



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

Rua Segura-Escola de Condução

André Fernandes A87938

Francisca Baptista A87967

José Machado A87979

Rui Faria A87957

dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Rua Segura-Escola de Condução

André Fernandes A87938,
Francisca Baptista A87967
José Machado A87979
Rui Faria A87957

dezembro,2020

Resumo

Neste trabalho, pretendemos criar uma base de dados para uma escola de condução.

Numa primeira fase identificamos o problema e o contexto em que este se enquadra. Na segunda fase procedemos ao levantamento dos requisitos necessários para a resolução dos problemas que tínhamos analisado e traçamos os objetivos principais.

Depois de estruturado o que pretendíamos e como iremos resolver desenvolvemos o modelo conceptual. De seguida fizemos a transição do modelo conceptual para o modelo logico onde o modelo foi manipulado conforme o que os diferentes utilizadores poderão manipular na base de dados.

Por fim procedemos à implementação física da base de dados, apresentando o Sistema de Gestão de Bases de Dados escolhido e também alguns exemplos de transações sobre BD.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados para uma escola de condução

Palavras-Chave: Alunos, Instrutores, Aulas, Exames, Horários, Condução, Código, Escolas. Funcionários, Categoria da carta, Base de Dados, Sistema de Gestão de Bases de Dados, Entidades, Atributos

Índice

Resumo	3
Índice de Anexos	6
Índice de Figuras	7
Índice de Tabelas	8
1. Definição do Sistema	9
1.1. Contextualização	9
1.2. Fundamentação da Base de Dados	10
1.3. Análise da Viabilidade do Projeto	10
2. Levantamento e Análise de Requisitos	11
2.1. Método de Levantamento e de análise de requisitos	11
2.2. Requisitos levantados	11
2.2.1. Requisitos de Descrição	11
2.2.2. Requisitos de Exploração	14
2.2.3 Requisitos de Controlo	16
3. Modelação Conceptual	17
3.1. Apresentação da abordagem de modelação realizada	17
3.2. Identificação e caracterização das entidades	17
3.3. Identificação e caracterização dos relacionamentos	19
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.	20
3.5. Detalhe ou generalização de entidades	26
3.6. Apresentação do Modelo Conceptual	27
3.7. Validação do modelo de dados produzido	28
4. Modelação Lógica	29
4.1. Construção e validação do modelo de dados lógico	29
4.2. Desenho do modelo lógico	30
4.3. Validação do modelo através da normalização	31
4.4. Validação do modelo com interrogações do utilizador	32

4.5. Revisão do modelo lógico produzido	32
5. Implementação Física	33
5.1. Seleção do sistema de gestão de bases de dados	33
5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	33
5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)	41
5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)	42
5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual	43
5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	44
5.7. Revisão do sistema implementado	45
6. Conclusões e Trabalho Futuro	46
7. Referências Bibliográficas	47
8. Anexos	48

Anexos

1.Anexo 1-Mensagem dos donos da empresa Rua Segura	46
2.Anexo 2-Base de Dados	47
3.Anexo 3-Script com algumas das funcionalidades pedidas pela Rua Segura	47

Índice de Figuras

Figura 1: Modelo Conceptual	26
Figura 2: Modelo Lógico	29

Índice de Tabelas

Tabela 1: Relacionamentos binários do modelo conceptual	19
Tabela 2: Atributos da entidade alunos	21
Tabela 3: Atributos da entidade Exames	22
Tabela 4: Atributos da entidade Exames	22
Tabela 5: Atributos da entidade Instrutores	22
Tabela 6: Atributos da entidade Viaturas	23
Tabela 7: Atributos da entidade Horários	24
Tabela 8: Atributos da entidade Funcionários	24
Tabela 9: Atributos da entidade Escolas	25
Tabela 10: Atributos da entidade Aulas	25

1. Definição do Sistema

1.1. Contextualização

Nos últimos anos, o uso de veículos individuais tem aumento de forma exponencial e em especial na população mais jovem. Isto deve-se ao facto de existir uma maior facilidade na aquisição destes tipos de veículos e no maior poder de compra das famílias.

Atualmente ter um veículo é visto como uma necessidade básica e não tanto como um luxo com acontecia anteriormente, pois este facilita em muito a vida de uma pessoa. Qualquer pessoa que possua um veículo individual pode fazer os seus próprios horários e deslocar-se facilmente para onde pretender sem grande dificuldade.

Mas para que as pessoas o possam utilizar é necessário que estas tenham uma carta de condução, para não o fazerem de forma ilegal. Esta carta tem de ser adquirida numa escola de condução e é através destas que se percebe que existe um aumento de pessoas com carta de condução, pois o número de escolas de condução por cidade é muito maior que a alguns anos atrás.

Apesar de não ser uma das principais razões pela qual as pessoas procuram tirar a carta, há um número significativo de pessoas que o procuram fazer pois é obrigatório para a profissão que praticam.

Devido a este aumento de pessoas que procuram este serviço, os donos de uma escola de condução chamada Rua Segura, desafiaram-nos a desenvolver uma base de dados para a sua escola de condução, pois pretendiam uma forma mais organizada e que permitisse guardar e alterar dados de forma mais eficaz e com maior capacidade.

1.2. Fundamentação da Base de Dados

Uma base de dados tem como função permitir uma organização detalhada de informação. O objetivo desta será permitir uma manipulação e consulta da dada (por parte dos gestores, funcionarios, e instrutores) robusta mas ao mesmo tempo acessível e de fácil procura. De forma a que possam ser guardados os dados dos alunos, funcionários, instrutores, viaturas, preços, horarios dos varios estabelecimentos que uma escola possa ter.

1.3. Análise da Viabilidade do Projeto

Apesar de ser possível guardar os dados em papel, uma forma mais eficaz de guardá-los e de acedê-los é através de uma base de dados, pois facilita a procura, sendo muito mais prático alterar dados e enviar para os centros de fiscalização e de provas (IMT).

2. Levantamento e Análise de Requisitos

2.1. Método de Levantamento e de análise de requisitos

Quando nos foi proposta a criação de uma base de dados para uma empresa de escolas de condução numa primeira fase, pesquisamos um pouco à cerca de como este negócio funciona e que tipo de informações são normalmente necessárias para este tipo de negócio.

Numa outra fase, encontramos com o proprietário e questionamo-lo sobre o que este pretendia que a base de dados tivesse capacidade para fazer e que informações eram necessárias que desse para guardar e adicionar.

Posto isto, listamos os requisitos necessários para realizar o que nos foi pretendido.

2.2. Requisitos levantados

2.2.1. Requisitos de Descrição

1. Guardar os nomes dos alunos, funcionários e instrutores.
2. Guardar a data de nascimento dos alunos.
3. Guardar a rua, localidade e código postal dos alunos, funcionários e instrutores.
4. Guardar os contactos dos alunos, instrutores e funcionários.

5. Guardar os emails, caso tenham, dos alunos, instrutores e funcionários.
6. Guardar o número do CC dos alunos, instrutores e funcionários.
7. Saber se o aluno possui atestado médico.
8. Saber o número de aulas de condução e teóricas que os alunos já tiveram.
9. Saber quanto o aluno pagou pela carta.
10. Saber se o aluno já passou no código.
11. Saber se o aluno já passou na condução.
12. Guardar o salário do instrutor e dos funcionários.
13. Guardar os preços de cada tipo de carta.
14. Guardar o tipo de carta.
15. Guardar o tipo de carta que o aluno quer tirar.
16. Guardar o tipo de carta que o instrutor tem.
17. Guardar os alunos, funcionários e instrutores pelo número de aluno, funcionário e instrutor, respetivamente.
18. Guardar as datas dos exames.
19. Guardar os horários das aulas teóricas.
20. Marcar as aulas de condução.
21. Guardar qual viatura usa cada instrutor.
22. Guardar que alunos o instrutor dá aulas.
23. Saber se o instrutor dá aulas de código ou condução.
24. Guardar matrícula da viatura.
25. Guardar modelo da viatura.
26. Guardar tipo da viatura (Carro, mota, ...).

- 27.** Guardar cor da viatura.
- 28.** Guardar ano da viatura.
- 29.** Guardar validade do selo, do seguro e da inspeção.
- 30.** Guardar dados caso algo precise de ser arranjado.
- 31.** Guardar estabelecimentos da escola por número.
- 32.** Guardar o email da escola.
- 33.** Guardar o contacto do estabelecimento.
- 34.** Guardar a rua, localidade e código postal do estabelecimento.
- 35.** Guardar quais funcionários, instrutores e alunos pertencem a cada estabelecimento.
- 36.** Alterar Contacto ou email de aluno, funcionário e instrutor
- 37.** Alterar instrutor de aluno.
- 38.** Alterar carro de instrutor.
- 39.** Alterar morada de aluno, instrutor e funcionário.
- 40.** Alterar salário de um instrutor ou funcionário.
- 41.** Apagar funcionário, instrutor ou aluno caso ele não pertença mais á escola.
- 42.** Mudar aluno de estabelecimento.
- 43.** Apagar carro caso este deixe de ser utilizado na escola.
- 44.** Pôr que o aluno entregou o atestado médico.
- 45.** Alterar preço de categoria.
- 46.** Alterar Horário.
- 47.** Alterar Aula.

2.2.2. Requisitos de Exploração

1. Ver os nomes dos alunos, funcionários e instrutores.
2. Ver a data de nascimento dos alunos.
3. Ver a rua, localidade e código postal dos alunos, funcionários e instrutores.
4. Ver os contactos dos alunos, instrutores e funcionários.
5. Ver os emails, caso tenham, dos alunos, instrutores e funcionários.
6. Ver o número do CC dos alunos, instrutores e funcionários.
7. Ver se o aluno possui atestado médico.
8. Ver o número de aulas de condução e teóricas que o aluno já fez.
9. Ver quanto o aluno pagou pela carta.
10. Ver se o aluno já passou no código.
11. Ver se o aluno já passou na condução.
12. Ver o salário do instrutor e dos funcionários.
13. Ver os preços de cada tipo de carta.
14. Ver o tipo de carta.
15. Ver o tipo de carta que o aluno quer tirar.
16. Ver o tipo de carta que o instrutor tem.
17. Ver os alunos, funcionários e instrutores pelo número de aluno, funcionário e instrutor, respetivamente.
18. Ver as datas dos exames de cada aluno.
19. Ver os horários das aulas teóricas.
20. Ver as aulas de condução marcadas.

- 21.** Ver qual viatura usa cada instrutor.
- 22.** Ver que alunos o instrutor dá aulas.
- 23.** Ver matrícula das viaturas.
- 24.** Ver modelos das viaturas.
- 25.** Ver tipo das viaturas (Carro, mota, ...).
- 26.** Ver cor das viaturas.
- 27.** Ver ano das viaturas.
- 28.** Ver validade do selo, do seguro e da inspeção das viaturas.
- 29.** Ver dados da viatura caso algo precise de ser arranjado.
- 30.** Ver estabelecimentos da escola por número.
- 31.** Ver o email da escola.
- 32.** Ver o contacto do estabelecimento.
- 33.** Ver a rua, localidade e código postal do estabelecimento.
- 34.** Ver quais funcionários, instrutores e alunos pertencem a cada estabelecimento.
- 35.** Ver quanto falta pagar ao aluno.
- 36.** Ver quantas aulas faltam para fazer as aulas obrigatórias.
- 37.** Ver os alunos que passaram no exame de código e condução.
- 38.** Quantos alunos estão a tirar cada tipo de carta.
- 39.** Faixa etária da maior parte dos alunos.
- 40.** Ver que carta um aluno quer tirar.
- 41.** Ver Contacto de um aluno, instrutor, funcionário.
- 42.** Ver Email de um aluno, instrutor, funcionário.
- 43.** Ver salario de um funcionário ou um instrutor.
- 44.** Ver Aulas de condução marcadas por cada aluno.

- 45. Ver Aulas de condução marcadas de cada instrutor.
- 46. Ver Datas do exame de um aluno.
- 47. Ver morada de um aluno, instrutor, funcionário.
- 48. Ver número de CC de um aluno, instrutor ou funcionário.

2.2.3 Requisitos de Controlo

Só poderá aceder às tabelas dos funcionários e dos instrutores os donos da empresa para não ser possível alterar valores dos seus salários apenas podendo ser os donos a fazê-lo.

Às outras tabelas poderam aceder os funcionários e os instrutores.

Os alunos poderam apenas aceder à tabela aulas e horários para verem que aulas foram marcadas não as podendo alterar.

2.3 Análise e validação geral dos requisitos

O aluno e o instrutor são as principais entidades deste sistema. O aluno é identificado pelo seu número de aluno e o instrutor é identificado pelo seu número de instrutor que é dado automaticamente.

Cada aluno tem um instrutor para as aulas de condução. As aulas estão associados ao aluno tal como ao instrutor.

O aluno marca aulas de condução com o funcionário identificado pelo seu número de funcionário. O aluno, o instrutor e os funcionários estão associados a um estabelecimento com o horário das aulas teóricas.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Para o desenvolvimento do Modelo Conceptual, foi importante para o grupo ter os requisitos já definidos. A partir deles, foram identificadas as entidades, os atributos e as relações entre atributos e entidades. Para identificar as entidades escolheram-se as chaves primárias.

3.2. Identificação e caracterização das entidades

Atendendo aos requisitos levantados, foram concebidas as seguintes entidades:

- **Alunos**

O conjunto de todos os alunos de estão inscritos nas diferentes escolas de condução da empresa Escola Segura.

- **Exames**

Todos os alunos têm obrigatoriamente de realizar o exame de condução e do de código, esta entidade vai guardar as datas da realização dos mesmo de cada aluno

- **Categorias**

Uma escola de condução tem vários tipos de carta conforme o tipo de viatura que o aluno pretende conduzir. Cada tipo de carta varia conforme o preço, sendo que já está estipulado pela escola.

- **Escolas**

Uma empresa pode ter mais do que um estabelecimento de escolas condução, mas um aluno quando se inscreve numa das escolas da empresa ele fica aluno só dessa escola em que se inscreveu.

- **Aulas**

São as aulas praticas que o aluno marca

- **Instrutores**

Para que as aulas de condução sejam dadas é necessário instrutores, que pertencem as diferentes escolas da empresa. A cada aluno é entregue um instrutor com o qual ele vai fazer pelo menos o número mínimo de aulas necessárias para que este possa realizar os exames. Os instrutores também dão as aulas de código.

- **Funcionários**

Uma escola também necessita de funcionários para a realização do resto das tarefas (estes não dão aulas) que são necessárias realizar numa escola de condução

- **Horários**

São a hora e o dia em que a escola tem horário para que os alunos possam ir às aulas de código dadas pelos instrutores

- **Viaturas**

Para que os alunos realizem as aulas de condução é necessário que a escola tenha viaturas. Cada viatura pertence a um instrutor e este pode ter mais do que uma viatura.

Como existem diferentes tipos de cartas então também existem diferentes tipos de viaturas como mota, carro, camião...

3.3. Identificação e caracterização dos relacionamentos

<i>Entidade</i>	<i>Relacionamento</i>	<i>Entidade</i>	<i>cardinalidade</i>	<i>Descrição</i>
Alunos	Tem	Categoria	1:1	Alunos tem uma categoria
Categoria	Tem	Alunos	0:n	As categorias tem 0 ou N alunos
Alunos	Tem	Exame	0:1	Um aluno não tem exame ou tem 1 exame
Exames	Tem	Alunos	0:n	Um exame tem 0 ou N alunos
Alunos	Tem	Instrutores	1:1	Cada aluno tem um só instrutor
Instrutores	Tem	Alunos	0:n	Um instrutor pode ter 0 ou vários alunos
Alunos	Tem	Aulas	0:n	Alunos tem zero ou várias aulas
Aulas	Tem	Alunos	1:1	Cada aula tem um aluno
Alunos	Tem	Escolas	1:1	Cada aluno pertence a uma só escola
Escolas	Tem	Alunos	0:n	As escolas têm 0 ou vários alunos
Instrutores	Tem	Viaturas	1:n	1 instrutor tem 1 ou várias viaturas
Viaturas	Tem	Instrutores	1:1	Todas a viaturas têm 1 e 1 só instrutor

Instrutores	Tem	Escolas	1:1	Cada instrutor pertence a uma só escola
Escolas	Tem	Instrutores	1:n	Escolas tem 1 ou vários instrutores
Instrutores	Tem	Aulas	0:n	Instrutor tem 0 ou varias aulas
Aulas	Tem	Instrutores	1:1	Aulas tem um e um só instrutor
Instrutores	Tem	Horários	0:n	Instrutores não tem aulas ou tem várias
Horários	Tem	Instrutores	0:1	Horários não tem instrutores ou se tem é dada por um só instrutor
Horários	Tem	Escolas	1:1	Cada horário tem uma escola
Escolas	Tem	Horários	1:n	Escolas pode ter mais que um horário
Funcionários	Tem	Escolas	1:1	1 Funcionário tem uma só escola
Escolas	Tem	Funcionários	0:n	Escolas tem 0 ou vários funcionários

Tabela 11: Relacionamentos binários do modelo conceptual

3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.

Após termos apresentado todas as entidades e relacionamentos com que iremos trabalhar, apresentaremos de seguida os atributos presentes em cada entidade e como estes se comportam.

Alunos

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
NrAluno	Número único que identifica unicamente o aluno	Número Inteiro Positivo	Não	Não
Nome	Nome do Aluno	Sequência de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Data_de_nascimento	Data de Nascimento do aluno	Date	Não	Não
Nr_CC	Número do cartão do cidadão do aluno	Número Inteiro Positivo	Não	Não
AtestadoMedico	Atestado Médico	Booleano (True or False)	Não	Sim
NrAulaT	Número de aulas de Código feitas pelo aluno até ao momento	Número Inteiro Positivo	Não	Sim
NrAulaP	Número de aulas de Condução feitas pelo aluno até ao momento	Número Inteiro Positivo	Não	Sim
Rua	Morada da residência atual do aluno	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Codigo_Postal	Código Postal da residência atual do aluno	Sequencia de Caracteres Tamanho Variável (Max: 8 caracteres)	Não	Não
Localidade	Localidade atual do aluno	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
QntPagou	Quanto o aluno pagou até ao momento da carta	Número Inteiro Positivo	Não	Sim
CodigoF	Código Feito	Booleano (True or False)	Não	Sim

ConducaoF	Condução Feita	Booleano (True or False)	Não	Sim
Contacto	Contacto telefónico do aluno	Número Inteiro Positivo	Não	Não
Email	Email do aluno	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Sim

Tabela 12: Atributos da entidade alunos

Exames

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
DataExameCodigo	Data do Exame de Código	DateTime	Não	Sim
DataExameCond	Data do Exame de Condução	DateTime	Não	Sim

Tabela 13: Atributos da entidade Exames

Categoria

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
Preço	Preço do custo dos diferentes tipos de carta	Número Float	Não	Sim
Tipo	Diferentes tipos de carta das escolas de condução	Sequencia de Caracteres Tamanho Variável (Max: 10 caracteres)	Sim	Não

Tabela 14: Atributos da entidade Exames

Instrutores

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
NrInstrutor	Número único que identifica unicamente o instrutor	Número Inteiro Positivo	Não	Não
Rua	Morada da residência atual	Sequencia de Caracteres	Não	Não

	do instrutor	Tamanho Variável (Max: 45 caracteres)		
Email	Email do instrutor	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Sim
Nome	Nome do Instrutor	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Contacto	Contacto telefónico do instrutor	Número inteiro positivo	Não	Não
Salário	Salário do instrutor	Número Float	Não	Não
Codigo_Postal	Código Postal da residência atual do aluno	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Localidade	Localidade atual do instrutor	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Nr_CC	Número do cartão de CC do instrutor	Número Inteiro Positivo	Não	Não

Tabela 15: Atributos da entidade Instrutores

Viaturas

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
Selo	Validade do Selo da viatura	Date	Não	Não
Tipo	Diferentes tipos de viaturas	Tamanho Variável (Max: 45 caracteres)	Não	Não
ValidadeInspecao	Validade da Inspeção da viatura	Date	Não	Não
Ano	Ano das viaturas	year	Não	Não
Modelo	Modelo da viatura	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Matricula	Matrícula da viatura	Sequencia de Caracteres Tamanho Variável (Max: 8 caracteres)	Não	Não
Observacoes	Acidentes,	Sequencia de	Não	Sim

	problemas e outras informações	Caracteres longtext		
Cor	Cor da viatura	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Sim
Seguro	Validade do Seguro da viatura	Date	Não	Não

Tabela 16: Atributos da entidade Viaturas

Horário

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
DiaSemana	Dia da Semana em que existe aulas teóricas	Sequencia de Caracteres Tamanho Variável (Max: 10 caracteres)	Não	Não
Hora	Hora em que existe aulas teóricas	Time	Não	Não

Tabela 17:Atributos da entidade Horários

Funcionários

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
NrFuncionario	Número único que identifica unicamente o funcionário	Número Inteiro Positivo	Não	Não
Nome	Nome do Funcionário	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Contacto	Contacto do funcionário	Número Inteiro Positivo	Não	Não
Salario	Salário do funcionário	Número float	Não	Não
Rua	Morada da residência atual do funcionário	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não

Codigo_Postal	Código Postal da residência atual do funcionário	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Localidade	Localidade atual do funcionário	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Nr_CC	Número de cartão CC do funcionário	Número Inteiro Positivo	Não	Não
Email	Email do funcionário	Sequencia de Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Sim

Tabela 18: Atributos da entidade Funcionários

Escolas

Atributo	Descrição	Tipo de Dados e Tamanho	Multivalor	Nulo
NrEscola	Número único que identifica unicamente a escola	Número Inteiro Positivo	Não	Não
Email	Email da escola	Sequencia de Caracteres Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Sim
Contacto	Contacto telefónico da escola	Número Inteiro Positivo	Não	Não
Rua	Morada da residência atual da escola	Sequencia de Caracteres Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não
Codigo_Postal	Código Postal da residência atual da escola	Sequencia de Caracteres Caracteres Tamanho Variável (Max: 8 caracteres)	Não	Não
Localidade	Localidade atual da escola	Sequencia de Caracteres Caracteres Tamanho Variável (Max: 45 caracteres)	Não	Não

		45 caracteres)		
--	--	----------------	--	--

Tabela 19: Atributos da entidade Escolas

Aulas

<i>Atributo</i>	<i>Descrição</i>	<i>Tipo de Dados e Tamanho</i>	<i>Multivalor</i>	<i>Nulo</i>
Horario	Horário das aulas praticas	DateTime	Não	Não

Tabela 20: Atributos da entidade Aulas

3.5. Detalhe ou generalização de entidades

A entidade alunos possui como atributos NrAluno, CofigoF, ConducaoF, Contacto, Email, Nome, Data_de_nascimento, Nr_CC, AtestadoMedico, Rua, Codigo_Postal, Localidade, NrAulaT, NrAulaP sendo o atributo chave o NrAluno visto que este vai ser sempre diferente.

A entidade Exames possui como atributos DataExameCodigo, DataExameCond não tendo atributo chave.

A entidade Categoria possui como atributos Tipo e Preco tendo com atributo chave o Tipo visto que este vai ser sempre diferente de Categoria para Categoria.

A entidade Aulas possui como atributo o Horario não tendo atributo chave.

A entidade Instrutores possui como atributos Salario, Contacto, Nome, NrInstrutor, Rua, Email, Nr_CC, Localidade, Codigo_Postal tendo como atributo o NrInstrutor visto que vai ser sempre diferente de Instrutor para Instrutor.

A entidade Funcionarios possui como atributos Salario, Contacto, Nome, NrFuncionario, Rua, Email, Nr_CC, Localidade, Codigo_Postal tendo como atributo o NrFuncionario visto que vai ser sempre diferente de Funcionario para Funcionario.

A entidade Escolas possui como atributos Contacto, Nome, NrEscola, Rua, Email, Localidade, Codigo_Postal tendo como atributo o NrEscola visto que vai ser sempre diferente de Escola para Escola.

A entidade Horario possui como atributos DiaSemana, Hora não tendo atributo chave.

A entidade Viaturas possui como atributos Matricula, Selo, Tipo, Validalnspecao, Ano, Modelo, Observacoes, Cor, Seguro tendo como atributo chave a Matricula pois esta é diferente de Viatura para Viatura.

3.6. Apresentação do Modelo Conceptual

Através dos requisitos que reconhecemos foi nos mais fácil elaborar o modelo conceptual.

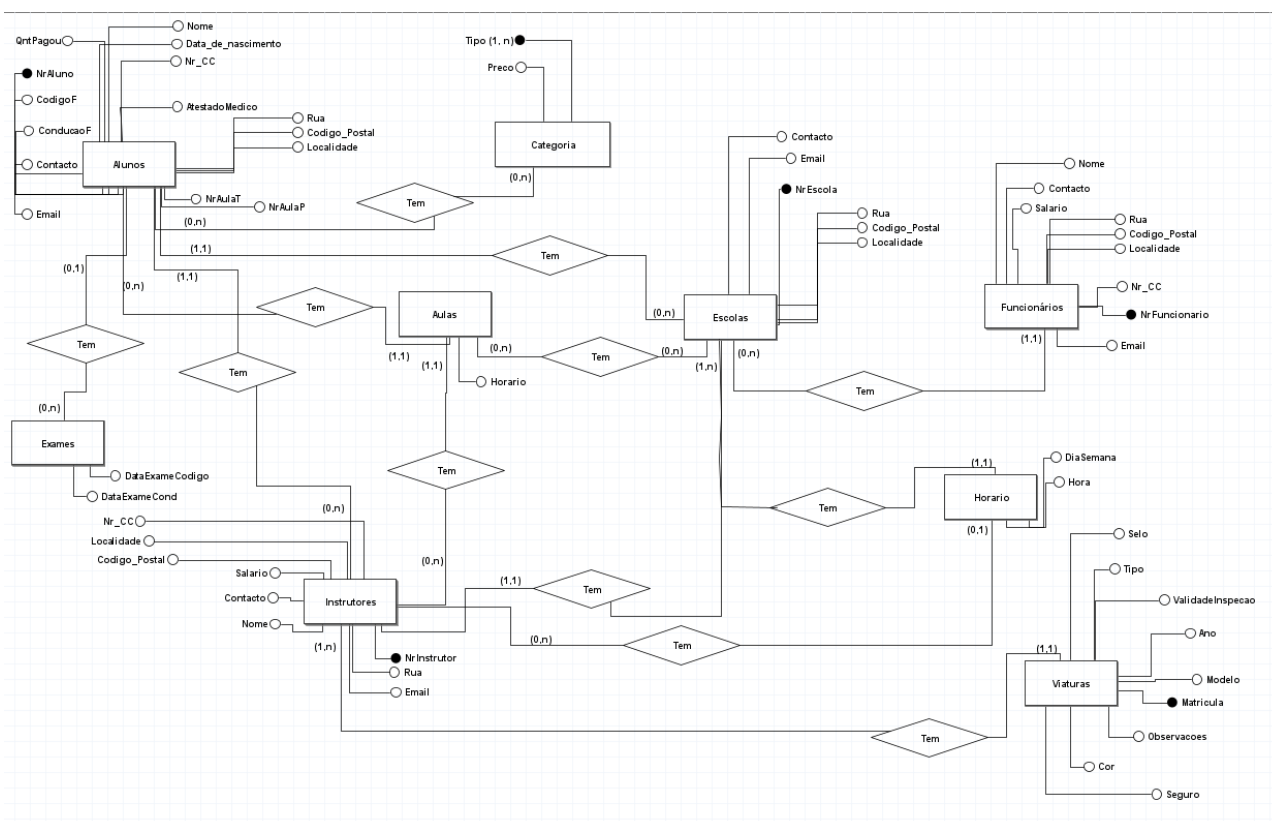


Figura 1: Modelo Conceptual

3.7. Validação do modelo de dados produzido

Depois de tudo decidido relativamente ao Modelo Conceptual, vamos ver se realmente com as nossas escolhas conseguimos responder aos pedidos que a empresa Rua Segura nos propôs.

Conseguimos guardar todas as entidades propostas Alunos, Instrutores, Funcionarios tendo estas ligação com a entidade Escolas.

Conseguimos estabelecer ligações entre a entidade Aulas e as entidades Alunos e Instrutores.

Também conseguimos ligar os Instrutores às suas viaturas e ao Horário de aulas teóricas.

Logo o modelo conceptual cumpre os pedidos da empresa Rua Segura então este é válido.

4. Modelação Lógica

4.1. Construção e validação do modelo de dados lógico

Depois de decidido qual o Modelo Conceptual a ser implementado, passamos à elaboração do nosso Modelo Lógico.

Criamos as tabelas correspondentes a cada entidade do modelo lógico tendo como colunas os seus atributos mantendo os tipos indicados nas tabelas do ponto 3.4.

As ligações do modelo conceptual foram representadas através de ligações por chaves estrangeiras, isto é, por exemplo, o aluno tem um professor então na tabela alunos possuímos uma chave estrangeira `instrutores_NrInstrutor` correspondendo á chave primária da tabela `instrutores`, acontece de forma semelhante nas outras tabelas como mostrado no Modelo Lógico.

O valor multivalorado Tipo de Categorias foi feito como um varchar isto é se possuímos uma categoria de tipo A e uma categoria de tipo B se o aluno quiser ter uma carta do tipo A,B tem de existir uma Categoria de Tipo A,B.

4.2. Desenho do modelo lógico

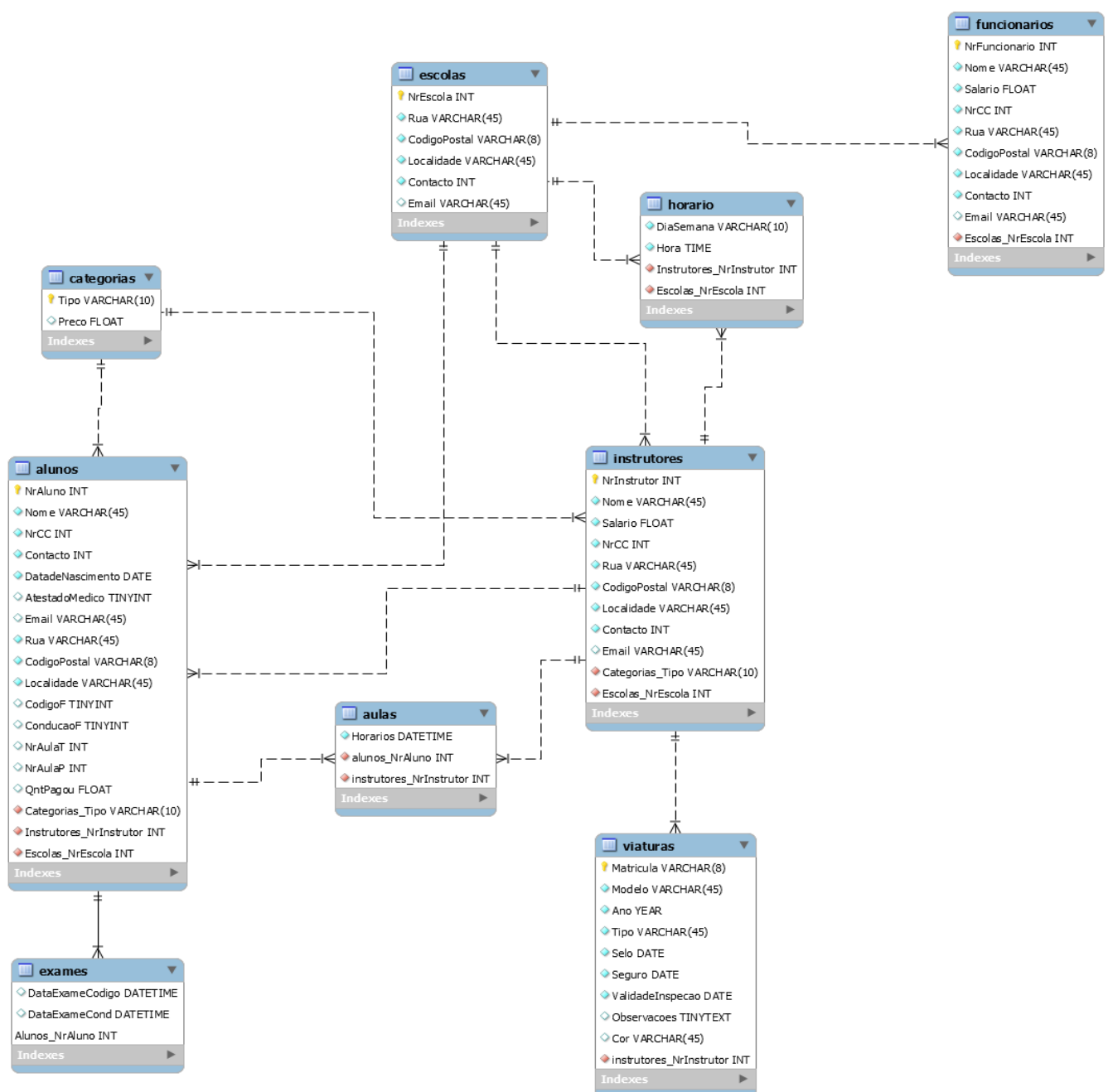


Figura 2: Modelo Lógico

4.3. Validação do modelo através da normalização

Com o objetivo de aumentar a coesão, coerência e o desempenho da base de dados desenvolvida pelo grupo, bem como a integridade dos dados, é feito de seguida a normalização dos dados, desde a primeira até à terceira forma normal (inclusive).

Primeira forma normal - as tabelas presentes no Modelo Lógico não têm atributos repetidos. Assim, todas as tabelas cumprem a primeira forma formal.

Segunda forma normal - em todas as tabelas presentes no Modelo Lógico, todos os atributos que não são chave primária apenas se encontram numa única tabela, quando querem ser referenciados noutra tabela usamos a chave primária da mesma. Sendo assim, não tivemos de fazer alteração nas tabelas ou mesmo criar outras para respeitar a segunda forma normal. Consequentemente, ao ser validada a primeira forma normal e verificada também a segunda, as tabelas cumprem a segunda forma normal.

Terceira forma normal - No Modelo Lógico desenvolvido, nenhum dos atributos não chave depende de outros igualmente classificáveis, para uma dada tabela. Não havendo dependências funcionais e sendo cumprida a segunda forma normal,, as relações encontram-se na terceira forma normal.

4.4. Validação do modelo com interrogações do utilizador

O modelo lógico é valido para os requisitos dos donos da Rua Segura, pois, é possível guardar os dados pedidos dos alunos, funcionários e instrutores, também é possível marcar exames, aulas.

É possível também guardar os preços de cada categoria e os dados das viaturas.

Ainda é possível guardar os dados dos estabelecimentos da Rua Segura.

Algumas das respostas às interrogações podem ser vistas no anexo 4.

4.5. Revisão do modelo lógico produzido

Após uma análise junto dos donos da empresa conclui-se que o Modelo Lógico desenvolvido pelo grupo aborda de forma correta os conceitos requeridos para o mesmo.

5. Implementação Física

5.1. Seleção do sistema de gestão de bases de dados

O sistema de gestão de base de dados que utilizamos foi o MySQL, pois é o que estamos mais familiarizados, visto que é o utilizado nas aulas

5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

```
USE `ruasegura` ;
```

```
-- -----  
-- Table `ruasegura`.`categorias`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ruasegura`.`categorias` (  
  `Tipo` VARCHAR(10) NOT NULL,  
  `Preco` FLOAT NULL DEFAULT NULL,  
  PRIMARY KEY (`Tipo`))  
ENGINE = InnoDB;
```



```

-----
-- Table `ruasegura`.`escolas`
-----

CREATE TABLE IF NOT EXISTS `ruasegura`.`escolas` (
  `NrEscola` INT NOT NULL AUTO_INCREMENT,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(8) NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Contacto` INT NOT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`NrEscola`))
ENGINE = InnoDB;

-----
-- Table `ruasegura`.`instrutores`
-----

CREATE TABLE IF NOT EXISTS `ruasegura`.`instrutores` (
  `NrInstructor` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `Salario` FLOAT NOT NULL,
  `NrCC` INT NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(8) NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Contacto` INT NOT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,

```

```

`Categorias_Tipo` VARCHAR(10) NOT NULL,

`Escolas_NrEscola` INT NOT NULL,

PRIMARY KEY (`NrInstrutor`),

INDEX `fk_Instrutores_Categorias1_idx` (`Categorias_Tipo`
ASC) VISIBLE,

INDEX `fk_Instrutores_Escolas1_idx` (`Escolas_NrEscola`
ASC) VISIBLE,

CONSTRAINT `fk_Instrutores_Categorias1`

FOREIGN KEY (`Categorias_Tipo`)

REFERENCES `ruasegura`.`categorias` (`Tipo`),

CONSTRAINT `fk_Instrutores_Escolas1`

FOREIGN KEY (`Escolas_NrEscola`)

REFERENCES `ruasegura`.`escolas` (`NrEscola`))

ENGINE = InnoDB;

```

```

-----
-- Table `ruasegura`.`alunos`
-----

```

```

CREATE TABLE IF NOT EXISTS `ruasegura`.`alunos` (

`NrAluno` INT NOT NULL AUTO_INCREMENT,

`Nome` VARCHAR(45) NOT NULL,

`NrCC` INT NOT NULL,

`Contacto` INT NOT NULL,

`DatadeNascimento` DATE NOT NULL,

`AtestadoMedico` TINYINT NULL DEFAULT NULL,

`Email` VARCHAR(45) NULL DEFAULT NULL,

```

```

`Rua` VARCHAR(45) NOT NULL,

`CodigoPostal` VARCHAR(8) NOT NULL,

`Localidade` VARCHAR(45) NOT NULL,

`CodigoF` TINYINT NULL DEFAULT NULL,

`ConducaoF` TINYINT NULL DEFAULT NULL,

`NrAulaT` INT NULL DEFAULT NULL,

`NrAulaP` INT NULL DEFAULT NULL,

`QntPagou` FLOAT NULL DEFAULT NULL,

`Categorias_Tipo` VARCHAR(10) NOT NULL,

`Instrutores_NrInstrutor` INT NOT NULL,

`Escolas_NrEscola` INT NOT NULL,

PRIMARY KEY (`NrAluno`),

INDEX `fk_Alunos_Categorias_idx` (`Categorias_Tipo` ASC)

VISIBLE,

INDEX `fk_Alunos_Instrutores1_idx`

(`Instrutores_NrInstrutor` ASC) VISIBLE,

INDEX `fk_Alunos_Escolas1_idx` (`Escolas_NrEscola` ASC)

VISIBLE,

CONSTRAINT `fk_Alunos_Categorias`

FOREIGN KEY (`Categorias_Tipo`)

REFERENCES `ruasegura`.`categorias` (`Tipo`),

CONSTRAINT `fk_Alunos_Escolas1`

FOREIGN KEY (`Escolas_NrEscola`)

REFERENCES `ruasegura`.`escolas` (`NrEscola`),

CONSTRAINT `fk_Alunos_Instrutores1`

FOREIGN KEY (`Instrutores_NrInstrutor`)

```

```

REFERENCES `ruasegura`.`instrutores` (`NrInstrutor`))

ENGINE = InnoDB;

-- -----
-- Table `ruasegura`.`aulas`
-- -----

CREATE TABLE IF NOT EXISTS `ruasegura`.`aulas` (

  `Horarios` DATETIME NOT NULL,

  `alunos_NrAluno` INT NOT NULL,

  `instrutores_NrInstrutor` INT NOT NULL,

  INDEX `fk_Aulas_alunos1_idx` (`alunos_NrAluno` ASC)
  VISIBLE,

  INDEX `fk_Aulas_instrutores1_idx`
  (`instrutores_NrInstrutor` ASC) VISIBLE,

  CONSTRAINT `fk_Aulas_alunos1`

    FOREIGN KEY (`alunos_NrAluno`)

      REFERENCES `ruasegura`.`alunos` (`NrAluno`),

  CONSTRAINT `fk_Aulas_instrutores1`

    FOREIGN KEY (`instrutores_NrInstrutor`)

      REFERENCES `ruasegura`.`instrutores` (`NrInstrutor`))

ENGINE = InnoDB;

```

```

-----
-- Table `ruasegura`.`exames`
-----

CREATE TABLE IF NOT EXISTS `ruasegura`.`exames` (
  `DataExameCodigo` DATETIME NULL DEFAULT NULL,
  `DataExameCond` DATETIME NULL DEFAULT NULL,
  `Alunos_NrAluno` INT NOT NULL,
  PRIMARY KEY (`Alunos_NrAluno`),
  CONSTRAINT `fk_Exames_Alunos1`
    FOREIGN KEY (`Alunos_NrAluno`)
    REFERENCES `ruasegura`.`alunos` (`NrAluno`))
ENGINE = InnoDB;

-----

-- Table `ruasegura`.`funcionarios`
-----

CREATE TABLE IF NOT EXISTS `ruasegura`.`funcionarios` (
  `NrFuncionario` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `Salario` FLOAT NOT NULL,
  `NrCC` INT NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(8) NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Contacto` INT NOT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,

```

```

`Escolas_NrEscola` INT NOT NULL,

PRIMARY KEY (`NrFuncionario`),

INDEX `fk_Funcionarios_Escolas1_idx` (`Escolas_NrEscola`
ASC) VISIBLE,

CONSTRAINT `fk_Funcionarios_Escolas1`

FOREIGN KEY (`Escolas_NrEscola`)

REFERENCES `ruasegura`.`escolas` (`NrEscola`))

ENGINE = InnoDB;

```

```

-----

-- Table `ruasegura`.`horario`

-----

CREATE TABLE IF NOT EXISTS `ruasegura`.`horario` (

`DiaSemana` VARCHAR(10) NOT NULL,

`Hora` TIME NOT NULL,

`Instrutores_NrInstrutor` INT NOT NULL,

`Escolas_NrEscola` INT NOT NULL,

INDEX                                `fk_Horario_Instrutores1_idx`

(`Instrutores_NrInstrutor` ASC) VISIBLE,

INDEX `fk_Horario_Escolas1_idx` (`Escolas_NrEscola` ASC)

VISIBLE,

CONSTRAINT `fk_Horario_Escolas1`

FOREIGN KEY (`Escolas_NrEscola`)

REFERENCES `ruasegura`.`escolas` (`NrEscola`),

CONSTRAINT `fk_Horario_Instrutores1`

FOREIGN KEY (`Instrutores_NrInstrutor`)

```

```

        REFERENCES `ruasegura`.`instrutores` (`NrInstrutor`))

ENGINE = InnoDB;

-- -----
-- Table `ruasegura`.`viaturas`
-- -----

CREATE TABLE IF NOT EXISTS `ruasegura`.`viaturas` (

    `Matricula` VARCHAR(8) NOT NULL,

    `Modelo` VARCHAR(45) NOT NULL,

    `Ano` YEAR NOT NULL,

    `Tipo` VARCHAR(45) NOT NULL,

    `Selo` DATE NOT NULL,

    `Seguro` DATE NOT NULL,

    `ValidadeInspecao` DATE NOT NULL,

    `Observacoes` TINYTEXT NULL DEFAULT NULL,

    `Cor` VARCHAR(45) NULL DEFAULT NULL,

    `instrutores_NrInstrutor` INT NOT NULL,

    PRIMARY KEY (`Matricula`),

    INDEX          `fk_viaturas_instrutores1_idx`
    (`instrutores_NrInstrutor` ASC) VISIBLE,

    CONSTRAINT `fk_viaturas_instrutores1`

    FOREIGN KEY (`instrutores_NrInstrutor`)

    REFERENCES `ruasegura`.`instrutores` (`NrInstrutor`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)

Quanto falta ao aluno pagar? É respondido pelo seguinte comando:

call alunopagar(1); sendo 1 o numero do aluno.

Função definida por:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `alunopagar` (Nr
int)

BEGIN

    select Nome, ((select Preco from categorias where
    tipo= (select categorias_tipo from alunos where
    Nr=NrAluno) ) - (select QntPagou from alunos where
    Nr=NrAluno)) as QntFaltaPagar from alunos where
    Nr=NrAluno ;

END
```

Quantas aulas faltam para o aluno poder fazer os exames? É respondido pelo seguinte comando:

call qntsaulas(1); sendo o 1 o numero do aluno.

Função definida por:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `qntsaulas` (Nr
int)

BEGIN
```



```

select Nome, (16 - (select NrAulaP from alunos))
as QntFaltaParaCodigo, (32 - (select NrAulaP from
alunos)) as QntFaltaParaConducao from alunos
where NrAluno=Nr;

END

```

Qual é Média de idades dos alunos? É respondido pelo seguinte comando:

```
select * from mediaidade;
```

Função definida por:

```

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `ruasegura`.`mediaidade` AS

SELECT

    (TIMESTAMPDIFF(YEAR,

        `ruasegura`.`alunos`.`DatadeNascimento`,

        NOW())) / COUNT(`ruasegura`.`alunos`.`NrAluno`))

AS `MediaIdades`

FROM `ruasegura`.`alunos`;

```

5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Visto que são muitos alunos decidimos criar um índice único no campo Contactos.

Alter table alunos Add UNIQUE INDEX Contacto(Contacto);

5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual

O tamanho ocupado no máximo por cada linha de uma dada tabela é o seguinte:

- Alunos: 253 bytes por linha (aproximadamente)
- Escolas: 155 bytes por linha (aproximadamente)
- Categorias: 20 bytes por linha (aproximadamente)
- Funcionarios: 222 bytes por linha (aproximadamente)
- Horario: 28 bytes por linha (aproximadamente)
- Instrutores: 244 bytes por linha (aproximadamente)
- Aulas: 16 bytes por linha (aproximadamente)
- Viaturas: 423 bytes por linha (aproximadamente)

Por ano inscrevem-se aproximadamente 300 alunos logo no máximo 75900 bytes para guardar a informação dos alunos.

Tendo por enquanto a empresa Rua Segura apenas um estabelecimento, três instrutores (e uma viatura por cada instructor) e um funcionário então não é expectável que sejam mais de 2223 bytes.

O horário se mudar não aumenta o número de linhas logo tendo 11 linhas então são necessários 308 bytes.

As categorias vão se mantendo anualmente logo tendo 9 linhas ocuparia 180 bytes.

Por fim havendo 30 aulas por dia marcando algumas com um mês de antecedência, sendo estas apagadas depois de terem ocorrido, então ocupariam 14400 bytes.

Por ano em média está ocupado em disco 93011 bytes.

5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

Saber média de idades dos alunos:

```
select * from mediaidade;
```

Função definida por:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `ruasegura`.`mediaidade` AS
    SELECT
        (TIMESTAMPDIFF(YEAR,
            `ruasegura`.`alunos`.`DatadeNascimento`,
            NOW())) / COUNT(`ruasegura`.`alunos`.`NrAluno`))
AS `MediaIdades`
FROM `ruasegura`.`alunos`;
```

Saber número de aulas feitas pelo aluno:

```
Select * from aulasalunos;
```

Função definida por:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `aulasalunos` AS
    SELECT
        `alunos`.`NrAluno` AS `NrAluno`,
        `alunos`.`Nome` AS `Nome`,
        `alunos`.`NrAulaT` AS `NrAulaT`,
        `alunos`.`NrAulaP` AS `NrAulaP`
    FROM `alunos`;
```

5.7. Revisão do sistema implementado

Depois de uma última revisão dada pelos donos da Rua Segura da base de dados mostraram-se bastante satisfeitos com as propriedades do mesmo. Após algumas semanas de utilização fomos contactados pelos donos por email mostrando-nos toda a sua satisfação com o resultado obtido para a escola através do Sistema de Base de Dados implementado. Podemos ler o que nos foi enviado pelos donos da Rua Segura no anexo 1.

6. Conclusões e Trabalho Futuro

No final deste trabalho, o grupo considera que conseguiu aplicar de forma correta o que foi lecionado até ao momento no semestre, relativamente ao desenvolvimento de uma base de dados relacional. Foi-nos transmitido o conceito de base de dados, qual a sua funcionalidade, quais as vantagens da sua utilização, formas de gerir a informação nela contida (codificando views, procedures, etc.). A partir daqui o grupo refletiu e chegou a um consenso em relação ao tema a escolher, base de dados de uma escola de condução.

Este é um tema que, considera o grupo, é suficiente para serem aplicados os conceitos que nos foram sendo transmitidos. Com o desenvolvimento do Modelo Conceptual e do Modelo Lógico, reparou-se na importância que é preciso dar às funcionalidades visto que é necessário desenvolver e especificar de forma muito concreta e concisa os requisitos da base de dados desenvolvida. O grupo teve o cuidado de, desde o início, construir ambos os modelos de forma a cumprirem os requisitos.

Todos os elementos do grupo cumpriram com o seu conhecimento para o desenvolvimento do trabalho.

O trabalho apresentado foi construído com o intuito de ser o mais legível possível, com identificação correta das entidades e respetivos atributos e com queries codificadas de modo a poder ser feita uma boa exploração da base de dados.

7. Referências Bibliográficas

<https://www.w3schools.com/sql/>

<https://stackoverflow.com/>

<https://controlauto.pt/seguranca-rodoviaria/educacao-rodoviaria/categorias-carta-de-conducao>

<https://www.rlsystem.com.br/tipos-dados-sql-server/>

8. Anexos

Boa tarde,

Como combinado, após 1 mês de utilização vamos vos dar a opinião que recolhemos junto dos nossos colaboradores sobre a base de dados que desenvolveram para a nossa empresa.

Como tínhamos relatado um dos principais problemas da base de dados com que a empresa trabalhava antigamente era a falta de organização e da informação que era possível armazenar, o que complicava o trabalho dos nossos funcionários e que por vezes provocava erros na altura de dar informações aos alunos como aos próprios funcionários da empresa.

Pela informação que os nossos funcionários me transmitiram, todos esses problemas foram resolvidos, elogiaram bastante a organização e facilidade de guardar e pesquisar informação.

Assim, só temos de elogiar o vosso trabalho e esforço que colocaram para desenvolverem uma base de dados que responde a todos os pedidos que vos tinham sido colocados.

Muito obrigado,

Cumprimentos,

Donos Escola Segura

Anexo 1-Mensagem dos donos da empresa Rua Segura



TPBD.mwb

Anexo 2-Base de Dados



Script.sql

Anexo 3-Script com algumas das funcionalidades pedidas pela Rua Segura