# Rent-A-Tool

Relatório de Projeto no âmbito da Unidade Curricular de **Bases de Dados**


**Licenciatura em Engenharia Informática e Computação**

Ano letivo 2021/2022

**| Grupo 104 |**

David Peixoto

Mafalda Magalhães

Salvador Moreira

# Índice

# Introdução

O objetivo deste projeto proposto no âmbito da unidade curricular de Bases de Dados é a criação e interrogação de uma base de dados de uma empresa de aluguer de ferramentas, tendo em vista o auxílio na gestão, organização e otimização de processos.

Assim, dentro do possível, iremos criar e desenvolver uma base de dados com os elementos relevantes para guardar e processar a informação necessária à execução e prossecução da atividade da empresa.

# Contexto

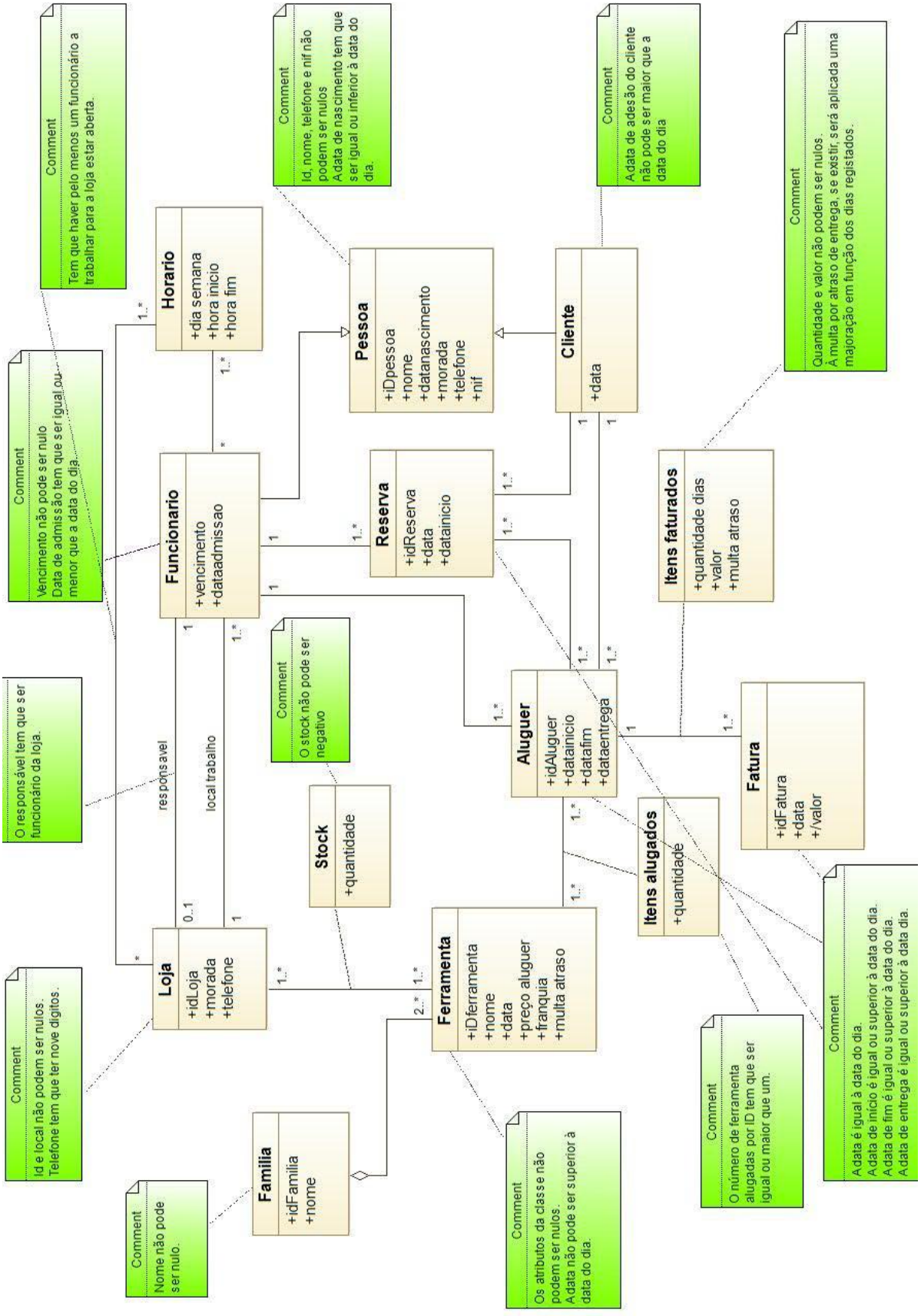
A empresa tem a cargo um sistema de aluguer, composto por uma rede de lojas, que servem os potenciais clientes do serviço.

Quando uma pessoa pretende usufruir do serviço, esta deve ser registada como cliente por um funcionário da loja em que se encontra. Posto isto, é permitido alugar uma ou mais ferramentas disponíveis em stock. Caso algum produto desejado pelo cliente não se encontre disponível no momento, este pode reservar a ferramenta para uma futura data disponível.

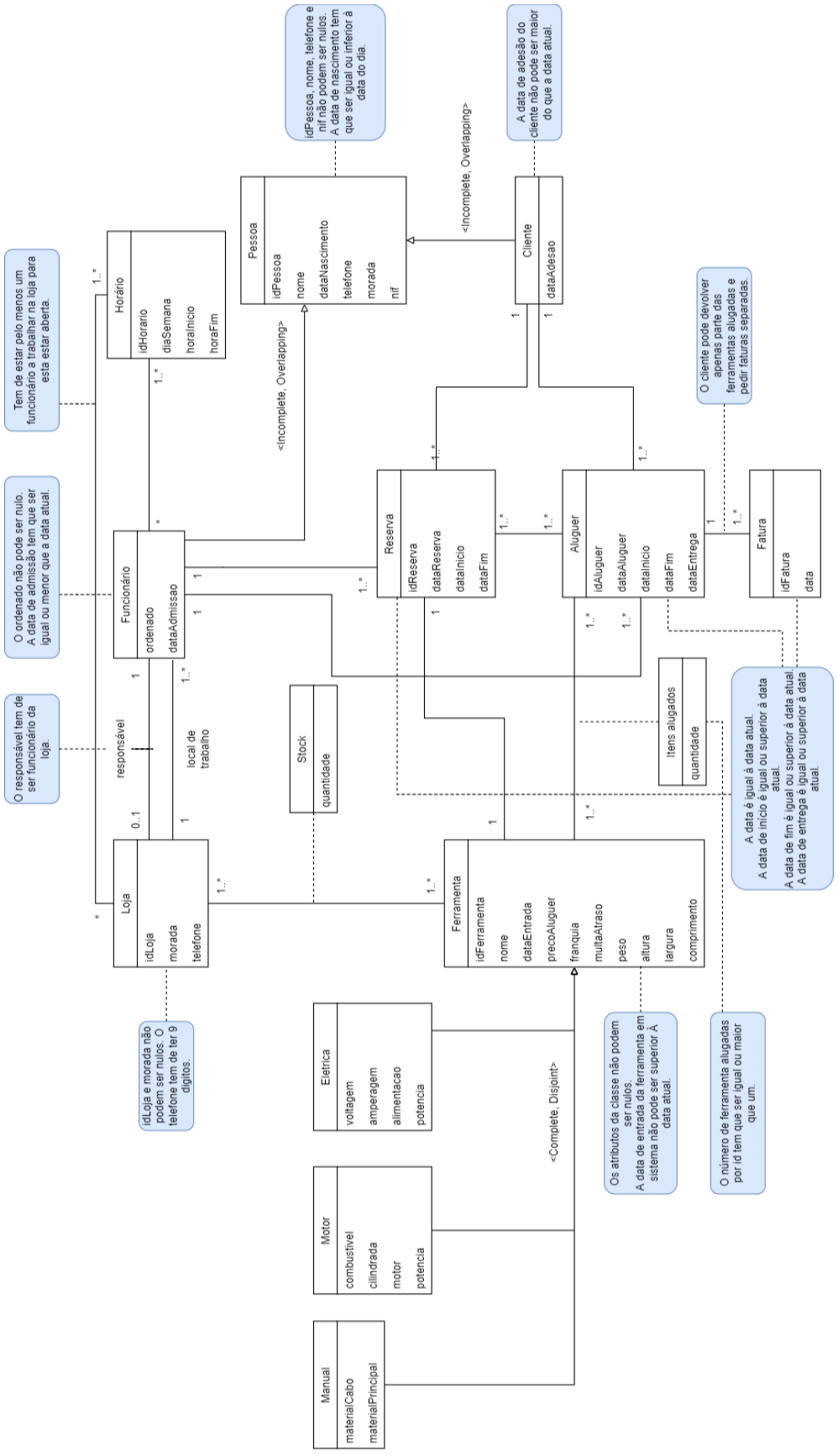
Sendo possível concretizar um pedido, o aluguer deve ser registado no sistema, assim como a data de início e de fim do mesmo. Neste caso, a informação de stock será atualizada, reduzindo, assim, o número de ferramentas disponíveis.

Após a devolução das ferramentas pelo cliente, o funcionário responsável deverá faturar o aluguer conforme a situação, é necessário verificar a data de entrega, confirmando, assim, se é cobrada uma sobretaxa de utilização pelo não cumprimento dos prazos.

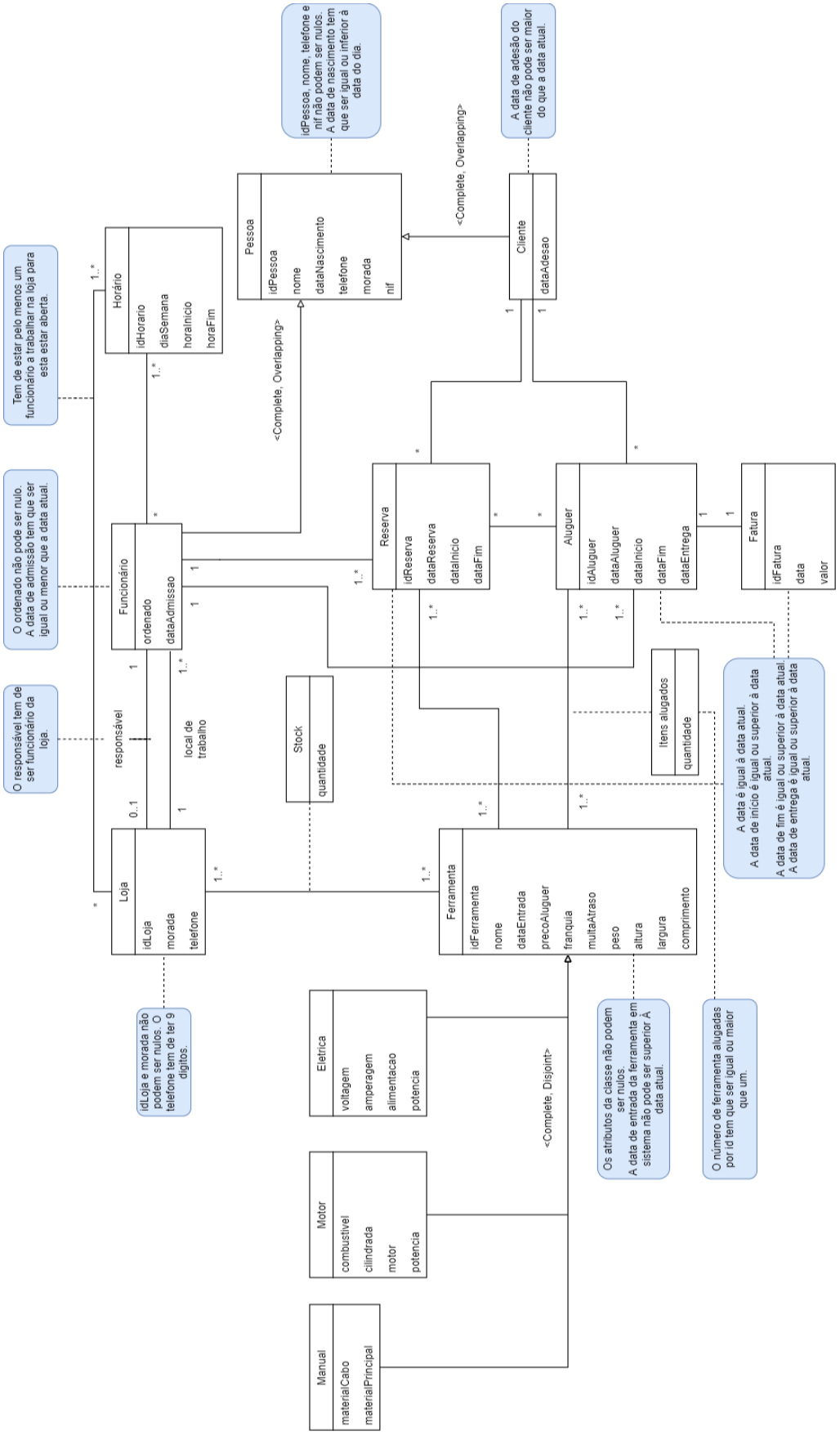
# Diagrama de Classes UML (1ª versão)



# Diagrama de Classes UML (2ª versão)



# Diagrama de Classes UML (3ª versão)



## Esquema relacional

- **Pessoa** (idPessoa, nome, dataNascimento, telefone, morada, nif)
- **Cliente** (idPessoa->Pessoa, dataAdesao)
- **Funcionario** (idPessoa->Pessoa, ordenado, dataAdmissao)
- **Horario** (idHorario, diaSemana, horaInicio, horaFim)
- **Loja** (idLoja, morada, telefone, idResponsavel->Funcionario)
- **Ferramenta** (idFerramenta, nome, dataEntrada, precoAluguer, franquia, multaAtraso, peso, altura, largura, comprimento)
- **Manual** (idFerramenta->Ferramenta, materialCabo, materialPrincipal)
- **Motor** (idFerramenta->Ferramenta, combustivel, cilindrada, motor, potencia)
- **Eletrica** (idFerramenta->Ferramenta, voltagem, amperagem, alimentacao, potencia)
- **Stock** (idLoja->Loja, idFerramenta->Ferramenta, quantidade)
- **Reserva** (idReserva, idCliente->Cliente, idFuncionario->Funcionario, idFerramenta->Ferramenta, dataReserva, dataInicio, dataFim)
- **LojaReserva** ( idLoja->Loja, idReserva->Reserva)
- **Aluguer** (idAluguer, idCliente->Cliente, idFuncionario->Funcionario, dataAluguer, dataInicio, dataFim, dataEntrega)
- **LojaAluguer** ( idLoja->Loja, idAluguer->Aluguer)
- **ItensAlugados** (idAluguer->Aluguer, idFerramenta->Ferramenta, quantidade, idFatura->Fatura)
- **Fatura** (idFatura, idAluguer->Aluguer, data, valor)
- **HorarioFuncionario** (idFuncionario-> Funcionario, idHorario->Horario)
- **HorarioLoja** (idLoja -> Loja, idHorario -> Horario)
- **LojaFuncionario** (idLoja -> Loja, idFuncionario-> Funcionario)
- **ReservaAluguer** (idReserva -> Reserva, idAluguer -> Aluguer)

# Análise de Dependências Funcionais e  Formas Normais

- **Pessoa** (idPessoa, nome, dataNascimento, telefone, morada, nif)
    - FDs:
        - idPessoa -> nome, dataNascimento, telefone, morada, nif
        - nif -> idPessoa, nome, dataNascimento, telefone, morada
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Cliente** (idPessoa->Pessoa, dataAdesao)
    - FDs:
        - idPessoa -> dataAdesao
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Funcionario** (idPessoa->Pessoa, ordenado, dataAdmissao)
    - FDs:
        - idPessoa -> ordenado, dataAdmissao
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Horario** (idHorario, diaSemana, horaInicio, horaFim)
    - FDs:
        - idHorario -> diaSemana, horaInicio, horaFim
        - diaSemana, horaInicio, horaFim -> idHorario
    - Formas
        - BCNF: sim
        - 3NF: sim

- **Loja** (idLoja, morada, telefone, idResponsavel->Funcionario)
    - FDs:
        - idLoja ->  morada, telefone, idResponsavel
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Ferramenta** (idFerramenta, nome, dataEntrada, precoAluger, franquia, multaAtraso, peso, altura, largura, comprimento)
    - FDs:
        - idFerramenta ->  nome, dataEntrada, precoAluger, franquia, multaAtraso, peso, altura, largura, comprimento
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Manual** (idFerramenta->Ferramenta, materialCabo, materialPrincipal)
    - FDs:
        - idFerramenta ->  materialCabo, materialPrincipal
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Motor** (idFerramenta->Ferramenta, combustivel, cilindrada, motor, potencia)
    - FDs:
        - idFerramenta ->  combustivel, cilindrada, motor, potencia
    - Formas
        - BCNF: sim
        - 3NF: sim

- **Eletrica** (idFerramenta->Ferramenta, voltagem, amperagem, alimentacao, potencia)
    - FDs:
        - idFerramenta -> voltagem, amperagem, alimentacao, potencia
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Stock** (idLoja->Loja, idFerramenta->Ferramenta, quantidade)
    - FDs:
        - idLoja, idFerramenta -> quantidade
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Reserva** (idReserva, idCliente->Cliente, idFuncionario->Funcionario, idFerramenta->Ferramenta, dataReserva, dataInicio, dataFim)
    - FDs:
        - idReserva -> idCliente, idFuncionario, idFerramenta, dataReserva, dataInicio, dataFim
        - idCliente, idFuncionario, idFerramenta, dataReserva, dataInicio, dataFim -> idReserva
    - Formas
        - BCNF: sim
        - 3NF: sim
- **LojaReserva** (idLoja-> Loja, idReserva->Reserva)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim

- **Aluguer** (idAluguer, idCliente->Cliente, idFuncionario->Funcionario, dataAluguer, dataInicio, dataFim, dataEntrega)
    - FDs:
        - idAluguer -> idCliente, idFuncionario, dataAluguer, dataInicio, dataFim, dataEntrega
        - idCliente, idFuncionario, dataAluguer, dataInicio, dataFim, dataEntrega -> idAluguer
    - Formas
        - BCNF: sim
        - 3NF: sim
- **LojaAluguer** (idLoja-> Loja, idAluguer->Aluguer)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim
- **ItensAlugados** (idAluguer->Aluguer, idFerramenta->Ferramenta, quantidade, idFatura->Fatura)
    - FDs:
        - idAluguer, idFerramenta -> quantidade, idFatura
    - Formas
        - BCNF: sim
        - 3NF: sim
- **Fatura** (idFatura, idAluguer->Aluguer, data, valor)
    - FDs:
        - idFatura -> idAluguer, data
    - Formas
        - BCNF: sim
        - 3NF: sim

- **HorarioFuncionario** (idFuncionario-> Funcionario, idHorario->Horario)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim
- **HorarioLoja** (idLoja -> Loja, idHorario -> Horario)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim
- **LojaFuncionario** (idLoja -> Loja, idFuncionario-> Funcionario)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim
- **ReservaAluguer** (idReserva -> Reserva, idAluguer -> Aluger)
    - FDs:
        - -
    - Formas
        - BCNF: sim
        - 3NF: sim

## Nota

- BCNF
  - A -> B não trivial
  - A é uma superkey/ key
- 3NF
  - A -> B não trivial
  - A é uma superkey/key

    OU

  - B tem apenas atributos primos (atributos que são membros de pelo menos uma chave da relação).
- Podemos assim concluir que em todas as relações e todas as FDs da nossa base de dados, o membro da esquerda de cada FD (correspondente a A) permite-nos descobrir todos os atributos da relação, implicando que A é (super)key.

# Restrições

- **Pessoa**
  - Não pode haver duas pessoas com o mesmo ID
    - idPessoa PRIMARY KEY
  - Todas as pessoas devem ter um id, um nome, data de nascimento,  telefone, morada e número de contribuinte
    - idPessoa NOT NULL
    - nome NOT NULL
    - dataNascimento NOT NULL
    - telefone NOT NULL
    - morada NOT NULL
    - nif NOT NULL
  - A data de nascimento tem de ser inferior ou igual à data atual
    - dataNascimento CHECK (dataNascimento <= CURRENT_DATE)
  - O nif tem de ter 9 dígitos
    - nif CHECK (nif >= 100000000 AND nif <= 999999999)
  - O telefone tem de ter 9 dígitos
    - telefone CHECK (telefone < 1000000000 AND telefone >= 100000000)
- **Cliente**
  - Não pode haver dois clientes com o mesmo ID e o ID deve corresponder a um ID de uma pessoa na tabela Pessoa
    - idPessoa PRIMARY KEY REFERENCES Pessoa (idPessoa)
  - Todos os clientes devem ter associados a eles a sua data de adesão
    - dataAdesao NOT NULL
  - A data de adesão tem de ser inferior ou igual à data atual
    - dataAdesao CHECK (dataAdesao <= CURRENT_DATE)
- **Funcionario**
  - Não pode haver dois funcionários com o mesmo ID e o ID deve corresponder a um ID de uma pessoa na tabela Pessoa
    - idPessoa PRIMARY KEY REFERENCES Pessoa (idPessoa)
  - Todos os funcionários devem ter um ordenado associado, bem como a data em que foram contratados
    - ordenado NOT NULL
    - dataAdmissao NOT NULL
  - O ordenado tem de ser maior ou igual a 705€
    - ordenado CHECK (ordenado >= 705)

- **Horario**
    - Não pode haver dois horários com o mesmo ID
        - idHorario PRIMARY KEY
    - Todos os horários necessitam de ter um id
        - idHorario NOT NULL
    - Deve existir apenas um horário para cada combinação diferente de dia da semana, hora de início e hora de fim
        - UNIQUE (diaSemana, horaInicio, horaFim)
    - Nenhum dos atributos pode ser nulo
        - diaSemana NOT NULL
        - horaInicio NOT NULL
        - horaFim NOT NULL
    - O dia da semana deve ser um dia da semana válido
        - diaSemana CHECK (diaSemana = 'SEGUNDA-FEIRA' OR diaSemana = 'TERCA-FEIRA' OR diaSemana = 'QUARTA-FEIRA' OR diaSemana = 'QUINTA-FEIRA' OR diaSemana = 'SEXTA-FEIRA' OR diaSemana = 'SABADO' OR diaSemana = 'DOMINGO')
    - A hora de fim tem de ser posterior à hora de início
        - horaFim CHECK (horaFim > horaInicio)
- **Loja**
    - Não pode haver duas lojas com o mesmo ID
        - idLoja PRIMARY KEY
    - Todas as lojas devem ter uma morada e um responsável atribuído
        - morada NOT NULL
        - idResponsavel NOT NULL
    - O ID do responsável da loja deverá corresponder ao ID de um funcionário
        - idResponsavel REFERENCES Funcionario(idPessoa)
    - O telefone tem de ter 9 dígitos
        - telefone CHECK (telefone < 1000000000 AND telefone >= 1000000000
- **Ferramenta**
    - Não pode haver duas ferramentas com o mesmo ID
        - idFerramenta PRIMARY KEY
    - Cada ferramenta deve ter um nome, uma data de entrada na loja, um preço de aluguer, uma franquia (caução), o valor da multa por atraso de entrega, peso, altura, largura e comprimento
        - nome NOT NULL
        - dataEntrada NOT NULL
        - precoAluguer NOT NULL
        - franquia NOT NULL
        - multaAtraso NOT NULL
        - peso NOT NULL

        - altura NOT NULL
        - largura NOT NULL
        - comprimento NOT NULL
    - A data de entrada tem que ser igual ou menor que a data atual
        - dataEntrada CHECK (dataEntrada <= CURRENT_DATE)
    - O preço de aluguer e a franquia têm que ser maiores que zero
        - precoAluguer CHECK (precoAluguer >= 0)
        - franquia CHECK (franquia >= 0)
    - A multa de atraso tem que ser maior ou igual a zero
        - multaAtraso CHECK (multaAtraso >= 0)
- **Manual**
    - Cada ferramenta manual deve ter identificado o material do cabo e o material principal
        - materialCabo NOT NULL
        - materialPrincipal NOT NULL
- **Motor**
    - Cada ferramenta com motor deve ter identificado o combustível, a cilindrada, o motor e a potência
        - combustivel NOT NULL
        - cilindrada NOT NULL
        - motor NOT NULL
        - potencia NOT NULL
- **Eletrica**
    - Cada ferramenta elétrica deve ter identificado a voltagem, a amperagem, a fonte de alimentação e a potência
        - voltagem NOT NULL
        - amperagem NOT NULL
        - alimentacao NOT NULL
        - potencia NOT NULL
- **Stock**
    - O ID da loja e da ferramenta devem corresponder a um ID de uma instância das tabelas Loja e Ferramenta
        - idLoja REFERENCES Loja (idLoja)
        - idFerramenta REFERENCES Ferramenta (idFerramenta)
    - Não deve haver duas instâncias com o mesmo par (idLoja, idFerramenta)
        - PRIMARY KEY (idLoja, idFerramenta)
    - A quantidade deve ser maior do que 0
        - quantidade CHECK (quantidade >= 0)
    - O stock necessita sempre de estar associado a uma loja e a uma ferramenta
        - idLoja NOT NULL
        - idFerramenta NOT NULL

- **Reserva**
  - Não pode haver duas reservas com o mesmo ID
    - idReserva PRIMARY KEY
  - Todos as reservas devem ter a data em que foram realizadas e a data de iníco do aluguer
    - dataReserva NOT NULL
    - dataInicio NOT NULL
    - dataFim NOT NULL
  - Todas as reservas necessitam de estar associadas a um funcionario, a uma ferramenta e a um cliente
    - idFuncionario NOT NULL
    - idFerramenta NOT NULL
    - idCliente NOT NULL
  - O ID de loja deve corresponder a um ID da tabela Loja, o ID do cliente deve corresponder a um ID da tabela Cliente, o ID do funcionário deve corresponder a um ID da tabela Funcionário e o ID da ferramenta deve corresponder a um ID da tabela Ferramenta
    - idLoja REFERENCES Loja (idLoja)
    - idCliente REFERENCES Cliente (idCliente)
    - idFuncionario REFERENCES Funcionario (idFuncionario)
    - idFerramenta REFERENCES Ferramenta (idFerramenta)
  - A data da reserva tem de ser inferior ou igual à data atua
    - dataReserva CHECK (dataReserva <= CURRENT_DATE)
  - A data de início tem de ser posterior ou igual à data da reserva
    - dataInicio CHECK ( dataInicio >= dataReserva)
  - A data de fim tem de ser igual ou superior à data de início
    - dataFim CHECK (dataFim >= dataInicio)
- **LojaReserva**
  - Não deve haver duas instâncias com o mesmo par (idLoja, idReserva)
    - PRIMARY KEY (idLoja, idReserva)
  - O ID de loja deve corresponder a um ID da tabela Loja e o ID da Reserva deve corresponder a um ID da tabela Reserva
    - idLoja REFERENCES Loja (idLoja)
    - idReserva REFERENCES Reserva (idReserva)
- **Aluguer**
  - Não pode haver dois alugueres com o mesmo ID
    - idAluguer PRIMARY KEY
  - Todos os alugueres devem ter uma data do aluguer, data de iníco e data de fim
    - dataAluguer NOT NULL
    - dataInicio NOT NULL
    - dataFim NOT NULL

    - Todos alugueres necessitam de estar associadas a um funcionario e a um cliente
        - idFuncionario NOT NULL
        - idCliente NOT NULL
    - O ID de loja deve corresponder a um ID da tabela Loja e o ID do cliente deve corresponder a um ID da tabela Cliente e o ID do funcionário deve corresponder a um ID da tabela Funcionário
        - idLoja REFERENCES Loja (idLoja)
        - idCliente REFERENCES Cliente (idCliente)
        - idFuncionario REFERENCES Funcionario (idFuncionario)
    - A data de aluguer tem de ser menor ou igual à data atual
        - dataAluguer CHECK (dataAluguer <= CURRENT_DATE)
    - A data de início tem de ser maior ou igual à data de aluguer
        - dataInicio CHECK (dataInicio >= dataAluguer)
    - A data de fim tem de ser superior à data de início
        - dataFim CHECK (dataFim >= dataInicio)
- **LojaAluguer**
    - Não deve haver duas instâncias com o mesmo par (idLoja, idReserva)
        - PRIMARY KEY (idLoja, idAluguer)
    - O ID de loja deve corresponder a um ID da tabela Loja e o ID do Aluguer deve corresponder a um ID da tabela Aluguer
        - idLoja REFERENCES Loja (idLoja)
        - idAluguer REFERENCES Aluguer (idAluguer)
- **ItensAlugados**
    - O ID do aluguer,  ferramenta  e fatura devem corresponder a um ID de uma instância das tabelas Aluguer, Ferramenta e Fatura
        - idAluguer REFERENCES Aluguer (idAluguer)
        - idFerramenta REFERENCES Ferramenta (idFerramenta)
        - idFatura REFERENCES Fatura (idFatura)
    - Todos os itens alugados necessitam de estar associadas a uma feramenta e a um aluguer
        - idFerramenta NOT NULL
        - idAluguerNOT NULL
    - Não deve haver duas instâncias com o mesmo par (idAluguer, idFerramenta)
        - PRIMARY KEY (idAluguer, idFerramenta)
    - A quantidade tem ser maior ou igual a 1
        - quantidade CHECK (quantidade >= 1)
        - o IdFatura será nulo enquanto o aluguer não for faturado
- **Fatura**
    - Não pode haver duas faturas com o mesmo ID
        - idFatura PRIMARY KEY

    - Todas faturas necessitam de estar associadas a um aluguer e ter um valor
        - idAluguer NOT NUL
        - valor NOT NULL
    - O ID do aluguer deve corresponder a um ID de uma instância da tabela Aluguer
        - idAluguer REFERENCES Aluguer (idAluguer)
    - A data tem de ser igual ou inferior à data atual
        - data CHECK ( data <= CURRENT_DATE)
- **HorarioFuncionario**
    - Não deve haver duas instâncias com o mesmo par (idFuncionario, idHorario)
        - PRIMARY KEY (idFuncionario, idHorario)
    - O ID de funcionário deve corresponder a um ID da tabela Funcionário e o ID do horáriodeve corresponder a um ID da tabela Horário
        - idFuncionario REFERENCES Funcionario (idFuncionario)
        - idHorario REFERENCES Horario (idHorario)
- **HorarioLoja**
    - Não deve haver duas instâncias com o mesmo par (idLoja, idHorario)
        - PRIMARY KEY (idLoja, idHorario)
    - O ID de loja deve corresponder a um ID da tabela Loja e o ID do horário deve corresponder a um ID da tabela Horário
        - idLoja REFERENCES Loja (idLoja)
        - idHorario REFERENCES Horário (idHorario)
- **LojaFuncionario**
    - Não deve haver duas instâncias com o mesmo par (idLoja, idFuncionario)
        - PRIMARY KEY (idLoja, idFuncionario)
    - O ID de funcionário deve corresponder a um ID da tabela Funcionário e o ID da loja deve corresponder a um ID da tabela Loja
        - idFuncionario REFERENCES Funcionario (idFuncionario)
        - idLoja REFERENCES Loja (idLoja)
- **ReservaAluguer**
    - Não deve haver duas instâncias com o mesmo par (idReserva, idAluguer)
        - PRIMARY KEY (idReserva, idAluguer)
    - O ID da reserva deve corresponder a um ID da tabela Reserva e o ID do aluguer deve corresponder a um ID da tabela Aluguer
        - idReserva REFERENCES Reserva (idReserva)
        - idAluguer REFERENCES Aluguer (idAluguer)

# Interrogações

1. Listagem dos alugueres em atraso;
2. Total de alugueres por Funcionário e por Loja;
3. Contagem de alugueres que chegaram mais cedo, no dia previsto de entrega ou atrasados;
4. Listagem das faturas emitidas em todas as lojas;
5. Cálculo do número total de faturas de novembro, do valor total e da média por fatura;
6. Loja com mais Alugueres;
7. Número de horas d etrabalho de cada Funcionário por semana;
8. Listagem de Clientes que já alugaram em todas as lojas;
9. Para cada par de pessoas, mostrar os alugueres em comum;
10. Cliente que gastou mais numa única compra.

## Gatilhos

1. Quando um aluguer é concluído (as ferramentas são devolvidas), o stock é atualizado e é emitada uma fatura com o valor a pagar;
2. Quando é feita uma reserva, é feito um aluguer derivado dessa mesma reserva, o stock é atualizado e, caso não haja stock disponível, é lançado um erro e a operação não é executada.
3. Quando é adicionado um funcionário ou quando alteramos os dados de um funcionário, verifica que este tem mais de 18 anos.