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

**Licenciatura em Engenharia Informática**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Projeto I

Relatório de projeto

Gestão de um ginásio

Alexandre Santos - 24585

Sofia Sousa - 23435

2022/2023

Sumário

[1. Introdução 4](#_Toc128477251)

[1.1 Apresentação do tema 4](#_Toc128477252)

[1.2 Objetivos do projeto 4](#_Toc128477253)

[2. Apresentação do negócio 4](#_Toc128477254)

[2.1 Âmbito e enquadramento do projeto 4](#_Toc128477255)

[2.2 Modelação dos Processos de negócio 4](#_Toc128477256)

[2.2.1 Processo 1 - Inscrição de um cliente 4](#_Toc128477257)

[2.2.2 Processo 2 - Consulta de avaliação física com Instrutor 5](#_Toc128477258)

[2.2.3 Processo 3 - Consulta de nutrição com Nutricionista 5](#_Toc128477259)

[2.2.4 Processo 4 – Inscrição em aula de grupo 5](#_Toc128477260)

[2.2.5 Processo 5 – Anular inscrição no ginásio 6](#_Toc128477261)

[3. Levantamento de Requisitos 7](#_Toc128477262)

[3.1 Tipos de utilizador 7](#_Toc128477263)

[3.1.1 Apresentação dos tipos de utilizador 7](#_Toc128477264)

[3.1.2 Ações de cada tipo de utilizador 7](#_Toc128477265)

[3.2 Requisitos Funcionais 7](#_Toc128477266)

[3.3 Requisitos não funcionais 8](#_Toc128477267)

[4. Design e Modelação 9](#_Toc128477268)

[4.1 Modelo de casos de uso 9](#_Toc128477269)

[4.1.1 Caso de uso: Inscrever cliente 10](#_Toc128477270)

[4.1.2 Caso de uso: Marcar avaliação física 10](#_Toc128477271)

[4.1.3 Caso de uso: Marcar consulta de nutrição 11](#_Toc128477272)

[4.1.4 Caso de uso: Marcar aula de grupo 11](#_Toc128477273)

[4.2 Modelo de classes 12](#_Toc128477274)

[4.3 Diagrama de Sequência 13](#_Toc128477275)

[4.3.1 Inscrever Aluno 13](#_Toc128477276)

[4.3.2 Verificar Disponibilidade 13](#_Toc128477277)

[4.4 Diagrama de Transição de Estados 14](#_Toc128477278)

[4.4.1 Inscrição Aluno 14](#_Toc128477279)

[4.4.2 Presenças nas Aulas Teóricas 14](#_Toc128477280)

[4.4.3 Presenças nas Aulas Práticas 15](#_Toc128477281)

[4.4.4 Exame Teórico 15](#_Toc128477282)

[4.4.5 Exame Prático 16](#_Toc128477283)

[5. Implementação da BD 17](#_Toc128477284)

[5.1 Diagrama Entidades-Relacionamentos 17](#_Toc128477285)

[5.2 Modelo de tabelas 17](#_Toc128477286)

[5.4 Modelo Final 20](#_Toc128477287)

[5.3 Código SQL 21](#_Toc128477288)

[5.3.1 Criação de tabelas 21](#_Toc128477289)

[5.3.2. Inserts 28](#_Toc128477290)

[5.3.3. Selects 34](#_Toc128477291)

[6. Conclusão e Trabalho Futuro 35](#_Toc128477292)

# Introdução

## Apresentação do tema

Foi-nos proposto pelos docentes da Unidade Curricular de Projeto 2 a continuação do desenvolvimento de uma aplicação desktop e web com o objetivo de ajudar a gerir um ginásio.

Ginásios são muito procurados devido à acessibilidade de serviços oferecidos como a ajuda de especialistas no alcance dos objetivos de cada um dos seus clientes.

## Objetivos do projeto

A aplicação a desenvolver tem como objetivo proporcionar aos seus utilizadores, um maior controlo das tarefas de gestão existentes na instituição. Os membros com acesso á aplicação desktop serão rececionistas, nutricionistas e instrutores. Os clientes terão acesso através de web.

# Apresentação do negócio

## Âmbito e enquadramento do projeto

Alimentação e atividade física são áreas fundamentais para manter um corpo e mente saudável.

No entanto, atualmente é bastante fácil estarmos sobrecarregados de informação, por vezes incorreta, que está sempre disponível nas nossas mãos, levando pessoas a criarem hábitos que funcionam mais como obstáculos do que ferramentas que poderiam ajudar qualquer um alcançar os seus objetivos.

Nos últimos anos, esses assuntos, por mais intimidantes que possam ser para alguns, têm visto um aumento de interesse por parte do público.

Este projeto visa simplificar esta área de conhecimento, oferecendo a clientes acesso a uma plataforma onde possam aprender e pôr em prática, cada vez mais de forma autónoma, atividade física e uma dieta equilibrada e saudável, ajudando a atingir os seus objetivos de forma mais eficaz.

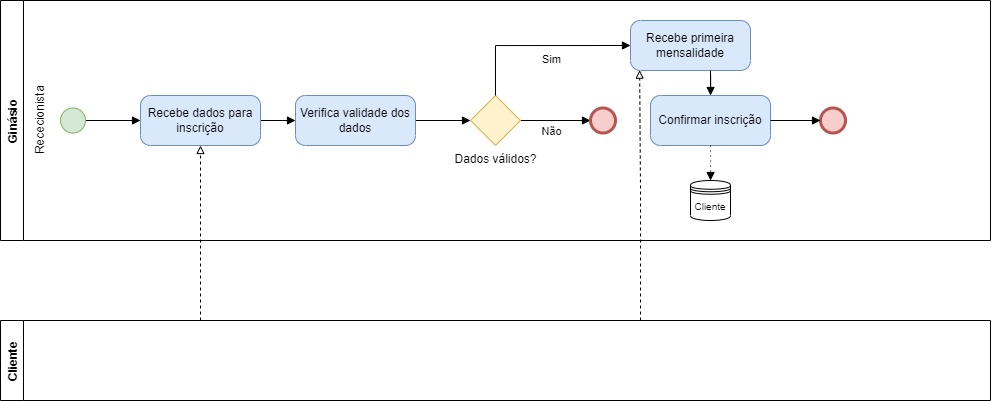
Para isso, instrutores e nutricionistas terão acesso a fichas de clientes, criadas e complementadas ao longo das suas sessões de treino e consultas, para determinar a melhor solução para qualquer dúvida que lhes possa vir a ser colocada.

Qualquer especialista irá concordar que um bom plano de exercício não sobrevive a um péssimo plano alimentar, e que o caminho de uma pessoa para atingir os seus objetivos de condicionamento físico depende de 80-90% da sua dieta.

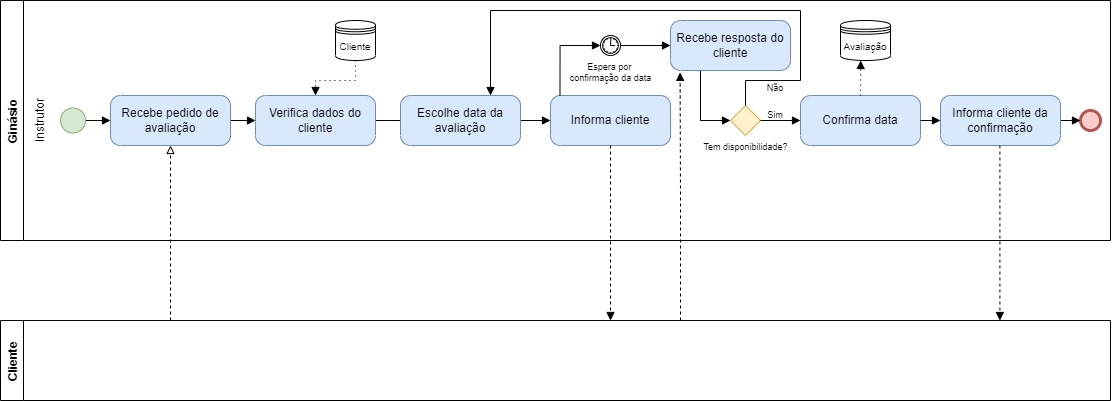
Por isso, o nosso serviço zelará não só pela rotina de exercícios do cliente, mas também pela sua rotina alimentar, dando-lhe a liberdade de escolha entre planos de exercício e alimentares personalizados feitos com a ajuda de especialistas que estarão sempre disponíveis aos clientes.

## Modelação dos Processos de negócio

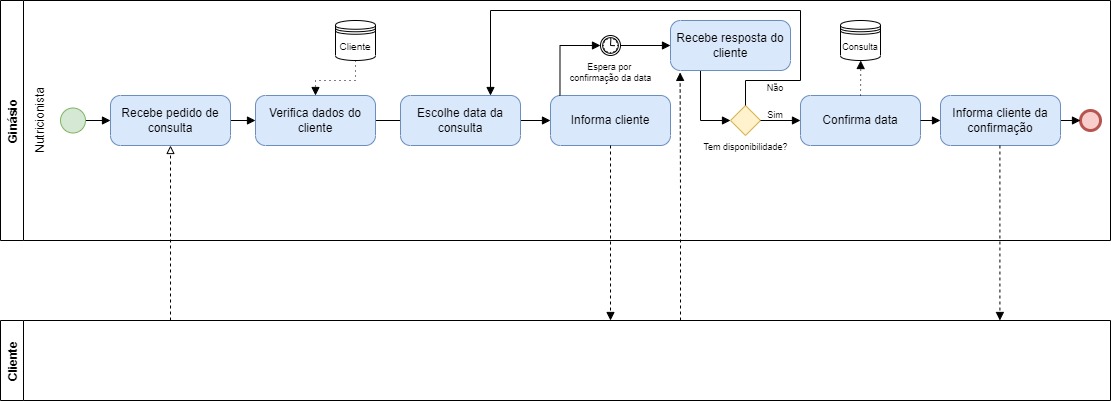
## Processo 1 - Inscrição de um cliente



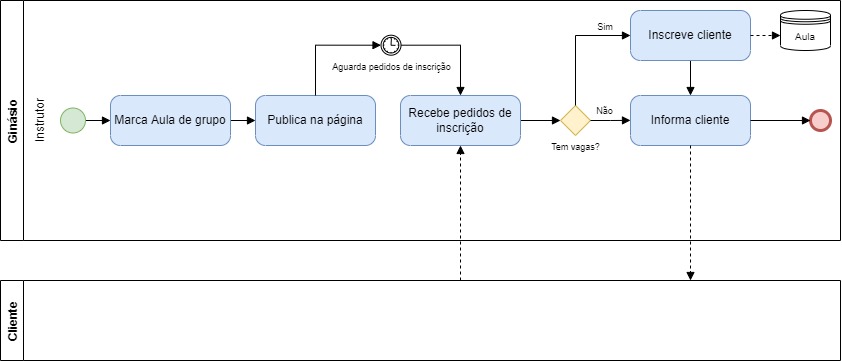
## Processo 2 - Consulta de avaliação física com Instrutor



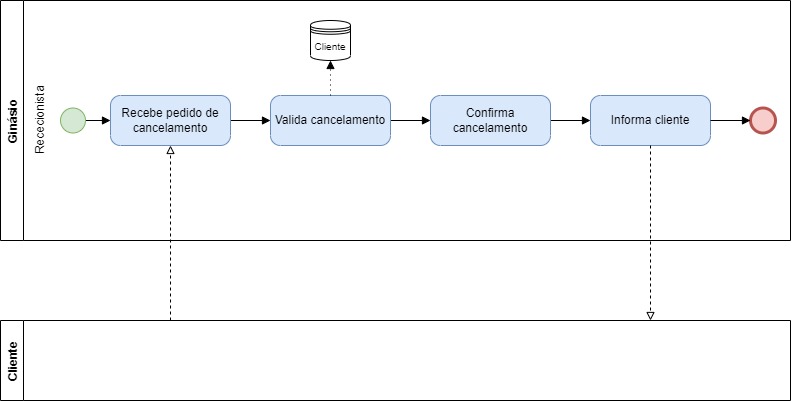
## Processo 3 - Consulta de nutrição com Nutricionista



## Processo 4 – Inscrição em aula de grupo



## Processo 5 – Anular inscrição no ginásio



# Levantamento de Requisitos

## Tipos de utilizador

## Apresentação dos tipos de utilizador

A versão desktop será usada pela Rececionista, Instrutor e Nutricionista. A versão web será usada pelo Cliente.

## Ações de cada tipo de utilizador

O Rececionista irá inscrever um cliente no ginásio, anular a inscrição do cliente e gerir pagamentos realizados pelos clientes.

O Nutricionista irá realizar consultas de acompanhamento alimentar com o cliente, tal como marcar a próxima consulta, tendo acesso aos relatórios passados.

O Instrutor irá realizar consultas de avaliação física com o cliente, tal como marcar a próxima consulta, sempre acedendo aos relatórios passados, também dando aulas de grupo.

O Cliente terá acesso a todos os dados do mesmo, realizar pagamentos, submeter pedidos para consultas e inscrever-se em aulas de grupo.

## Requisitos Funcionais

Como Rececionista quero inscrever um cliente no ginásio para o cliente ter acesso às instalações.

Como Rececionista quero confirmar um pagamento do cliente para o cliente ter as mensalidades em dia.

Como Rececionista quero anular a inscrição ao cliente para o cliente não estar fidelizado quando perde o interesse.

Como Cliente quero marcar uma consulta com nutricionista para ter um acompanhamento alimentar.

Como Nutricionista quero realizar uma consulta com o cliente para oferecer um acompanhamento alimentar.

Como Nutricionista quero realizar um relatório da consulta para continuar o acompanhamento alimentar.

Como Nutricionista quero ter acesso aos relatórios passados do cliente para analisar o progresso do cliente.

Como Instrutor quero realizar uma avaliação física com o cliente para acompanhar o seu estado físico.

Como Instrutor quero realizar a aula de grupo para os clientes terem uma aula de exercício físico em grupo.

Como Utilizador quero conseguir fazer login para poder ter acesso à aplicação.

## Requisitos não funcionais

Programa desenvolvido em java;

Base de dados desenvolvida em postgres;

Requer ligação á internet;

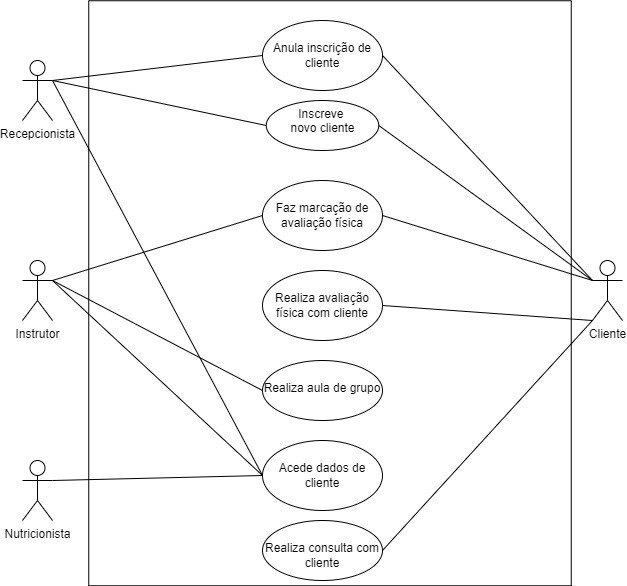
Desenvolvido para Windows e macOS;

Uso acessível;

Não requer computador de configurações elevadas.

# Design e Modelação

## Modelo de casos de uso



## Caso de uso: Inscrever cliente

|  |  |
| --- | --- |
| Id: | Caso de uso: **Inscrever cliente** |
| Ator principal: | Rececionista. |
| Pré-condições: | Rececionista está autenticada no sistema. |
| Pós-condições: | É criado cliente. |
| Cenário principal: | 1- Rececionista recebe pedido de inscrição;  2- Rececionista insere dados;  3- Sistema regista dados do cliente;  4- Secretária pede primeira mensalidade;  6- Sistema regista pagamento. |
| Cenários alternativos: | 1- Cliente não fornece dados suficientes;  2- Cliente não realiza o pagamento;  3- Cliente não fica inscrito. |

## Caso de uso: Marcar avaliação física

|  |  |
| --- | --- |
| Id: | Caso de uso: **Marcação de avaliação física** |
| Ator principal: | Instrutor. |
| Pré-condições: | Instrutor está autenticado no sistema;  Cliente registado. |
| Pós-condições: | Cliente ficará avaliado fisicamente. |
| Cenário principal: | 1- Recebe pedido de realização de avaliação;  2- Submete uma possível data;  3- Realiza a avaliação;  4- Escreve relatório da avaliação. |
| Cenários alternativos: | 1- Cliente não pode estar presente na data. |

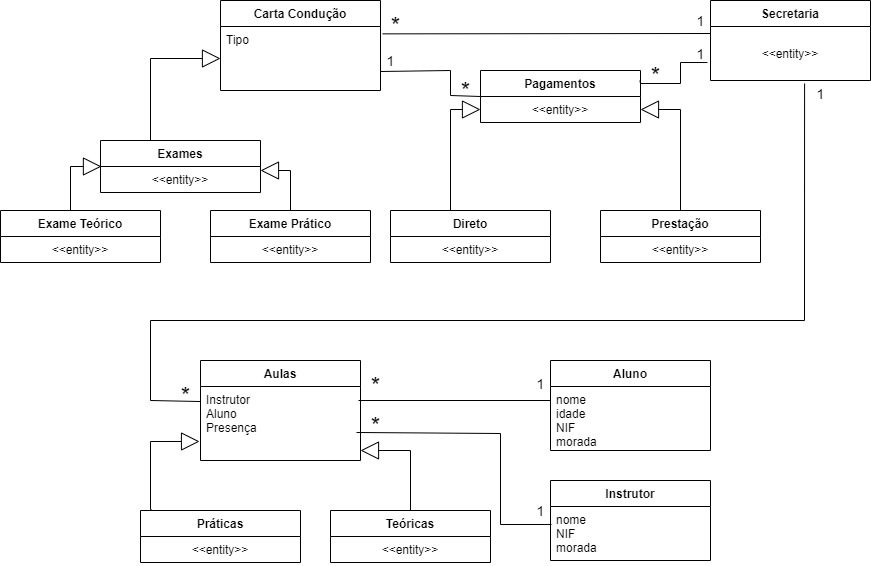
## Caso de uso: Marcar consulta de nutrição

|  |  |
| --- | --- |
| Id: | Caso de uso: **Marcar consulta de nutrição** |
| Ator principal: | Nutricionista. |
| Pré-condições: | Nutricionista está autenticado no sistema;  Cliente deve estar registado. |
| Pós-condições: | Cliente fica com plano nutricional á medida. |
| Cenário principal: | 1- Recebe pedido para consulta de nutrição;  2- Submete uma possível data;  3- Realiza a consulta;  4- Escreve plano alimentar;  4- Escreve relatório da consulta. |
| Cenários alternativos: | 1- Cliente não tem disponibilidade. |

## Caso de uso: Marcar aula de grupo

|  |  |
| --- | --- |
| Id: | Caso de uso: **Realizar aula de grupo** |
| Ator principal: | Secretária, instrutor. |
| Pré-condições: | Instrutor estar autenticado no sistema;  Cliente deve estar registado;  Ter vagas; |
| Pós-condições: | A aula fica realizada. |
| Cenário principal: | 1- Instrutor seleciona data para realização da aula;  2- Recebe pedidos dos alunos para inscrever;  3- Inscreve alunos até completar as vagas; |
| Cenários alternativos: | 1- Não existe vaga para o cliente. |

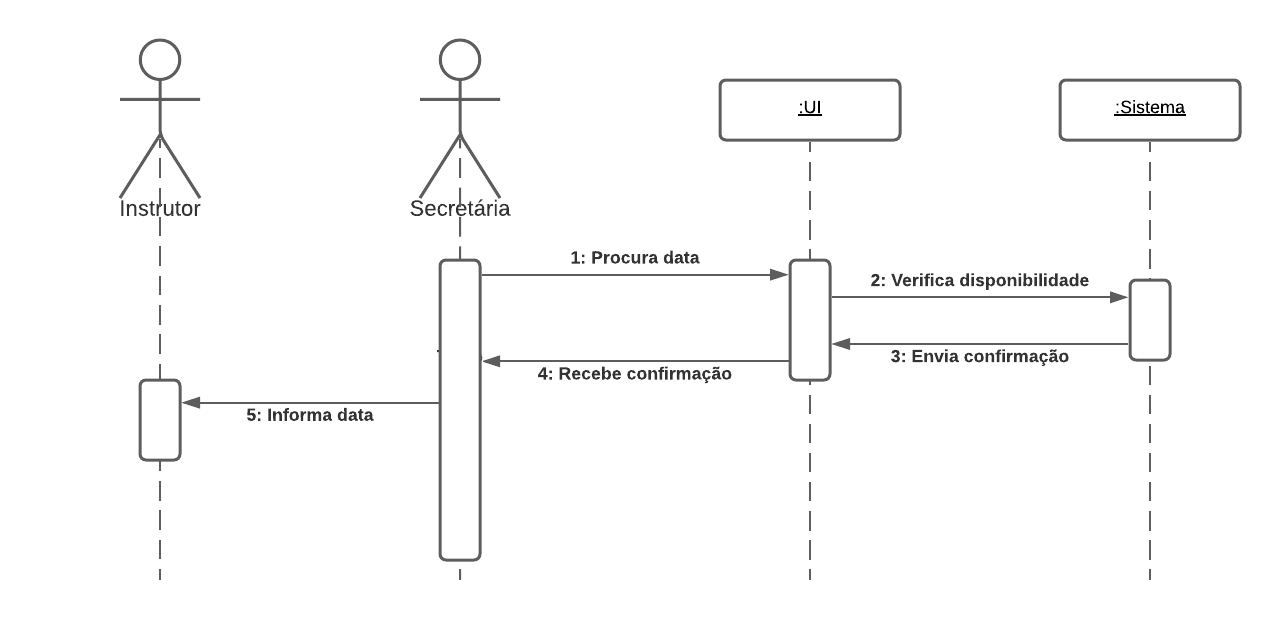
## Modelo de classes



## Diagrama de Sequência

## Inscrever Aluno

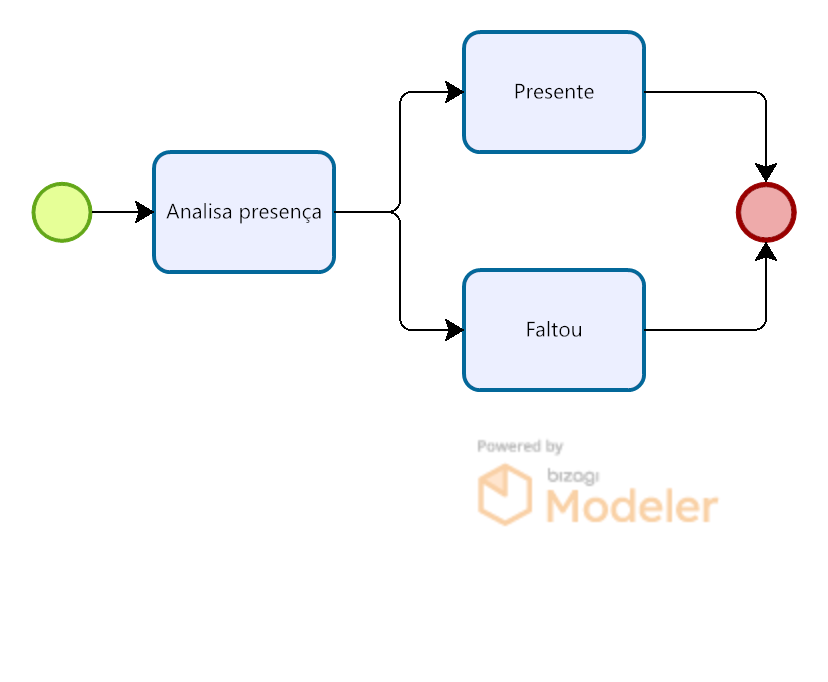
## Verificar Disponibilidade



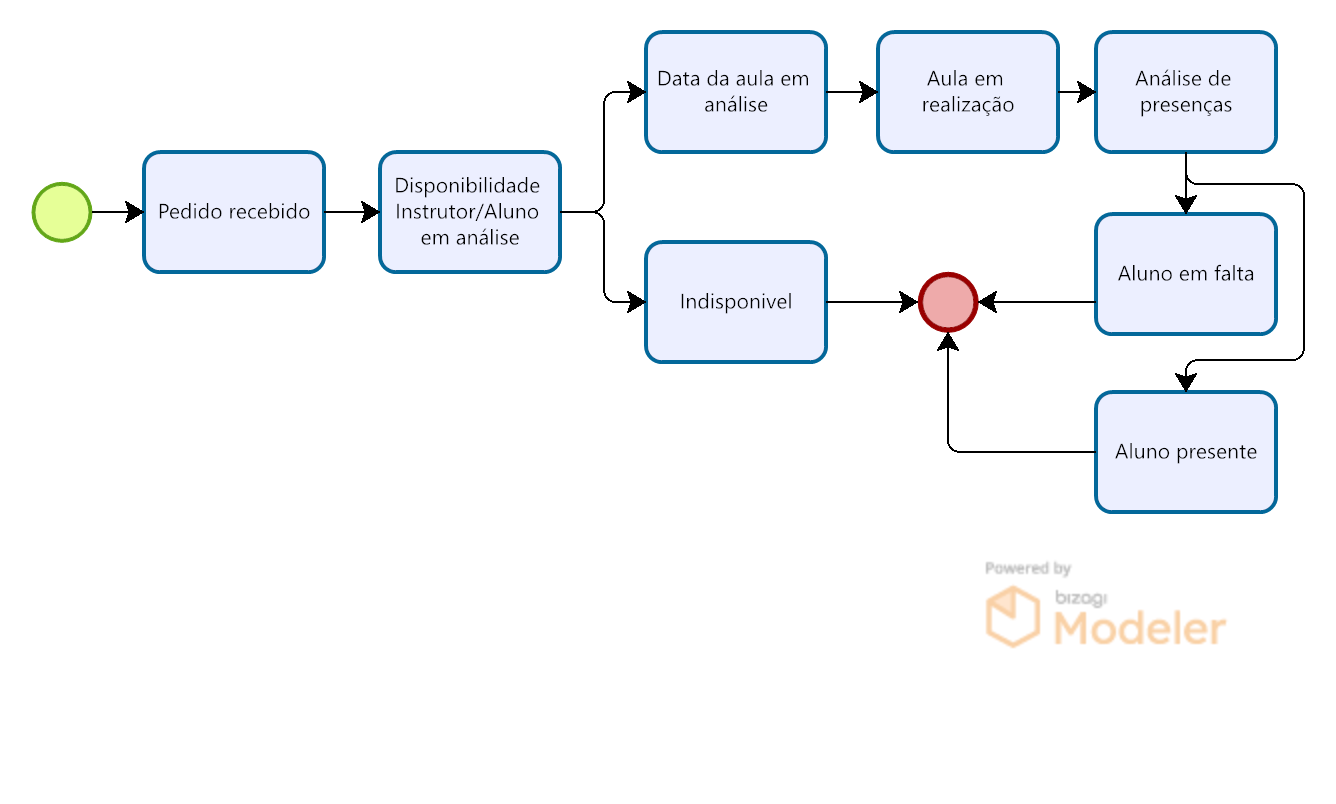
## Diagrama de Transição de Estados

## Inscrição Aluno

## Presenças nas Aulas Teóricas



## Presenças nas Aulas Práticas



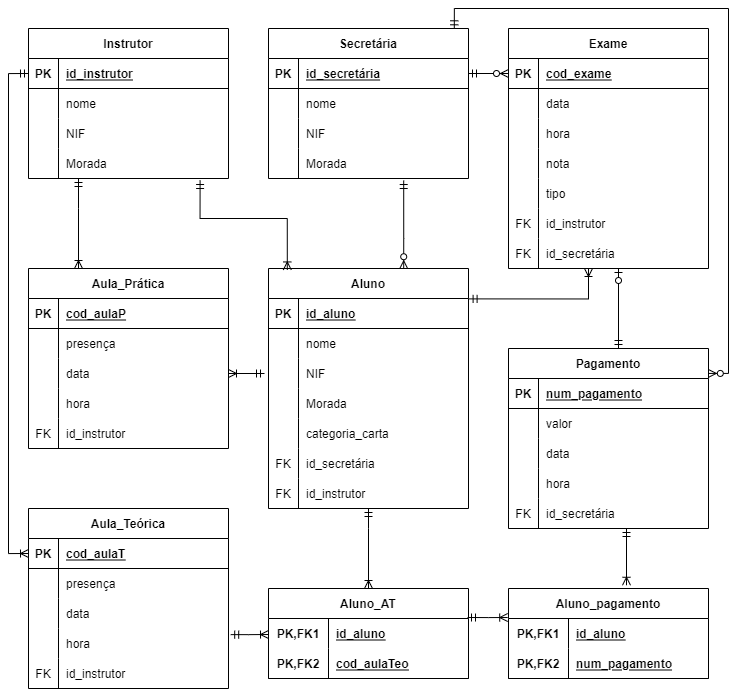
## Exame Teórico

## 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)**

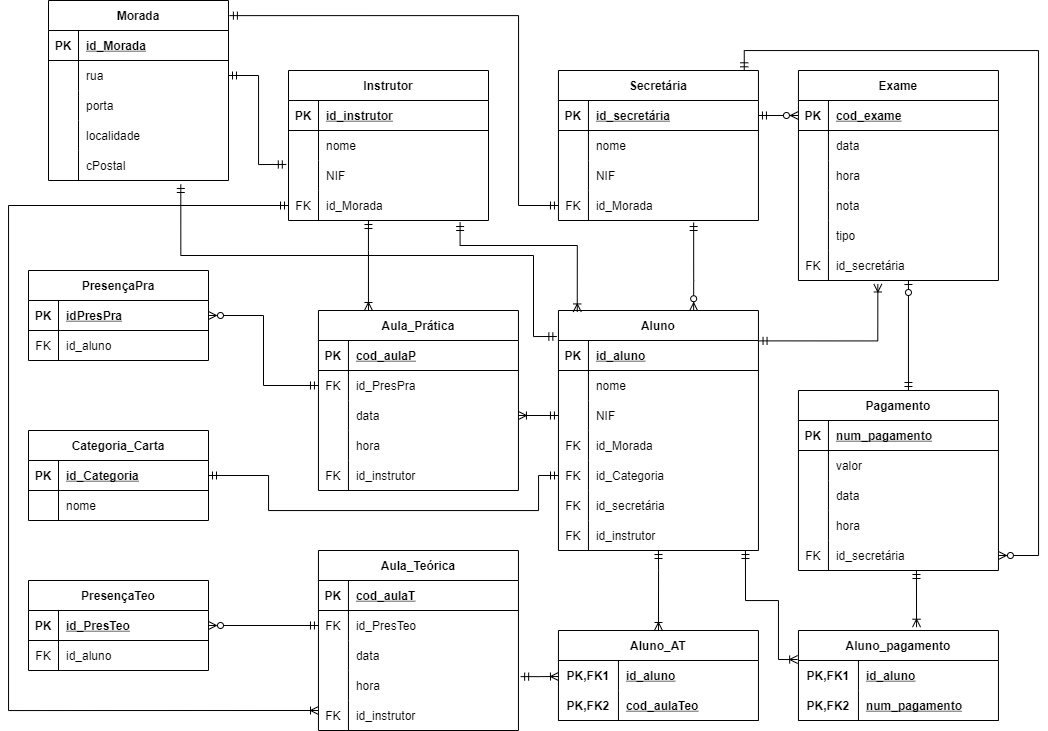
presençaTeo(idPresTeo, **id\_aluno)**

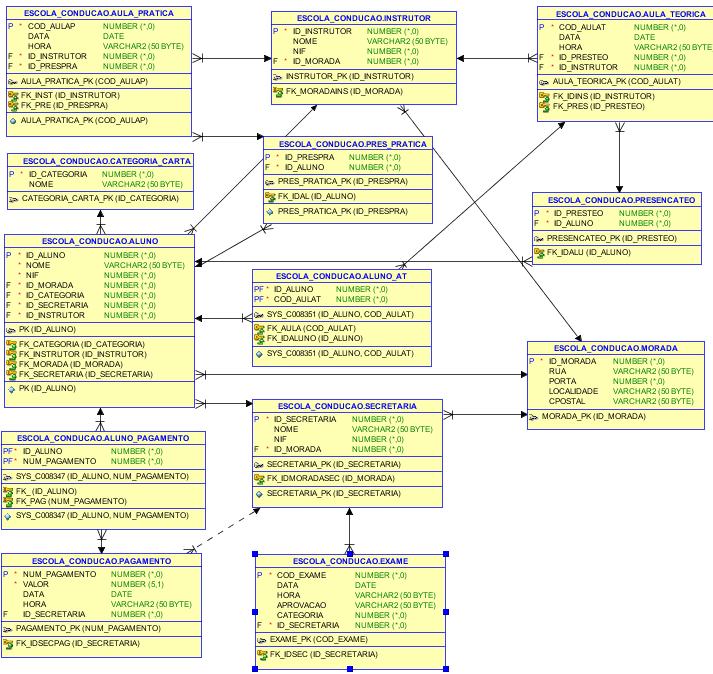
PresençaTeo (**idPresTeo**, **id\_aluno**)

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

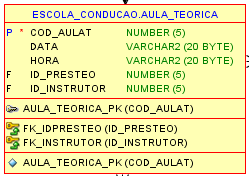
(num\_pagamento number(5) PRIMARY KEY NOT NULL,

valor number(7,2) NOT NULL

data date,

hora time,

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



CREATE TABLE Aula\_Teorica

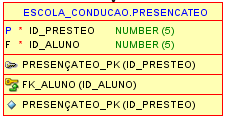
(cod\_aulaT number(5) PRIMARY KEY NOT NULL,

data date

hora time

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

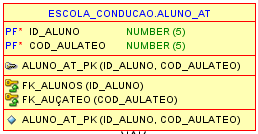
FOREIGN KEY (id\_instrutor) REFERENCES Instrutor (id\_instrutor) );



CREATE TABLE presençaTeo

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

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



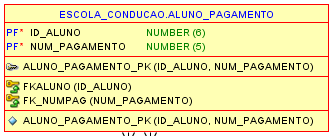
CREATE TABLE Aluno\_AT

(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));



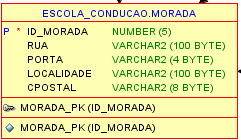
CREATE TABLE Aluno\_pagamento (

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

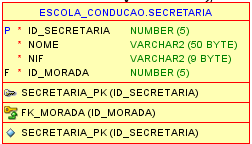
CREATE TABLE Instrutor

(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));



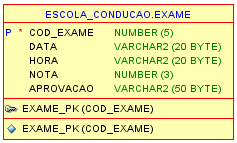
CREATE TABLE Secretaria (

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,

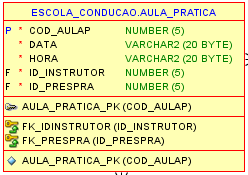
data date,

hora time,

nota number(3),

tipo varchar(8),

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



CREATE TABLE Aula\_Pratica

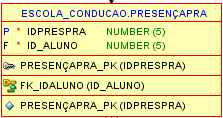
(cod\_aulaP number(5) PRIMARY KEY NOT NULL,

data date NOT NULL,

hora time NOT NULL,

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

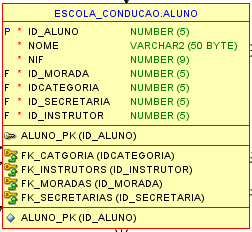
FOREIGN KEY (id\_PresPra) REFERENCES PresençaPra (id\_PresPra));



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.