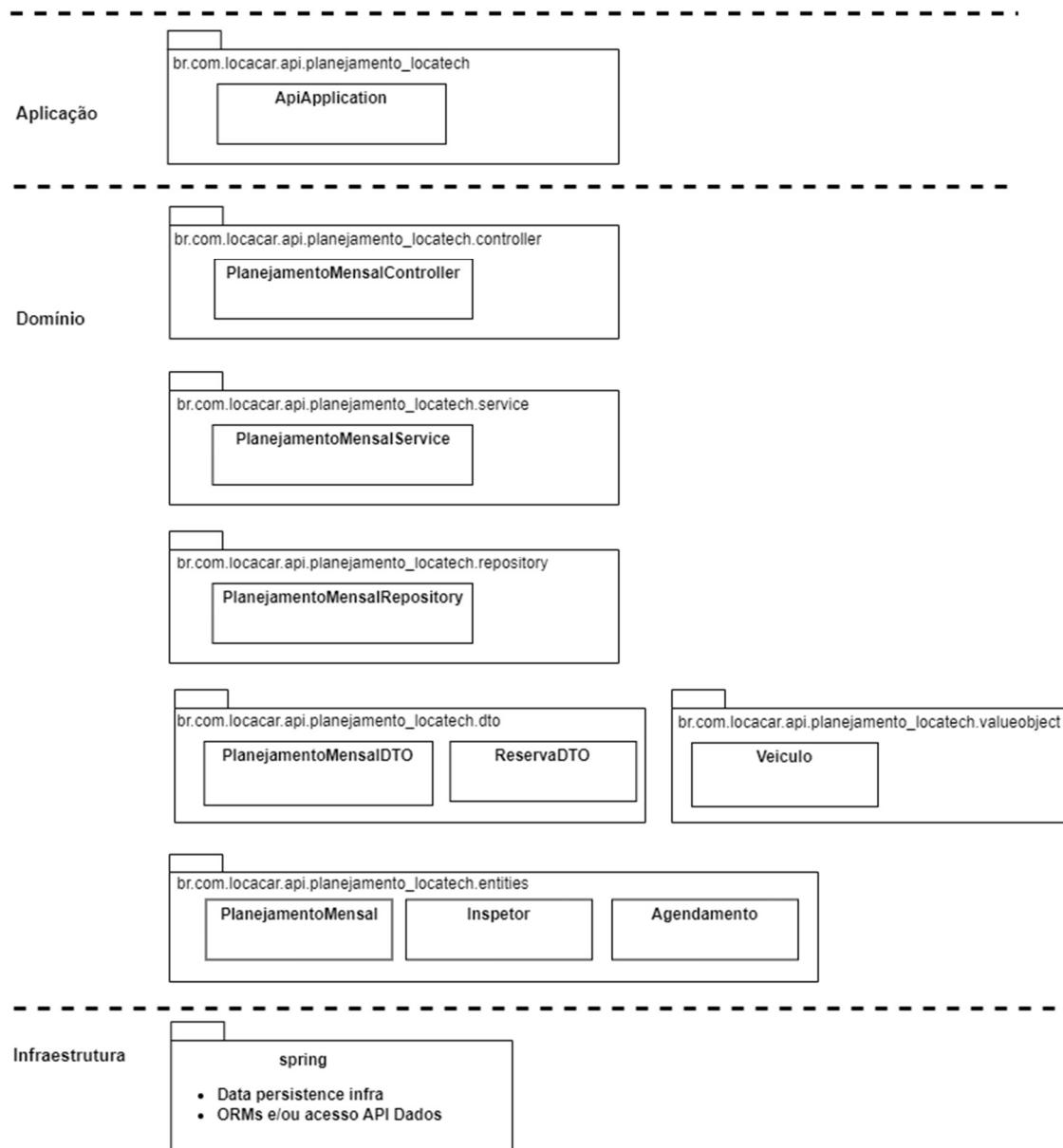


## Contexto Delimitado: Planejamento Mensal

## Arquitetura Aplicação Planejamento Mensal

*Aplicação PlanejamentoMensal*

Interface do Usuário



Agregados, Entidade e Objetos de Valor

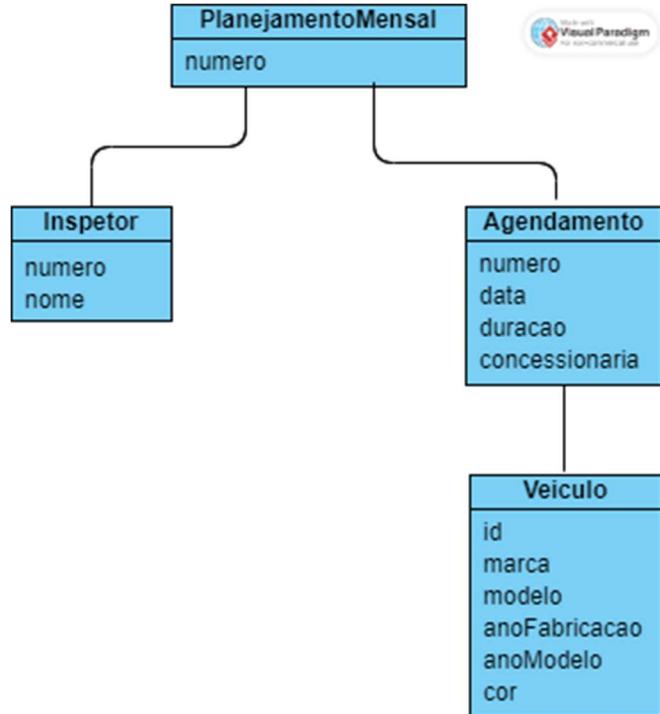
### Agregado Planejamento Mensal

#### Entidades

- Planejamento Mensal
- Inspetor
- Agendamento

#### Objetos de Valor

- Número
- Nome
- Data
- Duração
- Concessionária
- Veículo
- Id
- Marca
- Modelo
- Ano de Fabricação
- Ano do Modelo
- Cor

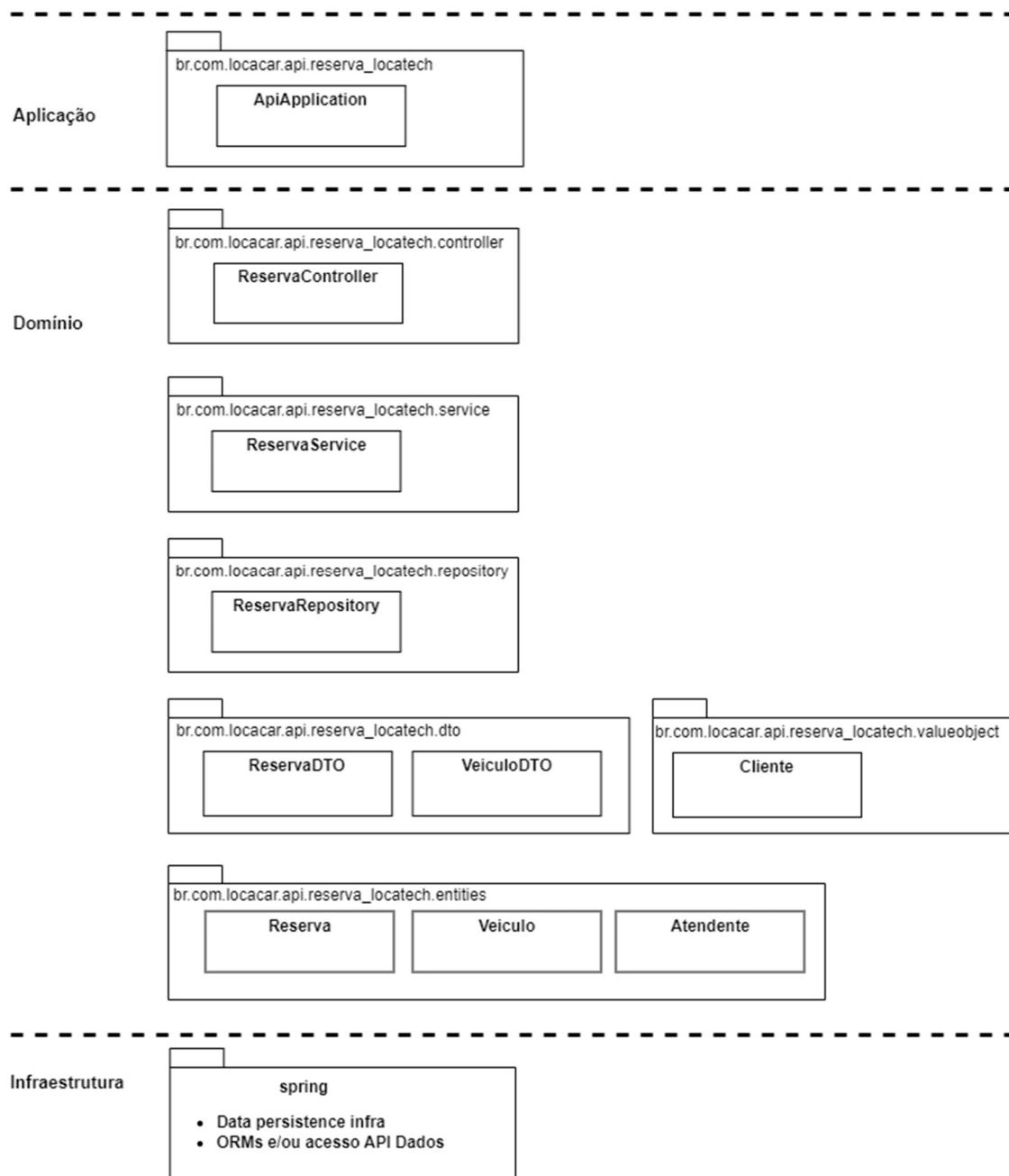


## Contexto Delimitado: Reserva

## Arquitetura Aplicação Reserva

*Aplicação Reserva*

Interface do Usuário



## Agregados, Entidade e Objetos de Valor

### *Aggregado Veículo*

#### Entidades

- Veículo

#### Objetos de Valor

- Id veículo
- Placa
- Marca
- Modelo
- Cor
- Ano fabricação
- Ano modelo
- Valor da diária
- Categoria

Veículo	
id	
placa	
marca	
modelo	
cor	
ano fabricacao	
ano modelo	
valor diaria	
categoria	

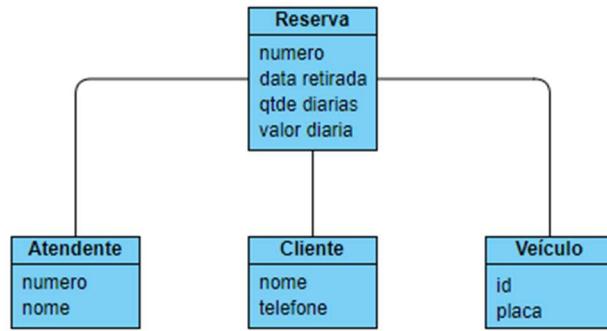
### *Aggregado Reserva*

#### Entidades

- Reserva
- Atendente
- Veículo

#### Objetos de Valor

- Número
- Data retirada
- Quantidade de diárias
- Valor da diária
- Nome
- Telefone
- Id veículo
- Placa
- Cliente

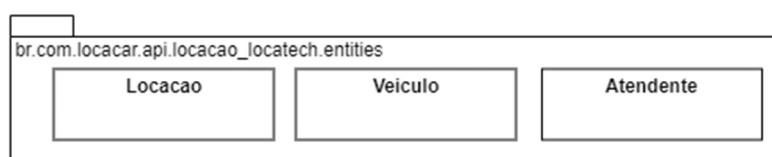
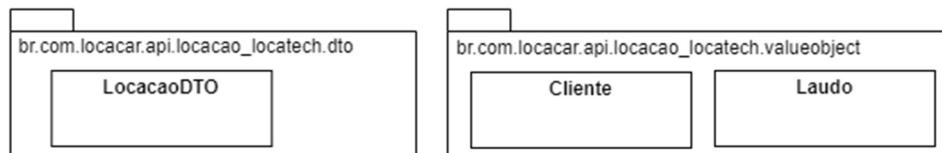
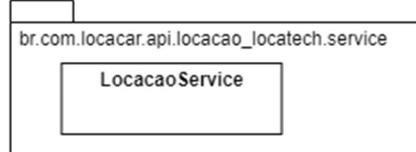
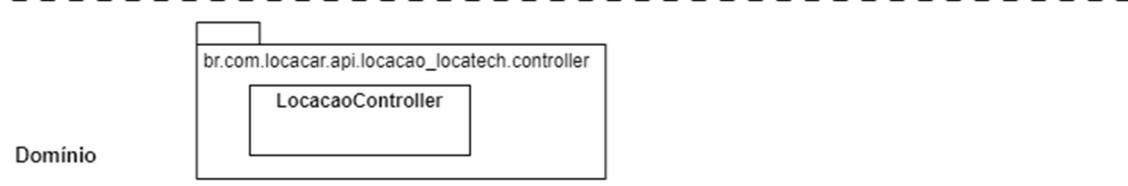
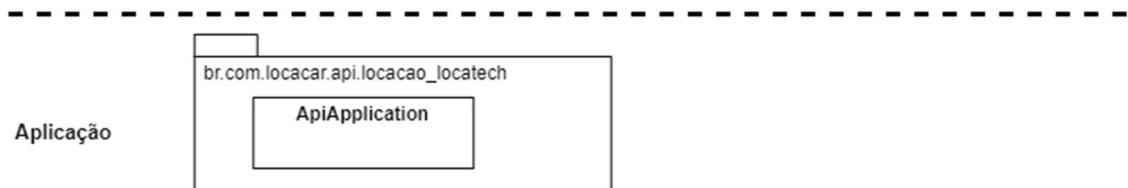


## Contexto Delimitado: Locação

Arquitetura Aplicação Locação

*Aplicação Locação*

Interface do Usuário



Agregados, Entidade e Objetos de Valor

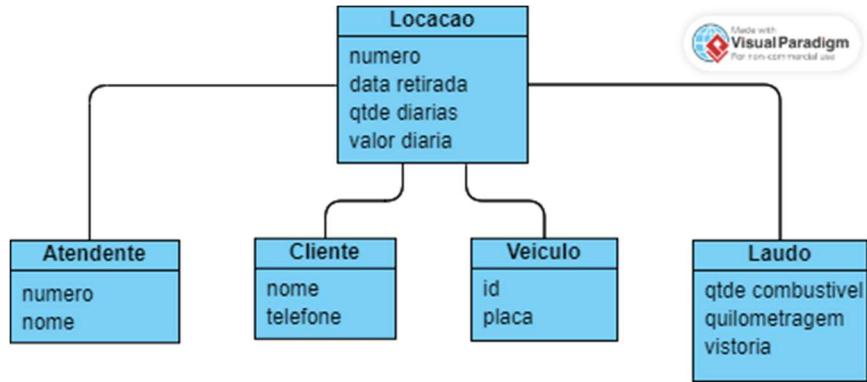
### *Agregado Locação*

#### Entidades

- Locação
- Atendente
- Veículo

#### Objetos de Valor

- Número
- Data início
- Quantidade de diárias
- Valor da diária
- Cliente
- Nome
- Telefone
- Laudo
- Id
- Placa
- Quantidade de combustível
- Quilometragem
- Vistoria



## Implementação

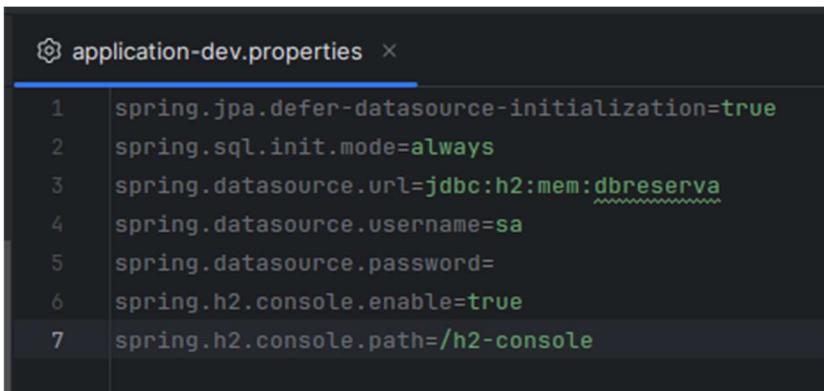
### Configurações do Hibernate

As entidades Atendente, Diretor, Administrativo e Inspetor são mantidas por um subdomínio genérico que será endereçado em versões futuras.

Estas entidades são persistidas pelas tabelas relacionais (H2) TB\_ATENDENTE, TB\_DIRETOR, TB\_ADMINISTRATIVO e TB\_INSPECTOR e nesta versão da aplicação são populadas por scripts “DATA.SQL”, localizados no diretório “resources”.

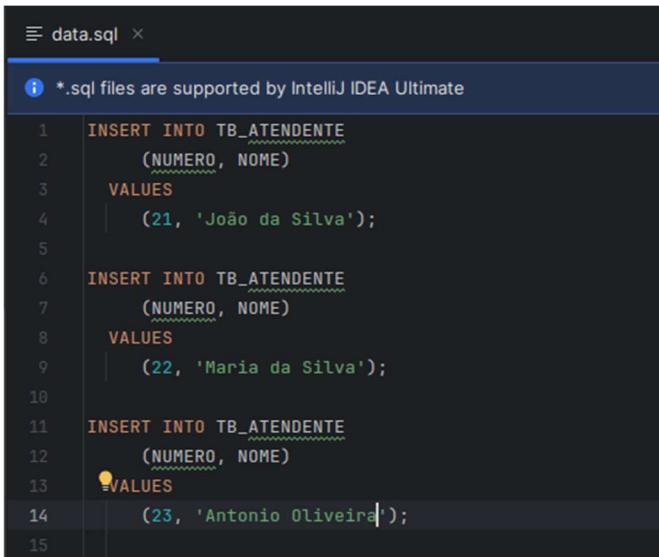
Por padrão os scripts “DATA.SQL” são executados antes da inicialização do Hibernate, para adiar esta carga precisamos incluir as propriedades abaixo no arquivo “applicaction.properties”, no nosso caso “application-dev.properties”. (Ref: Artigo - <https://www.baeldung.com/spring-boot-data-sql-and-schema-sql#thedatasqlfile> ).

**Arquivo:** application-dev.properties  
spring.jpa.defer-datasource-initialization=true  
spring.sql.init.mode=Always



```
application-dev.properties
1 spring.jpa.defer-datasource-initialization=true
2 spring.sql.init.mode=always
3 spring.datasource.url=jdbc:h2:mem:dbreserva
4 spring.datasource.username=sa
5 spring.datasource.password=
6 spring.h2.console.enable=true
7 spring.h2.console.path=/h2-console
```

**Arquivo:** data.sql  
INSERT INTO TB\_ATENDENTE  
(NUMERO, NOME)  
VALUES  
(21, 'João da Silva');  
...



```
*.sql files are supported by IntelliJ IDEA Ultimate
1 INSERT INTO TB_ATENDENTE
2   (NUMERO, NOME)
3   VALUES
4     (21, 'João da Silva');
5
6 INSERT INTO TB_ATENDENTE
7   (NUMERO, NOME)
8   VALUES
9     (22, 'Maria da Silva');
10
11 INSERT INTO TB_ATENDENTE
12   (NUMERO, NOME)
13   VALUES
14     (23, 'Antonio Oliveira');
```

## Considerações Finais

### Técnicas Utilizadas

**Domain Storytelling:** a utilização desta técnica permitiu a materialização dos processos de negócio e se mostrou de melhor entendimento pelo cliente. Os diagramas resultantes estavam mais próximos ao negócio.

**Event Storming:** a utilização desta técnica permitiu aprofundar os eventos de negócio se aproximando da solução técnica, permitiu uma passagem suave e direcionada do conceito do negócio para construção da solução.

### Design Patterns Utilizados

**Layered Architecture:** a adoção deste Design Pattern permitiu uma melhor organização dos componentes, uma divisão clara e assertiva de responsabilidades. Certamente facilitará futuras manutenções aumentando a expectativa de vida da solução.

### Reflexões sobre o Processo de Levantamento de Requisitos

Os dois principais pontos da abordagem DDD foram:

- a) proximidade com as áreas de negócio, oferecendo inclusive a oportunidade de revisão, alinhamento e descoberta de processos e
- b) respeito a realidade que empresas são constituídas por unidades independentes com processos e interesses próprios e às vezes conflitantes.

A abordagem DDD se contrapõe a abordagens top down onde a direção ou uma área de processos define a forma de trabalho de maneira centralizada para que as unidades sigam. Se por um lado a abordagem Top Down visa eliminar redundâncias e divergências, por vezes acaba por engessar processos, diagnosticar erroneamente problemas e falha na comunicação entre as partes.

A abordagem DDD promove a autonomia das áreas de negócio, melhora o engajamento e promete soluções mais ágeis e flexíveis. Há de se ponderar o risco de falta de integridade das informações.

### Benefícios Esperados

A informatização dos processos é um requisito fundamental para o crescimento e até a sobrevivência da empresa, marca a transição de uma empresa familiar para uma gestão mais profissional.

Esta mecanização deve reduzir a dependência de pessoas chave que hoje detêm o conhecimento do processo pondo em risco a continuidade dos negócios.

Também é esperada uma maior segurança das informações tanto contra a perda de dados, como por exemplo incêndio ou roubo, como eventual exposição indevida.

Também será possível extrair informações como o comportamento dos clientes, ROI (return of investment), monitorar gastos com manutenções, sazonalidade das locações e ociosidade. Métricas que antes eram obtidas de forma intuitiva ou com muito esforço e pouca precisão.

É esperada a redução da taxa de erros e enganos e melhor produtividade dos colaboradores poupando esforço em tarefas burocráticas e repetitivas e permitindo focar no negócio.

## Ferramentas Utilizadas

Egon.io – The Domain Story Modeler

Diagramação do storyteller

<https://github.com/>

Miro – Ferramenta para diagramação de fluxos de trabalho

Diagramação do Event Storming

<https://miro.com/pt/>

Draw.io Diagrams – Ferramenta para diagramação

Diagramação da Arquitetura de Solução

<https://app.diagrams.net/>

Visual Paradigm – Development Tool Suite

Diagramação das Classes

<https://www.visual-paradigm.com/>

IntelliJ IDEA – IDE para desenvolvimento Java

Construção e teste de aplicações Java

<https://www.jetbrains.com/idea/>

Spring – Framework para plataforma Java

Construção aplicações Java

<https://spring.io/>

H2 – SGDB relacional (embutido no Spring)

Banco de dados aplicações Java

<https://www.h2database.com/html/main.html>

Insomnia – Ferramenta API Client

Testes de chamadas REST

<https://insomnia.rest/>

JSON Server – Simulador de APIs REST

Testes de aplicações que consomem APIs REST

<https://www.npmjs.com/package/json-server>

<https://github.com/typicode/json-server>

## Execução de Fluxos

Aquisição e Venda

Aplicação Cotação

Solicita cotação de compra

The screenshot shows the Insomnia REST client interface. On the left, there's a sidebar with a tree view of API endpoints categorized by module: Aquisicao, Cotacao, Compra, Venda, Locacao, Reserva, Produto, and Usuario. Under the 'Cotacao' category, there are several methods listed: POST save\_SolicitaCotacao, POST save2, GET findAll, GET findById, PUT update\_ElaboraCo..., and PUT update-propostas2. The main panel shows a POST request to 'http://localhost:8080/aquisicao/cotacao'. The request body is a JSON object:

```
1v {
2v   "diretor": {
3v     "numero": 11,
4v     "nome": "Jair Menalau Bartolomeu"
5v   },
6v   "veiculo": {
7v     "marca": "FORD",
8v     "modelo": "FIESTA",
9v     "cor": "PRATA",
10v    "anoFabricacao": 2009,
11v    "anoModelo": 2009
12v  }
13v }
```

The response tab shows the result of the POST request: a 201 Created status with a response time of 810 ms, 205 B transferred, and the timestamp Just Now. The response body is also a JSON object:

```
1v {
2v   "numero": 1,
3v   "diretor": {
4v     "numero": 11,
5v     "nome": "Jair Menalau Bartolomeu"
6v   },
7v   "administrativo": null,
8v   "veiculo": {
9v     "marca": "FORD",
10v    "modelo": "FIESTA",
11v    "anoFabricacao": 2009,
12v    "anoModelo": 2009,
13v    "cor": "PRATA"
14v  },
15v   "propostas": null
16v }
```

The screenshot shows the H2 Console interface. The left sidebar lists database objects: jdbc:h2:mem:dbcotacao, TB\_ADMINISTRATIVO, TB\_COTACAO, TB\_DIRETOR, TB\_PROPOSTA, INFORMATION\_SCHEMA, and Users. The right panel shows a SQL statement being run:

```
SELECT * FROM TB_COTACAO
```

The results table shows the following data:

ADMINISTRATIVO_NUMERO	ANO_FABRICACAO	ANO_MODELO	DIRETOR_NUMERO	NUMERO	COR	MARCA	MODELO
null	2009	2009	11	1	PRATA	FORD	FIESTA

(1 row, 16 ms)

## Elabora cotação de compra

Insomnia

Application File Edit View Window Tools Help

Scratch Pad ▾

PUT ▾ http://localhost:8080/aquisicao/cotacao/1 Send ▾ 200 OK 597 ms 323 B Just Now ▾

Base Environment ▾ Auth ▾ Query Headers [?] Docs

Preview ▾ Headers [?] Cookies Timeline

```

1  {
2     "numero": 1,
3     "diretor": {
4         "numero": 11,
5         "nome": "Jair Menalau Bartolomeu"
6     },
7     "administrativo": {
8         "numero": 31,
9         "nome": "Marcos Petra"
10    },
11    "veiculo": {
12        "marca": "FORD",
13        "modelo": "FIESTA",
14        "cor": "PRATA",
15        "anoFabricacao": 2009,
16        "anoModelo": 2009
17    },
18    "propostas": [
19        {
20            "fornecedor": "Vimave",
21            "preco": 70000
22        },
23        {
24            "fornecedor": "Remaza",
25            "preco": 75000
26        }
27    ]
28 }

```

Beautify JSON \$.store.books[\*].author

Preferences Log in to see your projects Made with ❤ by Kong

H2 Console

localhost:8080/h2-console/login.do?jsessionid=c001c17c3f11479a93786e2a0cddea47

Auto commit Max rows: 1000 Auto complete Off Auto select On

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM TB\_COTACAO

ADMINISTRATIVO_NUMERO	ANO_FABRICACAO	ANO_MODELO	DIRETOR_NUMERO	NUMERO	COR	MARCA	MODELO
31	2009	2009	11	1	PRATA	FORD	FIESTA

(1 row, 1 ms)

Edit

The screenshot shows the H2 Console interface. In the top navigation bar, it says "H2 Console" and the URL is "localhost:8080/h2-console/login.do?jsessionid=c001c17c3f11479a93786e2a0cddea47". Below the URL, there are several configuration options: "Auto commit" (checked), "Max rows: 1000", "Run Selected", "Auto complete" (Off), "Auto select" (On), and a help icon. On the left, a sidebar lists database objects: "jdbc:h2:mem:dbcotacao", "TB\_ADMINISTRATIVO", "TB\_COTACAO", "TB\_DIRETOR", "TB\_PROPOSTA", "INFORMATION\_SCHEMA", "Users", and "H2 2.2.224 (2023-09-17)". The main area contains a SQL statement: "SELECT \* FROM TB\_PROPOSTA". Below the statement is the resulting table:

PRECO	COTACAO_NUMERO	ID	FORNECEDOR
70000	1	1	Vimave
75000	1	2	Remaza

(2 rows, 1 ms)

At the bottom of the result area is an "Edit" button.

## Aplicação Compra

A aplicação **Compra** informa aos SE - Sistemas Externos **Plano de Manutenção e Reserva** a aquisição de um novo veículo. Estas aplicações são simuladas com a ferramenta “json-server”.

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ json-server --port 3001 --watch db_planomanutencao.json\n\n  \{^_^\}/ hi!\n\n  Loading db_planomanutencao.json\n  Done\n\n  Resources\n  http://localhost:3001/veiculo\n\n  Home\n  http://localhost:3001\n\n  Type s + enter at any time to create a snapshot of the database\n  Watching...
```

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ json-server --port 3002 --watch db_locacao.json\n\n  \{^_^\}/ hi!\n\n  Loading db_locacao.json\n  Done\n\n  Resources\n  http://localhost:3002/veiculo\n\n  Home\n  http://localhost:3002\n\n  Type s + enter at any time to create a snapshot of the database\n  Watching...
```

## Solicita compra

Application File Edit View Window Tools Help

Scratch Pad ▾

Base Environment ▾

Add Cookies

Filter

Aquisicao

- Cotacao
- Compra
- POST save\_solicitac...**
- CET findAll**
- CET findByD...**
- PUT update\_compra\_...**
- PUT update\_fatura\_v...**
- PUT update\_licencia\_...**
- PUT update\_segura\_v...**
- PUT update-propostas**
- DEL delete**
- Venda
- Locacao
- Reserva

POST http://localhost:8080/aquisicao/compra

Send ▾

201 Created 809 ms 273 B Just Now ▾

Preview ▾ Headers<sup>3</sup> Cookies Timeline

```

1  {
2   "numero": 1,
3   "diretor": {
4     "numero": 11,
5     "nome": "Jair Menalau Bartolomeu"
6   },
7   "veiculo": {
8     "marca": "FORD",
9     "modelo": "FIESTA",
10    "cor": "PRATA",
11    "anoFabricacao": 2009,
12    "anoModelo": 2009
13  },
14  "proposta": {
15    "fornecedor": "Vimave",
16    "preco": 70000
17  }
18 }
```

\$store.books[\*].author

Beautify JSON

Preferences

Login Sign Up

H2 Console

localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:dbcompra

TB\_ADMINISTRATIVO

TB\_COMPRA

TB\_DIRETOR

TB\_VEICULO

INFORMATION\_SCHEMA

Users

H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM TB\_COMPRA

ADMINISTRATIVO_NUMERO	DIRETOR_NUMERO	PRECO	STATUS	NUMERO	VEICULO_ID	FORNECEDOR
null	11	70000	0	1	1	Vimave

(1 row, 16 ms)

Edit

The screenshot shows the H2 Console interface at `localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0`. The left sidebar lists tables: `TB_ADMINISTRATIVO`, `TB_COMPRA`, `TB_DIRETOR`, `TB_VEICULO`, `INFORMATION_SCHEMA`, and `Users`. The right panel contains a SQL statement editor with the query `SELECT * FROM TB_VEICULO;` and a results table.

ANO_FABRICACAO	ANO_MODELO	ID	COR	MARCA	MODELO
2009	2009	1	PRATA	FORD	FIESTA

(1 row, 1 ms)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM TB_VEICULO;
```

Edit

## Compra veículo

The screenshot shows the Insomnia REST client interface. The top bar displays "Insomnia" and the URL "PUT http://localhost:8080/aquisicao/compra/compraveiculo/1". The status bar indicates "200 OK", "150 ms", "302 B", and "Just Now". The left sidebar lists API endpoints under "Aquisicao" and "Compra". The main area shows a JSON payload being sent:

```
1 {  
2   "numero": 1,  
3   "diretor": {  
4     "numero": 11,  
5     "nome": "Jair Menalau Bartolomeu"  
6   },  
7   "administrativo": {  
8     "numero": 31,  
9     "nome": "Marcos Petra"  
10  },  
11  "veiculo": {  
12    "id": 1,  
13    "marca": "FORD",  
14    "modelo": "FIESTA",  
15    "cor": "PRATA",  
16    "anoFabricacao": 2009,  
17    "anoModelo": 2009  
18  },  
19  "proposta": {  
20    "fornecedor": "Vimave",  
21    "preco": 70000  
22  }  
23 }
```

The response preview shows the same JSON structure with some fields modified or added, indicating a successful update.

The screenshot shows the H2 Console interface. The top bar displays "H2 Console" and the URL "localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0". The left sidebar shows database schema with tables: TB\_ADMINISTRATIVO, TB\_COMPRA, TB\_DIRETOR, TB\_VEICULO, INFORMATION\_SCHEMA, and Users. The main area shows a SQL statement being run:

```
SELECT * FROM TB_COMPRA |
```

The results table shows one row of data:

ADMINISTRATIVO_NUMERO	DIRETOR_NUMERO	PRECO	STATUS	NUMERO	VEICULO_ID	FORNECEDOR
31	11	70000	1	1	1	Vimave

(1 row, 1 ms)

An "Edit" button is visible at the bottom of the results table.

## Fatura veículo

Insomnia

Application File Edit View Window Tools Help

Scratch Pad ▾

PUT http://localhost:8080/aquisicao/compra/faturaveiculo/1

Send ▾

200 OK 359 ms 303 B Just Now

Login Sign Up

Base Environment ▾

Add Cookies

Filter

Aquisicao

- Cotacao
- Compra
- POST save\_solicita\_compra**
- GET findAll**
- GET findById**
- PUT update\_compra\_veiculo**
- PUT update\_fatura\_veiculo**
- PUT update\_licenca\_veiculo**
- PUT update\_segura\_veiculo**
- PUT update\_propostas**
- DEL delete**
- Venda
- Locacao
- Reserva

JSON Auth Query Headers Docs

```

1  {
2    "numero": 1,
3    "diretor": [
4      {"numero": 11,
5       "nome": "Jair Menalau Bartolomeu"
6     },
7     {"administrativo": {
8       "numero": 31,
9       "nome": "Marcos Petra"
10    },
11    "veiculo": {
12      "id": 1,
13      "marca": "FORD",
14      "modelo": "FIESTA",
15      "cor": "PRATA",
16      "anoFabricacao": 2009,
17      "anoModelo": 2009
18    },
19    "proposta": {
20      "fornecedor": "Vimave",
21      "preco": 70000
22    }
23  }

```

Preview ▾

Headers Cookies Timeline

1 {
2 "numero": 1,
3 "diretor": [
4 {"numero": 11,
5 "nome": "Jair Menalau Bartolomeu"
6 },
7 {"administrativo": {
8 "numero": 31,
9 "nome": "Marcos Petra"
10 },
11 "veiculo": {
12 "id": 1,
13 "marca": "FORD",
14 "modelo": "FIESTA",
15 "anoFabricacao": 2009,
16 "anoModelo": 2009,
17 "cor": "PRATA"
18 },
19 "proposta": {
20 "fornecedor": "Vimave",
21 "preco": 70000
22 }
23 }
24 }

Beautify JSON

S.store.books[\*].author

Preferences

Log in to see your projects Made with ❤ by Kong

H2 Console

localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:dbcompra TB\_ADMINISTRATIVO TB\_COMPRA TB\_DIRETOR TB\_VEICULO INFORMATION\_SCHEMA Users H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM TB\_COMPRA

ADMINISTRATIVO_NUMERO	DIRETOR_NUMERO	PRECO	STATUS	NUMERO	VEICULO_ID	FORNECEDOR
31	11	70000	2	1	1	Vimave

(1 row, 1 ms)

Edit

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge Done Resources http://localhost:3001/veiculo Home http://localhost:3001 Type s + enter at any time to create a snapshot of the database Watching... db_planomanutencao.json has changed, reloading... Loading db_planomanutencao.json Done Resources http://localhost:3001/veiculo Home http://localhost:3001 POST /veiculo 201 92.536 ms - 118
```

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ cat db_planomanutencao.json {"veiculo": { "id": 1, "marca": "FORD", "modelo": "FIESTA", "anoFabricacao": 2009, "anoModelo": 2009, "cor": "PRATA" }}
```

## Licencia veículo

```

PUT http://localhost:8080/aquisicao/compra/licencaveiculo/1
200 OK
52.7 ms
305 B
Just Now

{
  "numero": 1,
  "diretor": {
    "numero": 11,
    "nome": "Jair Menalau Bartolomeu"
  },
  "administrativo": {
    "numero": 31,
    "nome": "Marcos Petra"
  },
  "veiculo": {
    "id": 1,
    "marca": "FORD",
    "modelo": "FIESTA",
    "cor": "PRATA",
    "anoFabricacao": 2009,
    "anoModelo": 2009
  },
  "proposta": {
    "fornecedor": "Vimave",
    "preco": 70000
  }
}

```

Atividades Navegador Web Firefox

H2 Console

localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0

Auto commit | Auto complete | Max rows: 1000 | Run | Run Selected | Auto complete | Clear | SQL statement:

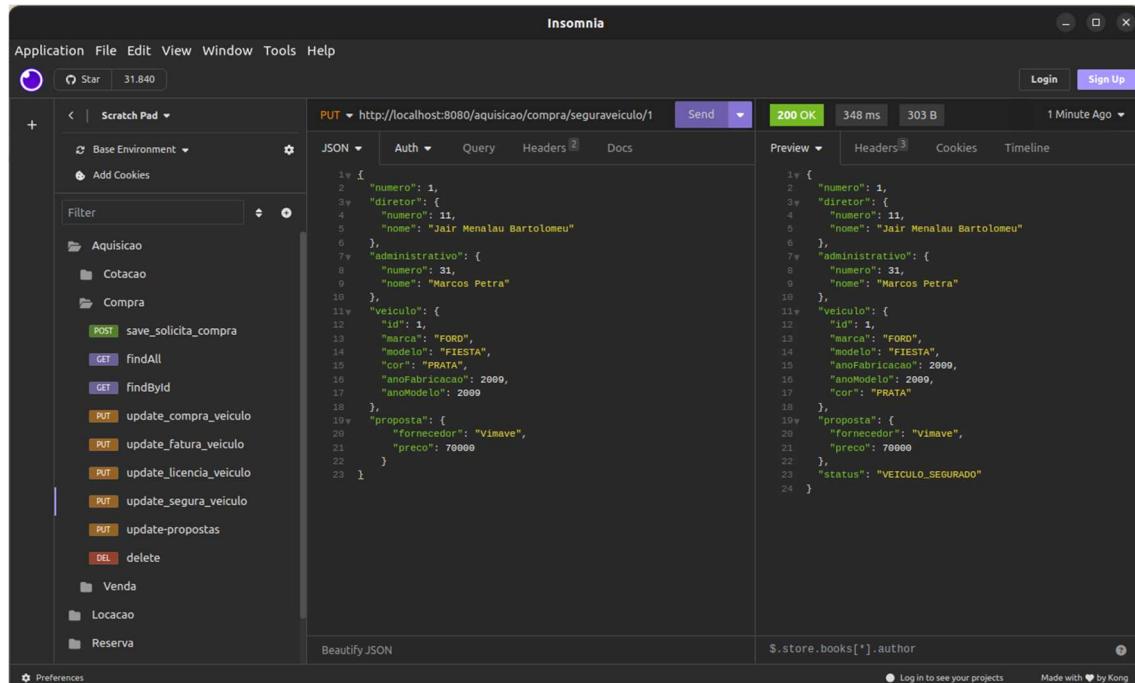
```
SELECT * FROM TB_COMPRA
```

ADMINISTRATIVO_NUMERO	DIRETOR_NUMERO	PRECO	STATUS	NUMERO	VEICULO_ID	FORNECEDOR
31	11	70000	3	1	1	Vimave

(1 row, 1 ms)

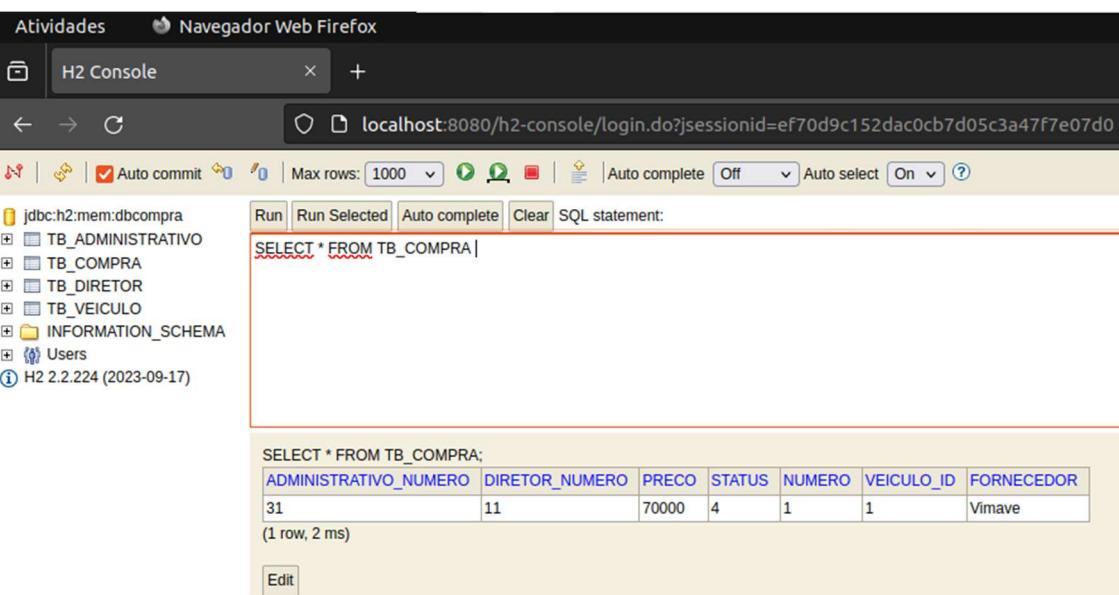
Edit

## Segura veículo



The screenshot shows the Insomnia REST client interface. The URL is `http://localhost:8080/aquisicao/compra/seguraveiculo/1`. The response code is **200 OK**, with a response time of 348 ms and a size of 303 B, timestamped 1 Minute Ago. The response body is a JSON object:

```
1 {
2   "numero": 1,
3   "diretor": {
4     "numero": 11,
5     "nome": "Jair Menalau Bartolomeu"
6   },
7   "administrativo": {
8     "numero": 31,
9     "nome": "Marcos Petra"
10 },
11 "veiculo": {
12   "id": 1,
13   "marca": "FORD",
14   "modelo": "FIESTA",
15   "cor": "PRATA",
16   "anoFabricacao": 2009,
17   "anoModelo": 2009
18 },
19 "proposta": {
20   "fornecedor": "Vimave",
21   "preco": 70000
22 }
23 }
```



The screenshot shows the H2 Console in Firefox. The URL is `localhost:8080/h2-console/login.do?jsessionid=ef70d9c152dac0cb7d05c3a47f7e07d0`. The query `SELECT * FROM TB_COMPRA;` was run, and the result is:

```
SELECT * FROM TB_COMPRA;
ADMINISTRATIVO_NUMERO DIRETOR_NUMERO PRECO STATUS NUMERO VEICULO_ID FORNECEDOR
31           11      70000    4       1         1        Vimave
(1 row, 2 ms)
```

```
Done

Resources
http://localhost:3002/veiculo

Home
http://localhost:3002

Type s + enter at any time to create a snapshot of the database
Watching...

db_locacao.json has changed, reloading...
Loading db_locacao.json
Done

Resources
http://localhost:3002/veiculo

Home
http://localhost:3002

POST /veiculo 201 124.470 ms - 118
```

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ cat db_planomanutencao.json
{
  "veiculo": {
    "id": 1,
    "marca": "FORD",
    "modelo": "FIESTA",
    "anoFabricacao": 2009,
    "anoModelo": 2009,
    "cor": "PRATA"
  }
}
pedro@VMLxUbt22DskDsv01:~/Dsv/fiap/tech_Challenge$ 
```

### Aplicação Venda

A aplicação **Venda** faz uma reserva e depois informa a venda de um veículo aos SE - Sistemas Externos **Plano de Manutenção** e **Reserva**. Estas aplicações são simuladas com a ferramenta “json-server”.

```
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ json-server --port 3001 --watch db_planomanutencao.json

\{^_^\}/ hi!

Loading db_planomanutencao.json
Done

Resources
http://localhost:3001/veiculo

Home
http://localhost:3001

Type s + enter at any time to create a snapshot of the database
.....
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ json-server --port 3002 --watch db_reserva.json

\{^_^\}/ hi!

Loading db_reserva.json
Done

Resources
http://localhost:3002/reserva

Home
http://localhost:3002

Type s + enter at any time to create a snapshot of the database
Watching...
pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech_Challenge$ json-server --port 3003 --watch db_veiculo.json

\{^_^\}/ hi!

Loading db_veiculo.json
Done

Resources
http://localhost:3003/veiculo

Home
http://localhost:3003

Type s + enter at any time to create a snapshot of the database
Watching...
```

## Solicita venda

The screenshot shows the Insomnia REST Client interface. On the left, the API structure is visible with endpoints for Aquisicao, Venda, Locacao, Reserva, Produto, and Usuario. The Venda endpoint has a POST method named 'save\_solicitacao'. The request body is a JSON object:

```

1  {
2    "diretor": {
3      "numero": 11,
4      "nome": "Jair Menalaus Bartolomeu"
5    },
6    "veiculo": {
7      "id": 1,
8      "marca": "FORD",
9      "modelo": "FIESTA",
10     "cor": "PRATA",
11     "anoFabricacao": 2009,
12     "anoModelo": 2009
13   }
14 }

```

The response tab shows a 201 Created status with a response body containing a single object:

```

1  {
2    "numero": 1,
3    "dataAgendada": null,
4    "diretor": {
5      "numero": 11,
6      "name": "Jair Menalaus Bartolomeu"
7    },
8    "administrativo": null,
9    "veiculo": {
10      "id": 4,
11      "marca": "FORD",
12      "modelo": "FIESTA",
13      "anoFabricacao": 2009,
14      "anoModelo": 2009,
15      "cor": "PRATA"
16    },
17    "status": "VENDA_SOLICITADA"
18  }

```

The screenshot shows the H2 Console interface. The left sidebar lists database schemas: jdbc:h2:mem:dbvenda, TB\_ADMINISTRATIVO, TB\_DIRETOR, TB\_VEICULO, TB\_VENDA, INFORMATION\_SCHEMA, and Users. The current schema is H2 2.2.224 (2023-09-17). The main area shows a SQL statement: "SELECT \* FROM TB\_VENDA". The results table displays one row of data:

ADMINISTRATIVO_NUMERO	DATA_AGENDADA	DIRETOR_NUMERO	STATUS	NUMERO	VEICULO_ID
null	null	11	0	1	1

(1 row, 3 ms)

**Edit**

## Agenda venda

```

PUT http://localhost:8080/aquisicao/venda/agenda
{
    "dataAgendada": "2024-01-01",
    "diretor": {
        "numero": 11,
        "nome": "Jair Menalau Bartolomeu"
    },
    "administrativo": {
        "numero": 31,
        "name": "Marcos Petra"
    },
    "veiculo": {
        "id": 1,
        "marca": "FORD",
        "modelo": "FIESTA",
        "cor": "PRATA",
        "anoFabricacao": 2009,
        "anoModelo": 2009
    }
}

```

```

1 {
2   "numero": 1,
3   "dataAgendada": "2024-01-01",
4   "diretor": {
5     "numero": 11,
6     "name": "Jair Menalau Bartolomeu"
7   },
8   "administrativo": {
9     "numero": 31,
10    "name": "Marcos Petra"
11  },
12  "veiculo": {
13    "id": 1,
14    "marca": "FORD",
15    "modelo": "FIESTA",
16    "cor": "PRATA",
17    "anoFabricacao": 2009,
18    "anoModelo": 2009
19  }
20 }

```

ADMINISTRATIVO_NUMERO	DATA_AGENDADA	DIRETOR_NUMERO	STATUS	NUMERO	VEICULO_ID
31	2024-01-01	11	1	1	1

```

\{^_^\} hi!
Loading db_reserva.json
Done

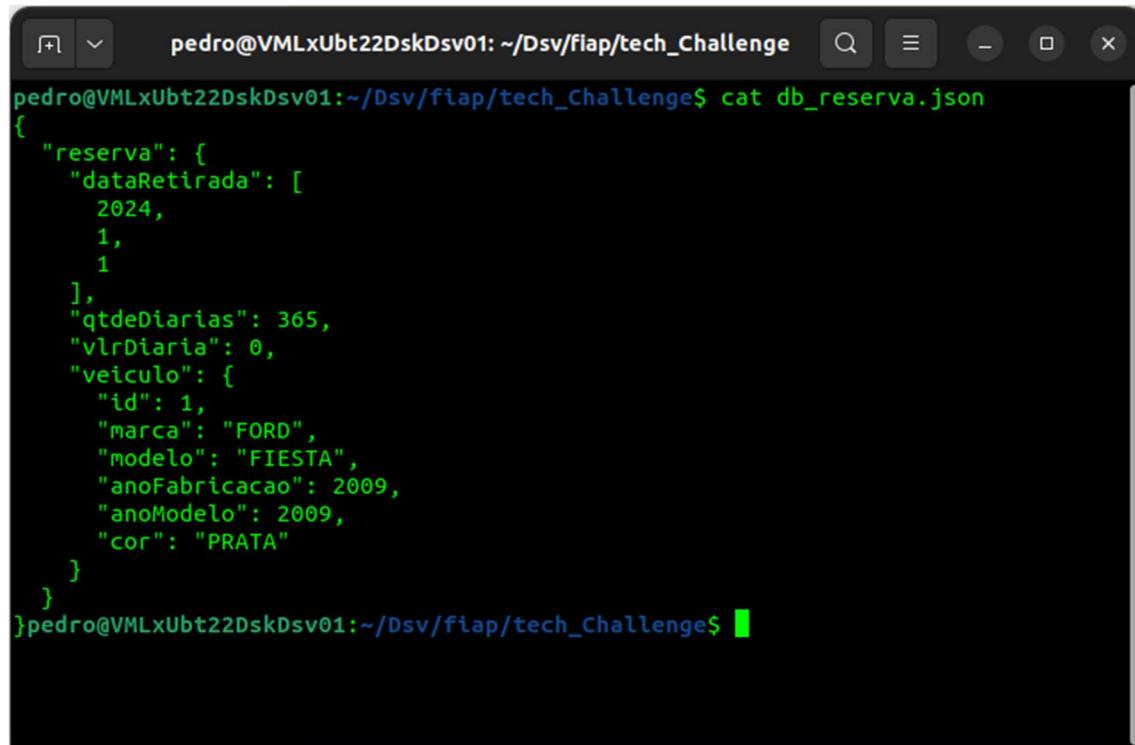
Resources
http://localhost:3002/reserva

Home
http://localhost:3002

Type s + enter at any time to create a snapshot of the database
Watching...

POST /reserva 201 93.935 ms - 237

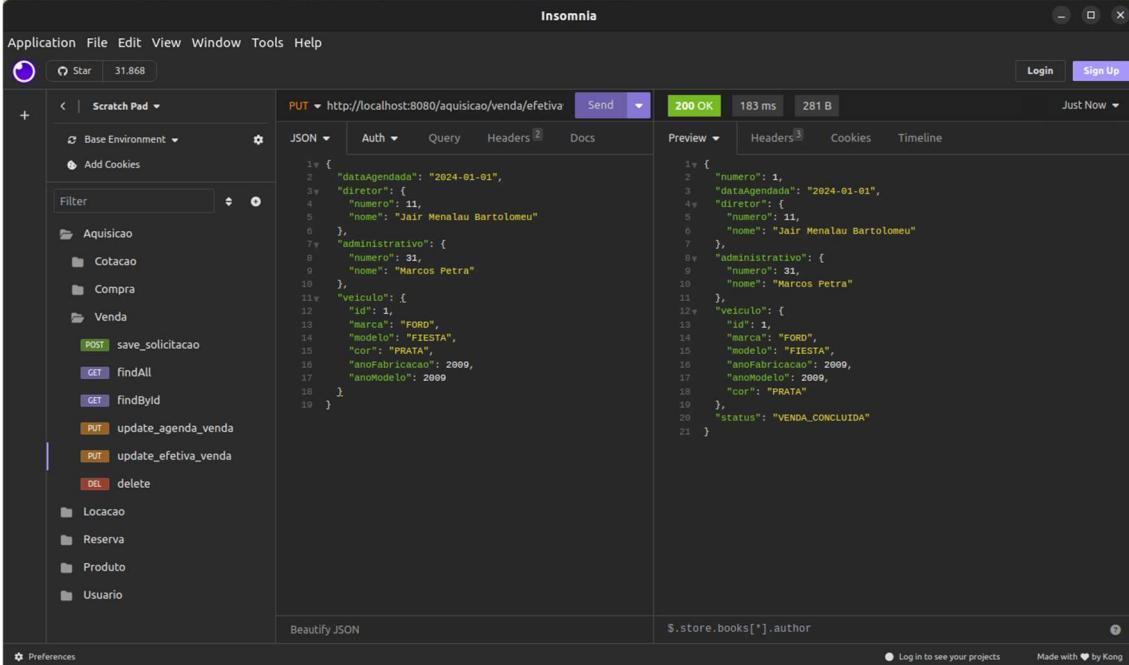
```



A screenshot of a terminal window titled "pedro@VMLxUbt22DskDsv01: ~/Dsv/fiap/tech\_Challenge". The window displays the following JSON content:

```
pedro@VMLxUbt22DskDsv01:~/Dsv/fiap/tech_Challenge$ cat db_reserva.json
{
  "reserva": [
    {
      "dataRetirada": [
        2024,
        1,
        1
      ],
      "qtdeDiarias": 365,
      "vlrDiaria": 0,
      "veiculo": {
        "id": 1,
        "marca": "FORD",
        "modelo": "FIESTA",
        "anoFabricacao": 2009,
        "anoModelo": 2009,
        "cor": "PRATA"
      }
    }
}
}pedro@VMLxUbt22DskDsv01:~/Dsv/fiap/tech_Challenge$ █
```

## Efetiva venda

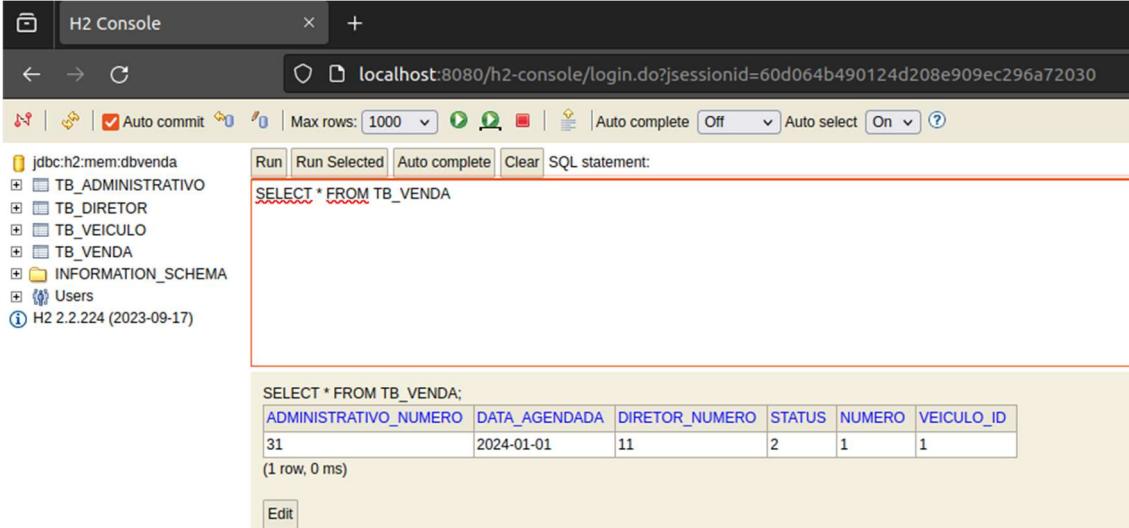


The screenshot shows the Insomnia REST client interface. The URL is `PUT http://localhost:8080/aquisicao/venda/efetiva`. The response status is `200 OK` with a response time of `183 ms` and a size of `281 B`. The response body is a JSON object representing a sale record:

```
1y {
2 "dataAgendada": "2024-01-01",
3 "diretor": {
4 "numero": 11,
5 "nome": "Jair Menalau Bartolomeu"
6 },
7 "administrativo": {
8 "numero": 31,
9 "nome": "Marcos Petra"
10 },
11 "veiculo": {
12 "id": 1,
13 "marca": "FORD",
14 "modelo": "FIESTA",
15 "cor": "PRATA",
16 "anoFabricacao": 2009,
17 "anoModelo": 2009
18 }
19 }
```

The response body continues with more details about the vehicle and the sale status:

```
1y {
2 "numero": 1,
3 "dataAgendada": "2024-01-01",
4 "diretor": {
5 "numero": 11,
6 "nome": "Jair Menalau Bartolomeu"
7 },
8 "administrativo": {
9 "numero": 31,
10 "nome": "Marcos Petra"
11 },
12 "veiculo": {
13 "id": 1,
14 "marca": "FORD",
15 "modelo": "FIESTA",
16 "anoFabricacao": 2009,
17 "anoModelo": 2009,
18 "cor": "PRATA"
19 },
20 "status": "VENDA_CONCLUIDA"
21 }
```



The screenshot shows the H2 Console interface. The URL is `localhost:8080/h2-console/login.do?jsessionid=60d064b490124d208e909ec296a72030`. The query window contains the following SQL statement:

```
SELECT * FROM TB_VENDA
```

The results of the query are displayed in a table:

ADMINISTRATIVO_NUMERO	DATA_AGENDADA	DIRETOR_NUMERO	STATUS	NUMERO	VEICULO_ID
31	2024-01-01	11	2	1	1

(1 row, 0 ms)

```
ch db_planomanutencao.json
\{^_^}/ hi!

Loading db_planomanutencao.json
Done

Resources
http://localhost:3001/veiculo

Home
http://localhost:3001

Type s + enter at any time to create a snapshot of the database
Watching...

DELETE /veiculo/1 200 44.553 ms - 2
```

```
\{^_^}/ hi!

Loading db_veiculo.json
Done

Resources
http://localhost:3003/veiculo

Home
http://localhost:3003

Type s + enter at any time to create a snapshot of the database
Watching...

DELETE /veiculo/1 200 34.142 ms - 2
```

## Retira seguro

```

PUT http://localhost:8080/aquisicao/venda/retiraseguro/1
{
  "dataAgendada": "2024-01-01",
  "diretor": {
    "numero": 11,
    "nome": "Jair Menalau Bartolomeu"
  },
  "administrativo": {
    "numero": 31,
    "nome": "Marcos Petra"
  },
  "veiculo": {
    "id": 1,
    "marca": "FORD",
    "modelo": "FIESTA",
    "cor": "PRATA",
    "anoFabricacao": 2009,
    "anoModelo": 2009
  }
}

```

ADMINISTRATIVO_NUMERO	DATA_AGENDADA	DIRETOR_NUMERO	STATUS	NUMERO	VEICULO_ID
31	2024-01-01	11	3	1	1

(1 row, 0 ms)

Miro:

[https://miro.com/app/board/uXjVNBqzMNU=/?share\\_link\\_id=855194456172](https://miro.com/app/board/uXjVNBqzMNU=/?share_link_id=855194456172)