

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.....	3
1.1 Apresentação do tema	3
1.2 Objetivos do projeto	3
2. Apresentação do negócio.....	3
2.1 Âmbito e enquadramento do projeto	3
2.2 Modelação dos Processos de negócio	4
2.2.1 Processo 1 - Inscrição de um cliente	4
2.2.2 Processo 2 - Consulta de avaliação física com Instrutor	4
2.2.3 Processo 3 - Consulta de nutrição com Nutricionista	4
2.2.4 Processo 4 – Inscrição em aula de grupo.....	5
2.2.5 Processo 5 – Anular inscrição no ginásio	5
3. Levantamento de Requisitos.....	6
3.1 Tipos de utilizador	6
3.1.1 Apresentação dos tipos de utilizador.....	6
3.1.2 Ações de cada tipo de utilizador	6
3.2 Requisitos Funcionais	6
3.3 Requisitos não funcionais	7
4. Design e Modelação	8
4.1 Modelo de casos de uso	8
4.1.1 Caso de uso: Inscrever cliente	9
4.1.2 Caso de uso: Marcar avaliação física.....	9
4.1.3 Caso de uso: Marcar consulta de nutrição	10
4.1.4 Caso de uso: Marcar aula de grupo.....	10
4.2 Modelo de classes	11
4.3 Diagrama de Sequência	12
4.3.1 Inscrever cliente	12
4.3.2 Marcar avaliação física.....	12
4.3.3 Inscrever em aula de grupo	13
4.4 Diagrama de Transição de Estados	13
4.4.1 Inscrição cliente.....	13
4.4.2 Marcar consulta de nutrição	14
4.4.3 Realizar pagamento de mensalidade	14
5. Implementação da BD.....	15
5.1 Diagrama Entidades-Relacionamentos	15
5.2 Modelo físico	16
5.3 Código SQL.....	19
5.3.1 Criação de tabelas	19
5.3.2. Inserts	23
5.3.3. Selects	27
5.3.4. Views.....	27
6. Conclusão e Trabalho Futuro	28

1. Introdução

1.1 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.

1.2 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.

2. Apresentação do negócio

2.1 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.

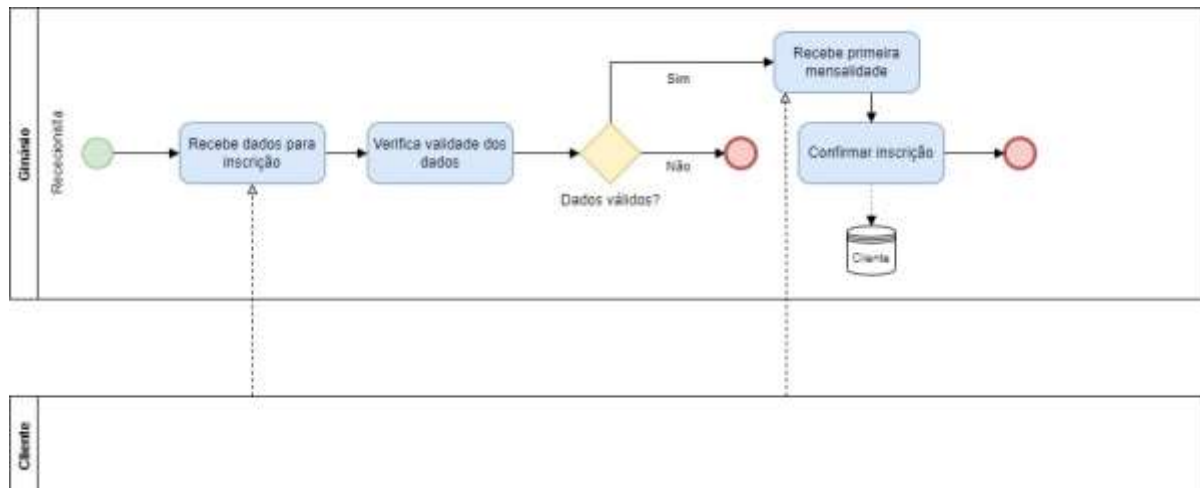
Para isso, instrutores e nutricionistas terão acesso a fichas de clientes, criadas e complementadas ao longo das suas sessões de treino e consultas, para determinar a melhor solução para qualquer dúvida que lhes possa vir a ser colocada.

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.

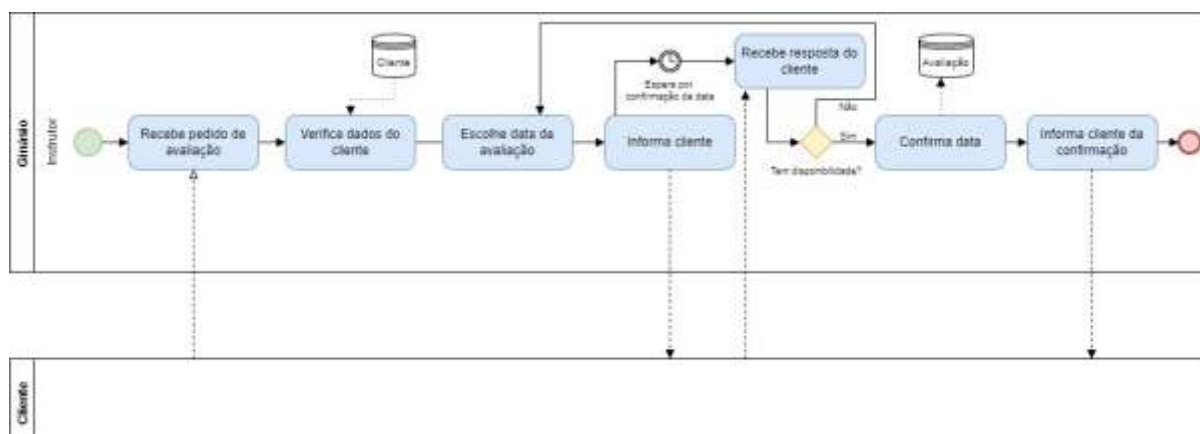
Por isso, o nosso serviço zelará não só pela rotina de exercícios do cliente, mas também pela sua rotina alimentar, dando-lhe a liberdade de escolha entre planos de exercício e alimentares personalizados feitos com a ajuda de especialistas que estarão sempre disponíveis aos clientes.

2.2 Modelação dos Processos de negócio

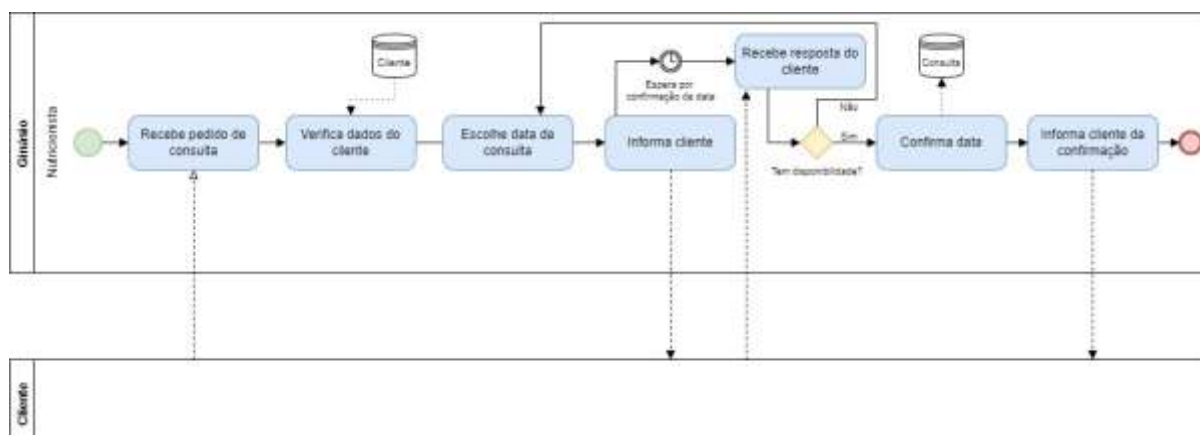
2.2.1 Processo 1 - Inscrição de um cliente



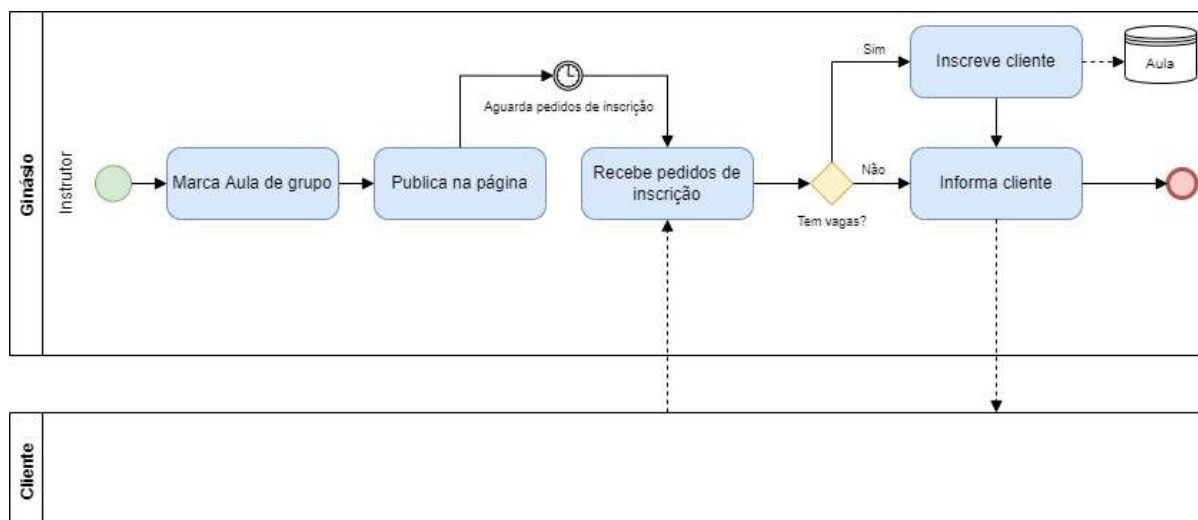
2.2.2 Processo 2 - Consulta de avaliação física com Instrutor



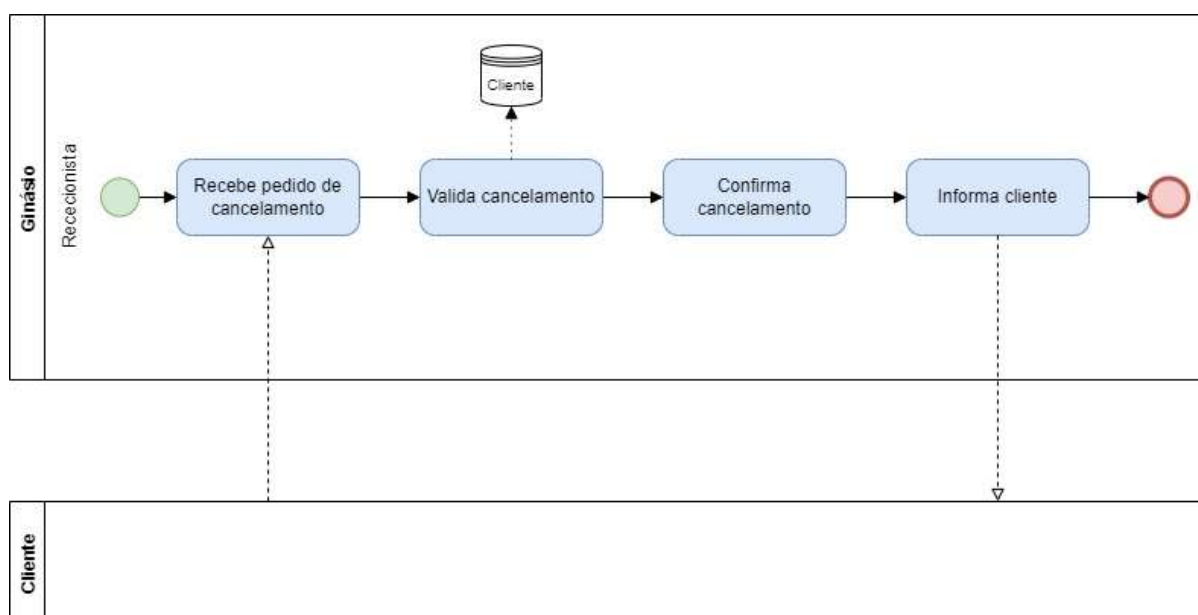
2.2.3 Processo 3 - Consulta de nutrição com Nutricionista



2.2.4 Processo 4 – Inscrição em aula de grupo



2.2.5 Processo 5 – Anular inscrição no ginásio



3. Levantamento de Requisitos

3.1 Tipos de utilizador

3.1.1 Apresentação dos tipos de utilizador

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

3.1.2 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.

3.2 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.

3.3 Requisitos não funcionais

Programa desenvolvido em java;

Base de dados desenvolvida em postgres;

Requer ligação á internet;

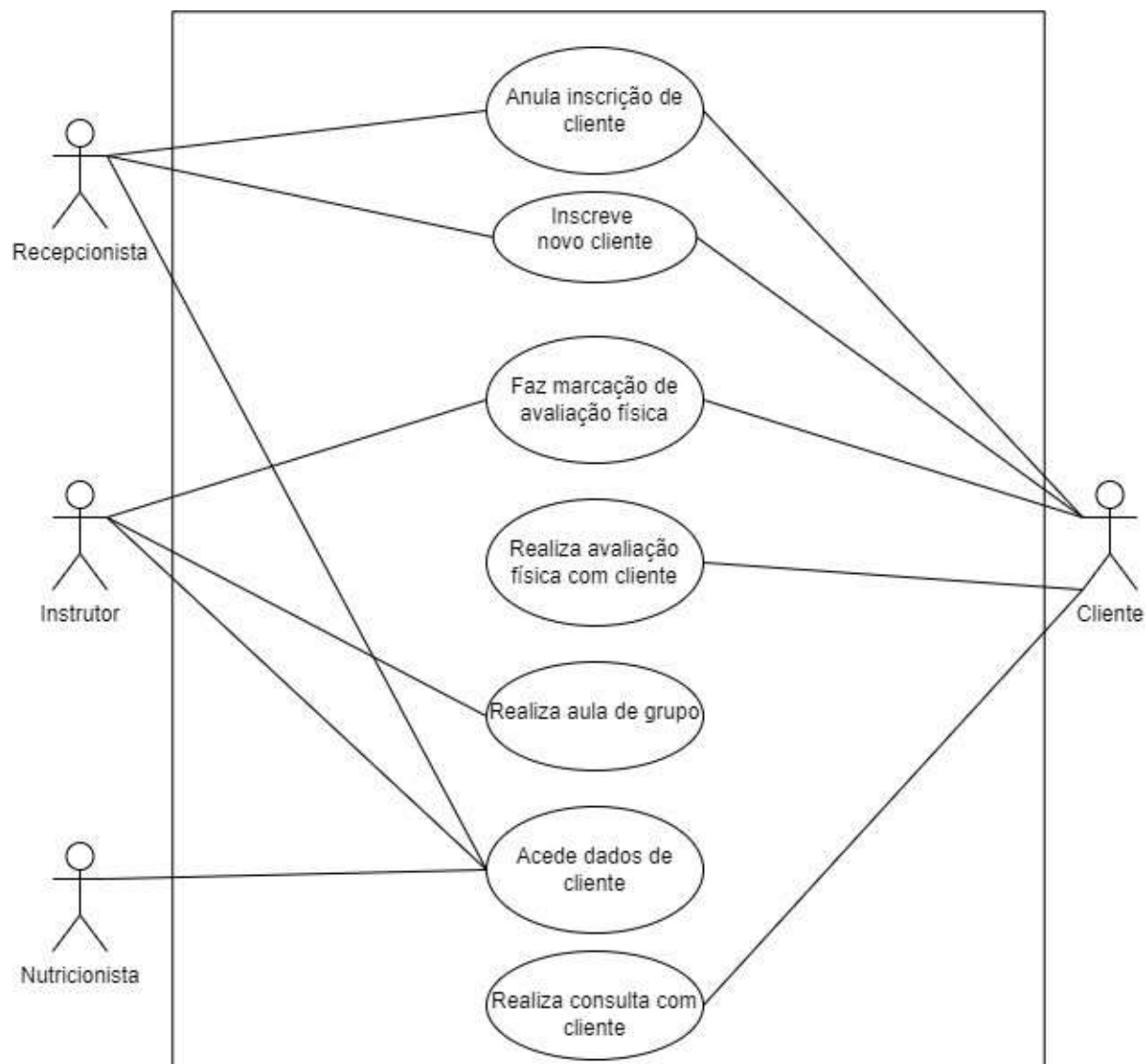
Desenvolvido para Windows e macOS;

Uso acessível;

Não requer computador de configurações elevadas.

4. Design e Modelação

4.1 Modelo de casos de uso



4.1.1 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.

4.1.2 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.

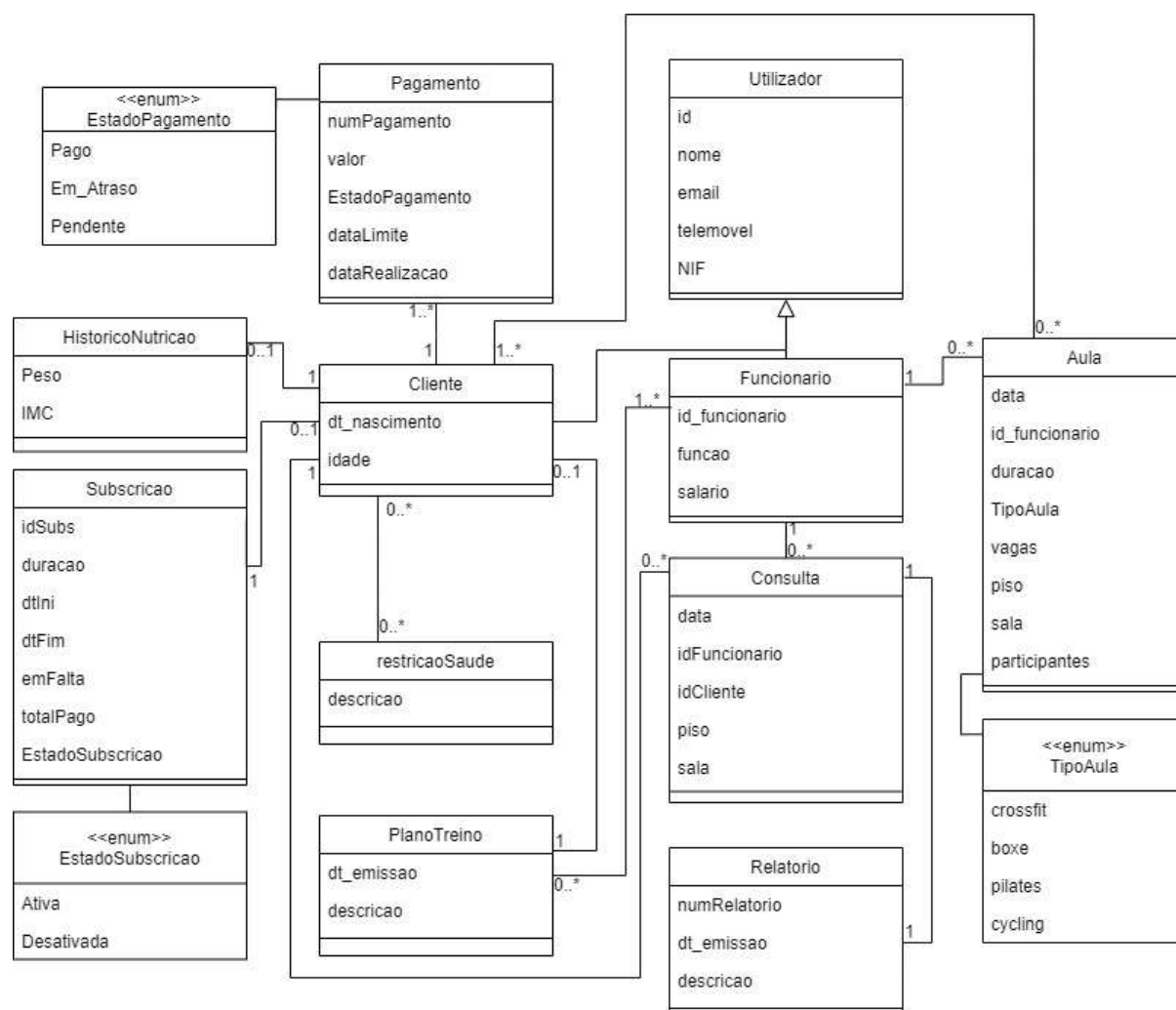
4.1.3 Caso de uso: Marcar consulta de nutrição

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.

4.1.4 Caso de uso: Marcar aula de grupo

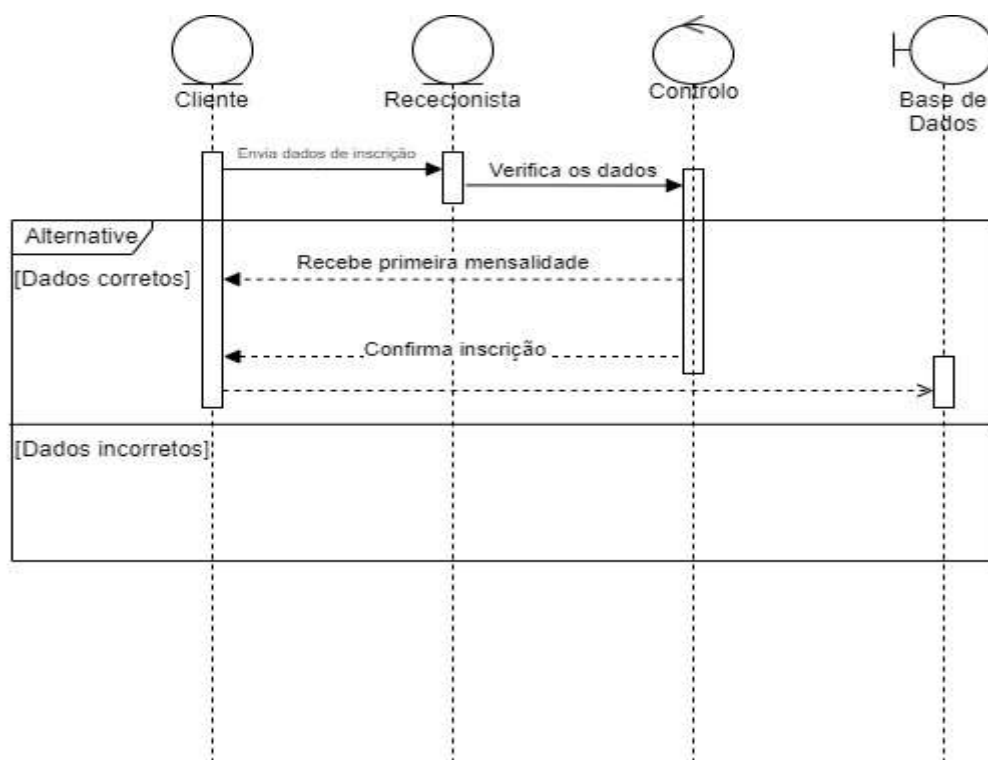
Id:	Caso de uso: Realizar aula de grupo
Ator principal:	Secretária, instrutor.
Pré-condições:	Instrutor estar autenticado no sistema; Cliente deve estar registado; Ter vagas;
Pós-condições:	A aula fica realizada.
Cenário principal:	1- Instrutor seleciona data para realização da aula; 2- Recebe pedidos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

4.2 Modelo de classes

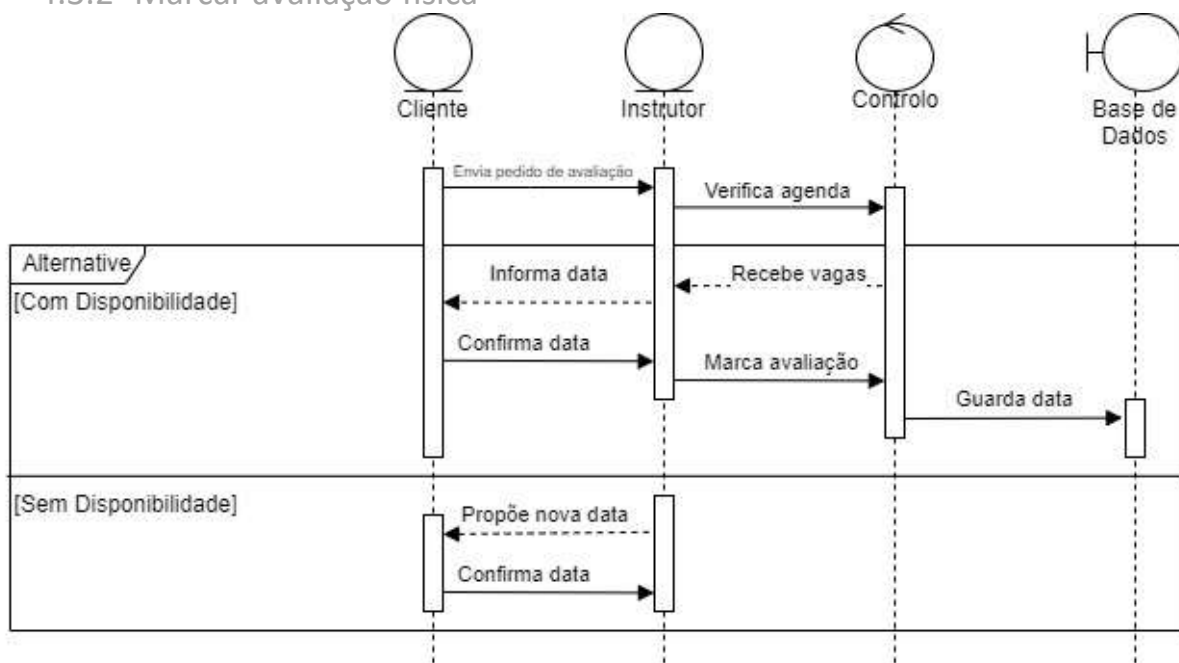


4.3 Diagrama de Sequência

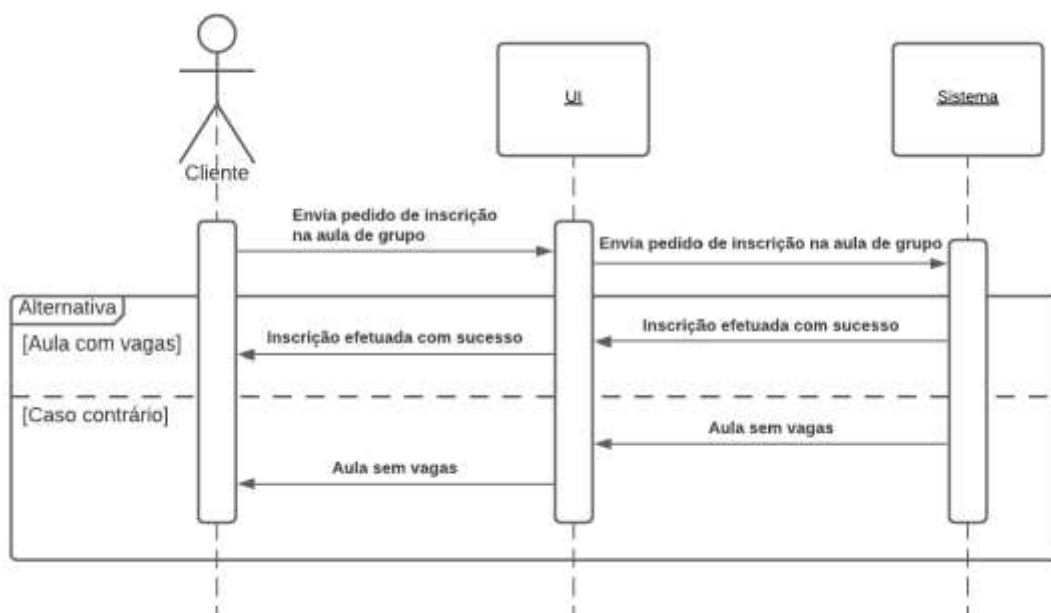
4.3.1 Inscrever cliente



4.3.2 Marcar avaliação física



4.3.3 Inscrever em aula de grupo

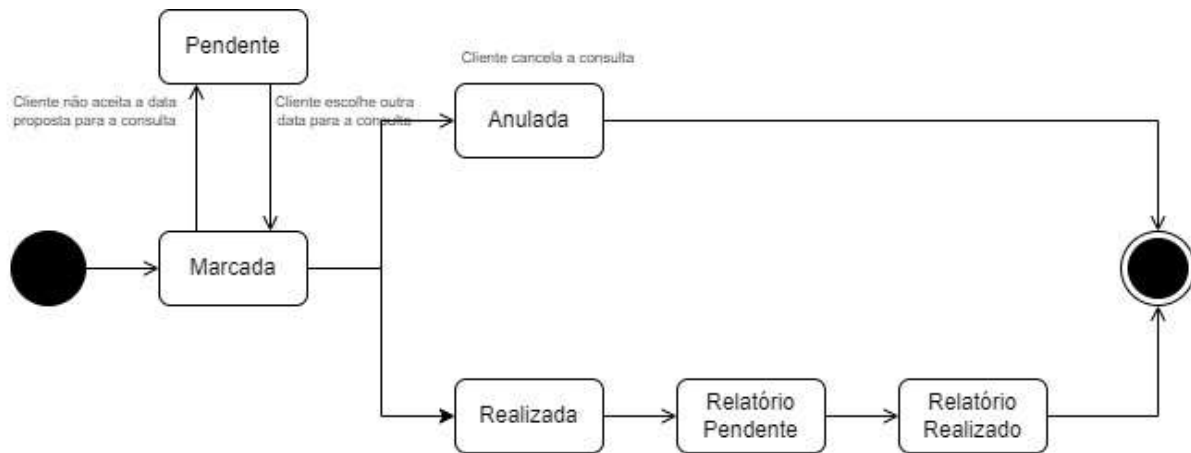


4.4 Diagrama de Transição de Estados

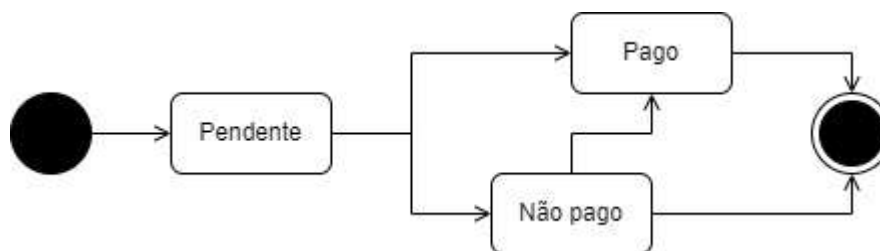
4.4.1 Inscrição cliente



4.4.2 Marcar consulta de nutrição

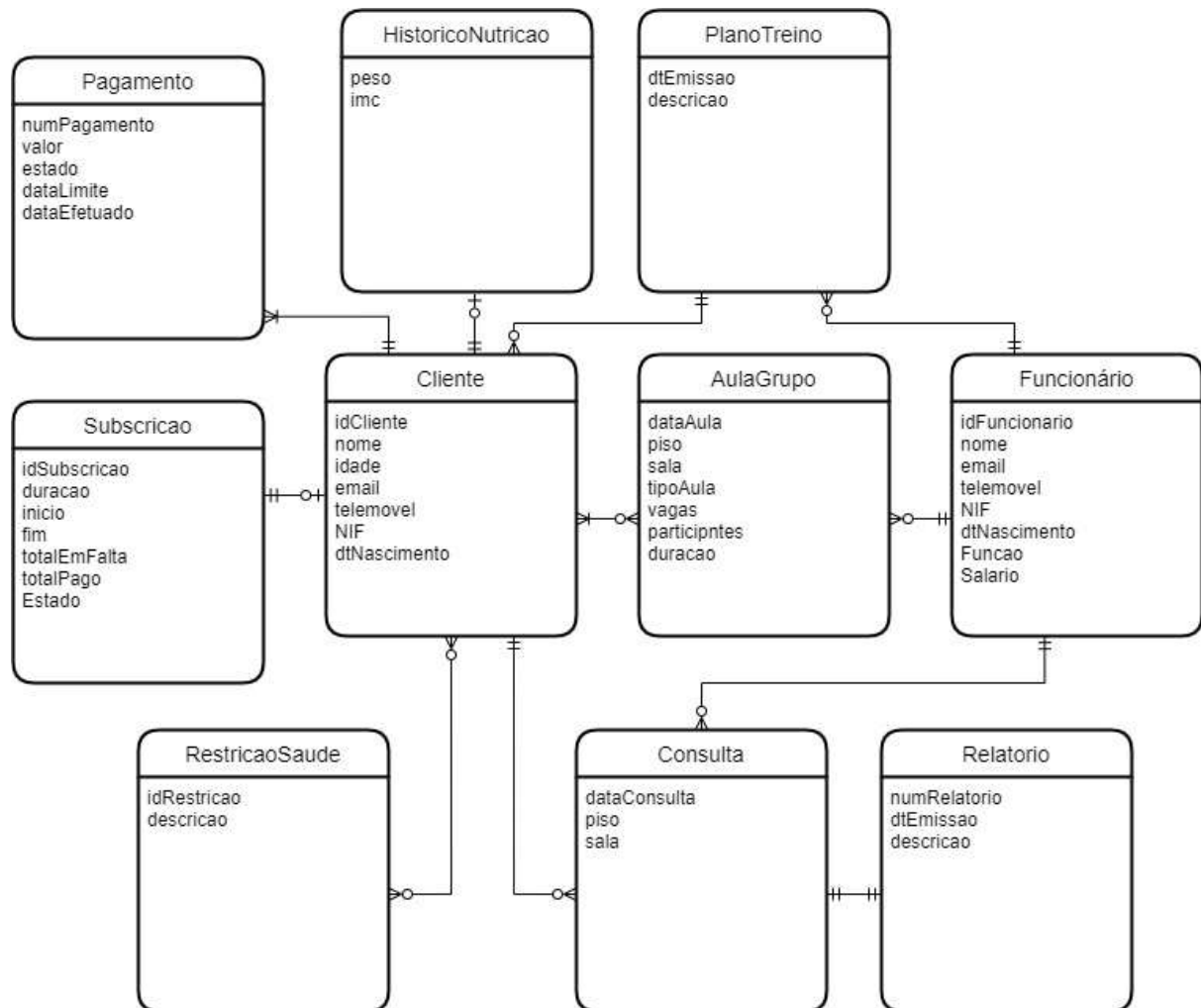


4.4.3 Realizar pagamento de mensalidade

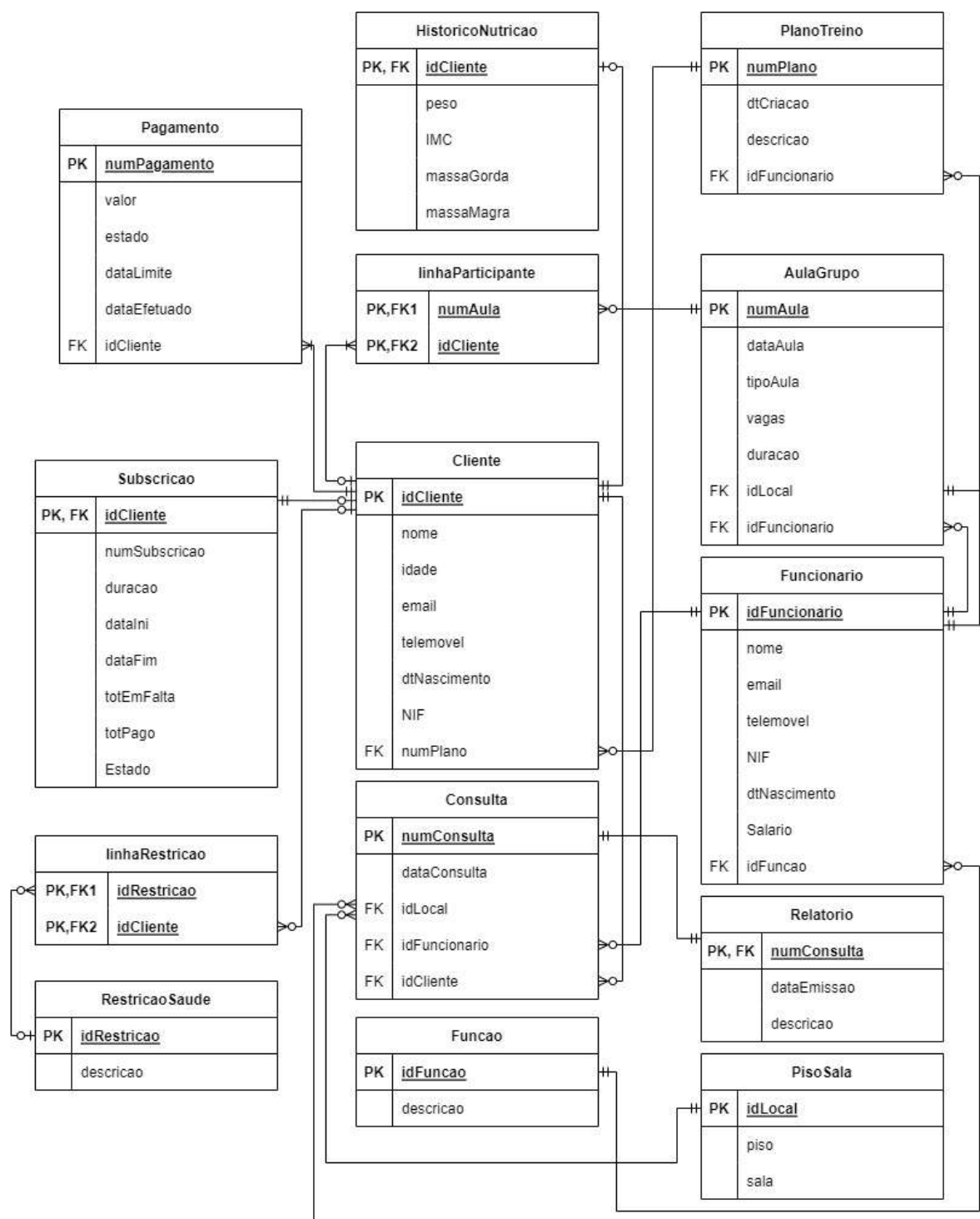


5. Implementação da BD

5.1 Diagrama Entidades-Relacionamentos



5.2 Modelo físico



Cliente (**idCliente**, nome, idade, email, telemóvel, dtNascimento, NIF, numPlano)

HistoricoNutricao (**idCliente**, peso, IMC, massaGorda, massaMagra)

Funcionario (**idFuncionario**, função, nome, email, telemóvel, NIF, dtNascimento, salario)

AulaGrupo (**numAula**, dataAula, piso, sala, tipoAula, vagas, duração, idFuncionario)

linhaParticipante (**numAula**, idCliente)

Consulta (**numConsulta**, dataConsulta, piso, sala, idFuncionario, idCliente)

Relatorio (**numConsulta**, dataEmissao, descricao)

PlanoTreino (**numPlano**, dtCriacao, descricao, idFuncionario)

RestricaoSaude (idRestricao, descricao)

LinhaRestricao (idRestricao, idCliente)

Subscricao (idCliente, numSubscricao, duracao, dataIni, dataFim, totEmFalta, totPago, estado)

Pagamento (**numPagamento**, valor, estado, dataLimite, dataEfetuado, idCliente)

1ª Forma Normal

Uma relação está na 1ª forma normal quando:

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

2ª Forma 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;

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

Normalização

Cliente (**idCliente**, nome, idade, email, telemóvel, dtNascimento, NIF, numPlano)

HistoricoNutricao (**idCliente**, peso, IMC, massaGorda, massaMagra)

Funcionario (**idFuncionario**, idFuncao, nome, email, telemóvel, NIF, dtNascimento, salario)

Funcao (**idFuncao**, descricao)

AulaGrupo (**numAula**, dataAula, idLocal, tipoAula, vagas, duracao, idFuncionario)

pisoSala (**idLocal**, piso, sala)

linhaParticipante (**numAula**, **idCliente**)

Consulta (**numConsulta**, dataConsulta, idLocal, idFuncionario, idCliente)

Relatorio (**numConsulta**, dataEmissao, descricao)

PlanoTreino (**numPlano**, dtCriacao, descricao, idFuncionario)

RestricaoSaude (**idRestricao**, descricao)

LinhaRestricao (**idRestricao**, **idCliente**)

Subscricao (**idCliente**, numSubscricao, duracao, dataIni, **valMensal**, **valTotal**, dataFim, totEmFalta, totPago, estado)

Pagamento (**numPagamento**, valor, estado, dataLimite, dataEfetuado, idCliente)

5.3 Código SQL

5.3.1 Criação de tabelas

```
CREATE TABLE Funcao (
```

```
    idFuncao INT PRIMARY KEY,
```

```
    descricao VARCHAR(100)
```

```
)
```

```
CREATE TABLE PisoSala (
```

```
    idLocal INT PRIMARY KEY,
```

```
    piso VARCHAR(1),
```

```
    sala VARCHAR(2)
```

```
)
```

```
CREATE TABLE Funcionario (
```

```
    idFuncionario INT PRIMARY KEY,
```

```
    idFuncao INT REFERENCES Funcao(idFuncao),
```

```
    nome VARCHAR(100),
```

```
    email VARCHAR(50),
```

```
    telemóvel INT,
```

```
    NIF INT,
```

```
    dtNascimento DATE,
```

```
    salario DECIMAL(7,2)
```

```
)
```

```
CREATE TABLE PlanoTreino (
```

```
    numPlano INT PRIMARY KEY,
```

```
    dtCriacao DATE,
```

```
    descricao VARCHAR(200),
```

```
    idFuncionario INT REFERENCES Funcionario(idFuncionario)
```

```
)
```

```
CREATE TABLE Cliente (  
    idCliente INT PRIMARY KEY,  
    nome VARCHAR(200),  
    idade INT,  
    email VARCHAR(100),  
    telemóvel INT,  
    dtNascimento DATE,  
    NIF INT,  
    numPlano INT REFERENCES PlanoTreino(numPlano)  
)  
  
CREATE TABLE HistoricoNutricao (  
    idCliente INT REFERENCES Cliente(idCliente),  
    peso DECIMAL(3,2),  
    IMC DECIMAL(2,1),  
    massaGorda INT,  
    massaMagra INT  
)  
  
CREATE TABLE AulaGrupo (  
    numAula INT PRIMARY KEY,  
    dataAula DATE,  
    idLocal INT REFERENCES PisoSala(idLocal),  
    tipoAula VARCHAR(1),  
    vagas INT,  
    duracao TIME,  
    idFuncionario INT REFERENCES Funcionario(idFuncionario)  
)
```

```
CREATE TABLE Consulta (  
    numConsulta INT PRIMARY KEY,  
    dataConsulta DATE,  
    idLocal INT REFERENCES PisoSala(idLocal),  
    idFuncionario INT REFERENCES Funcionario(idFuncionario),  
    idCliente INT REFERENCES Cliente(idCliente)  
)
```

```
CREATE TABLE Subscricao (  
    idCliente INT REFERENCES Cliente(idCliente),  
    duracao TIME,  
    dataIni DATE,  
    valMensal DECIMAL(3,2),  
    valTotal DECIMAL(3,2),  
    dataFim DATE,  
    totEmFalta DECIMAL(3,2),  
    totPago DECIMAL(4,2),  
    estado VARCHAR(20)  
)
```

```
CREATE TABLE linhaParticipante (  
    numAula INT REFERENCES AulaGrupo(numAula),  
    idCliente INT REFERENCES Cliente(idCliente)  
)
```

```
CREATE TABLE Relatorio (  
    numConsulta INT REFERENCES Consulta(numConsulta),  
    dataEmissao DATE,  
    descricao VARCHAR(200)  
)
```

```
CREATE TABLE RestricaoSaude (  
    idRestricao INT PRIMARY KEY,  
    descricao VARCHAR(200)  
)  
  
CREATE TABLE LinhaRestricao (  
    idRestricao INT REFERENCES RestricaoSaude(idRestricao),  
    idCliente INT REFERENCES Cliente(idCliente)  
)  
  
CREATE TABLE Pagamento (  
    numPagamento INT PRIMARY KEY,  
    valor decimal(3,2),  
    estado VARCHAR(20),  
    dataLimite DATE,  
    dataEfetuado DATE,  
    idCliente INT REFERENCES Cliente(idCliente)  
)
```

5.3.2. Inserts

insert into Funcao (idFuncao, descricao)

values (1, 'Rececionista'),

(2, 'Instrutor'),

(3, 'Nutricionista');

insert into PisoSala (idLocal, piso, sala)

values (1, 1, 1), (2, 1, 2), (3, 1, 3), (4, 1, 4),

(5, 1, 5), (6, 2, 1), (7, 2, 2), (8, 2, 3),

(9, 2, 4), (10, 2, 5);

insert into Funcionario (idFuncionario, idFuncao, nome, email, telemóvel, NIF, dtNascimento, salario)

values (1, 1, 'Dwain Snaden', 'dsnaden0@fc2.com', '9638680740', '2276607035', '21/03/1976', '€934,36'),

(2, 1, 'Emylee Grene', 'egrene1@tmall.com', '9677286053', '8733475878', '28/05/1981', '€934,36'),

(3, 2, 'Cos Rayne', 'crayne2@instagram.com', '9693503520', '2667036947', '06/02/1971', '€2177,38'),

(4, 2, 'Cammi Hellwing', 'chellwing3@mashable.com', '9642097925', '2259291286', '19/11/1999', '€2209,50'),

(5, 3, 'Morton Kagan', 'mkagan4@topsy.com', '9618831303', '8601579633', '23/06/1990', '€2430,86'),

(6, 3, 'Iosep Rosin', 'irosin5@illinois.edu', '9626521598', '2619357632', '09/12/1998', '€1617,60');

```
insert into PlanoTreino (numPlano, dtCriacao, descricao, idFuncionario)
```

```
values (1, '09/11/2020', 'Plano focado em Cardio para redução de massa gorda.', 3),  
(2, '19/02/2021', 'Plano focado em Musculação para aumento de massa magra.', 4);
```

```
insert into Cliente (idCliente, nome, email, telemóvel, dtNascimento, NIF, numPlano)
```

```
values (1, 'Alejandro Magnay', 'amagnay0@ibm.com', '964156411', '02/08/1995', '759500619', 2),  
(2, 'Maurine Thumann', 'mthumann1@etsy.com', '964948514', '16/06/2000', '869975297', 1),  
(3, 'Cosette Pleass', 'cpleass2@ed.gov', '964572474', '30/08/1962', '892590437', 1),  
(4, 'Elaina Fairholm', 'efairholm3@japanpost.jp', '967020440', '26/02/1986', '410472625', 2),  
(5, 'Verine Axcell', 'vaxcell4@psu.edu', '969662198', '12/10/1994', '456072561', 1),  
(6, 'Ax Mosby', 'amosby5@intel.com', '964659222', '11/05/1967', '236224630', 1),  
(7, 'Ashli Yellop', 'ayellop6@deliciousdays.com', '967568243', '25/05/1998', '991264043', 2),  
(8, 'Emily Groom', 'egroom7@sakura.ne.jp', '961976264', '16/04/1963', '861422027', 2),  
(9, 'Zandra Arbon', 'zarbon8@dmoz.org', '967333674', '12/04/1975', '421877824', 2),  
(10, 'Hildagard Busain', 'hbusain9@hp.com', '966467363', '11/11/1979', '103649512', 1);
```

```
insert into HistoricoNutricao (idCliente, peso, IMC, massaGorda, massaMagra)
```

```
values (2, 90.2, 20, 22, 17), (1, 55, 30, 19, 24), (4, 77, 37.1, 21, 18),  
(3, 56, 34, 19, 28), (6, 55, 36.3, 19, 25), (5, 84, 27, 31, 29),  
(8, 46.9, 31.2, 26, 20), (7, 79.5, 20.5, 19, 26), (9, 56, 27, 29, 29),  
(10, 51, 30, 16, 24);
```

```
insert into AulaGrupo (numAula, dataAula, idLocal, tipoAula, vagas, duracao, idFuncionario)
```

```
values (1, '27/12/2022', 4, 'Musculação', 10, 1, 3),  
(2, '23/12/2021', 4, 'Cycling', 20, 1, 4),  
(3, '20/07/2021', 1, 'Boxing', 15, 2, 3),  
(4, '01/05/2022', 2, 'Crossfit', 30, 3, 4);
```


insert into Consulta (numConsulta, dataConsulta, idLocal, idFuncionario, idCliente)

values (1, '27/11/2022', 7, 5, 2),
(2, '23/02/2021', 6, 5, 4),
(3, '20/01/2021', 9, 6, 6),
(4, '01/12/2022', 10, 6, 8);

insert into Subscricao (idCliente, duracao, dataIni, dataFim, valMensal, valTotal, totEmFalta, totPago, estado)

values (1, 6, '10/01/2022', '10/06/2022', 8, 48, 40, 8, 'Ativa'),
(2, 6, '01/01/2022', '01/06/2022', 8, 48, 0, 48, 'Cancelada'),
(3, 3, '04/01/2022', '04/03/2022', 15, 45, 0, 45, 'Ativa'),
(4, 12, '06/01/2022', '06/01/2023', 7, 84, 0, 84, 'Ativa'),
(5, 12, '03/01/2022', '03/01/2023', 7, 84, 7, 77, 'Ativa'),
(6, 12, '01/01/2022', '01/01/2023', 7, 84, 0, 84, 'Ativa'),
(7, 6, '04/01/2022', '04/06/2022', 8, 48, 40, 8, 'Ativa'),
(8, 6, '03/01/2022', '03/06/2022', 8, 48, 40, 8, 'Cancelada'),
(9, 3, '05/01/2022', '05/03/2022', 15, 45, 15, 30, 'Ativa'),
(10, 3, '07/01/2022', '07/03/2022', 15, 45, 0, 45, 'Ativa');

insert into linhaParticipante (numAula, idCliente)

values (2, 8), (2, 6), (4, 4), (1, 7), (2, 10), (4, 7), (2, 4), (3, 5),
(3, 7), (3, 8), (3, 8), (2, 8), (4, 1), (2, 1), (4, 10), (3, 8),
(1, 6), (4, 8), (4, 4), (1, 1);

```
insert into Relatorio (numConsulta, dataEmissao, descricao)
```

```
values (1, '27/11/2022', 'Peso aumentou 2 kg, ganhou massa magra, perdeu massa gorda.'),  
       (2, '23/02/2021', 'Peso reduziu 3 kg, manteve massa magra, perdeu massa gorda.'),  
       (3, '20/01/2021', 'Peso aumentou 6 kg, ganhou massa magra, ganhou massa gorda.'),  
       (4, '01/12/2022', 'Peso reduziu 1 kg, perdeu massa magra, manteve massa gorda.');
```

```
insert into RestricaoSaude (idRestricao, descricao)
```

```
values (1, 'Asma'), (2, 'Alergias'), (3, 'Diabetes');
```

```
insert into LinhaRestricao (idRestricao, idCliente)
```

```
values (1, 3), (2, 6), (3, 2);
```

```
insert into Pagamento (numPagamento, valor, estado, dataLimite, dataEfetuado, idCliente)
```

```
values (1, 20, 'pago', '27/11/2022', '27/11/2022', 2),  
       (2, 55.1, 'pendente', '05/01/2022', NULL, 3),  
       (3, 34, 'pago', '17/12/2021', '16/12/2021', 6);
```

5.3.3. Selects

(1) Mostrar todos os planos

```
SELECT numPlano, descricao FROM PlanoTreino;
```

(2) Mostrar clientes com pagamentos pendentes

```
SELECT c.nome, p.estado FROM Cliente as c, pagamento as p  
WHERE p.idCliente = c.idCliente AND p.estado = 'pendente';
```

(3) Mostrar clientes com pagamentos pagos

```
SELECT c.nome, p.estado FROM Cliente as c, pagamento as p  
WHERE p.idCliente = c.idCliente AND p.estado = 'pago';
```

5.3.4. Views

(1)Mostrar todos os clientes ativos

```
CREATE VIEW CLIATIV as  
SELECT c.nome, c.email  
FROM Cliente as c, Subscricao as s  
WHERE s.idCliente = c.idCliente AND s.Estado = 'Ativa';
```

(2)Mostrar todos os clientes que tem um plano

```
CREATE VIEW CLIENTEPLANO as  
SELECT c.nome, c.email, p.numPlano  
FROM Cliente as c AND PlanoTreino as p  
WHERE c.numPlano = p.numPlano;
```

6. Conclusão e Trabalho Futuro

No desenvolvimento deste projeto surgiram diversas dificuldades, nos 3 campos da resolução deste projeto: Modelação de Negócio, Modelação SW e implementação na base de dados.

Com a parte da Modelação de Negócio surgiram dificuldades na realização dos diagramas de sequência e de classes, mas conseguimos ainda assim pesquisar para completar as falhas de conhecimento.

Na implementação da base de dados surgiram algumas dificuldades em compreender como funciona o postgres.

Por fim, conseguimos atingir o objetivo que tínhamos e sabemos que ainda há muito espaço para melhorias que podem ser feitas neste trabalho.