

Anexo:

Projeto de Planeamento e Implementação de Base de Dados para a Alugadora

1 Introdução	3
1.1 Contextualização	3
1.2 Apresentação do Caso de Estudo	3
1.3 Motivação	3
1.4 Objectivos	4
1.5 Estrutura do Relatório	4
2 Levantamento de Requisitos	5
2.1 Requisitos Funcionais	5
2.2 Requisitos Operacionais	5
3 Modelo Concetual	7
3.1 Identificação e caracterização de entidades	7
3.2 Identificação e caracterização de relacionamentos	7
3.3 Identificação de atributos de entidades	9
3.4 Identificação de chaves primárias	10
3.5 Verificação de redundância	11
3.6 Validação do Modelo Concetual com requisitos	11
3.7 Modelo Concetual	14
3.8 Revisão do Modelo Conceptual	14
4 Escolha do Motor de Base de Dados	15
5 Modelo Lógico	16
5.1 Derivação dos Relacionamentos do Modelo Concetual	16
5.1.1 Transformação de relacionamentos e entidades em tabelas	16
5.1.2 Transformação de atributos em tabelas	16
5.1.3 Escolha de chaves estrangeiras	17
5.2 Validação com Regras de Normalização	19
5.2.1 Identificação de Dependências Funcionais	19
5.2.2 Formas Normais	20

<u>5.3 Modelo Lógico</u>	21
<u>5.4 Crescimento Futuro</u>	21

1 Introdução

Nesta introdução ao projeto começa-se com uma contextualização, seguida da motivação para a realização do projeto e os seus principais objetivos. Numa parte final da introdução sintetiza-se a estrutura deste relatório.

1.1 Contextualização

Este projecto consiste no estudo e modelação de uma base de dados relacional, capaz de armazenar as informações do negócio de uma cadeia de lojas de Alugueres de Automóveis.

A decisão da criação desta base de dados, reside no facto da empresa Alugadora estar em expansão e, com isto, surge a necessidade de criar uma base de dados capaz de centralizar a informação das diferentes lojas. Neste momento, a empresa suporta a sua informação em documentos Excel, o que se revela inadequado para a expansão, uma vez que iremos lidar com uma quantidade bem maior de dados e de fontes dispersas. Para isso, tem de existir um sistema novo capaz de lidar com esses dados de forma mais eficiente.

Ora, através do sistema de base de dados implementado, será possível para a empresa ter os seus dados centralizado das diferentes lojas, de forma segura e eficiente. Portanto torna-se a solução ideal para o problema da empresa, possibilitando uma maior facilidade na expansão e, mais importante, uma alta fidelidade do sistema de dados da Alugadora.

1.2 Apresentação do Caso de Estudo

A Alugadora é uma empresa de Aluguer de Carros que está neste momento em expansão e tem neste momento todos os dados dos automóveis, clientes e funcionários em Excel. Tem várias lojas, no norte da América junto aos aeroportos e pretende uma expansão para o estrangeiro e dentro do país. Para isso, pretende digitalizar toda informação do negócio e criar infra-estruturas que permitam potenciar o negócio.

1.3 Motivação

Sem um sistema de dados centralizados que permita uma gestão conjunta dos dados e um sistema de suporte à decisão para auxiliar na expansão da cadeia, todas as suas lojas usam folhas de cálculo em Excel para armazenar os dados de aluguer. Com alguns anos de existência, acumulou-se um grande volume de dados relativamente a alugueres, que torna ainda mais difícil a tarefa de manuseamento dos dados no sistema actual e ainda pior a extracção de qualquer conhecimento a partir dos mesmos.

Posto isto, não é possível gerir com eficiência a localização das viaturas, nem prever com o mínimo rigor as necessidades de alugueres de cada loja, pelo que é necessário contacto constante entre elas para satisfazerem as suas necessidades, o que por sua vez, torna o processo confuso e susceptível a falhas de comunicação. Numa cadeia em expansão como a Alugadora, torna-se difícil tomar decisões que estão permanentemente dependentes de contacto com as restantes lojas, impossibilitando assim um crescimento sustentável.

1.4 Objectivos

Com este sistema, a agência pretende melhorar todo o processo interno do aluguer. A agência pretende ter toda a informação das várias lojas centralizada, tornando muito mais fácil o esclarecimento de questões sobre o estado do serviço.

O sistema de base de dados da Alugadora tem como objetivo armazenar informação sobre clientes, alugueres, pagamentos, viaturas, lojas e funcionários, assim como os relacionamentos entre cada um deles. O motor de base de dados permitirá depois extrair e relacionar as várias informações constantes na base de dados.

1.5 Estrutura do Relatório

No relatório é feita uma análise de requisitos. Seguidamente desenvolve-se o modelo conceptual e a identificam-se de entidades, atributos e relacionamentos entre estes.

Aborda-se depois o modelo lógico, acabando por descrever o processo da implementação da base de dados e tradução do esquema lógico para modelo físico.

2 Levantamento de Requisitos

Para se definir ao certo quais as funcionalidades esperadas no Data Warehouse, foi feita uma análise de requisitos que contempla os requisitos funcionais, requisitos operacionais, caracterização das três fontes de dados e personas relevantes para a utilização do sistema.

2.1 Requisitos Funcionais

De modo a ser criada uma base de dados capaz de guardar as informações das lojas de alugueres de automóveis teriam que ser guardados os dados relativos a cada loja, aos seus clientes, automóveis, funcionários, alugueres e respectivos pagamentos. Para tal, foram levantados os seguintes requisitos funcionais para expressar e relacionar a devida informação corretamente.

1. Deve ser registada a informação relativa às lojas da Alugadora, sendo estas caracterizadas pelo seu número, designação e pela cidade onde se encontram;
2. A base de dados deverá guardar toda informação relativa aos alugueres, sendo esta constituída por número de aluguer, data do aluguer, local de início (levantamento do carro), local de fim (entrega do carro), data de início e data de fim, valor base do custo de aluguer, valor do seguro e valor total. Cada aluguer estará associado a um carro, a um pagamento, a um funcionário, a uma loja e a um cliente;
3. Cada aluguer tem um pagamento associado composto por número de pagamento, data de pagamento, valor de pagamento e avaliação do serviço prestado;
4. Serão guardados os dados relativos aos carros que são compostos pela matrícula do carro, o modelo, a marca, data de compra, cor, número de lugares, cilindrada e tipo (jipe, carrinha). Cada carro poderá estar também associado a uma loja (correspondente à loja onde o carro actualmente se encontra) e a vários alugueres (correspondente aos respectivos alugueres desse carro);
5. A informação respectiva aos clientes será guardada tal como o seu número de cliente, nome, sexo, data de nascimento, número de telemóvel, cidade, país e email. A cada cliente estarão também associados vários alugueres correspondentes aos alugueres que esse cliente fez;
6. Os dados relativos aos funcionários também serão registados, mais concretamente: o seu número de funcionário, número de telemóvel, data de nascimento e nome. Cada funcionário será associado a uma loja (a loja onde trabalha) e a vários alugueres (alugueres em que foi o funcionário de serviço).

2.2 Requisitos Operacionais

Deve ser possível:

1. Consultar o volume de alugueres e de faturação de cada loja e total da cadeia, por dia, por semana, por mês, trimestralmente, semestralmente e anualmente;
2. Consultar quais as viaturas mais alugadas e as informações da viatura e preço de alugar de forma a identificar possíveis ajustes a outras viaturas;
3. Consultar o volume de alugueres dos funcionários para detectar falhas e sucessos na sua performance;
4. Identificar quais os picos volume de alugueres semanalmente, mensalmente e anualmente;
5. Consultar os clientes com maior volume de alugueres e de valor de alugueres;

6. Consultar os veículos por número de alugueres e margem de lucro, considerando valor de aluguer e custos associados;
7. Consultar os locais onde se efetuam maior requisição e retorno de veículos;
8. Consultar os valores de feedback dados pelo cliente sobre o aluguer para medir a qualidade de serviço.

3 Modelo Concetual

Apresentam-se agora as entidades, atributos e relacionamentos do modelo conceptual; seguidas da identificação de chaves primárias, verificação de redundância e validação do modelo com os requisitos. Por fim é feita uma revisão ao modelo.

O modelo completo encontra-se em anexo.

3.1 Identificação e caracterização de entidades

Analisando os requisitos necessários da base de dados de aluguer de automóveis, é possível identificar as seguintes entidades:

- **Carro** - Viatura da companhia de aluguer de automóveis.
- **Loja** - Loja física onde podem ser feitos alugueres.
- **Funcionário** - Funcionário que trabalha na loja física.
- **Cliente** - Pessoa que faz o aluguer de um carro.
- **Aluguer** - Pedido e registo da acção de aluguer e devolução do automóvel.
- **Pagamento** - Pagamento do aluguer do automóvel.

3.2 Identificação e caracterização de relacionamentos

Analisando os requisitos necessários da base de dados de aluguer de automóveis, é possível identificar os seguintes relacionamentos entre entidades:

Carro - Loja

- **Relacionamento:** Um carro existe sempre associado a uma loja; Loja possui a sua frota de carros;
- **Descrição:** Identifica os carros que são responsabilidade de uma determinada loja. Esta informação pode mudar se as viaturas forem deslocadas para outra loja;
- **Cardinalidade:** Carro (N), Loja (1) - Um carro é responsabilidade de apenas uma loja e uma loja pode estar responsável por vários carros;
- **Participação:** Carro (T), Loja (P) - Poderão existir lojas sem viaturas associadas, mas todos os carros estão associados a uma loja;
- **Obrigatoriedade:** Todos os carros são responsabilidade de uma loja, mas uma loja pode não estar responsável por carros;
- **Atributos:** Este relacionamento não possui atributos.

Carro - Aluguer

- **Relacionamento:** Um carro é alugado; Um aluguer envolve sempre um e só um carro;
- **Descrição:** Identifica o carro que está associado a determinado aluguer;
- **Cardinalidade:** Carro (1), Aluguer (N) - Um aluguer envolve apenas um carro, mas um carro pode estar envolvido em vários alugueres;
- **Participação:** Carro (P), Aluguer (T) - Todos os alugueres envolvem um carro, mas nem todos os carros estiveram envolvidos em alugueres;
- **Obrigatoriedade:** Um aluguer tem obrigatoriamente um e só um carro associado, mas um carro pode não estar associado a nenhum aluguer;

- **Atributos:** Este relacionamento não possui atributos.

Loja - Funcionario

- **Relacionamento:** Um funcionario trabalha numa loja; Uma loja emprega um funcionário;
- **Descrição:** Identifica um funcionário que trabalha numa loja;
- **Cardinalidade:** Loja (N), Funcionario (1) - Um funcionário é empregado numa loja, mas uma loja pode ter vários funcionários que lá trabalham;
- **Participação:** Funcionario (T), Loja (P) - Um funcionário existe empregado por uma loja, mas uma loja pode ainda não ter funcionários;
- **Obrigatoriedade:** Um funcionário tem obrigatoriamente de existir associado a uma loja, o relacionamento é opcional no sentido inverso;
- **Atributos:** Este relacionamento não possui atributos.

Aluguer - Loja

- **Relacionamento:** Um aluguer é feito numa loja; Uma loja tem vários alugueres que lá foram feitos;
- **Descrição:** Associa um aluguer à loja onde foi celebrado;
- **Cardinalidade:** Aluguer (N), Loja (1) - Um aluguer é feito numa loja e numa loja podem ser (e ter sido) realizados vários alugueres;
- **Participação:** Aluguer (T), Loja (P) - Uma loja pode não ter celebrado alugueres, mas todos os alugueres estão associados à loja onde foram feitos;
- **Obrigatoriedade:** Um aluguer está obrigatoriamente associado a uma loja, enquanto que uma loja poderá não ter alugueres associados;
- **Atributos:** Este relacionamento não possui atributos.

Aluguer - Cliente

- **Relacionamento:** Um cliente faz um aluguer; Um aluguer é feito por um cliente;
- **Descrição:** Um aluguer está sempre associado a um cliente pois é o cliente que realiza a ação de alugar;
- **Cardinalidade:** Aluguer (N), Cliente (1) - Um cliente pode realizar vários alugueres, mas um aluguer está associado a apenas um cliente;
- **Participação:** Aluguer (T), Cliente (P) - Todos os alugueres estão associados a um cliente, mas um cliente pode não ter realizado qualquer aluguer;
- **Obrigatoriedade:** Todos os alugueres têm um cliente associado, mas um cliente pode não ter alugueres;
- **Atributos:** Este relacionamento não possui atributos.

Aluguer - Pagamento

- **Relacionamento:** Um aluguer tem um pagamento e um pagamento está associado a um aluguer;
- **Descrição:** Associa o aluguer ao seu pagamento;
- **Cardinalidade:** Aluguer (1), Pagamento (1) - Um aluguer tem um pagamento e um pagamento tem um aluguer;
- **Participação:** Aluguer (P), Pagamento (T) - Todos os pagamentos dizem respeito a um aluguer, mas um aluguer pode não ter ainda um pagamento associado;
- **Obrigatoriedade:** Um pagamento pode ainda não ter um pagamento associado, mas um pagamento tem obrigatoriamente de estar associado a um aluguer;
- **Atributos:** Este relacionamento não possui atributos.

Aluguer - Funcionario

- **Relacionamento:** Funcionário responsável por um aluguer; Aluguer é tratado por um funcionário;
- **Descrição:** Associa um funcionário ao aluguer pelo qual está responsável;
- **Cardinalidade:** Aluguer (N), Funcionario (1) - Um funcionário pode ser responsável por vários alugueres enquanto que um aluguer pode apenas ter um funcionário associado;
- **Participação:** Aluguer (T), Funcionario (P) - Todos os alugueres têm um funcionário associado; Um funcionário pode ainda não estar responsável por nenhum aluguer;
- **Obrigatoriedade:** Todos os alugueres têm obrigatoriamente um funcionário associado, a relação é opcional no sentido inverso;
- **Atributos:** Este relacionamento não possui atributos.

3.3 Identificação de atributos de entidades

Analisando os requisitos necessários da base de dados de aluguer de automóveis, é possível identificar as seguintes atributos para as entidades previamente identificadas.

Loja

- **nr** - chave primária da entidade Loja;
- **designação** - nome da loja;
- **cidade** - nome da cidade da loja;
- **pais** - nome do país da loja.

Funcionário

- **nr** - chave primária da entidade Loja;
- **nome** - nome do funcionário;
- **nascimento** - data de nascimento do funcionário;
- **sexo** - género do funcionário.
- **telemovel** - número de telemóvel do funcionário

Carro

- **matricula** - chave primária da entidade Carro, corresponde à sua matricula;
- **marca** - marca do fabricante do carro;
- **cor** - cor do carro;
- **modelo** - modelo do carro;
- **nr_lugares** - número de lugares para passageiros do carro;
- **cilindrada** - cilindrada do motor do carro;
- **tipo** - tipo de carro.

Cliente

- **nr** - chave primária da entidade Cliente;
- **email** - endereço de email do cliente;
- **nascimento** - data de nascimento do cliente;
- **sexo** - género do cliente;
- **nome** - nome do cliente;
- **telemovel** - telemovel do cliente;
- **cidade** - cidade de residência do cliente.
- **pais** - país de residência do cliente

Aluguer

- **nr** - chave primária da entidade Aluguer;
- **data** - data de aluguer;
- **periodo** - atributo composto definido por:
 - **data inicio** - data de início do aluguer do carro;
 - **data fim** - data de término do aluguer do carro;
- **locais** - atributo composto definido por:
 - **local inicio** - local de requisição da viatura;
 - **local fim** - local de entrega da viatura;
- **custos**: atributo composto definido por
 - **valor_base** - valor base de aluguer;
 - **valor_seguro** - valor de seguro do aluguer;
 - **valor_total** - atributo derivado da soma do valor_base e do valor_seguro;

Pagamento

- **nr** - chave primária da entidade Pagamento;
- **data** - data de realização do pagamento;
- **valor** - valor do pagamento realizado;
- **avaliação** - valor de feedback dado pela qualidade de serviço

3.4 Identificação de chaves primárias

Para as entidades definidas, de forma a garantir a restrição de integridade de cada uma, é necessário existir um identificador único a cada entidade. Para tal, decidiu-se que para cada uma destas seria utilizado um atributo único, como a matrícula no caso dos carros ou o “id” (identificador) do cliente, no caso dos clientes. Todas as restantes chaves foram classificadas como secundárias.

Na seguinte lista, podemos observar para cada uma das entidades qual foi a nossa escolha para sua chave primária. Outras chaves poderiam ser candidatas em algumas das entidades, no entanto achamos que as escolhidas são as que melhor garantem a integridade de cada entidade.

Entidade	Atributo
Carro	matricula
Aluguer	aluguer_id
Cliente	cliente_id
Pagamento	aluguer_id
Loja	loja_id
Funcionario	funcionario_id
País	pais_id
Cidade	cidade_id
Marca	marca_id

Modelo	modelo_id
--------	-----------

3.5 Verificação de redundância

No modelo concetual encontram-se alguns exemplos de redundância:

As entidades Loja e Cliente têm ambas os atributos Cidade e País. A cidade e país poderiam ser convertidos em duas entidades, cada uma com um atributo “nome” e criarem-se três novos relacionamentos:

- País (1) - (N) Cidade;
- Loja (N) - (1) Cidade;
- Cliente (N) - (1) Cidade.

Esta abordagem também seria válida, mas não foi escolhida porque a cidade e o país não foram considerados conceitos suficientemente importantes no contexto da empresa para que se possam considerar entidades.

Na criação do modelo lógico esta redundância foi removida através da criação de tabelas para Cidade e País e derivando relações como as descritas acima, evitando assim problemas de coerência da informação. A razão para isto é manter o modelo concetual focado no que é importante para o cumprimento dos requisitos, tratando depois deste tipo de ajustes na passagem para o modelo lógico. Ainda relativamente à Cidade e País, os atributos simples “local início” e “local fim”, que compõem o atributo complexo “locais” da entidade Aluguer, representam localizações no modelo concetual, mas na transformação para o modelo lógico foram substituídos por um relacionamento com a tabela “Cidade” para evitar redundância e melhorar a coerência da informação.

O atributo “valor total”, parte do atributo composto “custos” da entidade Aluguer, é um atributo derivado precisamente por ser obtido usando os valores presentes nos outros dois atributos simples do atributo composto “custos”. Manter o atributo “valor total” como atributo simples seria redundante.

3.6 Validação do Modelo Concetual com requisitos

1. Deve ser registada a informação relativa às lojas da Alugadora, sendo estas caracterizadas pelo seu número, designação e pela cidade onde se encontram.



Ilustração: Entidade Loja

2. A base de dados deverá guardar toda informação relativa aos alugueres, sendo está constituída por número de aluguer, data do aluguer, local de início (levantamento do carro), local de fim (entrega do carro), data de início e data de fim, valor base do custo de aluguer, valor do seguro e valor total. Cada aluguer estará associado a um carro, a um pagamento, a um funcionário, a uma loja e a um cliente.

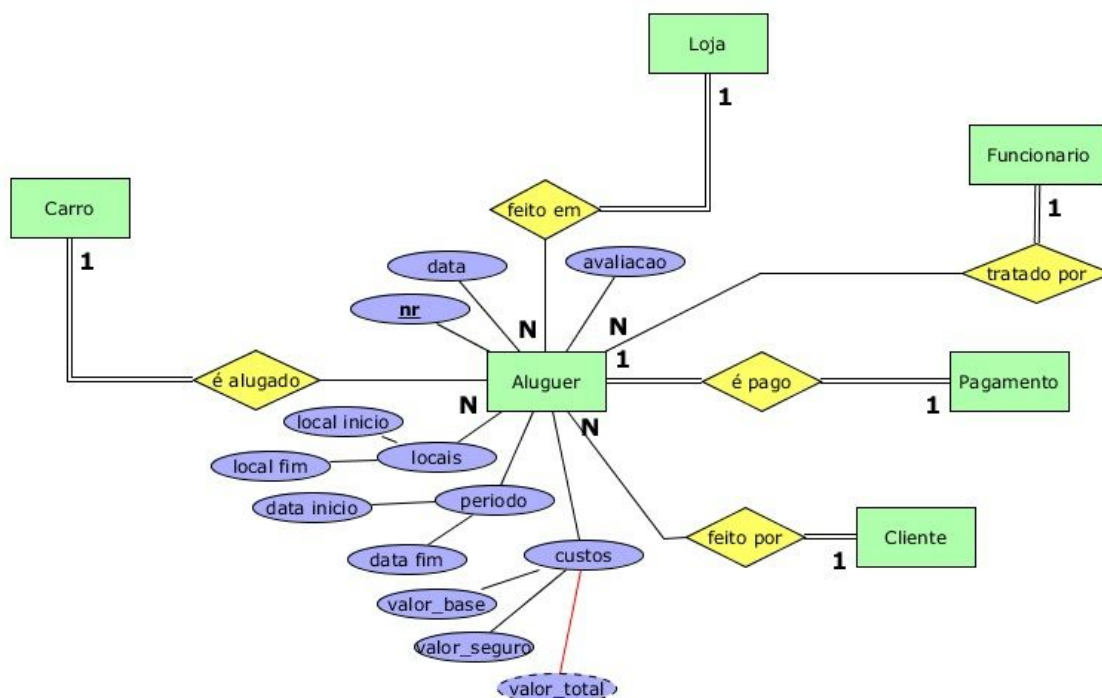


Ilustração: Entidade Aluguer

3. Cada aluguer tem um pagamento associado composto por número de pagamento, data de pagamento, valor de pagamento e avaliação do serviço prestado.

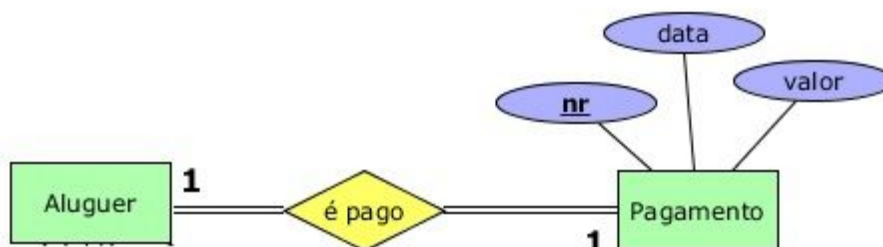


Ilustração: Relacionamento Aluguer-Pagamento

4. Serão guardados os dados relativos aos carros que são compostos pela matrícula do carro, o modelo, a marca, data de compra, cor, número de lugares, cilindrada e tipo (jipe, carrinha). Cada carro poderá estar também associado a uma loja (correspondente à loja onde o carro actualmente se encontra) e a vários alugueres (correspondentes aos respectivos alugueres desse carro).

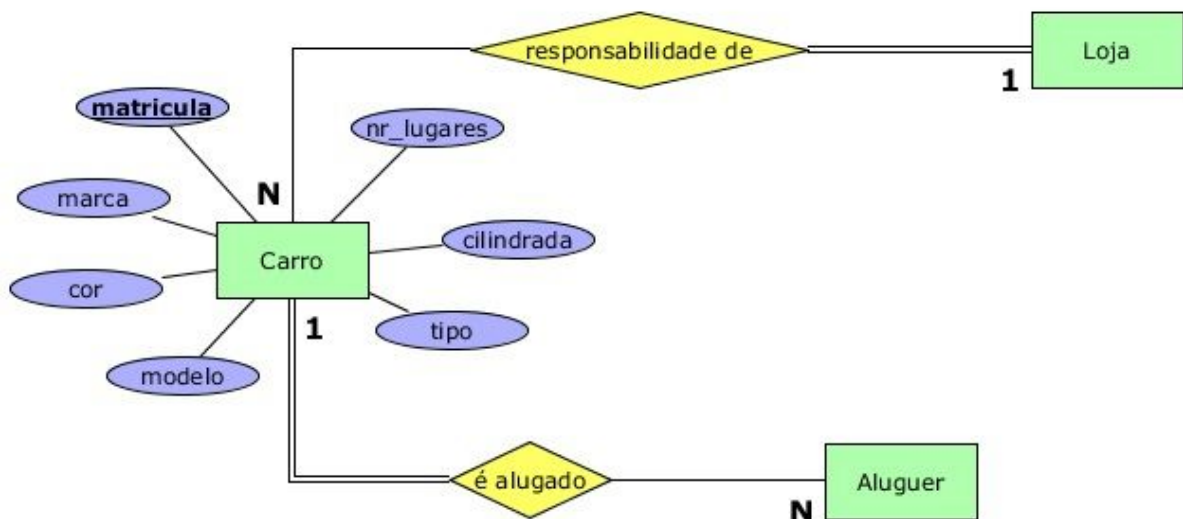


Ilustração: Entidade Carro

5. A informação respectiva aos clientes será guardada tal como o seu número de cliente, nome, sexo, data de nascimento, número de telemóvel, cidade, país e email. A cada cliente estarão também associados vários alugueres correspondentes aos alugueres que esse cliente fez.

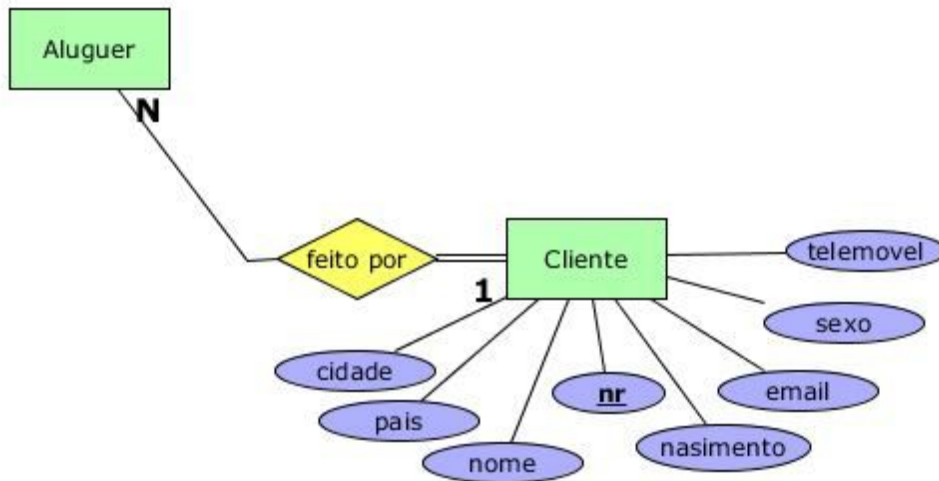


Ilustração: Entidade Cliente

6. Os dados relativos aos funcionários também serão registados, mais concretamente: o seu número de funcionário, número de telemóvel, data de nascimento e nome. Cada funcionário será associado a uma loja (a loja onde trabalha) e a vários alugueres (alugueres em que foi o funcionário de serviço).

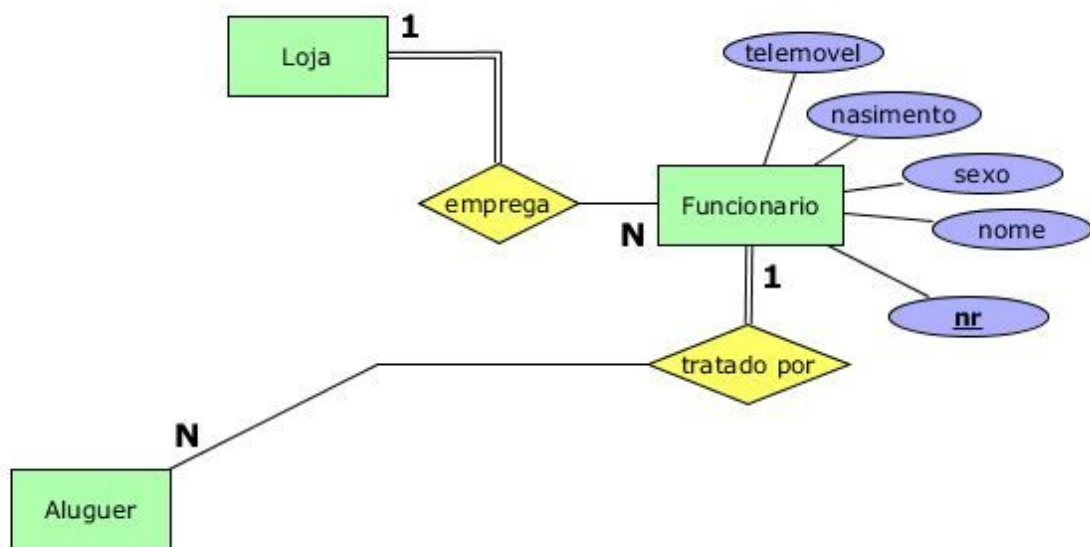


Ilustração: Entidade Funcionário

3.7 Modelo Concetual

O modelo concetual está presente no Anexo 1 deste relatório.

3.8 Revisão do Modelo Conceptual

4 Escolha do Motor de Base de Dados

O Sistema de Gestão de Base de Dados usado é MariaDB. Este SGBD é de obtenção gratuita e oferece garantias de segurança, coerência dos dados e tem compatibilidade com a maioria das ferramentas que operam sobre bases de dados. É também um sistema transacional.

5 Modelo Lógico

Segue-se agora uma descrição das derivações e transformações necessárias para chegar do modelo conceitual ao modelo lógico. Após essa descrição segue-se o modelo lógico final e validações ao modelo conseguido.

5.1 Derivação dos Relacionamentos do Modelo Conceitual

A derivação dos relacionamentos do modelo conceitual para o modelo lógico dividem-se em três principais grupos que serão agora abordados: a transformação de entidades e relacionamentos (N-M) em tabelas, a transformação de atributos em tabelas e a escolha das chaves estrangeiras.

5.1.1 Transformação de relacionamentos e entidades em tabelas

Analisando o modelo conceitual, convertemos todas as entidades em tabelas, sendo que obtivemos 6 tabelas, como descritas abaixo.

De referir que a notação usada implica o sublinhamento das chaves primárias e as chaves estrangeiras são identificadas textualmente.

Loja = <loja_id, designacao, cidade_id>

Chaves estrangeiras: cidade_id (refere cidade.cidade_id)

Carro = <matricula, cor, modelo_id, lugares, cilindrada, loja_id, tipo>

Chaves estrangeiras: modelo_id (refere modelo.modelo_id), loja_id (refere loja.loja_id)

Cliente = <cliente_id, email, nascimento, sexo, nome, cidade_id, telemovel>

Chaves estrangeiras: cidade_id (refere cidade.cidade_id)

Aluguer = <nr, data, data_inicio, data_fim, cidade_inicio, cidade_fim, valor_base, valor_seguro, cliente_id, funcionario_id, loja_id, matricula>

Chaves estrangeiras: cliente_id (refere cliente.cliente_id), funcionario_id (refere funcionario.funcionario_id), loja_id (refere loja.loja_id), matricula (refere carro.matricula), cidade_inicio e cidade_fim (ambas referem cidade.cidade_id)

Pagamento = <aluguer_id, data, valor, avaliacao>

Chaves estrangeiras: aluguer_id (refere aluguer.aluguer_id)

Funcionario = <funcionario_id, nome, sexo, nascimento, telemovel>

Chaves estrangeiras: Nenhuma

5.1.2 Transformação de atributos em tabelas

Analisando as dependências funcionais dos atributos em relação às entidades a que estavam associados no modelo conceitual, chegou-se à conclusão que alguns atributos não deviam ficar na tabela criada a partir da entidade à qual estavam associados.

Tal como anteriormente, na notação usada estão sublinhadas as chaves primárias e as chaves estrangeiras são identificadas textualmente.

Modelo = <modelo_id, nome, marca_id>

Chaves estrangeiras: marca_id (refere marca.marca_id)

Marca = <marca_id, nome>

Chaves estrangeiras: Nenhuma

Cidade = <cidade_id, nome, pais_id>

Chaves estrangeiras: pais_id

Pais = <pais_id, nome>

Chaves estrangeiras: Nenhuma

5.1.3 Escolha de chaves estrangeiras

Uma chave estrangeira é um atributo, que aponta para a chave primária de outra tabela ou da mesma tabela. Ou seja, passa a existir uma relação entre entidades de duas tabelas ou de uma apenas. A finalidade da chave estrangeira é garantir a integridade dos dados referenciais. Podemos observar na seguinte secção, as razões e explicação para a escolha das chaves estrangeiras das tabelas do sistema.

- Carro:

Marca - Relação de 1 para N com Marca. Como um Carro tem uma Marca e essa Marca pode aparecer em mais do que um Carro, o sentido da maior cardinalidade está do lado do Carro. Ou seja, a mesma Marca pode constar em mais do que uma entrada na tabela de Carro e como tal, esta última entidade terá um atributo com o identificador único da Marca.

Loja - Relação de N para 1, sendo que Carro é a tabela de maior cardinalidade. O que significa que é o Carro que terá o identificador da Loja a que pertence.

Modelo - O modelo segue precisamente a mesma lógica que a Marca, com uma relação de 1 para N de cardinalidade maior do lado do Carro, pois um Carro tem um e só um Modelo mas esse Modelo pode ser igual a mais do que um Carro. O identificador do Modelo tem de ser um dos atributos da tabela Carro.

- Aluguer:

Loja - A relação existente entre Aluguer e Loja é de N para 1, sendo que a maior cardinalidade fica no lado do Aluguer, pois como uma loja faz mais que um aluguer tem de ter a cardinalidade maior do lado do Aluguer. Significando que cada aluguer tem de ter a identificação da loja em que foi feito.

Cliente - A relação entre Aluguer e o Cliente é de N para 1, novamente ficando a maior cardinalidade do lado do Aluguer, como na relação anterior. O identificador do cliente fica como atributo na tabela de Aluguer.

Pagamento - Nesta relação existe uma relação de 1 para N, com maior cardinalidade do lado do Pagamento. Como o pagamento é único só existe um para cada Aluguer, na tabela de Pagamento fica um atributo com o identificador do Aluguer que corresponde.

Carro - Este caso é semelhante à relação com o Cliente e Loja. Um carro pode ser alugado várias vezes e portanto é uma relação de N para 1, sendo que a maior cardinalidade fica do lado do Aluguer. Esta tabela tem então de ter um identificador do Carro utilizado no Aluguer.

Cidade - Neste caso temos duas relações de 1 para N com a entidade Cidade. A maior cardinalidade está do lado do Aluguer, pois num aluguer existe uma cidade de início e outra de fim, como tal

correspondem a uma cidade na entidade Cidade e cada uma tem de ser identificada com um identificador na tabela Aluguer.

Funcionario - Existe uma relação de 1 para N com a entidade Funcionario, em que a maior cardinalidade está do lado do Aluguer. Como cada Aluguer existe um funcionário que o registou, então faz sentido haver um identificador desse funcionário na tabela Aluguer e como esse funcionário pode registar mais do que um Aluguer daí advém a cardinalidade maior do lado do Aluguer.

- **Cliente:**

Aluguer - A relação do Cliente com o Aluguer é de 1 para N, com a maior cardinalidade do lado do Aluguer. Como vimos anteriormente, um Cliente pode fazer vários Aluguers e portanto tem de ser identificado na tabela de Aluguers para que haja correspondência o Cliente.

Cidade - Existe uma relação de N para 1 com a entidade Cidade, com a maior cardinalidade do lado do Cliente. Para identificar a cidade a que o cliente pertence temos de ter um identificador da cidade na tabela do Cliente. Desta forma, podemos identificar os clientes de uma determinada cidade.

- **Pagamento:**

Aluguer - A relação do Pagamento com o Aluguer é uma relação de 1 para 1. Como existe apenas um pagamento por cada Aluguer, então na tabela de Pagamento tem de existir um atributo que relacione e identifique o Aluguer que lhe corresponde.

- **Loja:**

Aluguer - A relação da Loja com o Aluguer é de 1 para N com a cardinalidade de maior grau do Aluguer. Como uma loja faz vários Aluguers, então faz sentido na tabela do Aluguer existir um identificador único da Loja correspondente.

Carro - Seguindo a mesma lógica da relação anterior, a relação entre estas duas entidades é de 1 para N com a cardinalidade de maior grau do lado do Carro e assim cada carro tem de ter um atributo que identifique e relacione a loja a que pertence.

Cidade - A relação entre Loja e Cidade é de N para 1, com maior cardinalidade do lado da Loja. Para que a mesma cidade possa identificar várias lojas temos de ter um identificador único da cidade na tabela Loja.

- **Funcionario:**

Aluguer - A relação entre o Funcionario e o Aluguer é uma relação de 1 para N, respectivamente. Um Funcionario pode fazer muitos alugueres e dessa forma, devemos então ter um identificador na tabela de Aluguer que identifique o Funcionario que tratou do Aluguer.

Loja - Na relação com a entidade Loja, temos uma relação de 1 para N em que a maior cardinalidade está do lado do Funcionario, pois como numa loja há mais do que um funcionário e cada funcionário apenas trabalha numa loja, temos de ter um atributo identificador da Loja na tabela de Funcionario.

- **Cidade:**

País - A relação é de N para 1, com a cardinalidade maior do lado da Cidade. Para identificarmos o país a que pertence a cidade, temos então de na tabela Cidade adicionar um identificador que relacione e identifique o País a que pertence.

Cliente - A relação é de 1 para N, com maior cardinalidade do lado Cliente para que seja possível haver vários clientes identificáveis pela mesma cidade. Para tal, deve haver um identificador da Cidade na tabela Cliente.

Loja - Da mesma forma que a Cidade se relaciona com a entidade Cliente, com a Loja temos uma relação igual de 1 para N, em que na tabela Loja temos um identificador único da cidade a que pertence.

Aluguer - Como vimos o Aluguer tem associadas duas cidades, uma de partida e outra de chegada. Assim, temos duas relações de 1 para N com a entidade Aluguer, nas quais a maior cardinalidade está do lado do Aluguer, portanto para identificar as cidades temos de ter um identificador para cada uma na tabela Aluguer.

- **Pais:**

Cidade - A relação entre o Pais e a entidade Cidade é de 1 para N, em que a maior cardinalidade está do lado da Cidade, pois temos várias cidades associadas a um país. Assim sendo, na tabela da cidade tem de existir um identificador do Pais a que corresponde.

- **Modelo:**

Carro - A relação é de 1 para N com a entidade Carro, em que a maior cardinalidade está do lado do Carro, pois como existem vários carros com o mesmo modelo, faz sentido haver um identificador do modelo na tabela Carro.

- **Marca:**

Carro - A relação entre a Marca e o Carro é de 1 para N, em que a maior cardinalidade é do lado do Carro. Da mesma forma que existem vários carros com o mesmo modelo, existem vários carros da mesma Marca e analogamente, temos que identificar a marca com um identificador na tabela Carro.

5.2 Validação com Regras de Normalização

Nesta secção são identificadas as dependências funcionais e são depois usadas para auxiliar a validação do modelo com as regras de normalização.

5.2.1 Identificação de Dependências Funcionais

Para permitir uma correta validação através das regras de normalização, é necessário identificar corretamente as dependências. Em seguida, será abordada a validação da escolha das chaves primárias através das dependências funcionais.

Todos os atributos de uma tabela são determinados funcionalmente pela sua chave primária. Assim, serão omitidas essas dependências triviais, mas a título de exemplo representaremos algumas dessas dependências de cada tabela.

- **Aluguer:**

aluguer_id -> data
data -> aluguer_id
(...)

- **Cliente:**

cliente_id -> email
cliente_id -> nome
(...)

- **Loja:**
loja_id -> cidade
loja_id -> designacao
- **Carro:**
matricula -> marca
matricula -> modelo
(...)
- **Pagamento:**
aluguer_id -> data
aluguer_id -> valor
- **Funcionario:**
funcionario_id -> nome
funcionario_id -> sexo
(...)
- **Pais:**
pais_id -> nome
- **Marca:**
marca_id -> nome
- **Modelo:**
modelo_id -> nome
- **Cidade:**
cidade_id -> nome

Cada um destes determinantes, dado que determina funcionalmente a chave primária, por transitividade, também determina todos os restantes atributos.

Tendo identificado as dependências funcionais, estas permitem validar corretamente a escolha das chaves primárias. Supondo que um atributo A, determina funcionalmente atributos B e C e o atributo B determina funcionalmente C, vemo-nos confrontados com duas chaves candidatas. Uma avaliação correta das dependências funcionais ajuda a mostrar que A é a chave primária, uma vez que todos os outros atributos dependem funcionalmente de A.

5.2.2 Formas Normais

O modelo lógico encontra-se normalizado nas três formas normais.

- 1ª Forma Normal

A 1ª Forma Normal consiste na identificação e remoção de tipos repetidos. Analisando todas as tabelas e pondo de parte atributos que poderão causar repetição esporádica, é possível que haja, por exemplo, repetição no atributo marca ou modelo presente na tabela Carro, assim faz sentido existir uma tabela para as marcas e outra para os modelos, tal como temos no nosso sistema. Outro aspecto importante e requer uma normalização é a questão das cidade e moradas. Pelo mesmo motivo das marcas e modelos estas são também representados numa tabela própria.

- 2ª Forma Normal

Uma tabela está na Segunda Forma Normal 2FN se ela estiver na 1FN e se todos os atributos forem totalmente dependentes da chave primária, assim, procura eliminar dependências parciais.

Se o nome do funcionário já existe na tabela Funcionario, então não é necessário que exista na tabela de Aluguer um atributo com o nome dele. Esta é mais uma anomalia que a segunda forma normal trata e evita que valores fiquem em redundância na base de dados. Posto isto, podemos afirmar que o nosso sistema obedece a estas normas e encontra-se validado pela 2ª Forma Normal.

- **3ª Forma Normal**

A 3ª Forma Normal procura eliminar as dependências transitivas. Uma tabela está na 3ª Forma Normal se ela estiver na 2ª Forma Normal e se nenhum atributo depender de outro que não seja a chave primária. Na terceira forma normal temos de eliminar aqueles campos que podem ser obtidos pela equação de outros campos da mesma tabela. Analisando as dependências funcionais, é possível concluir que nenhuma é transitiva no nosso sistema, estando assim validada de acordo com a 3ª Forma Normal.

5.3 Modelo Lógico

O modelo concetual está presente no Anexo 2 deste relatório.

5.4 Crescimento Futuro

Segundo a Alugadora, no ultimo ano foram realizados 400 alugueres. Assumindo que 80% desses alugueres são feitos por novos clientes e que são pagos numa única vez, chega-se a uma previsão de crescimento de 400 registos de alugueres, 400 registos de pagamentos e 320 registos de clientes por ano. Sabendo o espaço que cada um destes registos ocupa no sistema operacional é possível calcular a taxa aproximada de crescimento da base de dados.

