

Instituto Social Nossa Senhora de Fátima

TK Soluções De Informática



Concessionária de Carro

São Paulo

2020

INTEGRANTES:

João Victor Mendes de Souza INFO-B Nº15

Kauã Cardoso da Mata Nº18

Matheus Silva de Carvalho Vieira INFO-B Nº23

Pedro Costa de Carvalho INFO-B Nº27

Samuel Dos Santos Silva Fiuza Nº32

Link Trello:

<https://trello.com/invite/b/hCu7iMuu/04d3b8d648f7df4007db43be6be680af/concessionaria-de-carro>

Link Git:

<https://github.com/JVZer0/Concessionaria-de-Carro>

ÍNDICE

Módulos.....	3
Casos de uso.....	3
Prototipação.....	9
MER.....	23
Script Banco de Dados.....	25

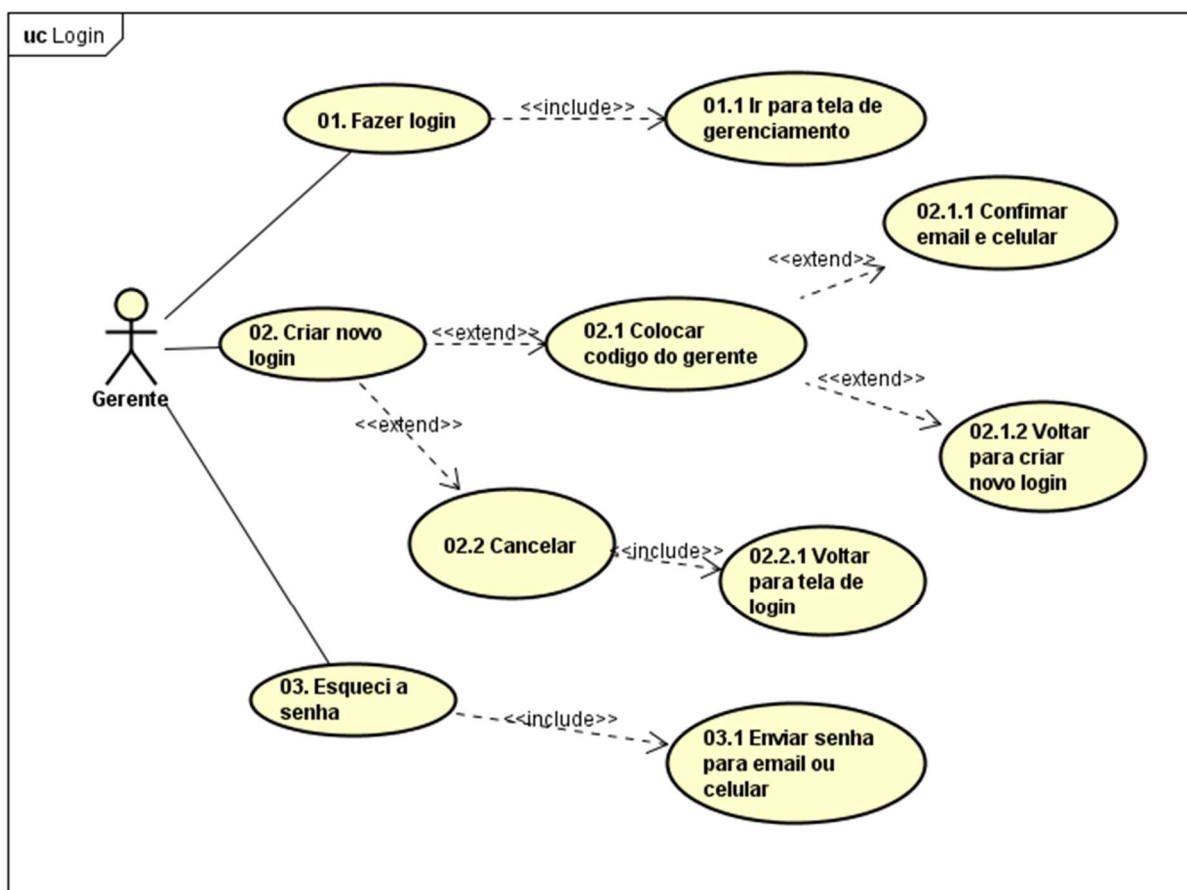
Sistema Concessionaria de Carros

Módulos:

- Login
- Gerenciamento da concessionaria e dos serviços prestados.
- Consulta dos carros
- Simulado de parcelas.
- Agendamento de serviços.

Especificações:

USE CASE LOGIN

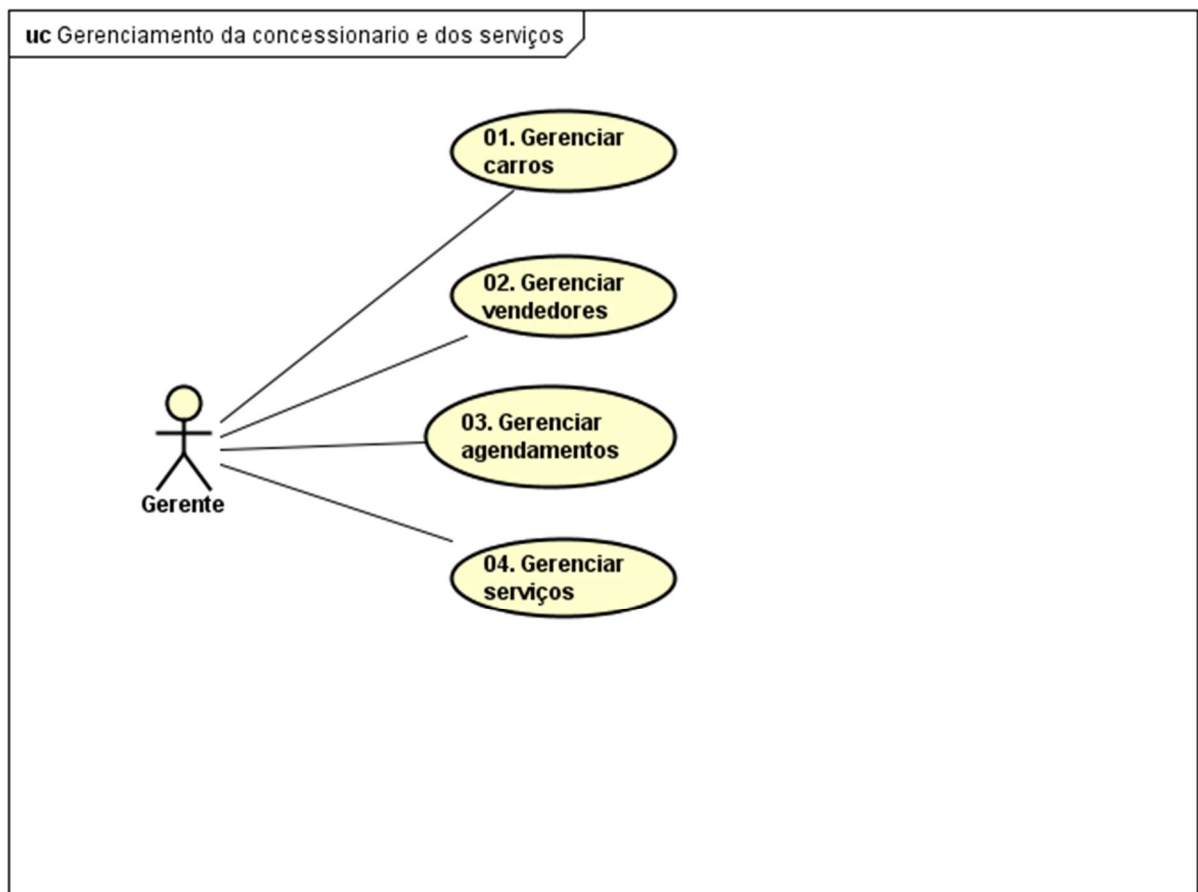


UC-LOGIN.01: O gerente que trabalha na concessionaria deve conseguir entrar no sistema depois que já tiver feito o uma conta no sistema. Para efetuar o login ele deve informar o e-mail e a senha que ele cadastrou na hora do login. Depois de efetuar o login o sistema entrara para a tela de gerenciamento.

UC-LOGIN.02: O gerente deve conseguir criar o login informando seu nome, CPF, número de telefone, e-mail e senha. A senha deve conter no mínimo 8 caracteres. O sistema proíbe o cadastro de e-mails já cadastrados. O sistema irá pedir um código de segurança que somente os gerentes tem, se o gerente não tem o código ele pode cancelar o processo todo do login. Depois disso o sistema enviara um e-mail ou um SMS para a confirmação do e-mail informado no login.

UC-LOGIN.03: Caso o gerente esqueça a senha ele receberá um e-mail ou SMS com a sua nova senha. Para receber essa nova senha ele deve informar seu e-mail, CPF e número de telefone. Ele poderá escolher por onde quer receber esta nova senha.

USE CASE GERENCIAMENTO:



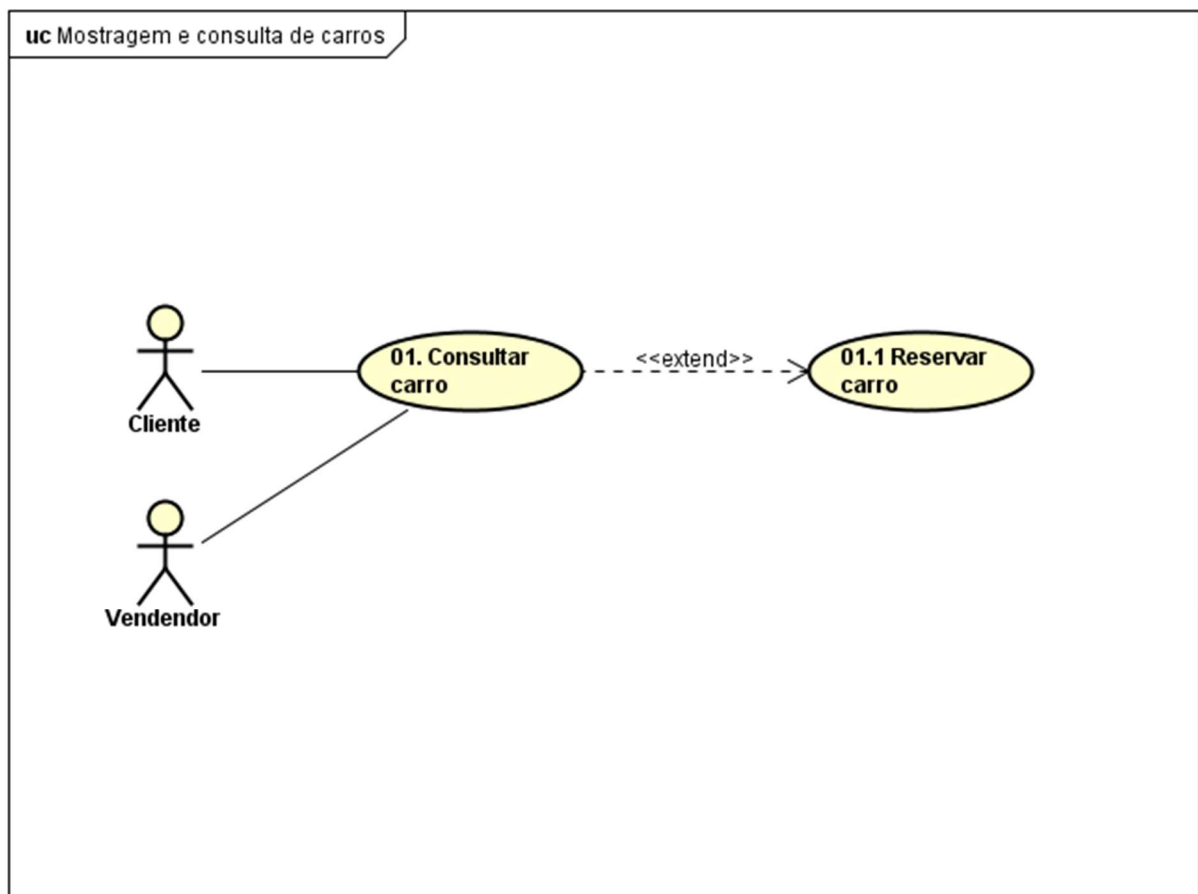
UC-GERENCIAMENTO.01: O gerente poderá adicionar, alterar, consultar e remover os carros. Para isso ele deve informar o nome do Carro, modelo, ano, KM, tipo combustível e preço.

UC-GERENCIMENTO.02: O gerente poderá adicionar, alterar, consultar e remover os vendedores. Para isso ele deve informar o nome, e-mail, CPF e o número de telefone.

UC-GERENCIMENTO.03: O gerente poderá adicionar, alterar, consultar e remover os agendamentos. Para isso ele deve informar as informações dos carros, os tipos de serviços, os horários, os pagamentos e o nome do cliente.

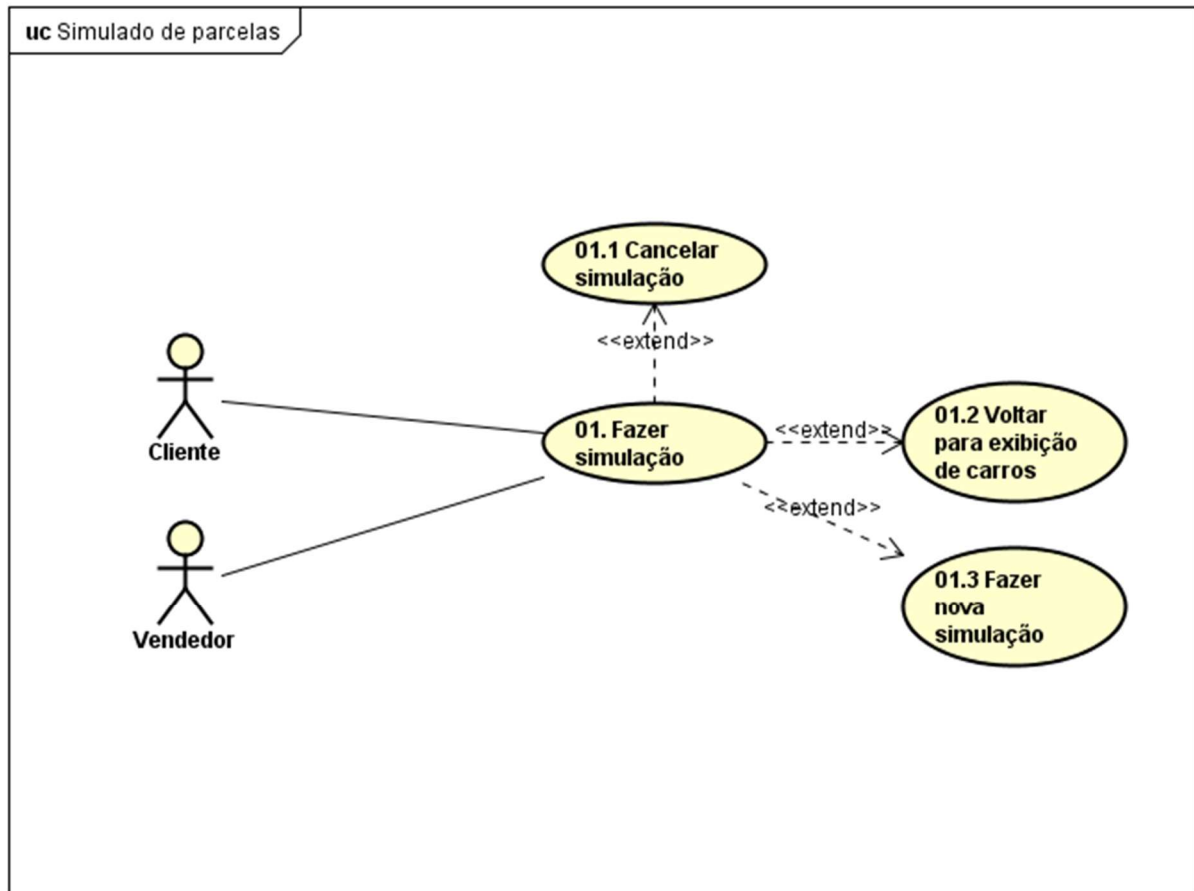
UC-GERENCIMENTO.04: O gerente poderá adicionar, alterar, consultar e remover os serviços prestados pela concessionária. Para isso ele deve informar os tipos de serviços, os preços e os custos.

USE CASE CONSULTA/MOSTRAGEM DE CARROS:



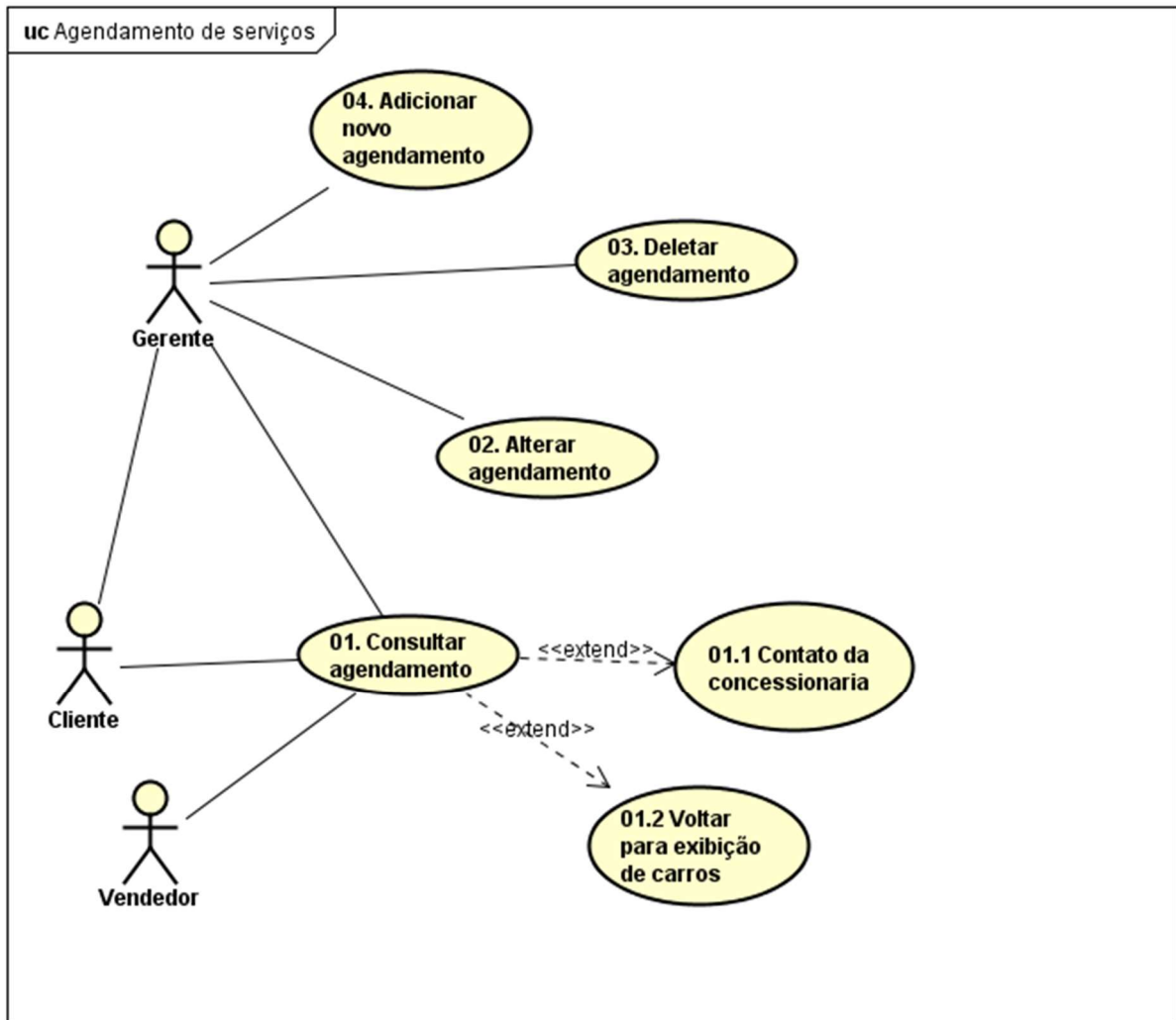
UC-CONSULTADECARROS.01: O vendedor e cliente poderão consultar os carros disponíveis e se o cliente tiver interesse no carro ele pode reservar o carro e entrar em contato com o vendedor.

USE CASE SIMULAÇÃO DE PARCELAS:



UC-SIMULAÇÃO DE PARCELAS.01: O vendedor e cliente poderão fazer a simulação que funciona da seguinte forma, o cliente informa a quantidades de parcelas que ele irá pagar, com isso o sistema calcula os juros e dá um resultado final para o cliente. Se ele não gostar desse resultado ele pode cancelar, voltar para a exibição de carros ou fazer uma nova simulação.

USE CASE AGENDAMENTOS DE SERVIÇOS:



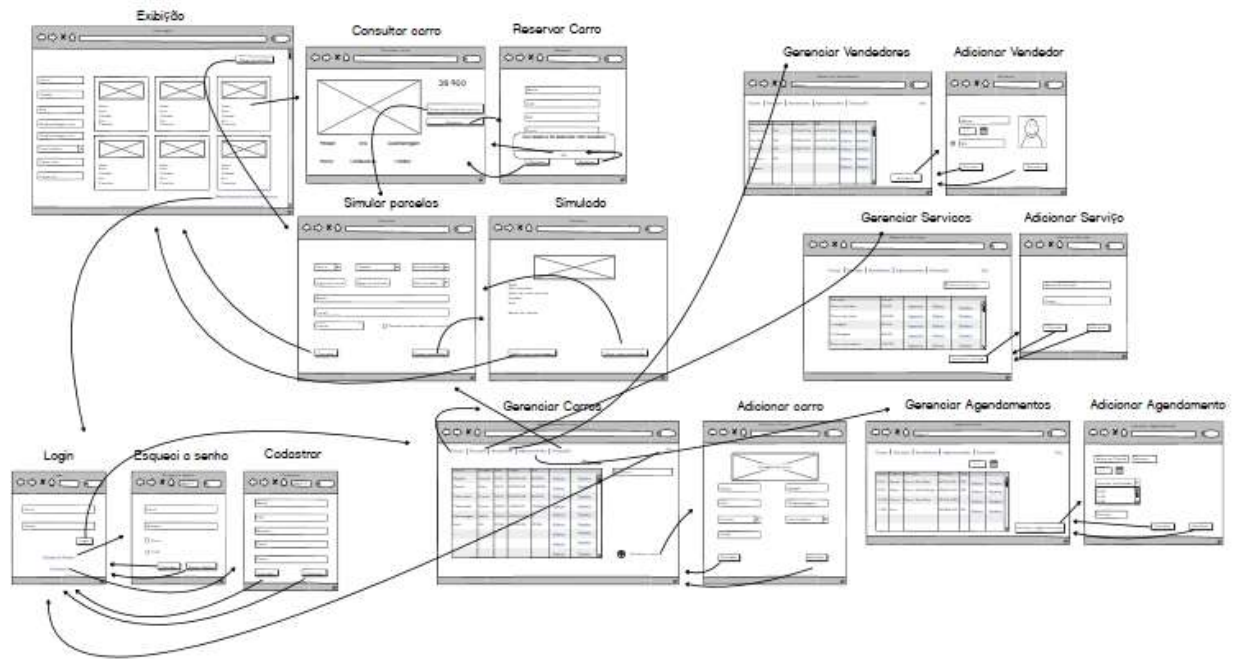
UC-AGENDAMENTOS DE SERVIÇOS.01: O gerente, cliente e o vendedor poderão consultar os agendamentos. Se o cliente quiser alterar ou deletar seus agendamentos ele terá o contato da concessionaria para informar isto.

UC-AGENDAMENTOS DE SERVIÇOS.02: O gerente pode alterar os agendamentos informando o tipo de serviço, informações dos carros, informações dos clientes, datas e horários marcados, e pagamento.

UC-AGENDAMENTOS DE SERVIÇOS.03: O gerente pode deletar os agendamentos informando o tipo de serviço, informações dos carros, informações dos clientes, datas e horários marcados, e pagamento. Caso o cliente já tenha pago o serviço a concessionaria pode fazer a devolução do dinheiro.

UC-AGENDAMENTOS DE SERVIÇOS.04: O gerente pode adicionar um novo agendamento informando o tipo de serviço, informações dos carros, informações dos clientes, datas e horários marcados, e pagamento.

THUMBANIL PROTOTIPAÇÃO



PROTOTIPAÇÃO

PRTP001.MOSTRAGEM

Evento: Clicar no botão de fazer simulação

Entrada: ...

Saída: Ir para tela de *[[Simulação de parcelas]]*

Evento: Clicar no anúncio do carro

Entrada: ...

Saída: Ir para tela de *[[Consultar carro]]*

Evento: Clicar no botão de filtro

Entrada: Marca, modelo, ano, quilometragem, tipo de combustível, Preço.

Saída: Filtrar os carros disponíveis de acordo

PRTP002.CONSULTARCARRO

Evento: Clicar no botão Fazer simulação

Entrada: ...

Saída: Ir para tela de *[[Simulação de parcelas]]* do carro selecionado

Evento: Clicar no botão Reservar

Entrada: ...

Saída: Ir para tela de *[[Reservar]]*

P RTP003.RESERVAR

Reservar

Nome

CPF

RG

Email

Número

Codigo vendedor

Cancelar Reservar

Evento: Clicar no botão Reservar

Entrada: Nome, CPF, RG, E-mail, Número, código do cliente.

Saída: Receber a mensagem que *[[Sua reserva foi efetuada com sucesso]]*

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela de *[[Consulta de carro]]*

Evento: Clicar no botão OK

Entrada: ...

Saída: Fechar a mensagem recebida.

PRTP004.SIMULADO

Simulado

Marca: [dropdown] Modelo: [dropdown] Ano do modelo: [dropdown]

Preço do carro: [text] Valor da entrada: [text] Qtd parcelas: [dropdown]

Nome: [text]

E-mail: [text]

Celular: [text] ☐ Deseja receber ofertas no email?

Cancelar Fazer simulação

Evento: Clicar no botão Fazer simulação

Entrada: Marca do carro, o modelo, ano, preço do carro, valor da entrada, quantidade de parcelas, nome, e-mail, celular, confirmar a checkbox(opcional)

Saída: Ir para tela [[*de Simulação efetuada*]]

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela de [[*Exibição dos carros*]]

PRTP005.SIMULAÇÃO EFETUADA.

Simulado

[Image placeholder with X]

Total:
Qtd parcelas:
Valor de cada parcela:
Modelo:
Ano:
Nome do cliente:

Voltar para mostragem Fazer nova simulação

Evento: Apertar no botão Fazer nova simulação

Entrada: ...

Saída: Ir para tela de [[*Simulado*]]

Evento: Apertar no botão Voltar para mostragem.

Entrada: ...

Saída: Ir para tela de [[*Consulta de carros*]]

PRTP006.LOGINDOGERENTE

Evento: Apertar no botão de Login

Entrada: Email, senha.

Saída: Ir para tela de *[[Gerenciamento de serviços]]*

Evento: Apertar no link Esqueci a Senha

Entrada: ...

Saída: Ir para tela de *[[Esqueci a senha]]*

Evento: Apertar no link Cadastrar

Entrada: ...

Saída: Ir para tela de *[[Cadastro]]*

PRTP007.ESQUECIASENHA

Evento: Apertar no botão de Enviar Senha

Entrada: Email, Número, Marcar a checkbox de email ou de SMS

Saída: Mandar a nova senha para o email ou para o celular do usuário, e redireciona-lo para tela de *[[Login]]*

Evento: Apertar no botão Cancelar

Entrada: ...

Saída: Ir para tela de *[[Login]]*

P RTP008.CADASTRAR

Evento: Apertar no botão Cadastrar

Entrada: Nome, CPF, Numero, Email, Senha, checkbox de ser é gerente ou não.

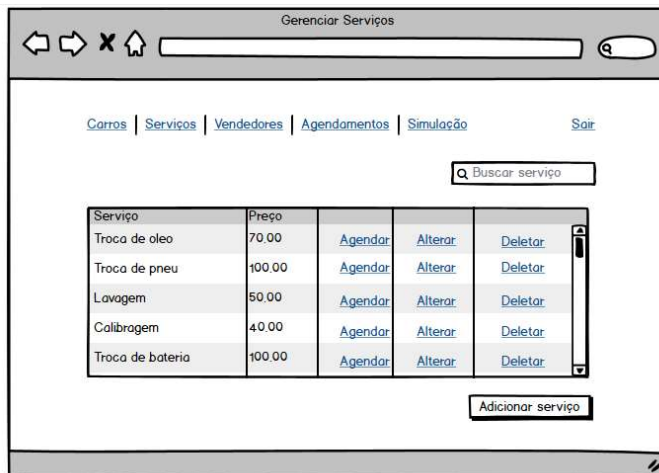
Saída: Cadastrar o usuário e redireciona-lo para tela de *[[Login]]*

Evento: Apertar no botão Cancelar

Entrada: ...

Saída: Ir para tela de *[[Login]]*

PRTP009.GERENCIAR SERVIÇOS



Evento: Clicar em Carros

Entrada: ...

Saída: Ir para tela [[Gerenciamento carros]]

Evento: Clicar em Serviços

Entrada: ...

Saída: Ir para tela [[Gerenciamento serviços]]

Evento: Clicar em Vendedores

Entrada: ...

Saída: Ir para tela [[Gerenciamento de vendedores]]

Evento: Clicar em Agendamentos

Entrada: ...

Saída: Ir para tela [[Gerenciamento de agendamentos]]

Evento: Clicar em Simulação

Entrada: ...

Saída: Ir para tela [[Simulado]]

Evento: Clicar em Sair

Entrada: ...

Saída: Ir para tela [[Login]]

Evento: Clicar no botão de Agendar.

Entrada: ...

Saída: Ir para tela [[Adicionar Agendamento]]

Evento: Clicar no botão de Alterar.

Entrada: ...

Saída: Ir para tela [[Adicionar serviços]]

Evento: Clicar no botão de Deletar.

Entrada: ...

Saída: Mensagem de deletado com sucesso

Evento: Clicar no botão de Adicionar serviço.

Entrada: ...

Saída: Ir para tela [[Adicionar serviços]]

PRTP010. ADICIONAR SERVIÇOS

The wireframe shows a mobile application screen titled "Adicionar Serviço". The top bar is grey and contains four navigation icons (left arrow, right arrow, close 'X', and home house) followed by a search bar with a magnifying glass icon. The main content area is white and contains two text input fields. The first field is labeled "Nome do serviço" and the second is labeled "Preço". Below these fields are two buttons: "Cancelar" on the left and "Adicionar" on the right. The bottom of the screen has a grey bar with a small double-slash icon in the bottom right corner.

Evento: Clicar no botão de Adicionar.

Entrada: Nome do Serviço, Preço.

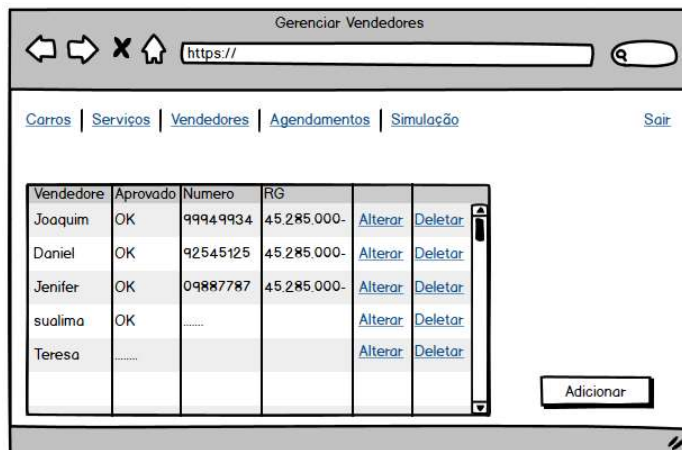
Saída: Mensagem de adicionado com sucesso com botão ok que retorna para tela *[[Gerenciamento serviços]]*.

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela *[[Gerenciamento serviços]]*.

PRTP011.GERENCIAR VENDEDORES



Evento: Clicar em Carros

Entrada: ...

Saída: Ir para tela [[Gerenciamento carros]]

Evento: Clicar em Serviços

Entrada: ...

Saída: Ir para tela [[Gerenciamento serviços]]

Evento: Clicar em Vendedores

Entrada: ...

Saída: Ir para tela [[Gerenciamento de vendedores]]

Evento: Clicar em Agendamentos

Entrada: ...

Saída: Ir para tela [[Gerenciamento de agendamentos]]

Evento: Clicar em Simulação

Entrada: ...

Saída: Ir para tela [[Simulado]]

Evento: Clicar em Sair

Entrada: ...

Saída: Ir para tela [[Login]]

Evento: Clicar no botão de Alterar.

Entrada: ...

Saída: Ir para tela [[Adicionar Vendedores]]

Evento: Clicar no botão de Deletar.

Entrada: ...

Saída: Mensagem de deletado com sucesso

Evento: Clicar no botão de Adicionar.

Entrada: ...

Saída: Ir para tela [[Adicionar Vendedores]]

P RTP012. ADICIONAR VENDEDORES

Adicionar vendedores

Nome

/ /

RG

Cancelar

Adicionar

Evento: Clicar no botão de Adicionar carro.

Entrada: Nome do vendedor, data de nascimento, Numero RG

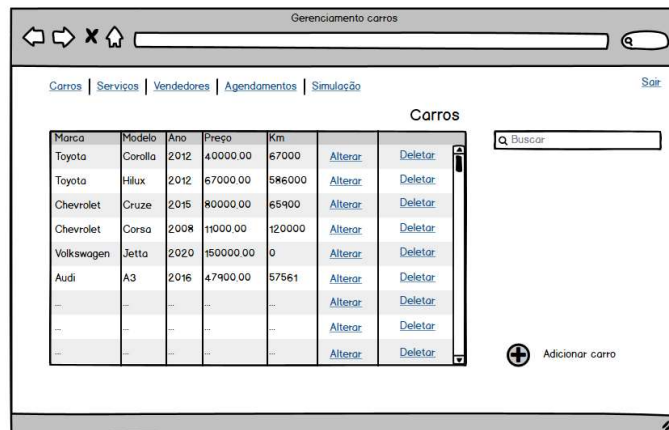
Saída: Mensagem de adicionado com sucesso com botão ok que retorna para tela *[[Gerenciamento de vendedores]]*.

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela *[[Gerenciamento de vendedores]]*.

PRTP013.GERENCIAMENTO DE CARROS



Evento: Clicar em Carros

Entrada: ...

Saída: Ir para tela *[[Gerenciamento carros]]*

Evento: Clicar em Serviços

Entrada: ...

Saída: Ir para tela *[[Gerenciamento serviços]]*

Evento: Clicar em Vendedores

Entrada: ...

Saída: Ir para tela *[[Gerenciamento de vendedores]]*

Evento: Clicar em Agendamentos

Entrada: ...

Saída: Ir para tela *[[Gerenciamento de agendamentos]]*

Evento: Clicar em Simulação

Entrada: ...

Saída: Ir para tela *[[Simulado]]*

Evento: Clicar em Sair

Entrada: ...

Saída: Ir para tela *[[Login]]*

Evento: Clicar em busca

Entrada: Modelo do carro, ou marca.

Saída: Carros com aquele modelo ou marca

Evento: Clicar no botão de Alterar.

Entrada: ...

Saída: Ir para tela *[[Adicionar carros]]*

Evento: Clicar no botão de Deletar.

Entrada: ...

Saída: Mensagem de deletado com sucesso

Evento: Clicar no botão de Adicionar carro.

Entrada: ...

Saída: Ir para tela *[[Adicionar carros]]*

P RTP014.ADICIONAR CARROS

Adicionar Carros

Imagem do carro

Marca

Modelo

Ano

Quilometragem

Cambio

Combustivel

Preço

Cancelar

Adicionar

Evento: Clicar no botão de Adicionar carro.

Entrada: Imagem do carro, Marca do carro, Modelo do carro, Ano do carro, Quilometragem do carro, Tipo de câmbio, Tipo de combustível, Preço

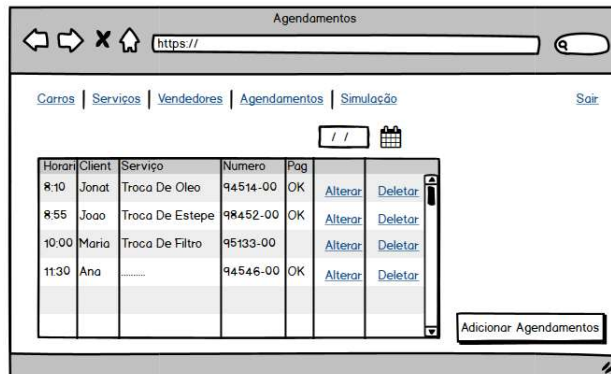
Saída: Mensagem de adicionado com sucesso com botão ok que retornara para tela *[[Gerenciamento carros]]*.

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela *[[Gerenciamento carros]]*.

PRTP015.AGENDAMENTOS



Evento: Clicar em Carros

Entrada: ...

Saída: Ir para tela *[[Gerenciamento carros]]*

Evento: Clicar em Serviços

Entrada: ...

Saída: Ir para tela *[[Gerenciamento serviços]]*

Evento: Clicar em Vendedores

Entrada: ...

Saída: Ir para tela *[[Gerenciamento de vendedores]]*

Evento: Clicar em Agendamentos

Entrada: ...

Saída: Ir para tela *[[Gerenciamento de agendamentos]]*

Evento: Clicar em Simulação

Entrada: ...

Saída: Ir para tela *[[Simulado]]*

Evento: Clicar em Sair

Entrada: ...

Saída: Ir para tela *[[Login]]*

Evento: Digitar data

Entrada: Data

Saída: Agendamentos para aquela data

Evento: Clicar no botão de Alterar.

Entrada: ...

Saída: Ir para tela *[[Adicionar agendamentos]]*

Evento: Clicar no botão de Deletar.

Entrada: ...

Saída: Mensagem de deletado com sucesso

Evento: Clicar no botão de Adicionar Agendamentos.

Entrada: ...

Saída: Ir para tela *[[Adicionar agendamentos]]*

PRTP016.ADICIONAR AGENDAMENTOS

Adicionar Agendamento

Nome do Cliente Numero Marca

/ /

Modelo

horarios disponiveis

8,40
5,30
3,45

Ano

Placa

Serviço ☐ Pago

Cancelar Adicionar

Evento: Clicar no botão Adicionar

Entrada: Nome do cliente, Número de telefone, Marca do carro, Modelo do carro, Ano do carro, Placa do carro, Horário, Data, Serviço, check de pago ou não

Saída: Mensagem de adicionado com sucesso com botão ok que retornara para tela [[Gerenciamento de agendamentos]]

Evento: Clicar no botão Cancelar

Entrada: ...

Saída: Ir para tela [[Gerenciamento de agendamentos]]

MODELAGEM:

TB_LOGIN: Armazena os dados dos usuários que podem entrar no sistema.

TB_CARRO: Armazena os dados dos carros que estão disponíveis na concessionária.

TB_SERVICO: Armazena os serviços e os preços que nossa concessionária oferece para nossos clientes e serviços que podem ser feitos nos carros que estão na concessionária.

TB_CARRO_SERVICO: Guarda as informações do carro da concessionária que precisa de serviço, e do serviço que irá ser feito. Tem relação **N:N** com tb_servico e tb_carro, pois um serviço pode estar em vários carros, e um carro pode ter vários serviços.

TB_CLIENTE: Guarda as informações caso o cliente escolha a opção de receber promoções.

TB_RESERVA: Guarda as informações do cliente caso ele reserve um carro. Tem relação **1:1** com tb_carro, pois uma reserva pode ter um carro. Tem relação **1:N** com tb_vendedor pois uma reserva pode ter um vendedor, e um vendedor pode estar em várias reservas. Tem relação **1:N** com tb_cliente pois um cliente pode ter várias reservas.

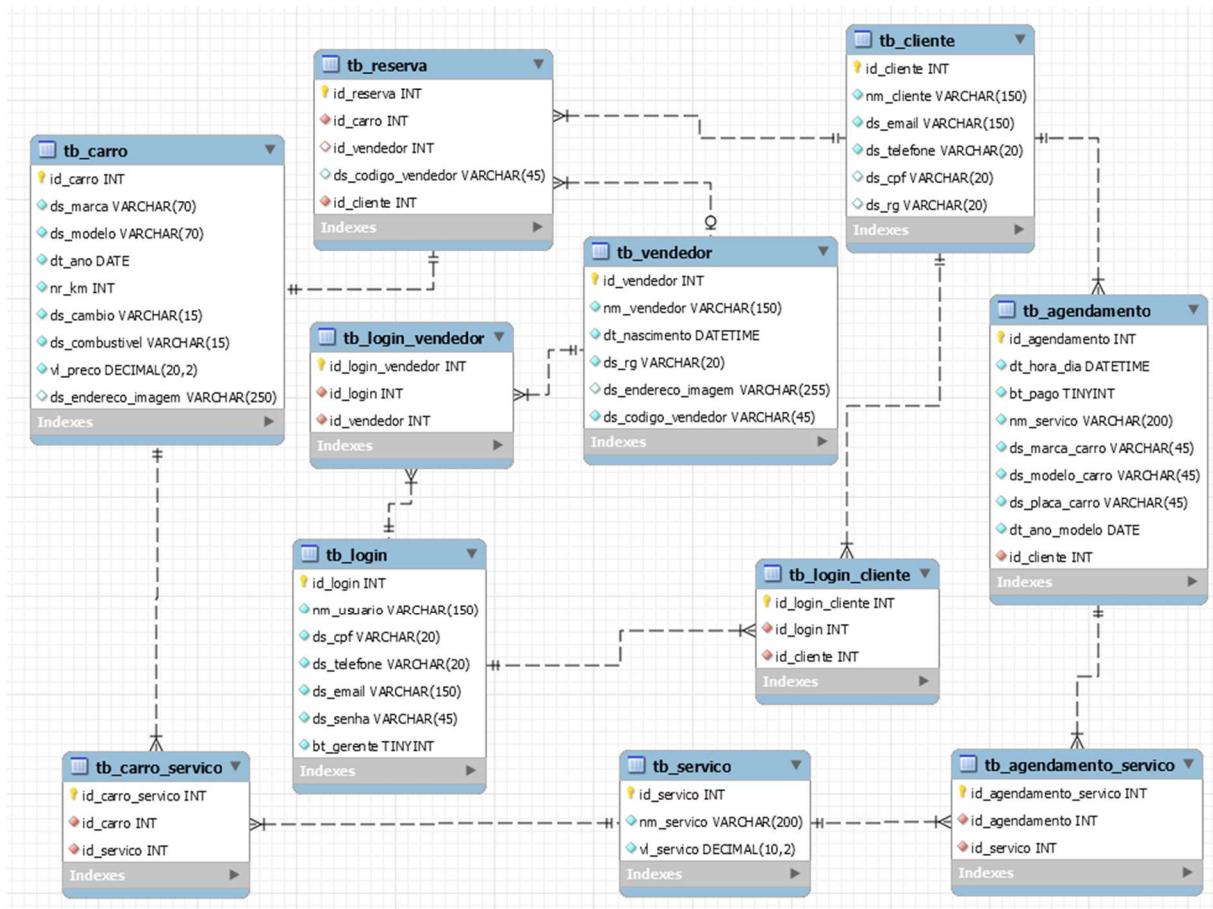
TB_VENDEDOR: Armazena os dados dos vendedores que atuam na concessionária.

TB_AGENDAMENTO: Guarda as informações do agendamento que o cliente fez. Tem relação **1:N** com tb_cliente pois um cliente pode ter vários agendamentos.

TB_AGENDAMENTO_SERVICO: Guarda as informações do agendamento e do serviço que irá ser feito no carro do cliente. Tem relação **N:N** com tb_servico e tb_agendamento, pois um serviço pode estar em vários agendamentos, e um agendamento pode ter vários serviços.

TB_LOGIN_CLIENTE: Tem relação **N:N** com tb_login e tb_cliente, pois um cliente pode estar em vários logins e um login pode ter vários clientes.

TB_LOGIN_VENDEDOR Tem relação **N:N** com **tb_login** e **tb_vendedor**, pois um login pode ter vários vendedores, e um vendedor pode estar em vários logins.



SCRIPT BANCO DE DADOS:

```

1 • create database concessionariadecarro;
2
3 • use concessionariadecarro;
4
5 • create table tb_carro(
6     id_carro int primary key auto_increment,
7
8     ds_marca varchar(70),
9     ds_modelo varchar(70),
10    dt_ano datetime,
11    nr_km int,
12    ds_cambio varchar(15),
13    ds_combustivel varchar(15),
14    vl_preco decimal(20,2),
15    ds_endereco_imagem varchar(250)
16 );
17
18 • create table tb_reserva(
19     id_reserva int primary key auto_increment,
20     id_carro int,
21     id_vendedor int,
22     ds_codigo_vendedor varchar(45),
23     id_cliente int,
24     foreign key (id_cliente) references tb_cliente (id_cliente)
25 );
26
27 • create table tb_vendedor(
28     id_vendedor int primary key auto_increment,
29     nm_vendedor varchar(150),
30     dt_nascimento datetime,
31     dt_rg varchar(20),
32     ds_endereco_imagem varchar(255),
33     ds_codigo_vendedor varchar (45)
34 );
35
36 • create table tb_login(
37     id_login int primary key auto_increment,
38     nm_usuario varchar(150),
39     ds_cpf varchar(20),
40     ds_telefone varchar(45),
41     ds_email varchar(150),

```

```
41     ds_email varchar(150),
42     ds_senha varchar(45),
43     bt_gerente boolean
44 );
45
46 • create table tb_login_vendedor(
47     id_login_vendedor int primary key auto_increment,
48     id_login int,
49     id_vendedor int,
50     foreign key (id_login) references tb_login (id_login),
51     foreign key(id_vendedor) references tb_vendedor (id_vendedor)
52 );
53
54 • create table tb_cliente(
55     id_cliente int primary key auto_increment,
56     nm_cliente varchar (150),
57     ds_email varchar(150),
58     ds_telefone varchar(20),
59     ds_cpf varchar(20),
60     ds_rg varchar(20)
61 );
62
63 • create table tb_login_cliente(
64     id_login_cliente int primary key auto_increment,
65     id_login int,
66     id_cliente int,
67     foreign key (id_login) references tb_login (id_login),
68     foreign key(id_cliente) references tb_cliente (id_cliente)
69 );
70
71 • create table tb_servico(
72     id_servico int primary key auto_increment,
73     nm_servico varchar(200),
74     vl_servico decimal(10,2)
75 );
76
77
78 • create table tb_carro_servico(
79     id_carro_servico int primary key auto_increment,
80     id_carro int,
81     id_servico int,
```

```

81     id_servico int,
82     foreign key (id_carro) references tb_carro (id_carro),
83     foreign key (id_servico) references tb_servico (id_servico)
84 );
85
86
87 • create table tb_agendamento(
88     id_agendamento int primary key auto_increment,
89     dt_hora_dia datetime,
90     bt_pago boolean,
91     nm_servico varchar(200),
92     ds_marca_carro varchar(45),
93     ds_modelo_carro varchar(45),
94     ds_placa_carro varchar(45),
95     dt_ano_modelo date,
96     id_cliente int,
97     foreign key(id_cliente) references tb_cliente (id_cliente)
98 );
99
100 • create table tb_agendamento_servico(
101     id_agendamento_servico int primary key auto_increment,
102     id_agendamento int,
103     id_servico int,
104     foreign key (id_servico) references tb_servico (id_servico),
105     foreign key (id_agendamento) references tb_cliente (id_agendamento)
106 );
107
108
109 • insert into tb_carro(ds_marca, ds_modelo, dt_ano, nr_km, ds_cambio, ds_combustivel, vl_preco, ds_endereco_imagem)
110 values("Ford","1.6 FREESTYLE 16V FLEX 4P",'2013-01-01', 90.00,"Manual","Gasolina",41.890,null);
111
112 • insert into tb_carro(ds_marca, ds_modelo, dt_ano, nr_km, ds_cambio, ds_combustivel, vl_preco, ds_endereco_imagem)
113 values("Toyota Corolla","2.0 XEI 16V FLEX 4P",'2018-01-01', 28.162,"Automatica","Gasolina",88.490,null);
114
115
116 • insert into tb_login(nm_usuario, ds_cpf, ds_email, ds_senha, bt_gerente)
117 values("Joao","646.107.649-20","Joaodasilva@.com","joaosilva",true);
118
119 • insert into tb_login(nm_usuario, ds_cpf, ds_email, ds_senha, bt_gerente)
120 values("Paulo","163.946.072-15","paulodacosta@.com","wifi",true);
121

```

```

121
122
123 • insert into tb_reserva(id_carro, id_vendedor,ds_codigo_vendedor, id_cliente ) values (1, 2,"542910",1);
124 • insert into tb_reserva(id_carro, id_vendedor,ds_codigo_vendedor, id_cliente ) values (1, 1,"122110",2);
125
126
127 • insert into tb_vendedor(nm_vendedor, dt_nascimento, dt_rg, ds_endereco_imagem, ds_codigo_vendedor)
128 values("Pedro", '2001-03-1','11 345 678-9",null,"521110");
129
130 • insert into tb_vendedor( nm_vendedor , dt_nascimento ,dt_rg, ds_endereco_imagem, ds_codigo_vendedor)
131 values( "Pedro", '1997-06-05','11 217 198-6",null,"187260");
132
133
134 • insert into tb_login_vendedor(id_login_vendedor, id_login , id_vendedor ) values (1, 2, 3);
135
136 • insert into tb_login_vendedor(id_login_vendedor, id_login , id_vendedor ) values (5, 6, 2);
137
138
139 • insert into tb_cliente(nm_cliente , ds_email , ds_telefone , ds_cpf , ds_rg )
140 values("Cassiano", "Cassianocostacp@gmail.com", "5989-7612", "683.240.302-04", "82.965.531-1");
141
142 • insert into tb_cliente(nm_cliente , ds_email , ds_telefone , ds_cpf , ds_rg )
143 values("Marcelo", "marceloanocostacp@gmail.com", "5169-3622", "173.845.402-14", "32.666.237-9");
144
145
146
147 • insert into tb_login_cliente (id_login, id_cliente) values (2, 1);
148 • insert into tb_login_cliente(id_login, id_cliente) values ( 2, 1);
149
150
151
152 • insert into tb_servico( nm_servico ,vl_servico)
153 values("troca de oleo", 20.5 );
154
155 • insert into tb_servico( nm_servico ,vl_servico)
156 values("troca de escapamento", 12.5);
157
158
159
160
161 • insert into tb_carro_servico(id_carro_servico ,id_carro , id_servico) values(1,4,2);

```

```

161 • insert into tb_carro_servico(id_carro_servico ,id_carro , id_servico) values(1,4,2);
162
163 • insert into tb_carro_servico(id_carro_servico ,id_carro , id_servico) values(3,5,6);
164
165
166 • insert into tb_agendamento( dt_hora_dia ,bt_pago, nm_servico, ds_marca_carro, ds_modelo_carro,ds_placa_carro,dt_ano_modelo,id_cliente)
167 values( '2020-12-02 12:00', true, "troca de pneu", "Chevrolet", "6.2 2SS CONVERSÍVEL V", "RI02A18", '2014-01-01',1);
168
169 • insert into tb_agendamento( dt_hora_dia ,bt_pago, nm_servico, ds_marca_carro, ds_modelo_carro,ds_placa_carro,dt_ano_modelo,id_cliente)
170 values( '2020-01-01 16:00', true, "troca de oleo", "Ford", "2.0 TITANIUM PLUS 16V FLEX 4P POWERSHIFT", "FJ01B98", '2013-01-01',9);
171
172
173
174 • insert into tb_agendamento_servico(id_agendamento_servico ,id_agendamento, id_servico) values(1,3,7);
175
176 • insert into tb_agendamento_servico(id_agendamento_servico ,id_agendamento, id_servico) values(8,2,5);
177
178
179
180
181
182 • select *
183     from tb_reseva
184     inner
185     join tb_cliente
186     on tb_reserva.id_cliente = tb_reserva.id_cliente;
187
188
189 • select *
190     from tb_login_vendedor
191     inner
192     join tb_login
193     on tb_login_vendedor.id_login = tb_login_vendedor.id_login
194     inner
195     join tb_cliente
196     on tb_login_vendedor.id_cliente=tb_login_vendedor.id_cliente;
197
198
199 • select *
200     from tb_carro_servico
201     inner

```



```

201     inner
202     | join tb_carro
203       on tb_carro_servico.id_carro = tb_carro_servico.id_carro
204     inner
205       join tb_servico
206         on tb_carro_servico.id_servico=tb_carro_servico.id_servico;
207
208
209
210 • select *
211     from tb_login_cliente
212     inner
213       join tb_login
214         on tb_login_cliente.id_login = tb_login.id_login
215     inner
216       join tb_cliente
217         on tb_login_cliente.id_cliente = tb_login_cliente.id_cliente;
218
219
220 • select *
221     from tb_agendamento
222     inner
223       join tb_cliente
224         on agendamento.id_cliente = agendamento.id_cliente;
225
226
227 • select *
228     from tb_agendamento_servico
229     inner
230       join tb_servico
231         on tb_agendamento_servico.id_servico = tb_agendamento_servico.id_servico
232     inner
233       join tb_servico
234         on tb_agendamento_servico.id_agendamento = tb_agendamento_servico.id_agendamento;
235
236
237
---
```