|  |  |
| --- | --- |
|  |  |

**Licenciatura em Engenharia Informática**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Unidade curricular de

Projeto I

Relatório de projeto

2021/2022

*Gestão de uma escola de condução*

24585 Alexandre Santos

24586 Bruno Oliveira

Sumário

[1. Introdução 3](#_Toc90740476)

[1.1 Apresentação do tema 3](#_Toc90740477)

[1.2 Objetivos do projeto 3](#_Toc90740478)

[1.3 Atores 3](#_Toc90740479)

[2. Apresentação do negócio 3](#_Toc90740480)

[2.1 Âmbito e enquadramento do projeto 3](#_Toc90740481)

[2.2 Modelação dos Processos de negócio 4](#_Toc90740482)

[2.2.1 Processo 1 - Inscrição de um aluno 4](#_Toc90740483)

[2.2.2 Processo 2 - Inscrição a exame teórico 4](#_Toc90740484)

[2.2.3 Processo 3 - Inscrição a exame prático 5](#_Toc90740485)

[2.2.4 Processo 4 - Controlo de assiduidade aulas teóricas 5](#_Toc90740486)

[2.2.5 Processo 5 – Controlo de assiduidade aulas práticas 6](#_Toc90740487)

[3. Levantamento de Requisitos 7](#_Toc90740488)

[3.1 Tipos de utilizador 7](#_Toc90740489)

[3.1.1 Apresentação dos tipos de utilizador 7](#_Toc90740490)

[3.1.2 Ações de cada tipo de utilizador 7](#_Toc90740491)

[3.2 Requisitos Funcionais 7](#_Toc90740492)

[3.3 Requisitos não funcionais 8](#_Toc90740493)

[4. Design e Modelação 9](#_Toc90740494)

[4.1 Modelo de casos de uso 9](#_Toc90740495)

[4.1.1 Caso de uso: Inscrever aluno 9](#_Toc90740496)

[4.1.2 Caso de uso: Gerir pagamento inicial 10](#_Toc90740497)

[4.1.3 Caso de uso: Marcar aulas práticas 10](#_Toc90740498)

[4.1.4 Caso de uso: Controlar assiduidade 11](#_Toc90740499)

[4.2 Modelo de classes 11](#_Toc90740500)

[4.3 Diagrama de Sequência 13](#_Toc90740501)

[4.3.1 Inscrever Aluno 13](#_Toc90740502)

[4.3.2 Verificar Disponibilidade 13](#_Toc90740503)

[4.4 Diagrama de Transição de Estad 14](#_Toc90740504)

[4.4.1 Inscrição Aluno 14](#_Toc90740505)

[4.4.2 Presenças nas Aulas Teóricas 14](#_Toc90740506)

[4.4.3 Presenças nas Aulas Práticas 15](#_Toc90740507)

[4.4.4 Exame Teórico 15](#_Toc90740508)

[4.4.5 Exame Prático 16](#_Toc90740509)

[5. Implementação da BD 17](#_Toc90740510)

[5.1 DER 17](#_Toc90740511)

[5.2 Modelo de tabelas 17](#_Toc90740512)

[5.3 Normalização 18](#_Toc90740513)

[5.4 Código SQL 19](#_Toc90740514)

[5.4.1 Criação de tabelas 19](#_Toc90740515)

[6. Conclusão e Trabalho Futuro 23](#_Toc90740516)

# Introdução

## Apresentação do tema

Neste projeto será apresentado o desenvolvimento de uma aplicação para gestão de uma escola de condução, onde serão apresentados os modelos de processo de negócio, constituídos pela inscrição inicial do aluno, controlo do percurso dos alunos, marcação de exames, controlo de pagamentos e controlo de assiduidade nas aulas práticas e nas aulas teóricas. Modelação SW(levantamento de requisitos, design e Modelação) e Modelação de Dados e Implementação da base de dados.

## Objetivos do projeto

O objetivo do projeto será, proporcionar aos membros de uma escola de condução, uma melhor organização dos processos envolvidos na manutenção e gestão da escola em questão.

## Atores

No nosso projeto consideramos dois atores:

Secretaria-A secretaria tem um dos papeis principais da escola de condução pois e responsável pela gestão dos alunos desde a inscrição até concluir a carta de condução.

Instrutor – O instrutor tem a função de gerir a assiduidade dos alunos nas aulas teóricas e práticas para mais tarde realizar a verificação para ir a exame.

# Apresentação do negócio

## Âmbito e enquadramento do projeto

A aplicação será direcionada para uso na secretaria da escola, onde todos os processos são geridos.

Numa primeira fase um aluno poderá dirigir-se á secretaria para se inscrever numa determinada categoria, facultando primeiro os seus dados pessoais, um atestado médico e por fim escolhendo qual o método como deseja pagar a carta de condução com o preço a variar entre categorias. Os dados serão depois armazenados numa base de dados e o aluno passa a ter a possibilidade de frequentar as aulas teóricas.

A seguinte componente do projeto é a gestão do progresso dos alunos, a secretaria neste caso pode submeter um aluno a exame teórico quando completa as 28 horas de aulas, caso o aluno seja reprovado, terá que pagar a nova inscrição e só aí poderá voltar a realizar o exame.

Quando aprovado, passa a frequentar as aulas práticas e ao finalizar as 32 horas de aulas obrigatórias, a secretaria submete um pedido á instituição de exames para o aluno realizar o exame. Ao receber a resposta do I.M.T. (Instituto da Mobilidade dos Transportes), a escola notifica novamente o aluno tal como no exame teórico e fica marcado para a data indicada pelo I.M.T.. Mais uma vez, se o aluno reprovar terá que pagar um determinado valor para repetir o exame, mas, se for aprovado termina e fica inserido numa base de dados como apto a conduzir veículos da categoria que lhe corresponde.

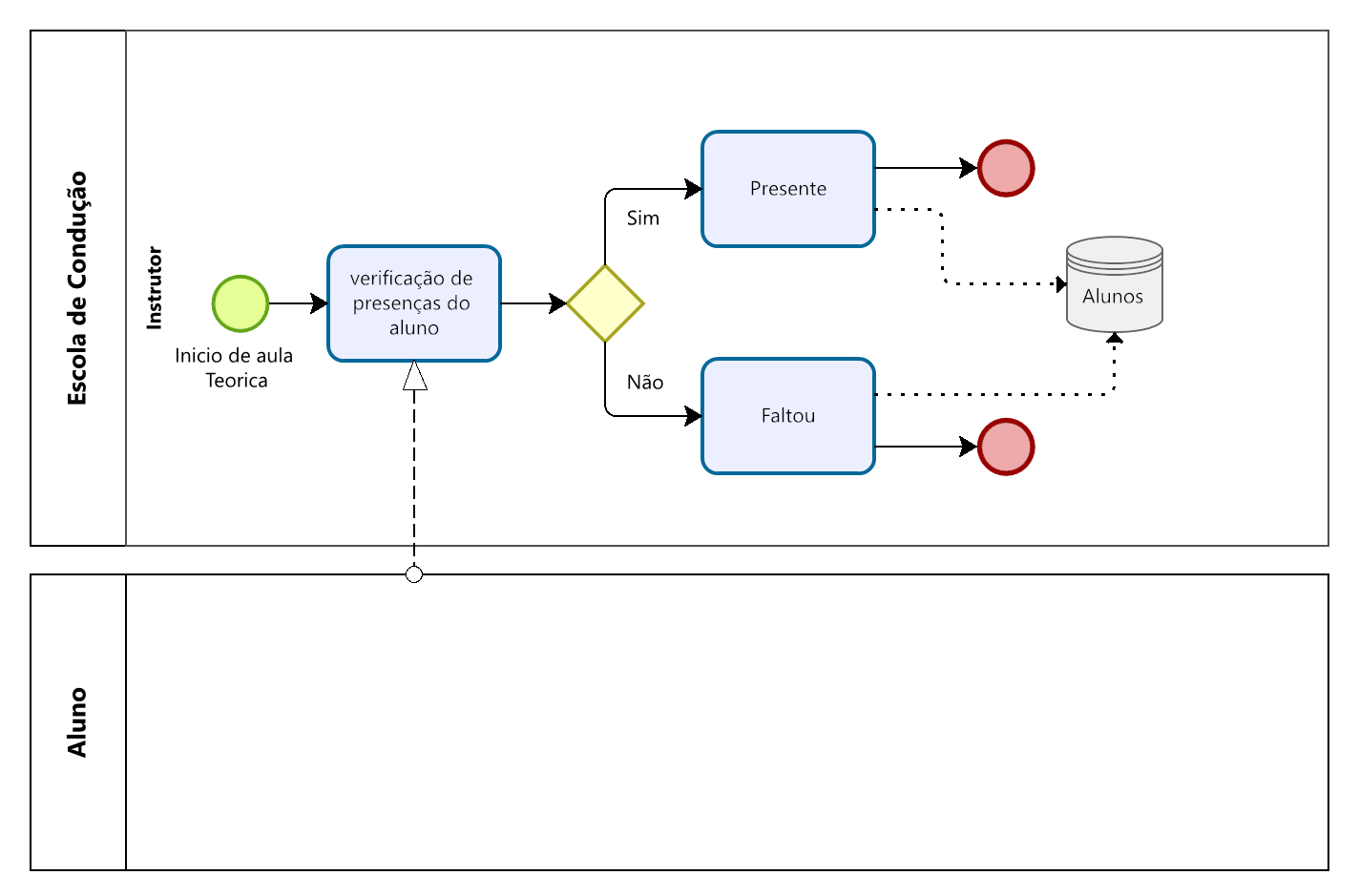
## Modelação dos Processos de negócio

## Processo 1 - Inscrição de um aluno

## Processo 2 - Inscrição a exame teórico

## Processo 3 - Inscrição a exame prático

## Processo 4 - Controlo de assiduidade aulas teóricas



## Processo 5 – Controlo de assiduidade aulas práticas

# Levantamento de Requisitos

## Tipos de utilizador

## Apresentação dos tipos de utilizador

A aplicação será usada pela secretaria e pelos instrutores para controlar o percurso dos alunos da escola.

## Ações de cada tipo de utilizador

A secretaria é a entidade principal com controlo total sobre a aplicação, pode inscrever alunos, manipular os dados, realizar pedidos de inscrição a exame, verificar presenças dos alunos e gerir os pagamentos dos alunos.

Os instrutores apenas poderão controlar a assiduidade dos alunos nas aulas, com o controlo da assiduidade depois será possivel avaliar se um aluno poderá ou não realizar os exames.

## Requisitos Funcionais

Como Secretaria, quero inscrever um aluno, para fazer a carta de conduçao de um veículo;

Como Secretaria, quero pedir atestado médico ao aluno, para garantir que pode ser inscrito na carta de condução;

Como Secretaria, quero registar pagamento de inscrição, para gerir faturas e pagamentos;

Como Secretaria, quero registar pagamentos restantes, para gerir faturas e pagamentos;

Como Secretaria, quero verificar a disponibilidade de um instrutor, para marcar aula de condução do aluno;

Como Instrutor, quero verificar a presença do aluno, para controlar assiduidade;

Como Secretaria, quero fazer um pedido de exame de codigo ao IMT, para o aluno concluir a parte teorica;

Como Secretaria, quero fazer um pedido de exame de condução ao IMT, para o aluno concluir a parte prática;

Como Secretaria, quero fazer um pedido de exame de codigo ao IMT, para o aluno repetir caso reprove no anterior.

## Requisitos não funcionais

Programa desenvolvido em java;

Base de dados desenvolvida em oracle;

Requer ligação á internet;

Desenvolvido para Windows e macOS;

Uso acessível;

Não requer computador de configurações elevadas.

# Design e Modelação

## Modelo de casos de uso



## Caso de uso: Inscrever aluno

|  |  |
| --- | --- |
| Id: | Caso de uso: **Inscrever aluno** |
| Ator principal: | Secretária. |
| Pré-condições: | Secretária está autenticada no sistema. |
| Pós-condições: | É criado novo aluno. |
| Cenário principal: | 1- Secretária recebe pedido de inscrição;  2- Secretária insere dados;  3- Sistema regista dados do aluno;  4- Secretária questiona acerca do método de pagamento;  5- Secretária insere tipo de pagamento;  6- Sistema regista método de pagamento; |
| Cenários alternativos: | 1- Aluno não fornece dados suficientes;  2- Aluno não fica inscrito. |

## Caso de uso: Gerir pagamento inicial

|  |  |
| --- | --- |
| Id: | Caso de uso: **Gerir pagamento inicial** |
| Ator principal: | Secretária. |
| Pré-condições: | Secretária está autenticada no sistema;  Aluno registado; |
| Pós-condições: | Aluno poderá começar a assistir a aulas. |
| Cenário principal: | 1- É finalizada inserção dos dados do aluno no sistema;  2- Secretária questiona aluno sobre qual método de pagamento deseja utilizar;  3- Secretária recebe pagamento;  4- É registado na base de dados que pagamento foi feito; |
| Cenários alternativos: | 1- Aluno não finaliza inscrição. |

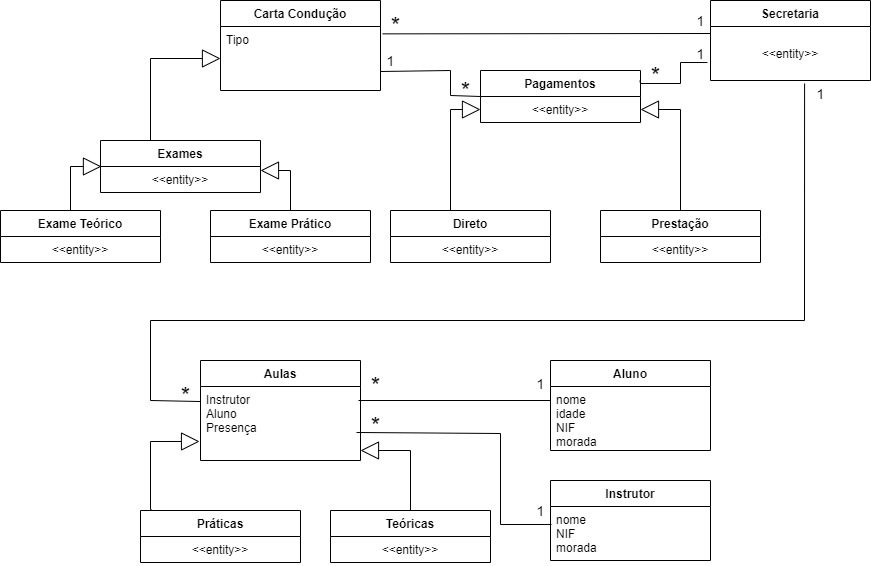
## Caso de uso: Marcar aulas práticas

|  |  |
| --- | --- |
| Id: | Caso de uso: **Marcar aulas práticas** |
| Ator principal: | Secretária, instrutor. |
| Pré-condições: | Secretária está autenticada no sistema;  Aluno ter sido aprovado á componente teórica; |
| Pós-condições: | Aluno fica apto a exame prático ao completar número minimo de horas em aula. |
| Cenário principal: | 1- Secretária verifica disponibilidade do instrutor no sistema;  2- Instrutor é notificado;  3- Secretária Informa aluno da marcação; |
| Cenários alternativos: | 1- Professor não tem disponibilidade. |

## Caso de uso: Controlar assiduidade

|  |  |
| --- | --- |
| Id: | Caso de uso: **Controlar assiduidade** |
| Ator principal: | Secretária, instrutor. |
| Pré-condições: | Secretária está autenticada no sistema;  Instrutor está autenticado no sistema;  Aluno está registado; |
| Pós-condições: |  |
| Cenário principal: | 1- Secretária ou instrutor selecionam o aluno;  2- Verificam presenças do aluno; |
| Cenários alternativos: | 1- Caso o aluno não tenha comparecido ás aulas num longo período de tempo é contactado. |

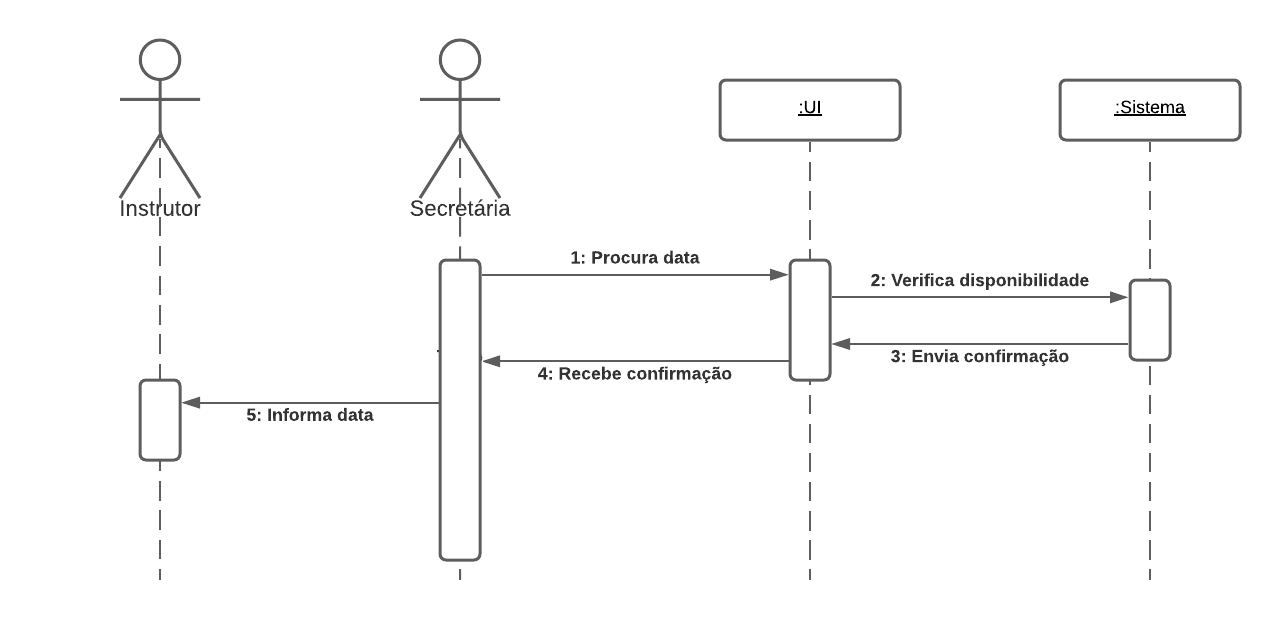
## Modelo de classes



## Diagrama de Sequência

## Inscrever Aluno

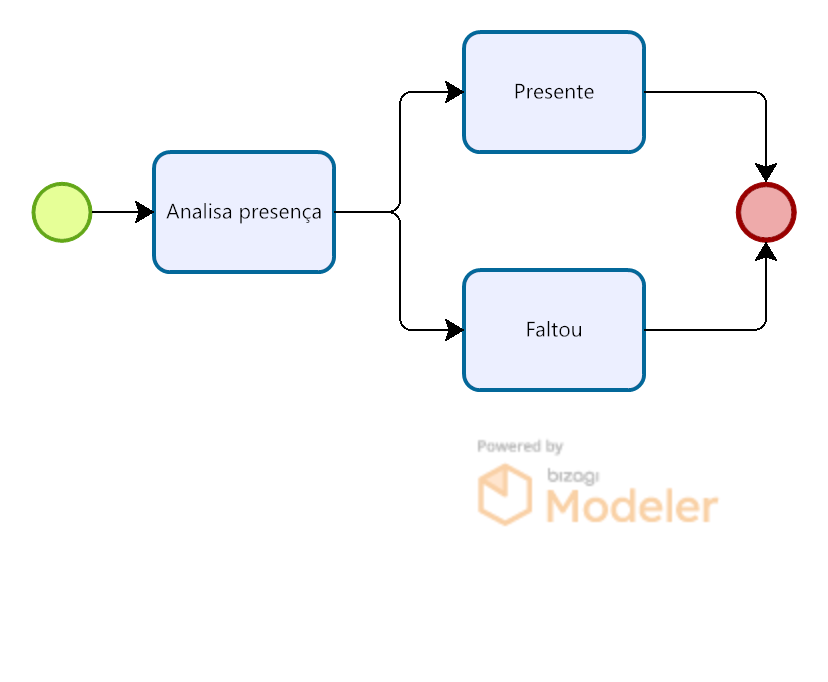
## Verificar Disponibilidade



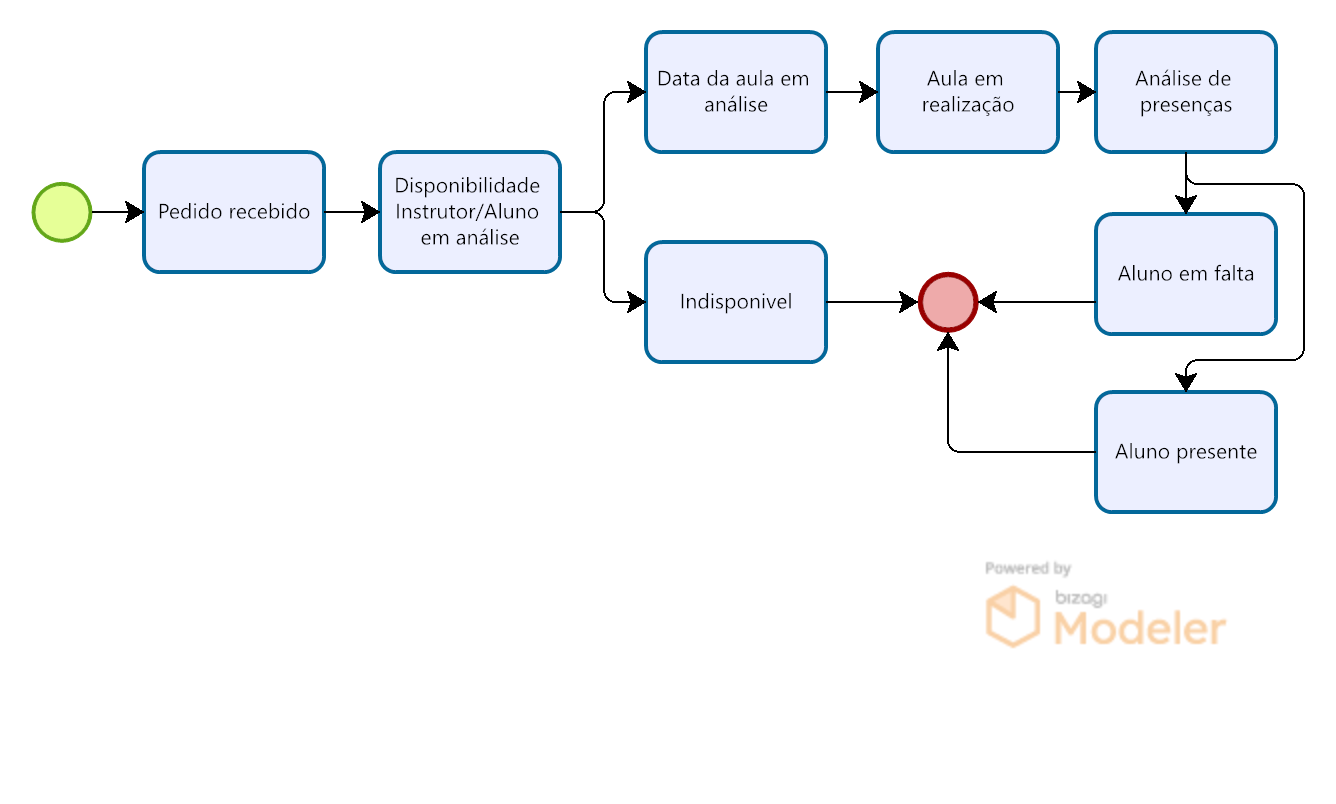
## Diagrama de Transição de Estados

## Inscrição Aluno

## Presenças nas Aulas Teóricas



## Presenças nas Aulas Práticas



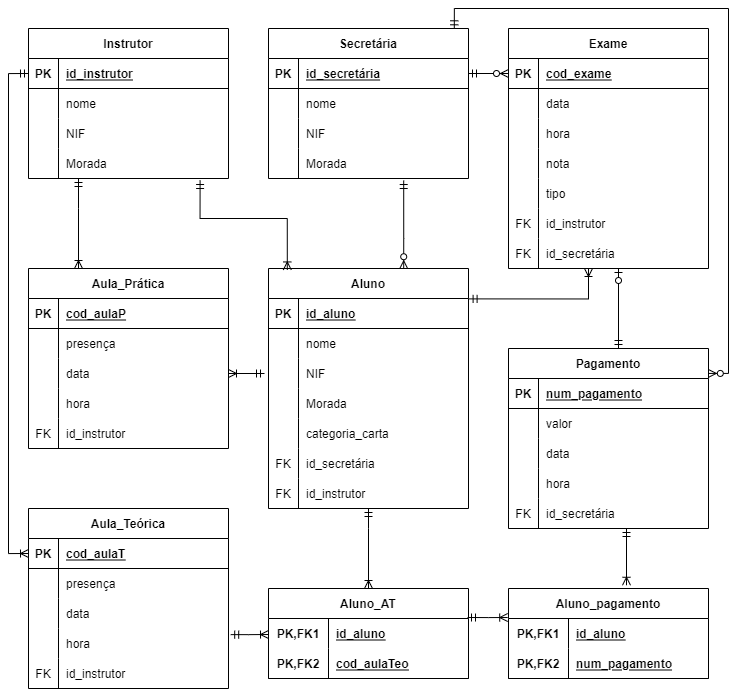
## Exame Teórico

## Exame Prático

# Implementação da BD

## Diagrama Entidades-Relacionamentos

## Modelo de tabelas



Instrutor (**idinstrutor**, nome, nif, morada);

Secretaria ( **idsecretaria**, nome, nif, morada);

Exame (**codexame**, data, hora, nota, tipo, **idsecretaria**);

Aula\_Prática (**cod\_aulaP**, presença, data, hora, **id\_instrutor**);

Aluno (**id\_aluno, nome**, NIF, Morada, categoria\_carta, **id\_secretaria**, **id\_instrutor**);

Pagamento (**num\_Pagamento**, valor, data, hora, **id\_secretaria**);

Aula\_Teórica (**cod\_aulaT**, presença, data, hora, **id\_instrutor**);

Aluno\_AT (**id\_aluno**, **cod\_aulaT**);

Aluno\_Pagamenro (**id\_aluno**, **num\_pagamento**);

**1º Forma Normal**

Uma relação está na 1ª forma normal quando:

* Não contém atributos compostos;
* Não contém atributos repetitivos (atributos multivalor);

Instrutor (**idinstrutor**, nome, nif, rua, porta, localidade, cPostal);

Secretária (**idsecretaria**, nome, nif, rua, porta, localidade, cPostal);

Exame (**codexame**, data, hora, nota, tipo, **idsecretaria**);

Aula\_Prática (**cod\_aulaP**, **idPresPra**, data, hora, **idinstrutor**);

PresençaPra (**idPresPra**, **id\_aluno**);

Aluno (**id\_aluno**, nome, NIF, rua, porta, localidade, cPostal**, idCategoria, id\_secretaria, id\_instrutor**);

Categoria\_carta ( **idCategoria**, nome);

Pagamento (**num\_Pagamento**, valor, data, hora**, id\_secretaria**);

Aula\_Teorica (**cod\_aulaT**, **idPresTeo**, data, hora**, id\_instrutor**);

PresençaTeo (**idPresTeo**, **id\_aluno**);

Aluno\_AT (**id\_aluno**, **cod\_aulaTeo**);

Aluno\_Pagamenro (**id\_aluno**, **num\_pagamento**);

**2ªForma Normal**

Uma relação está na 2ª forma normal quando:

* Está na 1FN;
* Todos os atributos não chave dependem por completo da totalidade da chave primária;

Instrutor (**idinstrutor**, nome, nif, rua, porta, localidade, cPostal);

Secretária (**idsecretaria**, nome, nif, rua, porta, localidade, cPostal);

Exame (**codexame**, data, hora, nota, tipo, **idsecretaria**);

Aula\_Prática (**cod\_aulaP**, **idPresPra**, data, hora, **idinstrutor**);

PresençaPra (**idPresPra**, **id\_aluno**);

Aluno (**id\_aluno**, nome, NIF, rua, porta, localidade, cPostal**, idCategoria, id\_secretaria, id\_instrutor**);

Categoria\_carta ( **idCategoria**, nome);

Pagamento (**num\_Pagamento**, valor, data, hora**, id\_secretaria**);

Aula\_Teorica (**cod\_aulaT**, **idPresTeo**, data, hora**, id\_instrutor**);

PresençaTeo (**idPresTeo**, **id\_aluno**);

Aluno\_AT (**id\_aluno**, **cod\_aulaTeo**);

Aluno\_Pagamenro (**id\_aluno**, **num\_pagamento**);

**3ª Forma Normal**

Uma relação está na 3ª forma normal quando:

* Está na 2FN;
* Não existem dependências funcionais entre atributos não chave (dependências transitivas), isto é:
  + Nenhum atributo não-chave depende por transitividade da chave primária
  + Os atributos não chave não dependem funcionalmente uns dos outros
  + Cada atributo deve depender apenas da chave primária

Instrutor( **idinstrutor**, nome, nif, **idMorada)**

Secreteria( idsecretaria, nome, nif, **idMorada**)

Exame(codexame, data, hora, nota, tipo, **idsecretaria)**

Aula\_Pratica(cod\_aulaP, **idPresPra**, data, hora, **id\_instrutor)**

presençaPra(idPresPra, **id\_aluno)**

Aluno(id\_aluno, nome, NIF**, idMorada, idCategoria, id\_secretaria, id\_instrutor)**

categoria\_carta( idCategoria, nome)

Pagamento(num\_Pagamento, valor, data, hora**, id\_secretaria)**

Aula\_Teorica(cod\_aulaT, **idPresTeo**, data, hora, **id\_instrutor)**

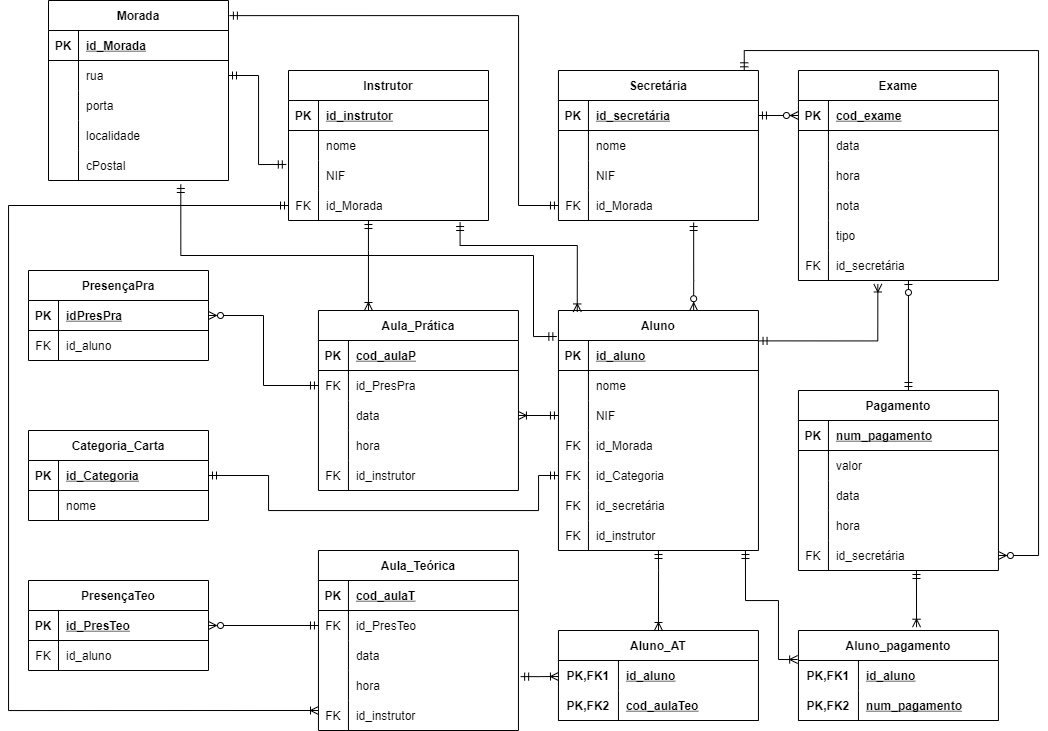
presençaTeo(idPresTeo, **id\_aluno)**

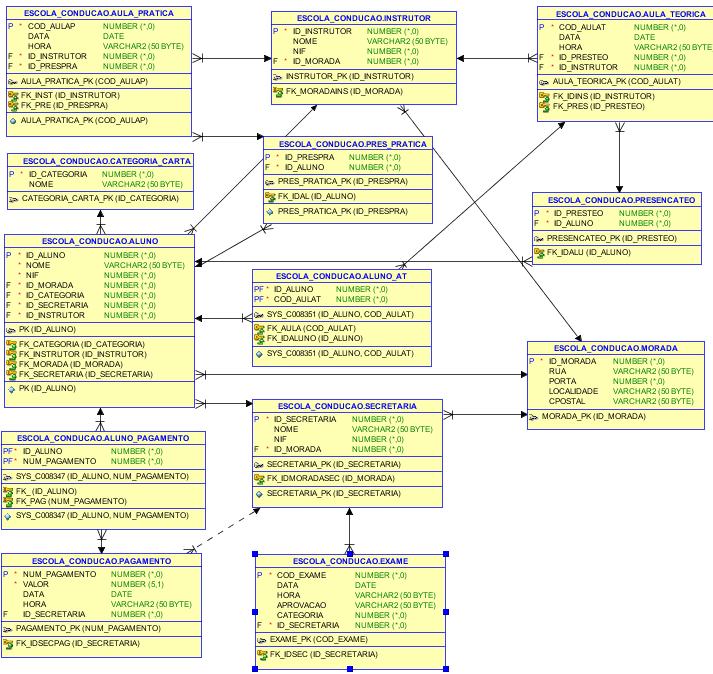
PresençaTeo (**idPresTeo**, **id\_aluno**)

Aluno\_AT (**id\_aluno**, **cod\_aulaTeo**)

Aluno\_Pagamenro (**id\_aluno**, **num\_pagamento**)

## 5.4 Modelo Final



****

## Código SQL

## Criação de tabelas

Uma imagem com texto, mesa

Descrição gerada automaticamente

CREATE TABLE categoria\_carta

(id\_categoria number(5) PRIMARY KEY NOT NULL,

nome varchar(50));

Uma imagem com mesa

Descrição gerada automaticamente

CREATE TABLE pagamento

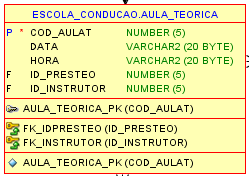
(num\_pagamento number(5) PRIMARY KEY NOT NULL,

valor number(7,2) NOT NULL

data date,

hora time,

FOREIGN KEY (id\_secretaria) REFERENCES secretária (id\_secretaria) );



CREATE TABLE Aula\_Teorica

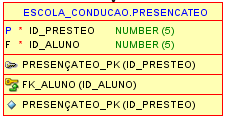
(cod\_aulaT number(5) PRIMARY KEY NOT NULL,

data date

hora time

FOREIGN KEY (Id\_presTeo) REFERENCES PresençaTeo (Id\_PresTeo)

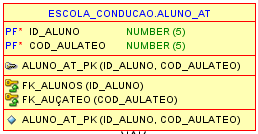
FOREIGN KEY (id\_instrutor) REFERENCES Instrutor (id\_instrutor) );



CREATE TABLE presençaTeo

(id\_presTeo number(5) PRIMARY KEY NOT NULL,

FOREIGN KEY (id\_presTeo) REFERENCES Aula\_Teorica (id\_aluno));



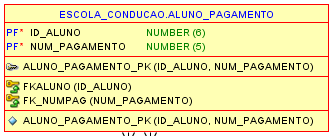
CREATE TABLE Aluno\_AT

(id\_aluno number(5) PRIMARY KEY NOT NULL,

cod\_aulaTeo number(5) PRIMARY KEY NOT NULL,

FOREIGN KEY (id\_aluno) REFERENCES Aluno (id\_aluno),

FOREIGN KEY (cod\_aulaTeo) REFERENCES Aula\_teorica (cod\_aulaT));



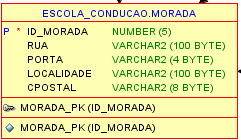
CREATE TABLE Aluno\_pagamento (

id\_aluno number(5) PRIMARY KEY NOT NULL,

num\_pagamento number(5) PRIMARY KEY NOT NULL,

FOREIGN KEY (id\_aluno) REFERENCES Aluno (id\_aluno),

FOREIGN KEY (num\_pagamento) REFERENCES pagamento (num\_pagamento));



CREATE TABLE Morada (

id\_morada number(5) PRIMARY KEY NOT NULL,

rua varchar2(100),

porta number(4),

localidade varchar2(100),

cPostal varchar2(8));

Uma imagem com mesa

Descrição gerada automaticamente

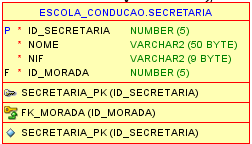
CREATE TABLE Instrutor

(idInstrutor number(5) PRIMARY KEY NOT NULL,

nome varchar(50) NOT NULL,

NIF number(9) NOT NULL,

FOREIGN KEY (id\_Morada) REFERENCES Morada (id\_Morada));



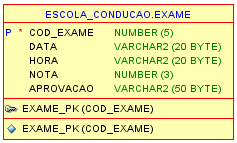
CREATE TABLE Secretaria (

idSecretaria number(5) PRIMARY KEY NOT NULL,

nome varchar(50) NOT NULL,

NIF number(9) NOT NULL,

FOREIGN KEY (id\_Morada) REFERENCES Morada (id\_Morada));



CREATE TABLE Exame

(Cod\_exame number(5) PRIMARY KEY NOT NULL,

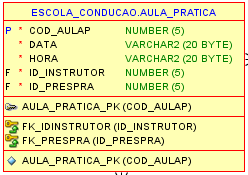
data date,

hora time,

nota number(3),

tipo varchar(8),

FOREIGN KEY (id\_secretaria) REFERENCES Secretaria (id\_secretaria));



CREATE TABLE Aula\_Pratica

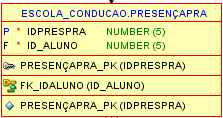
(cod\_aulaP number(5) PRIMARY KEY NOT NULL,

data date NOT NULL,

hora time NOT NULL,

FOREIGN KEY (id\_instrutor) REFERENCES Instrutor (id\_instrutor),

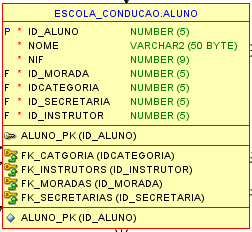
FOREIGN KEY (id\_PresPra) REFERENCES PresençaPra (id\_PresPra));



CREATE TABLE PresencaPra

(idPresPra NUMBER (5) PRIMARY KEY NOT NULL,

FOREIGN KEY (Aluno) REFERENCES Aluno (id\_aluno));



CREATE TABLE Aluno

(id\_aluno number(5) PRIMARY KEY NOT NULL,

nome varchar(50) NOT NULL,

NIF number(9) NOT NULL,

FOREIGN KEY (id\_Morada) REFERENCES Morada (id\_Morada),

FOREIGN KEY (idCategoria) REFERENCES categoria\_carta (idCategoria),

FOREIGN KEY (id\_secretaria) REFERENCES Secretaria (id\_secretaria),

FOREIGN KEY (id\_instrutor) REFERENCES Instrutor (id\_instrutor));

# 5.3.2. Inserts

**Exame**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."EXAME" (COD\_EXAME, DATA, HORA, APROVACAO, CATEGORIA, ID\_SECRETARIA) VALUES ('65846', TO\_DATE('02-10-2011', 'DD-MM-YYYY'), '11:49', 'true', '5', '00010');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."EXAME" (COD\_EXAME, DATA, HORA, APROVACAO, CATEGORIA, ID\_SECRETARIA) VALUES ('84922', TO\_DATE('06-09-2014', 'DD-MM-YYYY'), '12:53', 'true', '2', '00011');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."EXAME" (COD\_EXAME, DATA, HORA, APROVACAO, CATEGORIA, ID\_SECRETARIA) VALUES ('79813', TO\_DATE('28-03-2011', 'DD-MM-YYYY'), '14:36', 'true', '3', '00012');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."EXAME" (COD\_EXAME, DATA, HORA, APROVACAO, CATEGORIA, ID\_SECRETARIA) VALUES ('89332', TO\_DATE('16-09-2020', 'DD-MM-YYYY'), '11:29', 'true', '3', '00013');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."EXAME" (COD\_EXAME, DATA, HORA, APROVACAO, CATEGORIA, ID\_SECRETARIA) VALUES ('32835', TO\_DATE('31-01-2011', 'DD-MM-YYYY'), '9:14', 'false', '2', '00014');

**Aula Prática**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_PRATICA" (COD\_AULAP, DATA, HORA, ID\_INSTRUTOR, ID\_PRESPRA) VALUES ('26358', TO\_DATE('27-04-2010', 'DD-MM-YYYY'), '16:40', '00101', '78346');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_PRATICA" (COD\_AULAP, DATA, HORA, ID\_INSTRUTOR, ID\_PRESPRA) VALUES ('37778', TO\_DATE('16-11-2010', 'DD-MM-YYYY'), '08:10', '00102', '48761');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_PRATICA" (COD\_AULAP, DATA, HORA, ID\_INSTRUTOR, ID\_PRESPRA) VALUES ('89418', TO\_DATE('25-01-2014', 'DD-MM-YYYY'), '16:17', '00103', '40818');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_PRATICA" (COD\_AULAP, DATA, HORA, ID\_INSTRUTOR, ID\_PRESPRA) VALUES ('57401', TO\_DATE('10-08-2019', 'DD-MM-YYYY'), '12:21', '00104', '89861');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_PRATICA" (COD\_AULAP, DATA, HORA, ID\_INSTRUTOR, ID\_PRESPRA) VALUES ('68746', TO\_DATE('17-03-2018', 'DD-MM-YYYY'), '11:25', '00105', '82864');

**Aula Teórica**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_TEORICA" (COD\_AULAT, DATA, HORA, ID\_PRESTEO, ID\_INSTRUTOR) VALUES ('001', TO\_DATE('02-07-2010', 'DD-MM-YYYY'), '16:05', '91765', '00101');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_TEORICA" (COD\_AULAT, DATA, HORA, ID\_PRESTEO, ID\_INSTRUTOR) VALUES ('002', TO\_DATE('04-02-2012', 'DD-MM-YYYY'), '9:34', '19551', '00102');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_TEORICA" (COD\_AULAT, DATA, HORA, ID\_PRESTEO, ID\_INSTRUTOR) VALUES ('003', TO\_DATE('06-12-2014', 'DD-MM-YYYY'), '10:10', '46627', '00103');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_TEORICA" (COD\_AULAT, DATA, HORA, ID\_PRESTEO, ID\_INSTRUTOR) VALUES ('004', TO\_DATE('08-11-2011', 'DD-MM-YYYY'), '9:56', '24961', '00104');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."AULA\_TEORICA" (COD\_AULAT, DATA, HORA, ID\_PRESTEO, ID\_INSTRUTOR) VALUES ('005', TO\_DATE('10-10-2013', 'DD-MM-YYYY'), '14:40', '65752', '00105');

**Aluno AT**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_AT" (ID\_ALUNO, COD\_AULAT) VALUES ('01001', '001');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_AT" (ID\_ALUNO, COD\_AULAT) VALUES ('01002', '002');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_AT" (ID\_ALUNO, COD\_AULAT) VALUES ('01003', '003');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_AT" (ID\_ALUNO, COD\_AULAT) VALUES ('01004', '004');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_AT" (ID\_ALUNO, COD\_AULAT) VALUES ('01005', '005');

**Presença Prática**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."PRES\_PRATICA" (id\_presPra, id\_aluno) VALUES (78346, 01001);

(2) INSERT INTO "ESCOLA\_CONDUCAO"."PRES\_PRATICA" (id\_presPra, id\_aluno) VALUES (48761, 01002);

(3) INSERT INTO "ESCOLA\_CONDUCAO"."PRES\_PRATICA" (id\_presPra, id\_aluno) VALUES (40818, 01003);

(4) INSERT INTO "ESCOLA\_CONDUCAO"."PRES\_PRATICA" (id\_presPra, id\_aluno) VALUES (89861, 01004);

(5) INSERT INTO "ESCOLA\_CONDUCAO"."PRES\_PRATICA" (id\_presPra, id\_aluno) VALUES (82864, 01005);

**Presença Teórica**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."PRESENCATEO" (id\_presTeo, id\_aluno) VALUES (91765, 01001);

(2) INSERT INTO "ESCOLA\_CONDUCAO"."PRESENCATEO" (id\_presTeo, id\_aluno) VALUES (19551, 01002);

(3) INSERT INTO "ESCOLA\_CONDUCAO"."PRESENCATEO" (id\_presTeo, id\_aluno) VALUES (46627, 01003);

(4) INSERT INTO "ESCOLA\_CONDUCAO"."PRESENCATEO" (id\_presTeo, id\_aluno) VALUES (24961, 01004);

(5) INSERT INTO "ESCOLA\_CONDUCAO"."PRESENCATEO" (id\_presTeo, id\_aluno) VALUES (65752, 01005);

**Morada**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00006, '2192 Ohio Trail', 268, 'Lisboa', '4912-135');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00007, '2 Memorial Crossing', 363, 'Lisboa', '4910-335');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00008, '34068 Chive Hill', 323, 'Porto', '4810-765');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00009, '5021 Sommers Crossing', 298, 'Viana do Castelo', '4712-255');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00010, '9317 Summer Ridge Lane', 398, 'Braga', '4611-315');

(6) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00001, '2192 Ohio Trail', 121, 'Braga', '4633-455');

(7) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00002, '2 Memorial Crossing', 234, 'Lisboa', '4940-135');

(8) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00003, '34068 Chive Hill', 292, 'Lisboa', '4950-035');

(9) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00004, '5021 Sommers Crossing', 248, 'Braga', '4660-235');

(10) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00005, '9317 Summer Ridge Lane', 521, 'Braga', '4610-335');

(11) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00011, '2192 Ohio Trail', 456, 'Lisboa', '4914-123');

(12) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00012, '2 Memorial Crossing', 654, 'Viana do Castelo', '4734-321');

(13) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00013, '34068 Chive Hill', 218, 'Lisboa', '4934-234');

(14) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00014, '5021 Sommers Crossing', 275, 'Viana do Castelo', '4778-534');

(15) INSERT INTO "ESCOLA\_CONDUCAO"."MORADA" (id\_morada, rua, porta, localidade, cPostal) VALUES (00015, '9317 Summer Ridge Lane', 241, 'Lisboa', '4987-556');

**Instrutor**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."INSTRUTOR" (id\_instrutor, nome, NIF, id\_morada) VALUES (00101, 'Vinson Horley', 257123456, 00006);

(2) INSERT INTO "ESCOLA\_CONDUCAO"."INSTRUTOR" (id\_instrutor, nome, NIF, id\_morada) VALUES (00102, 'Toni Cancutt', 386654321, 00007);

(3) INSERT INTO "ESCOLA\_CONDUCAO"."INSTRUTOR" (id\_instrutor, nome, NIF, id\_morada) VALUES (00103, 'Emmalynn Paris', 235987765, 00008);

(4) INSERT INTO "ESCOLA\_CONDUCAO"."INSTRUTOR" (id\_instrutor, nome, NIF, id\_morada) VALUES (00104, 'Conan Wein', 315756597, 00009);

(5) INSERT INTO "ESCOLA\_CONDUCAO"."INSTRUTOR" (id\_instrutor, nome, NIF, id\_morada) VALUES (00105, 'Noe Antonovic', 205134575, 00010);

**Secretaria**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."SECRETARIA" (id\_secretaria, nome, NIF, id\_morada) VALUES (00010, 'Sheila-kathryn Kalb', 674603758, 00001);

(2) INSERT INTO "ESCOLA\_CONDUCAO"."SECRETARIA" (id\_secretaria, nome, NIF, id\_morada) VALUES (00011, 'Meredeth de Leon', 937571669, 00002);

(3) INSERT INTO "ESCOLA\_CONDUCAO"."SECRETARIA" (id\_secretaria, nome, NIF, id\_morada) VALUES (00012, 'Dusty Feldhammer', 654502551, 00003);

(4) INSERT INTO "ESCOLA\_CONDUCAO"."SECRETARIA" (id\_secretaria, nome, NIF, id\_morada) VALUES (00013, 'Jase Locksley', 722641814, 00004);

(5) INSERT INTO "ESCOLA\_CONDUCAO"."SECRETARIA" (id\_secretaria, nome, NIF, id\_morada) VALUES (00014, 'Flem Starbeck', 202068856, 00005);

**Aluno**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO" (ID\_ALUNO, NOME, NIF, ID\_MORADA, ID\_CATEGORIA, ID\_SECRETARIA, ID\_INSTRUTOR) VALUES ('01001', 'Judith Deakins', '378611932', '00011', '5', '00010', '00101');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO" (ID\_ALUNO, NOME, NIF, ID\_MORADA, ID\_CATEGORIA, ID\_SECRETARIA, ID\_INSTRUTOR) VALUES ('01002', 'Rycca Szimoni', '532940661', '00012', '2', '00011', '00102');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO" (ID\_ALUNO, NOME, NIF, ID\_MORADA, ID\_CATEGORIA, ID\_SECRETARIA, ID\_INSTRUTOR) VALUES ('01003', 'Carter Durbyn', '934396896', '00013', '3', '00012', '00103');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO" (ID\_ALUNO, NOME, NIF, ID\_MORADA, ID\_CATEGORIA, ID\_SECRETARIA, ID\_INSTRUTOR) VALUES ('01004', 'Nate Ibel', '917514795', '00013', '3', '00013', '00104');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO" (ID\_ALUNO, NOME, NIF, ID\_MORADA, ID\_CATEGORIA, ID\_SECRETARIA, ID\_INSTRUTOR) VALUES ('01005', 'Marthena Wadham', '144613164', '00013', '2', '00014', '00105');

**Categoria carta**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('1', 'A');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('2', 'A1');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('3', 'A2');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('4', 'AM');

(5) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('5', 'B');

(6) INSERT INTO "ESCOLA\_CONDUCAO"."CATEGORIA\_CARTA" (id\_categoria, nome) VALUES ('6', 'B1');

**Pagamento**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."PAGAMENTO" (NUM\_PAGAMENTO, VALOR, DATA, HORA, ID\_SECRETARIA) VALUES ('0002', '1337.4', TO\_DATE('2011-07-11', 'YYYY-MM-DD'), '17:25', '00011');

(2) INSERT INTO "ESCOLA\_CONDUCAO"."PAGAMENTO" (NUM\_PAGAMENTO, VALOR, DATA, HORA, ID\_SECRETARIA) VALUES ('0003', '1033.3', TO\_DATE('2012-12-07', 'YYYY-MM-DD'), '10:35', '00012');

(3) INSERT INTO "ESCOLA\_CONDUCAO"."PAGAMENTO" (NUM\_PAGAMENTO, VALOR, DATA, HORA, ID\_SECRETARIA) VALUES ('0004', '1742.6', TO\_DATE('2013-09-21', 'YYYY-MM-DD'), '10:10', '00013');

(4) INSERT INTO "ESCOLA\_CONDUCAO"."PAGAMENTO" (NUM\_PAGAMENTO, VALOR, DATA, HORA, ID\_SECRETARIA) VALUES ('0005', '1726.0', TO\_DATE('2014-06-25', 'YYYY-MM-DD'), '12:25', '00014');

**Aluno Pagamento**

(1) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_PAGAMENTO" (id\_aluno, num\_pagamento) VALUES (01001, 0001);

(2) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_PAGAMENTO" (id\_aluno, num\_pagamento) VALUES (01002, 0002);

(3) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_PAGAMENTO" (id\_aluno, num\_pagamento) VALUES (01003, 0003);

(4) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_PAGAMENTO" (id\_aluno, num\_pagamento) VALUES (01004, 0004);

(5) INSERT INTO "ESCOLA\_CONDUCAO"."ALUNO\_PAGAMENTO" (id\_aluno, num\_pagamento) VALUES (01005, 0005);

# 5.3.3. Selects

**(1) Quantos utilizadores vivem em Lisboa:**

SELECT COUNT (id\_morada) AS LISBOETAS

FROM morada

WHERE localidade = 'Lisboa';

**(2)** **Quantos alunos estão inscritos na categoria 2:**

SELECT COUNT (id\_aluno) AS CAT2

FROM aluno

WHERE id\_categoria = 2;

**(3)** **Informação sobre todos os utilizadores inscritos na categoria 3:**

SELECT id\_aluno, nome, id\_categoria

FROM aluno

WHERE id\_categoria = 3;

**(4)** **Quantidade de utilizadores com NIF:**

SELECT COUNT (id\_aluno) AS ComNIF

FROM aluno

WHERE nif IS NOT NULL;

**(5)** **Informação sobre os alunos que realizaram exame de categoria 5:**

SELECT a.id\_aluno, a.nome, e.categoria

FROM aluno a, exame e

WHERE a.id\_aluno = e.id\_aluno

AND e.categoria = 5;

# Conclusão e Trabalho Futuro

No desenvolvimento deste projeto surgiram algumas dificuldades em maior e menor grau, nos 3 campos da resolução deste projeto: Modelação de Negócio, Modelação SW e implementação na base de dados.

Após termos entregue a parte da Modelação de Negócio chegamos a conclusão que podíamos ainda acrescentar o controlo de assiduidade nas aulas práticas e nas aulas teóricas pois completaria essa parte do projeto. Continuando para modelação de SW sentimos algumas dificuldades no desenvolvimento do diagrama de sequência.

Na implementação da base de dados tivemos algumas dificuldades mais especificamente na inserção de dados na base de dados e na criação dos selects.

Por fim, conseguimos atingir o objetivo que tinhamos e sabemos que ainda há muito espaço para melhorias que podem ser feitas neste trabalho. Assim com este projeto conseguimos perceber que um engenheiro informático não se trata só de programadores mas sim de um vasto leque de áreas de planeamento e execução.