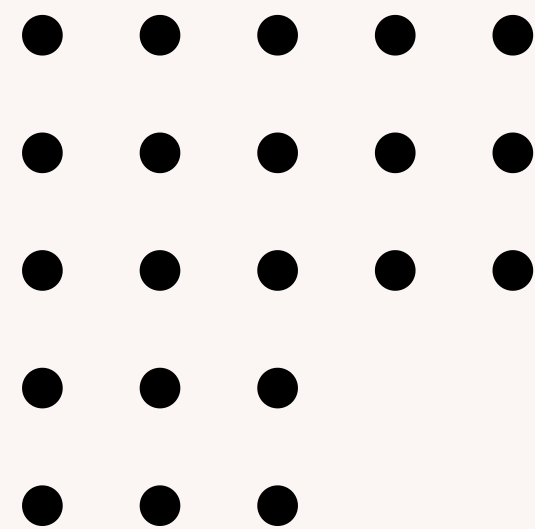




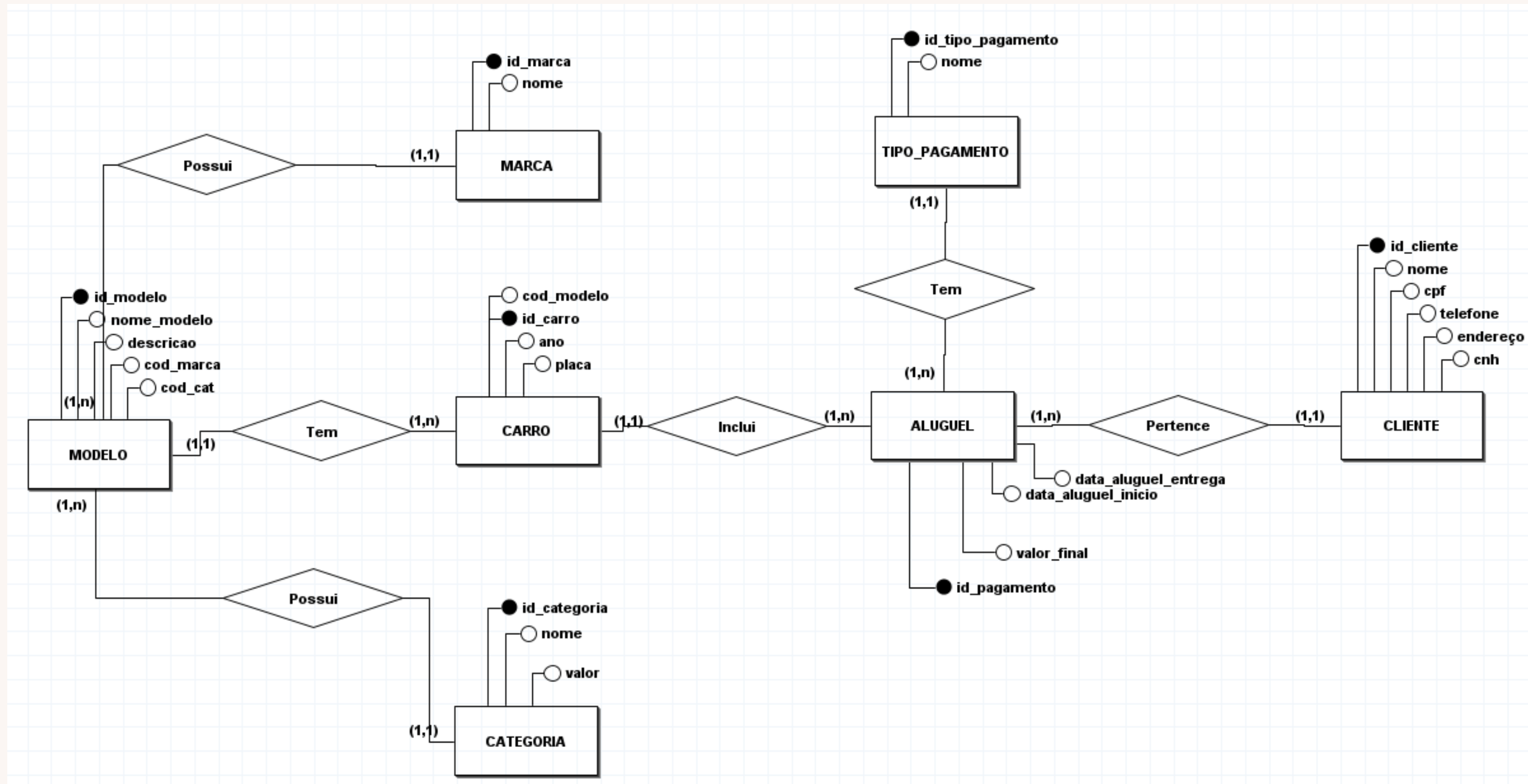
# SISTEMA DE ALUGUEL DE CARROS

# DriveEasy

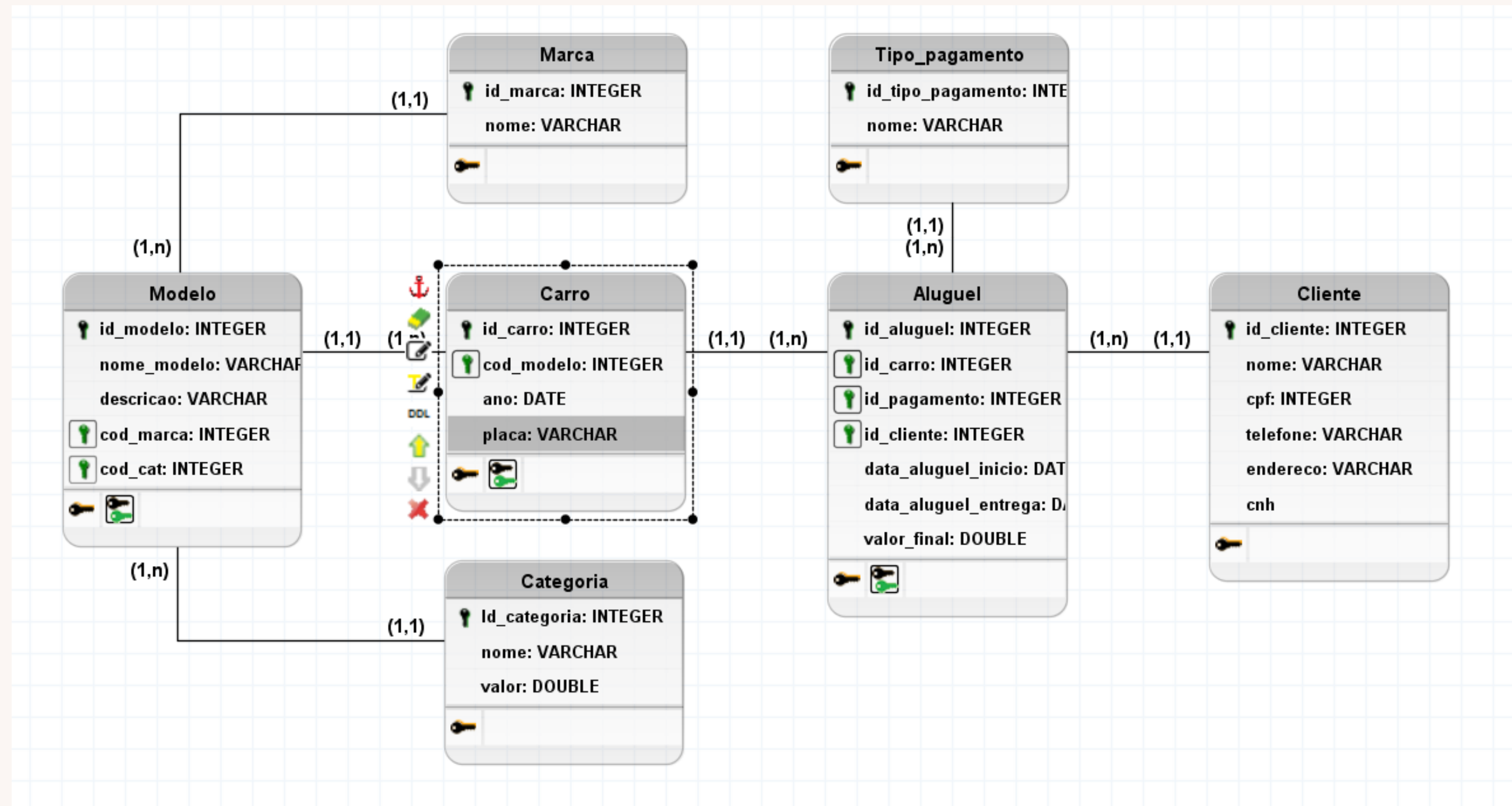
por Breno, Eduardo, Matheus, Ulysses



# Conceitual



# Lógico



# Físico e Insert

```
CREATE TABLE MODELO(  
  ID_MODELO SERIAL PRIMARY KEY,  
  NOME_MODELO VARCHAR(50) NOT NULL,  
  DESCRICAO VARCHAR(200),  
  FK_ID_MARCA INTEGER REFERENCES MARCA(ID_MARCA),  
  FK_ID_CATEGORIA INTEGER REFERENCES CATEGORIA(ID_CATEGORIA))
```

```
CREATE TABLE ALUGUEL(  
  ID_ALUGUEL SERIAL PRIMARY KEY,  
  FK_ID_CARRO INTEGER REFERENCES CARRO(ID_CARRO),  
  FK_ID_CLIENTE INTEGER REFERENCES CLIENTE(ID_CLIENTE),  
  FK_PAGAMENTO INTEGER REFERENCES TIPO_PAGAMENTO(ID_TIPO_PAGAMENTO),  
  DATA_ALUGUEL_INICIAL DATE NOT NULL,  
  DATA_ALUGUEL_ENTREGA DATE NOT NULL,  
  VALOR_FINAL FLOAT NOT NULL)
```

```
INSERT INTO MODELO (id_modelo, nome_modelo, descricao, fk_id_marca, fk_id_categoria) VALUES  
(1, 'Corolla', 'Sedan compacto', 1, 1),  
(2, 'Civic', 'Sedan médio', 2, 1),  
(3, 'Mustang', 'Esportivo', 3, 2),  
(4, 'Cruze', 'Sedan médio', 4, 1),  
(5, 'Altima', 'Sedan luxo', 5, 2),  
(6, 'X5', 'SUV grande', 6, 3),  
(7, 'A4', 'Sedan luxo', 7, 2),  
(8, 'Golf', 'Hatchback', 8, 1),  
(9, 'C-Class', 'Sedan luxo', 9, 2),  
(10, 'Elantra', 'Sedan compacto', 10, 1);
```

```
INSERT INTO ALUGUEL (id_aluguel, fk_id_carro, fk_id_cliente, fk_pagamento,  
  data_aluguel_inicial, data_aluguel_entrega, valor_final) VALUES  
(1, 1, 1, 1, '2022-01-01', '2022-01-10', 500),  
(2, 2, 2, 2, '2022-02-02', '2022-02-09', 600),  
(3, 3, 3, 3, '2022-03-05', '2022-03-15', 700),  
(4, 4, 4, 4, '2022-04-10', '2022-04-20', 550),  
(5, 5, 5, 5, '2022-05-20', '2022-05-30', 650),  
(6, 6, 6, 6, '2022-06-01', '2022-06-11', 750),  
(7, 7, 7, 7, '2022-07-10', '2022-07-20', 850),  
(8, 8, 8, 8, '2022-08-15', '2022-08-25', 600),  
(9, 9, 9, 3, '2022-09-05', '2022-09-15', 650),  
(10, 10, 10, 7, '2022-10-01', '2022-10-11', 700);
```

# Group By e Funções de Agrupamento

```
df_alugueis_por_categoria = pd.read_sql_query("""
SELECT
    Categoria.nome AS categoria_nome,
    COUNT(Aluguel.id_aluguel) AS total_alugueis
FROM
    Aluguel
INNER JOIN Carro ON Aluguel.fk_id_carro = Carro.id_carro
INNER JOIN Modelo ON Carro.fk_id_modelo = Modelo.id_modelo
INNER JOIN Categoria ON Modelo.fk_id_categoria = Categoria.id_categoria
GROUP BY
    Categoria.nome
ORDER BY
    total_alugueis DESC;
""", conn)
df_alugueis_por_categoria
```

	categoria_nome	total_alugueis
0	Sedan	5
1	LUXO	4
2	SUV	1

# Group By e Funções de Agrupamento

```
df_media_valor_pagamento = pd.read_sql_query("""
SELECT
    Tipo_pagamento.nome AS tipo_pagamento_nome,
    AVG(Aluguel.valor_final) AS valor_medio
FROM
    Aluguel
INNER JOIN Tipo_pagamento ON Aluguel.fk_pagamento = Tipo_pagamento.id_tipo_pagamento
GROUP BY
    Tipo_pagamento.nome
ORDER BY
    valor_medio DESC;
""", conn)
df_media_valor_pagamento
```

	tipo_pagamento_nome	valor_medio
0	Pix	775.0
1	Cheque	750.0
2	Transferência	675.0
3	Dinheiro	650.0
4	Financiamento	600.0
5	Boleto Bancário	600.0
6	Cartão de Débito	550.0
7	Cartão de Crédito	500.0

# Left ou Right ou Self ou full Join

```
df_modelos_mesmo_ano = pd.read_sql_query("""SELECT A.ano, A.fk_id_modelo FROM Carro A,  
Carro B WHERE A.fk_id_modelo <> B.fk_id_modelo AND A.ano = B.ano ORDER BY A.ano;""", conn)  
df_modelos_mesmo_ano
```

	ano	fk_id_modelo
0	2019	8
1	2019	5
2	2020	10
3	2020	4
4	2021	1
5	2021	1
6	2021	3
7	2021	3
8	2021	9

# Left, Right, Self ou full Join

```
df_modelos_mesma_descricao = pd.read_sql_query("""SELECT DISTINCT A.nome_modelo,  
B.descricao FROM Modelo A,Modelo B WHERE A.nome_modelo <> B.nome_modelo AND A.descricao  
= B.descricao ORDER BY B.descricao;""", conn)  
df_modelos_mesma_descricao
```

	nome_modelo	descricao
0	Corolla	Sedan compacto
1	Elantra	Sedan compacto
2	A4	Sedan luxo
3	Altima	Sedan luxo
4	C-Class	Sedan luxo
5	Civic	Sedan médio
6	Cruze	Sedan médio



# View com Subconsultas

```
df_aluguel_valor = pd.read_sql_query("""
SELECT
    Aluguel.id_aluguel,
    (SELECT valor FROM Categoria WHERE id_categoria = Modelo.fk_id_categoria) AS valor
FROM
    Modelo
INNER JOIN Carro ON Modelo.id_modelo = Carro.fk_id_modelo
INNER JOIN Aluguel ON Carro.id_carro = Aluguel.fk_id_carro
GROUP BY
    Aluguel.id_aluguel,
    valor
""", conn)
df_aluguel_valor
```

	id_aluguel	valor
0	1	500.0
1	2	500.0
2	3	800.0
3	4	500.0
4	5	700.0
5	6	750.0
6	7	700.0
7	8	500.0
8	9	700.0
9	10	500.0
10	11	500.0
11	12	700.0
12	13	700.0

# Relatórios

## 1. Relatório de Inadimplência e Recuperação de Pagamentos

- Objetivo: Monitorar a saúde financeira do negócio em relação aos pagamentos.

## 2. Relatório de Manutenção Preventiva de Carros

- Objetivo: Garantir a integridade da frota por meio de manutenção preventiva.

## 3. Relatório de Satisfação do Cliente pós-Aluguel

- Objetivo: Avaliar a satisfação dos clientes após a experiência de aluguel.

## 4. Relatório de Categorias de Veículos Mais Populares

- Objetivo: Analisar as preferências de pagamento dos clientes e correlacionar com suas escolhas de veículos.

## 5. Relatório de Análise de Pagamento e Preferências do Cliente

- Objetivo: Entender quais categorias de veículos são mais demandadas pelos clientes.

## 4.Relatório de Categorias de Veículos Mais Populares

```
rel4 = pd.read_sql_query("""SELECT
    CATEGORIA.nome AS "Categoria",
    COUNT(ALUGUEL.id_aluguel) AS "Quantidade de Alugueis",
    AVG(ALUGUEL.valor_final) AS "Receita Média",
    AVG(AGE(ALUGUEL.data_aluguel_entrega, ALUGUEL.data_aluguel_inicial)) AS "Tempo Médio de Aluguel (dias)",
    (COUNT(ALUGUEL.id_aluguel) * 100.0 / (SELECT COUNT(*) FROM ALUGUEL)) AS "Percentual do Total de Alugueis"
FROM
    ALUGUEL
JOIN
    CARRO ON ALUGUEL.fk_id_carro = CARRO.id_carro
JOIN
    MODELO ON CARRO.fk_id_modelo = MODELO.id_modelo
JOIN
    CATEGORIA ON MODELO.fk_id_categoria = CATEGORIA.id_categoria
GROUP BY
    CATEGORIA.nome
ORDER BY
    "Quantidade de Alugueis" DESC;

""",conn)
rel4
```

	Categoria	Quantidade de Alugueis	Receita Média	Tempo Médio de Aluguel (dias)	Percentual do Total de Alugueis
0	Sedan	24	652.333333	14.791667	48.0
1	Luxo	16	690.062500	15.937500	32.0
2	Esportivo	5	662.800000	15.800000	10.0
3	SUV	5	655.400000	15.000000	10.0

# Gráfico Tempo x ID\_Categoria Relatório 4

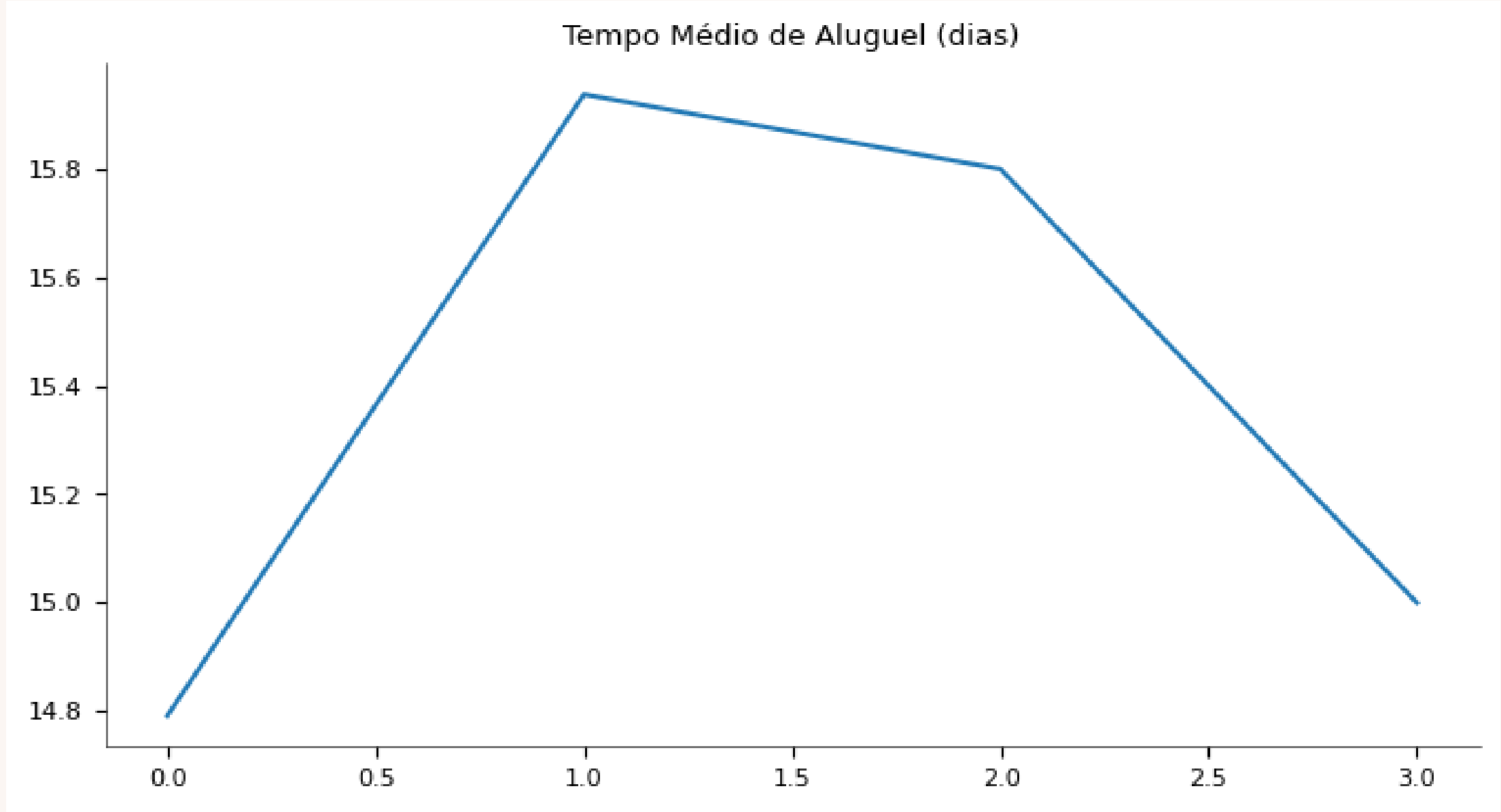


Gráfico Tempo médio x ID\_Categoria:

- ID 0 - Sedan
- ID 1 - Luxo
- ID 2 - Esportivo
- ID 3 - SUV

# 5. Relatório de Análise de Pagamento e Preferências do Cliente

```
rel5 = pd.read_sql_query("""SELECT
    TIPO_PAGAMENTO.nome,
    COUNT(ALUGUEL.id_aluguel),
    AVG(ALUGUEL.valor_final),
    STRING_AGG(DISTINCT CATEGORIA.nome, ', ') AS categorias_preferidas,
    STRING_AGG(DISTINCT CAST(ALUGUEL.fk_id_cliente AS VARCHAR), ', ') AS id_clientes
FROM
    ALUGUEL
JOIN
    TIPO_PAGAMENTO ON ALUGUEL.fk_pagamento = TIPO_PAGAMENTO.id_tipo_pagamento
JOIN
    CARRO ON ALUGUEL.fk_id_carro = CARRO.id_carro
JOIN
    MODELO ON CARRO.fk_id_modelo = MODELO.id_modelo
JOIN
    CATEGORIA ON MODELO.fk_id_categoria = CATEGORIA.id_categoria
GROUP BY
    TIPO_PAGAMENTO.nome
ORDER BY
    COUNT(ALUGUEL.id_aluguel) DESC;
""",conn)
rel5
```

	nome	count	avg	categorias_preferidas	id_clientes
0	Transferência	9	668.222222	Esportivo, Luxo, Sedan, SUV	1, 2, 3, 4, 5, 6, 7, 9
1	Cartão de Crédito	8	670.000000	Luxo, Sedan, SUV	1, 2, 3, 4, 5, 7
2	Dinheiro	7	657.000000	Esportivo, Luxo, Sedan	1, 10, 2, 5, 6, 7
3	Pix	6	710.166667	Luxo, Sedan	1, 10, 3, 6, 7, 9
4	Cheque	5	639.000000	Esportivo, Luxo, Sedan, SUV	5, 6, 7, 8
5	Boleto Bancário	5	691.200000	Luxo, Sedan	2, 3, 6, 8, 9
6	Financiamento	5	628.400000	Luxo, Sedan	10, 3, 4, 8, 9
7	Cartão de Débito	5	652.200000	Luxo, Sedan, SUV	4, 5, 7, 9

# Gráfico de barras horizontal Relatório 5

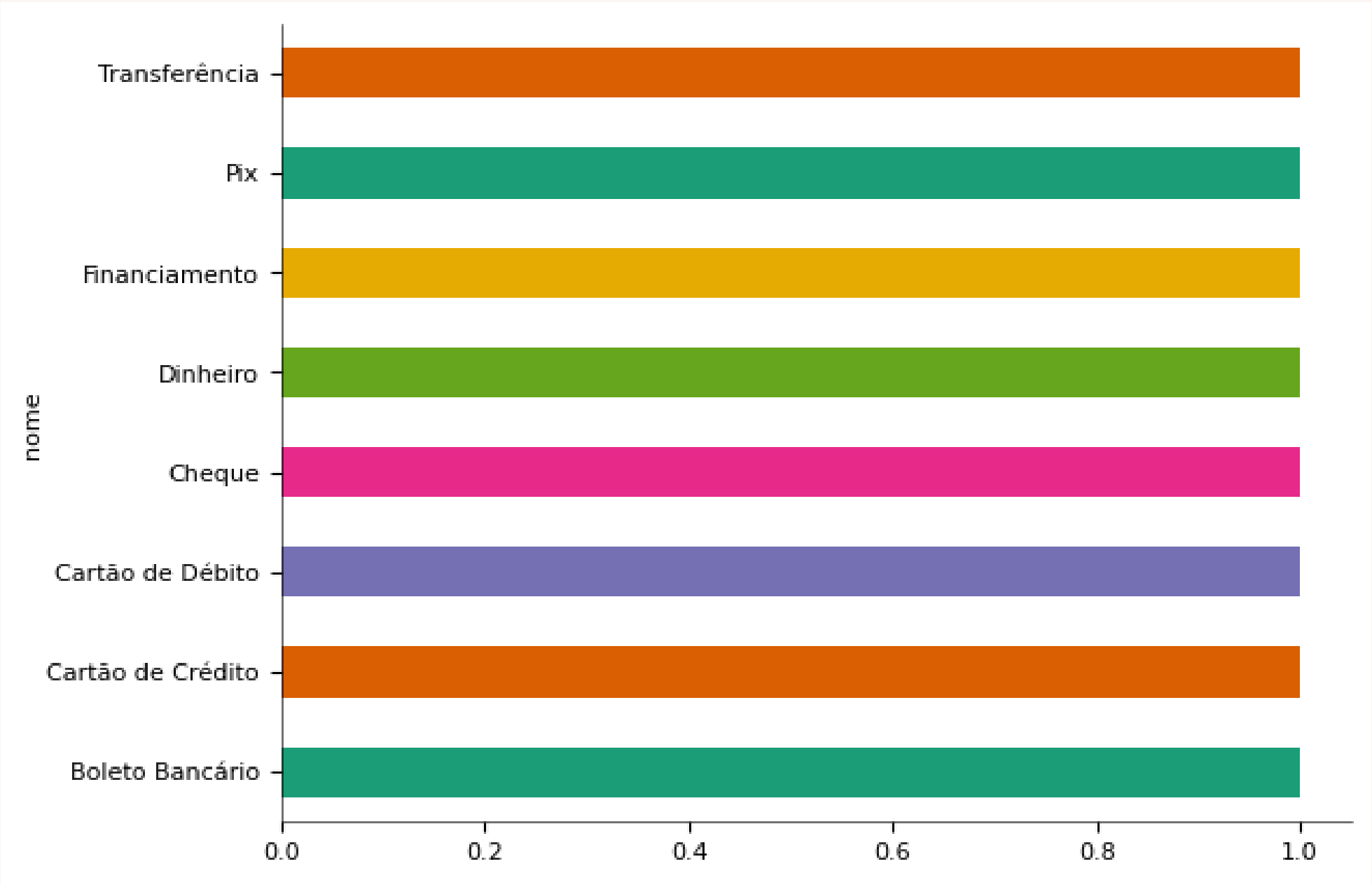


Gráfico ilustrativo de todos os nomes de transferências disponíveis e utilizados