



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DA PARAÍBA**  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JOZIMAR SOARES DA COSTA  
RÔMULO SOARES BEZERRA

**SISTEMA DE GERENCIAMENTO DE ACADEMIA FITNESS**

Cajazeiras  
2017

JOZIMAR SOARES DA COSTA  
RÔMULO SOARES BEZERRA

## **SISTEMA DE GERENCIAMENTO DE ACADEMIA FITNESS**

Trabalho apresentado ao Curso Superior Tecnológico em Análise e Desenvolvimento de Sistemas do IFPB – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, para a disciplina Banco de Dados.

Prof. Dr. Fabio Gomes de Andrade

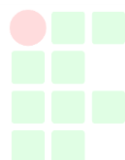
Cajazeiras  
2017

# Sumário

<b>1. Introdução</b>	<b>4</b>
<b>2. Modelagem Conceitual</b>	<b>5</b>
2.1 Levantamento de Requisitos .....	6
2.2 Modelo Conceitual .....	8
2.3 Dicionário de dados Conceitual .....	9
2.4 Levantamento de consultas .....	16
<b>3. Modelagem Lógica</b>	<b>17</b>
3.1 Mapeamento Entidade Relacionamento .....	19
3.2 Dicionário de Dados Lógico .....	20
3.3 Scripts de Criação SQL .....	39
3.4 Scripts de Inserção SQL .....	48

## **1. Introdução**

Com extensão das academias de musculação e o avanço tecnológico, gerir um negócio tão abrangente sem o auxílio da tecnologia se tornou uma atividade inviável. Com base nisso, no intuito de prevenir más ocorrências evitando possíveis dores de cabeça com o descaminho dos dados, o dono da Top Fitness Academia requisitou um sistema que o ajude a otimizar o gerenciamento de seu estabelecimento - informações de clientes, compras, gastos, etc. Todos os seus dados e de seus alunos são armazenados em fichas de papelão que estão sujeitas a deterioração por agentes naturais (poeira e umidade) que podem ocasionar a perda dessas informações e gerar um retrabalho excessivo e desagradável.



**MODELAGEM CONCEITUAL**  
SISTEMA DE GERENCIAMENTO DE ACADEMIA FITNESS

## 2.1 Levantamento de Requisitos

O sistema precisa cadastrar alunos. Do aluno precisam ser armazenados: código, nome, sexo, telefone, peso, idade, altura, IMC, BF, estado de saúde e objetivo. Também precisam ser armazenadas informações extras do aluno com relação a mensalidade. Em mensalidade são registradas o valor, data de pagamento e data de recibo em que o aluno fez o pagamento. O aluno tem também matrícula que possui código, data de abertura e data de trancamento.

Ao se cadastrar, o aluno, é feita uma mensuração de medidas corporais que são vistas a cada mês, guardando-se a data, desde a data de entrada, para o acompanhamento de seu desenvolvimento. As medidas corporais do aluno são ponderadas em centímetros e compostas por medidas de braços: esquerdo e direito, antebraços: esquerdo e direito, deltoides, peitoral, abdome, coxas: esquerda e direita e panturrilhas: esquerda e direita.

A academia possui funcionários que podem ser classificados em dois tipos: gerente que administra todo o empreendimento e/ou professor. Para professor e gerente são mantidos: nome, código, sexo, CPF e data de pagamento do salário. Em especificação, professor tem especialidade (s) e salário, e o gerente: lucro mensal que varia mensalmente conforme a quantidade de alunos frequentes e renda de vendas realizadas.

O gerente gerencia despesas, com descrição, valor e data, que são geradas a partir de necessidades integrantes. O gerente pode realizar compras, assim como o aluno, podendo adquirir produtos como por exemplo suplementação e gourmet, sendo necessário guardar a data da realização. É importante armazenar informações da compra também como: descrição, valor, CNPJ ou CPF e código. As compras podem ser a vista ou a prazo. Se a vista, guarda-se o desconto e se a prazo, guarda-se as informações de quantidade de parcelas, valor de cada parcela e um atributo para verificar se foi quitada. Para produto, devem ser guardados o nome, marca, quantidade e preço.

Para cada aluno é montado um treino e exercícios pelo professor. Cada treino tem código, descrição, uso de carga, o tipo - que é a combinação dos tipos distintos

trabalhados nos exercícios, tempo de intervalo de descanso em dias quando completo o ciclo do tipo, tempo de intervalo entre as séries, tempo de intervalo entre as repetições, data de início e duração até a mudança de treino.

Exemplo: - O aluno Abcelino está com o treino de código 001, descrito como: útil para hipertrofia muscular, com uso de carga pesada, tipo ABC, com intervalo de descanso: um dia após a realização dos exercícios da divisão “C” para novamente executar o treino “A”, 3 minutos de descanso para execução de uma nova série, com 30 segundos entre as repetições. Ele está trabalhando uma ou duas vezes por semana o mesmo músculo. Iniciou o treino dia 24/02/2017, e tem como duração 2 meses até a troca ou adaptação do novo treino.

O treino tem exercícios que são particulares a grupos musculares. Os exercícios têm músculo trabalhado, nome, número de séries, quantidade de repetições, se é Drop Set e o tipo, se: A, B, C, D, E ou F. Cada tipo é uma divisão de exercícios de grupos musculares e treinado por dia. Os treinos e os exercícios são montados e supervisionados pelo professor.

Logo, concluindo com o exemplo de Abcelino; ele numa segunda-feira vai executar o primeiro exercício para bíceps braquial de nome “rosca direta” com 3 vezes de 12 repetições, não Drop Set e referente a sigla “A” do tipo de treino “ABC”. Como segundo exercício, também para bíceps braquial ele fará a “rosca martelo”, com 4 vezes de 10 repetições, é Drop Set e também de “A” já que é praticado no mesmo dia. Terminado esses exercícios, Abcelino agora vai malhar o exercício para peitoral, com nome supino reto, 3 de 15, não Drop Set e de “A”, e em seguida com mais um de peitoral “supino declinado com halteres”, 4 de 8, não Drop Set, e de “A”.

Para melhor entendimento, o aluno Abcelino numa terça-feira vai malhar um exercício para dorsal “remada baixa”, com 4 vezes de 12 repetições, não Drop Set e referente a sigla do dia “B”. Como segundo exercício: “remada curvada”, 3 de 10, Drop Set e da sigla do tipo “B”. E assim segue o mesmo para a quarta-feira; cada dia terá vários exercícios trabalhando músculos com características e execuções diferentes.

## 2.2 Modelo Conceitual

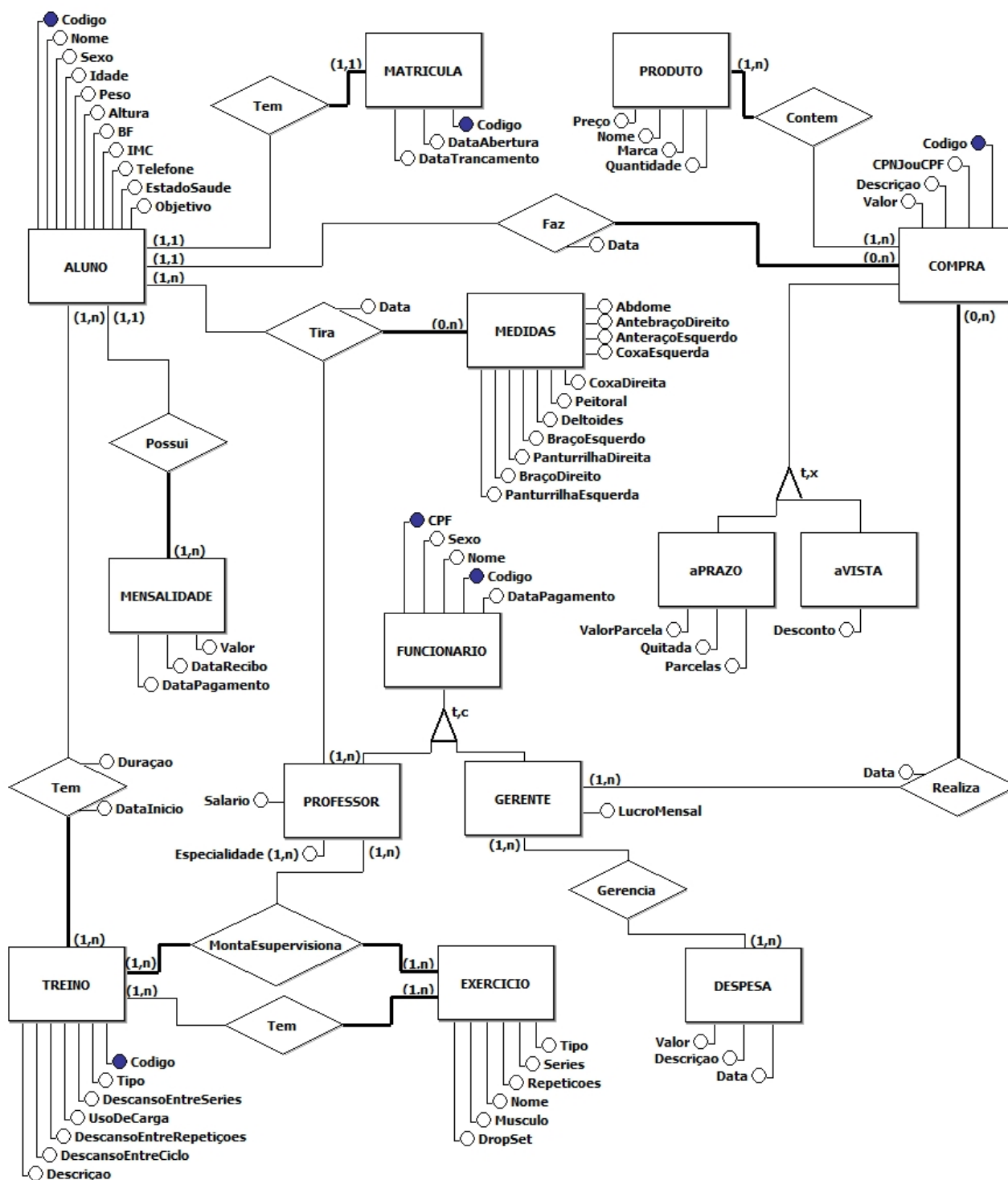


Figura 1: Modelo Conceitual



## 2.3 Dicionário de Dados Conceitual

### — Entidade ALUNO —

---

- ALUNO: criada para guarda informações sobre aluno.

#### Atributo (s):

- Código: criado para guardar o número de código único de cada aluno.
- Nome: criado para armazenar o nome de cada aluno.
- Sexo: criado para guardar o sexo de um determinado aluno.
- Idade: criado para guardar a idade de um determinado aluno.
- Peso: criado para manter o dado do peso de um aluno.
- Altura: criado para manter a altura de um aluno.
- BF: derivado, criado para armazenar o índice da gordura corporal (*body fat*) do aluno.
- IMC: derivado, criado para reservar o índice da massa corpórea de um aluno.
- Telefone: criado para guardar o número de telefone do aluno.
- EstadoSaude: derivado, criado para guardar a descrição do estado de saúde de um aluno de acordo com o IMC: abaixo do peso, peso normal, acima do peso...
- Objetivo: criado para guardar a informação do objetivo do aluno: ganho de massa muscular, definição muscular, emagrecimento...

#### Relacionamento (s):

- Tem: relaciona Aluno com Matricula. Um aluno tem somente uma matrícula assim como uma determinada matrícula pode pertencer a apenas um aluno.
- Faz: relaciona Aluno com a entidade Compra. Cada aluno pode ou não fazer uma ou várias compras e uma dada compra é feita por somente um aluno, guardando-se a data de realização.
- Tira: relaciona Aluno à Professor e Medidas. Do aluno pode-se ou não tirar uma ou mais medidas por um ou mais professores e um professor pode ou não tirar uma ou mais medidas de um ou mais alunos, guardando-se a data da mensuração.
- Possui: relaciona Aluno com Mensalidade. Um aluno tem uma ou várias mensalidades (a cada mês uma nova) e cada mensalidade pertence a apenas

um aluno.

- Tem: relaciona Aluno à Treino. Um aluno tem um ou vários treinos e um treino pode pertencer a um ou mais alunos, guardando a data de início e duração do treino.

### — Entidade MATRICULA —

---

- MATRICULA: criada para guardar informações de matricula de aluno.

#### **Atributo (s):**

- DataAbertura: criado para guardar a data da abertura da matrícula.
- Codigo: criado para manter o número da matricula.
- DataTrancamento: criado para guardar a data de trancamento da matrícula.

#### **Relacionamento (s):**

- Tem: relaciona Matricula à entidade Aluno. Uma matricula pode pertencer a unicamente um aluno e um aluno tem unicamente uma matricula.

### — Entidade MENSALIDADE —

---

- MENSALIDADE: criada para guardar informações de mensalidades de aluno.

#### **Atributo (s):**

- Valor: criado para manter o valor da mensalidade.
- DataRecibo: criado para armazenar a data de recebimento do pagamento da mensalidade.
- DataPagamento: criado para guardar a próxima data de pagamento da mensalidade.

#### **Relacionamento (s):**

- Possui: relaciona Mensalidade à entidade Aluno. Uma mensalidade pertence a um único aluno e um determinado aluno tem uma ou muitas mensalidades (a cada mês uma nova).

### — Entidade TREINO —

---

- TREINO: criado para guardar informações de treinos.

#### **Atributo (s):**

- **Codigo:** criado para guardar o número de identificação do código do treino.
- **Tipo:** criado para armazenar informação do tipo de treino do aluno: AB, ABC ou ABCD...
- **DescansoEntreSeries:** criado para guardar o tempo de descanso entre cada série.
- **UsoDeCarga:** criado para armazenar o uso de carga padrão do treino: leve, moderada ou pesada.
- **DescansoEntreRepeticoes:** criado para guardar o tempo de descanso entre repetições do treino.
- **DescansoEntreCiclo:** criado para armazenar os dias de descanso ao término de um ciclo do tipo do treino.
- **Descrição:** criado para armazenar a descrição do treino: ganho de massa magra, definição...

#### **Relacionamento (s):**

- **MontaEsupervisiona:** relaciona Treino a entidade Professor e Exercício. O treino é montado e supervisionado por um ou mais professores que possui um ou mais exercícios e os exercícios são montados e supervisionados por um ou mais professores e estão presentes em um ou mais treinos.
- **Tem:** relaciona Treino a entidade Exercício. Um treino tem um ou muitos exercícios, assim como um exercício está contido em um ou mais treinos.
- **Tem:** relaciona Treino à Aluno. Um treino é tido por um ou mais alunos, bem como alunos têm um ou mais treinos guardando a data de início de prática do treino e a duração.

---

#### **Entidade EXERCICIO**

- **EXERCICIO:** criado para guardar informações de exercícios.

#### **Atributo (s):**

- **Tipo:** criado para guardar a sigla do dia do exercício executado: A, B, ou C...
- **Series:** criado para armazenar o número de séries de um determinado exercício.
- **Repeticoes:** criado para armazenar o número de repetições da série de um dado exercício.

- Nome: criado para guardar o nome do exercício.
- Musculo: criado para armazenar o nome do musculo trabalhado pelo exercício.
- DropSet: criado para armazenar uma informação da forma de execução do exercício: se Drop Set ou não.

#### **Relacionamento (s):**

- MontaEsupervisiona: relaciona Exercicio à entidade Treino e Professor. Um exercício é montado e supervisionado por um ou mais professores e são pertencentes a um ou muitos treinos e os treinos têm um ou muitos exercícios que são montados por um ou mais professores.
- Tem: relaciona Exercicio à Treino. Os exercícios estão presentes em um ou muitos treinos, assim como os treinos possuem um ou mais exercícios.

### **— Entidade FUNCIONARIO —**

- FUNCIONARIO: generalização das entidades Professor e Gerente, criada para guardar informações gerais do funcionário.

#### **Atributo (s):**

- CPF: criado para guardar o número do CPF do funcionário.
- Sexo: criado para guardar a informação do sexo do funcionário.
- Nome: criado para armazenar o nome do funcionário.
- Codigo: criado para armazenar o número do código de identificação do funcionário.
- DataPagamento: criado para guardar a data do pagamento do salário do funcionário.

#### **Relacionamento (s):**

- *Sem relacionamento.*

### **— Entidade PROFESSOR —**

- PROFESSOR: especialização da entidade Funcionario criada para especificar e guardar informações do professor (tipo de funcionário).

#### **Atributo (s):**

- Salario: criado para guardar a informação do valor do salário do professor.

- Especialidade: criado para guardar as especialidades do professor.

**Relacionamento (s):**

- Tira: relaciona Professor à entidade Aluno e à Medidas. Um professor pode ou não tirar uma ou mais medidas de um ou mais alunos, todavia as medidas de um ou mais alunos são tiradas por um ou mais professores, guarda-se a data da mensuração.
- MontaEsupervisiona: relaciona Professor com a entidade Treino e Exercício. Um professor monta e supervisiona um ou mais treinos com um ou mais exercícios e os exercícios são montados e supervisionados por um ou mais professores que estão presentes em um ou mais treinos.

— **Entidade GERENTE** —

- GERENTE: especialização da entidade Funcionario criada para especificar e guardar informações do gerente (tipo de funcionário).

**Atributo (s):**

- LucroMensal: derivado, criado para guardar o valor do lucro mensal do gerente, calculado pelas mensalidades de alunos frequentes e total do lucro de vendas realizadas em um determinado mês.

**Relacionamento (s):**

- Gerencia: relaciona Gerente à entidade Despesa. Um gerente gerencia uma ou várias despesas e as despesas são geridas por um ou mais gerentes.
- Realiza: relaciona Gerente com Compra. Um gerente pode ou não realizar uma ou várias compras e as compras são feitas por um ou mais gerentes, guardando-se a data da realização.

— **Entidade DESPESA** —

- DESPESA: criada para guardar informações de despesa.

**Atributo (s):**

- Descrição: criado para armazenar a descrição da despesa.
- Valor: criado para armazenar o valor de determinada despesa.
- Data: criado para reservar a data em que a despesa foi adquirida.

### **Relacionamento (s):**

- Gerencia: relaciona Despesa a entidade Gerente. Uma despesa é gerida por um ou vários gerentes e um gerente gerencia uma ou várias despesas.

### **— Entidade COMPRA —**

---

- COMPRA: generalização das entidades Avista e Aprazo criada para guardar informações gerais da compra.

### **Atributo (s):**

- Código: criado para armazenar o código de identificação de uma compra.
- CNPJouCPF: criado para armazenar o CNPJ do fornecedor de uma compra feita pelo gerente, ou CPF do aluno que fez a compra.
- Descrição: criado para manter dados da descrição da compra.
- Valor: criado para guardar o valor de uma determinada compra.

### **Relacionamento (s):**

- Faz: relaciona Compra com a entidade Aluno. Uma compra é feita por um e somente um aluno e um aluno faz ou não uma ou muitas compras, guardando a data da realização.
- Contem: relaciona Compra à Produto. Uma compra possui um ou mais produtos e um produto pode estar presente em uma ou mais compras.
- Realiza: relaciona Compra à entidade Gerente. Uma compra pode ser realizada por um ou vários gerentes e um gerente pode fazer zero ou muitas compras, guardando-se a data da compra.

### **— Entidade aPRAZO —**

---

- aPRAZO: especialização da entidade Compra criada para especificar e guardar informações do tipo de compra a prazo.

### **Atributo (s):**

- ValorParcela: criado para guardar o valor das parcelas de cada compra.
- Parcelas: criado para guardar o número de parcelas em que uma compra foi dividida.
- Quitada: criado para armazenar a informação de quitação do valor de todas as

parcelas de uma compra, se totalmente paga ou não.

**Relacionamento (s):**

→ *Sem relacionamento.*

---

— **Entidade aVISTA** —

- aVISTA: especialização da entidade Compra criada para especificar e guardar informações do tipo de compra à vista.

**Atributo (s):**

- Desconto: criado para guardar o valor do desconto de uma determinada compra.

**Relacionamento (s):**

→ *Sem relacionamento.*

---

— **Entidade PRODUTO** —

- PRODUTO: criada para guardar informações de produto.

**Atributo (s):**

- Preço: criado para manter o preço de um determinado produto.
- Quantidade: criado para guardar a quantidade de um mesmo produto.
- Nome: criado para guardar o nome de um dado produto.
- Marca: criado para guardar a marca de um certo produto.

**Relacionamento (s):**

→ Contém: relaciona Produto à entidade Compra. Um ou mais produtos pode estar presente em uma ou mais compras e uma compra pode conter um ou mais produtos.

---

— **Entidade MEDIDAS** —

- MEDIDAS: criada para guardar informações sobre medidas de aluno.

**Atributo (s):**

- Peitoral: criado para guardar a informação da medida em centímetros do peitoral.
- Abdome: criado para guardar informação da medida em centímetros do abdome.
- Deltoides: criado para guardar a informação das medidas em centímetros dos deltoides.

- CoxaEsquerda: criado para armazenar a informação da medida em centímetros da coxa esquerda.
- CoxaDireita: criado para armazenar a informação da medida em centímetros da coxa direita.
- AntebraçoEsquerdo: criado para guardar o dado da medida em centímetros do antebraço esquerdo.
- AntebraçoDireito: criado para guardar o dado da medida em centímetros do antebraço direito.
- BraçoEsquerdo: criado para armazenar o dado da medida em centímetros do braço esquerdo.
- BraçoDireito: criado para armazenar o dado da medida em centímetros do braço direito.
- PanturrilhaEsquerda: criado para guardar o dado da medida em centímetros da panturrilha esquerda.
- PanturrilhaDireita: criado para guardar o dado da medida em centímetros da panturrilha direita.

**Relacionamento (s):**

- Tira: relaciona Medida à entidade Aluno e Professor. As medidas de um ou mais alunos são feitas por um ou mais professores e um professor pode ou não tirar uma ou mais medidas de um ou mais alunos, guarda-se a data da realização.



## 2.4 Levantamento de Consultas

O sistema deve informar:

1. As compras que um aluno fez com o mês de referência.
2. As compras que um gerente fez com o mês de referência.
3. As compras pendentes de pagamento que um aluno fez com o mês de referência.
4. As compras pendentes de pagamento que um gerente fez com o mês de referência.
5. Os produtos comprados por um aluno de uma datada compra.
6. Os produtos comprados por um gerente de uma datada compra.
7. A data de pagamento da mensalidade do aluno.
8. A data que foi recebido o pagamento da mensalidade.
9. As medidas de um aluno em determinado mês.
10. As características do treino que um aluno está treinando em determinada data.
11. Os exercícios presentes no treino que um aluno está praticando.
12. O salário do professor.
13. A quantidade de alunos ativos na academia.
14. O lucro de um certo mês do gerente.
15. Informações das despesas de um determinado mês geridas pelo gerente.



**MODELAGEM LÓGICA**  
SISTEMA DE GERENCIAMENTO DE ACADEMIA FITNESS

### 3.1 Mapeamento Entidade Relacionamento

ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo);

MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento);

MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo, DataPagamento);

TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga, DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao);

ALUNO\_Tem\_TREINO (CodigoAluno, CodigoTreino, Duracao, DataInicio);

EXERCICIO (Codigo, Tipo, Series, Repeticoes, Nome, Musculo, DropSet);

TREINO\_Tem\_EXERCICIO (CodigoTreino, CodigoExercicio);

PROFESSOR (Codigo, CPF, Sexo, Nome, DataPagamento, Salario);

EspecialidadePROFESSOR (CodigoProfessor, Especialidade);

PROFESSOR\_EXERCICIO (CodigoProfessor, CodigoTreino, CodigoExercicio);

GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal);

DESPESA (Codigo, Descricao, Valor, Data);

GERENTE\_Gerencia\_DESPESA (CodigoGerente, CodigoDespesa);

COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descricao, Valor, Data);

COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas);

COMPRAaVISTA (Codigo, Desconto);

GERENTE\_Realiza\_COMPRA (CodigoGerente, CodigoCompra, Data);

PRODUTO (Codigo, Preço, Nome, Marca, Quantidade);

COMPRA\_Contem\_PRODUTO (CodigoCompra, CodigoProduto);

MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita);

PROFESSOR\_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno, Data);

### 3.2 Dicionário Lógico de Dados

ALUNO: Relação que armazena os dados de cada aluno da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que representa o código de identificação do aluno	Int	Números inteiros positivos	▪ Chave Primária
Nome	Atributo que representa o nome do aluno	String (100)	String (100)	▪ Não Nulo
Sexo	Atributo que representa a inicial do sexo do aluno	Char	Char	▪ Não Nulo
Idade	Atributo que representa a idade do aluno	Int	Números inteiros positivos	▪ Não Nulo
Peso	Atributo que representa o peso do aluno	Real	Números reais positivos	▪ Não Nulo
Altura	Atributo que representa a altura do aluno	Real	Números reais positivos	▪ Não Nulo
BF	Atributo que representa o percentual do índice de gordura corporal do aluno	String (4)	String (4)	▪ Não Nulo

IMC	Atributo que representa o índice de massa corporal do aluno	Real	Números reais positivos	▪ Não Nulo
Telefone	Atributo que representa o número de telefone do aluno	String (20)	String (20)	▪ Não Nulo
EstadoSaude	Atributo que representa o estado de saúde em que o aluno se encontra	String (50)	String (50)	▪ Não Nulo
Objetivo	Atributo que representa o objetivo principal do aluno	String (100)	String (100)	▪ Não Nulo

Tabela 1: Relação ALUNO

<b>MATRÍCULA:</b> Relação que armazena os dados da matrícula de cada aluno da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que representa o código da matrícula do aluno	Int	Números inteiros positivos	▪ Chave Primária
DataAbertura	Atributo que representa a data de realização da matrícula do aluno	Date	Date	▪ Não Nulo
DataTrancament o	Atributo que representa a data de trancamento de matrícula do aluno	Date	Date	▪ Sem Restrição

CodigoAluno	Atributo que armazena o código do aluno que realiza a matrícula	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> <li>▪ Único</li> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "ALUNO"</li> </ul>
-------------	---	-----	----------------------------	--

Tabela 2: Relação MATRICULA

MENSALIDADE: Relação que armazena os dados da mensalidade de cada aluno da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
Valor	Atributo que representa o valor da mensalidade paga pelo aluno	Real	Números reais positivos	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
DataRecibo	Atributo que representa a data de recebimento do valor correspondente à mensalidade do aluno	Date	Date	<ul style="list-style-type: none"> <li>▪ Sem restrição</li> </ul>
DataPagamento	Atributo que representa a data prevista para o pagamento da mensalidade do aluno	Date	Date	<ul style="list-style-type: none"> <li>▪ Sem Restrição</li> </ul>
CodigoAluno	Atributo que armazena o código do aluno que realiza	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> <li>▪ Único</li> <li>▪ Chave</li> </ul>

	a matrícula			estrangeira que referencia o atributo "Codigo" da relação "ALUNO"
Codigo	Atributo que representa o código da mensalidade do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave substituta criada para representar o código da mensalidade de um aluno</li> <li>▪ Chave Primária</li> </ul>

Tabela 3: Relação MENSALIDADE

TREINO: Relação que armazena os treinos de cada aluno da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que representa o código do treino do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave Primária</li> </ul>
Tipo	Atributo que representa o tipo de treino que será realizado pelo aluno	String (10)	String (10)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
DescansoEntreSeries	Atributo que representa o tempo de descanso entre as series realizadas pelo aluno	String (5)	String (5)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
UsoDeCarga	Atributo que representa a descrição do uso de carga do aluno	String (10)	String (10)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>

DescansoEntreRepeticoes	Atributo que representa o tempo de descanso entre as repetições realizadas pelo aluno	String (5)	String (5)	▪ Não Nulo
DescansoEntreCiclo	Atributo que representa a quantidade de dias de descanso entre os ciclos de exercícios realizados pelo aluno	Int	Int	▪ Não Nulo
Descricao	Atributo que representa uma breve descrição sobre os treinos realizados pelo aluno	String (200)	String (200)	▪ Sem Restrição

Tabela 4: Relação TREINO

ALUNO_Tem_TREINO: Relação que armazena os dados do relacionamento entre o aluno e o treino				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoAluno	Atributo que representa o código do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "ALUNO"</li> <li>▪ Chave Primária</li> </ul>
CodigoTreino	Atributo que representa o código do treino do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "TREINO"</li> </ul>



				▪ Chave Primária
Duração	Atributo que representa o tempo de duração do treino aluno	Time	Time	▪ Não Nulo
DataInicio	Atributo que representa a data que o aluno iniciou os treinos	Date	Date	▪ Não Nulo

Tabela 5: Relação ALUNO\_Tem\_TREINO

EXERCICIO: Relação que armazena os dados de cada exercício do aluno				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que representa o código do exercício do Aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave substituta criada para representar o código do exercício de um aluno</li> <li>▪ Chave Primária</li> </ul>
Tipo	Atributo que representa o tipo de exercício que o aluno vai realizar	String (5)	String (5)	▪ Não Nulo
Series	Atributo que representa o número de series realizadas pelo aluno em um exercício	Int	Números inteiros positivos	▪ Não Nulo

Repetições	Atributo que armazena o número de repetições de uma série de exercícios	Int	Números inteiros positivos	▪ Não Nulo
Nome	Atributo que representa o nome do exercício	String (100)	String (100)	▪ Não Nulo
Musculo	Atributo que representa o nome do músculo trabalhado no exercício	String (20)	String (20)	▪ Não Nulo
DropSet	Atributo que armazena a forma de execução do exercício	Boolean	Boolean	▪ Não Nulo

Tabela 6: Relação EXERCICIO

TREINO_Tem_EXERCICIO: Relação que armazena os dados do relacionamento entre o treino e o exercício				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoTreino	Atributo que representa o código da relação treino	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "TREINO"</li> <li>▪ Chave Primária</li> </ul>

CodigoExercicio	Atributo que representa o código da relação exercício	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “EXERCICIO”</li> <li>▪ Chave Primária</li> </ul>
-----------------	---	-----	----------------------------	---

Tabela 7: Relação TREINO\_Tem\_EXERCICIO

PROFESSOR: Relação que armazena os dados de cada professor da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
CPF	Atributo que representa o CPF do professor	String (14)	String (14)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> <li>▪ Único</li> </ul>
Sexo	Atributo que representa a sigla do sexo do professor	Char	Char	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Nome	Atributo que representa o nome do professor	String (100)	String (100)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Codigo	Atributo que armazena o código do professor	Int	Int	<ul style="list-style-type: none"> <li>▪ Chave Primária</li> </ul>
DataPagamento	Atributo que representa a data referente ao pagamento do professor	Date	Date	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Salario	Atributo que representa o valor do	Real	Números reais positivos	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>

	salário pago ao professor			
--	---------------------------	--	--	--

Tabela 8: Relação PROFESSOR

<b>EspecialidadePROFESSOR:</b> Relação que armazena os dados das especialidades do professor da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoProfessor	Atributo que armazena o código do professor	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "PROFESSOR"</li> <li>▪ Chave Primária</li> </ul>
Especialidade	Atributo que representa as especialidades do professor	String (100)	String (100)	<ul style="list-style-type: none"> <li>▪ Chave Primária</li> </ul>

Tabela 9: Relação EspecialidadePROFESSOR

<b>PROFESSOR_EXERCICIO:</b> Relação que armazena os dados do relacionamento entre o professor, o treino e o exercício				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoProfessor	Atributo que armazena o código do professor	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "PROFESSOR"</li> <li>▪ Chave Primária</li> </ul>

CodigoTreino	Atributo que armazena o código do treino	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “TREINO”</li> <li>▪ Chave Primária</li> </ul>
CodigoExercicio	Atributo que armazena o código do exercício	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “EXERCICIO”</li> <li>▪ Chave Primária</li> </ul>

Tabela 10: Relação PROFESSOR\_EXERCICIO

GERENTE: Relação que armazena os dados de cada gerente da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
CPF	Atributo que armazena o CPF do gerente da academia	String (14)	String (14)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> <li>▪ Único</li> </ul>
Sexo	Atributo que representa a sigla do sexo do gerente da academia	Char	Char	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Nome	Atributo que representa o nome do gerente da academia	String (100)	String (100)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Codigo	Atributo que representa o código do gerente da	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave Primária</li> </ul>

	academia			
DataPagamento	Atributo que representa a data do pagamento do gerente	Date	Date	▪ Não Nulo
LucroMensal	Atributo que armazena o lucro mensal gerado pela academia	Real	Números reais positivos	▪ Não Nulo

Tabela 11: Relação GERENTE

DESPESA: Relação que armazena os dados das despesas da academia				
Atributo	Descrição	Tipo	Domínio	Restrição
Descrição	Atributo que representa a descrição das despesas da academia	String (100)	String (100)	▪ Não Nulo
Valor	Atributo que representa o valor das despesas da academia	Real	Números reais positivos	▪ Não Nulo
Codigo	Atributo que armazena o código da despesa da academia	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave substituta criada para representar o código da despesa da academia</li> <li>▪ Chave Primária</li> </ul>

Data	Atributo que armazena a data em que a despesa foi realizada	Date	Date	▪ Não Nulo
------	---	------	------	------------

Tabela 12: Relação DESPESA

GERENTE_Gerencia_DESPESA: Relação que armazena os dados do relacionamento entre o gerente e a despesa				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoGerente	Atributo que armazena o código do gerente da academia	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "GERENTE"</li> <li>▪ Chave Primária</li> </ul>
CodigoDespesa	Atributo que armazena o código da despesa gerada pela academia	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "DESPESA"</li> <li>▪ Chave Primária</li> </ul>

Tabela 13: Relação GERENTE\_Gerencia\_DESPESA

COMPRA: Relação que armazena os dados de cada compra realizada				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que armazena o código da compra realizada	Int	Números inteiros positivos	▪ Chave Primária
CNPJouCPF	Atributo que representa o CNPJ	String (20)	String (20)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> <li>▪ Único</li> </ul>

	ou o CPF de quem realiza a compra			
Descrição	Atributo que representa a descrição de uma compra realizada	String (100)	String (100)	▪ Não Nulo
Valor	Atributo que armazena o valor de uma compra realizada	Real	Números reais positivos	▪ Não Nulo
CodigoAluno	Atributo que armazena o código do aluno	Int	Números inteiros positivos	▪ Chave estrangeira que referencia o atributo "Codigo" da relação "ALUNO"
Data	Atributo que armazena a data em que a compra foi realizada	Date	Date	▪ Não Nulo

Tabela 14: Relação COMPRA

COMPRAaPRAZO: Relação que armazena os dados das compras a prazo				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que armazena o código da compra a prazo realizada	Int	Números inteiros positivos	▪ Chave Primária
ValorParcela	Atributo que armazena o valor de cada parcela da	Real	Números reais positivos	▪ Não Nulo



	compra a prazo			
Quitada	Atributo que representa se a compra a prazo foi quitada ou não	Boolean	Boolean	▪ Não Nulo
Parcelas	Atributo que armazena a quantidade de parcelas da compra a prazo	Int	Números inteiros positivos	▪ Não Nulo;

Tabela 15: Relação COMPRAaPRAZO

COMPRAaVISTA: Relação que armazena os dados das compras a prazo				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que armazena o código da compra avista realizada	Int	Números inteiros positivos	▪ Chave Primária
Desconto	Atributo que representa o valor do desconto da compra avista	Real	Números reais positivos	▪ Não Nulo

Tabela 16: Relação COMPRAaVISTA

GERENTE_Realiza_COMPRA: Relação que armazena os dados do relacionamento entre o gerente e a compra				
Atributo	Descrição	Tipo	Domínio	Restrição

CodigoGerente	Atributo que armazena o código do gerente da academia	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “GERENTE”</li> <li>▪ Chave Primária</li> </ul>
CodigoCompra	Atributo que armazena o código da compra realizada	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “COMPRA”</li> <li>▪ Chave Primária</li> </ul>
Data	Atributo que representa a data em que o gerente efetuou a compra	Date	Date	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>

Tabela 17: Relação GERENTE\_Realiza\_COMPRA

PRODUTO: Relação que armazena os dados de cada produto				
Atributo	Descrição	Tipo	Domínio	Restrição
Codigo	Atributo que armazena o código identificador de cada produto	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave substituta criada para representar o código do produto</li> <li>▪ Chave Primária</li> </ul>
Preço	Atributo que armazena o valor do produto	Real	Números reais positivos	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>
Nome	Atributo que representa o nome do produto	String (100)	String (100)	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>

Marca	Atributo que representa a marca do produto	String (50)	String (50)	▪ Não Nulo
Quantidade	Atributo que representa a quantidade de produtos feitos na compra	Int	Números inteiros positivos	▪ Não Nulo

Tabela 18: Relação PRODUTO

COMPRA_Contem_PRODUTO: Relação que armazena os dados do relacionamento entre Compra e Produto				
Atributo	Descrição	Tipo	Domínio	Restrição
CodigoCompra	Atributo que armazena o código da compra	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "COMPRA"</li> <li>▪ Chave Primária</li> </ul>
CodigoProduto	Atributo que armazena o código do produto	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo "Codigo" da relação "PRODUTO"</li> <li>▪ Chave Primária</li> </ul>

Tabela 19: Relação COMPRA\_Contem\_PRODUTO

MEDIDAS: Relação que armazena os dados de todas as medidas do aluno				
Atributo	Descrição	Tipo	Domínio	Restrição

Abdome	Atributo que representa as medidas em cm do abdome do aluno	Real	Números reais positivos	▪ Não Nulo
CoxaDireita	Atributo que representa as medidas em cm da coxa direita do aluno	Real	Números reais positivos	▪ Não Nulo
CoxaEsquerda	Atributo que representa as medidas em cm da coxa esquerda do aluno	Real	Números reais positivos	▪ Não Nulo
BraçoDireito	Atributo que representa as medidas em cm do braço direito do aluno	Real	Números reais positivos	▪ Não Nulo
BraçoEsquerdo	Atributo que representa as medidas em cm do braço esquerdo do aluno	Real	Números reais positivos	▪ Não Nulo
Peitoral	Atributo que representa as medidas em cm do peitoral do aluno	Real	Números reais positivos	▪ Não Nulo
AntebraçoDireito	Atributo que representa as medidas em cm do antebraço direito do aluno	Real	Números reais positivos	▪ Não Nulo

AntebraçoEsquerdo	Atributo que representa as medidas em cm do antebraço esquerdo do aluno	Real	Números reais positivos	▪ Não Nulo
Deltoides	Atributo que representa as medidas em cm dos deltoides do aluno	Real	Números reais positivos	▪ Não Nulo
PanturrilhaEsquerda	Atributo que representa as medidas em cm da panturrilha esquerda do aluno	Real	Números reais positivos	▪ Não Nulo
PanturrilhaDireita	Atributo que representa as medidas em cm da panturrilha direita do aluno	Real	Números reais positivos	▪ Não Nulo
Codigo	Atributo que representa o código das medidas do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave substituta criada para representar o código das medidas</li> <li>▪ Chave Primária</li> </ul>

Tabela 20: Relação MEDIDAS

PROFESSOR_ALUNO: Relação que armazena os dados do relacionamento entre Professor, Medidas e Aluno				
Atributo	Descrição	Tipo	Domínio	Restrição

CodigoProfessor	Atributo que armazena o código do professor	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “PROFESSOR”</li> <li>▪ Chave Primária</li> </ul>
CodigoAluno	Atributo que armazena o código do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “ALUNO”</li> <li>▪ Chave Primária</li> </ul>
CodigoMedidas	Atributo que armazena o código das medidas do aluno	Int	Números inteiros positivos	<ul style="list-style-type: none"> <li>▪ Chave estrangeira que referencia o atributo “Codigo” da relação “MEDIDAS”</li> <li>▪ Chave Primária</li> </ul>
Data	Atributo que armazena a data em que as medidas foram tiradas	Date	Date	<ul style="list-style-type: none"> <li>▪ Não Nulo</li> </ul>

Tabela 21: Relação PROFESSOR\_ALUNO

### 3.3 Scripts de Criação SQL

```
CREATE TABLE ALUNO(  
    Codigo INT,  
    Nome VARCHAR(100) NOT NULL,  
    Sexo CHAR NOT NULL,  
    Idade INT NOT NULL,  
    Peso REAL NOT NULL,  
    Altura REAL NOT NULL,  
    BF VARCHAR(4) NOT NULL,  
    IMC REAL NOT NULL,  
    Telefone VARCHAR(20),  
    EstadoSaude VARCHAR(50) NOT NULL,  
    Objetivo VARCHAR(100) NOT NULL,  
    CONSTRAINT AlunoPK PRIMARY KEY (Codigo),  
    CHECK(Peso>0),  
    CHECK(Altura>0)  
);
```

```
CREATE TABLE MATRICULA(  
    Codigo INT,  
    CodigoAluno INT NOT NULL UNIQUE,  
    DataAbertura DATE NOT NULL,  
    DataTrancamento DATE,  
    CONSTRAINT MatriculaPK PRIMARY KEY (Codigo),  
    CONSTRAINT MatriculaFK FOREIGN KEY (CodigoAluno)  
        REFERENCES ALUNO(Codigo)  
    ON UPDATE CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE MENSALIDADE(  

```

```

Codigo INT,
CodigoAluno INT NOT NULL UNIQUE,
Valor REAL NOT NULL,
DataRecibo DATE,
DataPagamento DATE,
CONSTRAINT MensalidadePK PRIMARY KEY (Codigo),
CONSTRAINT MensalidadeFK FOREIGN KEY (CodigoAluno)
    REFERENCES ALUNO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
CHECK(Valor>0)
);

```

```

CREATE TABLE TREINO(
    Codigo INT,
    Tipo VARCHAR(10) NOT NULL,
    DescansoEntreSeries VARCHAR(5) NOT NULL,
    UsoDeCarga VARCHAR(10) NOT NULL,
    DescansoEntreRepeticoes VARCHAR(5) NOT NULL,
    DescansoEntreCiclo INT NOT NULL,
    Descricao VARCHAR(200),
    CONSTRAINT TreinoPK PRIMARY KEY (Codigo)
);

```

```

CREATE TABLE ALUNO_Tem_TREINO(
    CodigoAluno INT,
    CodigoTreino INT,
    Duraçao TIME NOT NULL,
    DataInicio DATE NOT NULL,
    CONSTRAINT Aluno_Tem_TreinoPK PRIMARY KEY (CodigoAluno,
    CodigoTreino),
    CONSTRAINT Aluno_Tem_TreinoFK FOREIGN KEY (CodigoAluno)

```



```

        REFERENCES ALUNO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT Aluno_Tem_TreinoFK2 FOREIGN KEY (Codigotreino)
        REFERENCES TREINO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE EXERCICIO(
    Codigo INT,
    Tipo VARCHAR(5) NOT NULL,
    Series INT NOT NULL,
    Repetições INT NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    Musculo VARCHAR(20) NOT NULL,
    DropSet BOOLEAN NOT NULL,
    CONSTRAINT ExercicioPK PRIMARY KEY (Codigo),
    CHECK(Series>0),
    CHECK(Repetições>6)
);

```

```

CREATE TABLE TREINO_Tem_EXERCICIO(
    CodigoTreino INT,
    CodigoExercicio INT,
    CONSTRAINT Treino_Tem_ExercicioPK PRIMARY KEY (CodigoTreino,
CodigoExercicio),
    CONSTRAINT Treino_Tem_ExercicioFK FOREIGN KEY (CodigoTreino)
        REFERENCES TREINO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT Treino_Tem_ExercicioFK2 FOREIGN KEY (CodigoExercicio)
        REFERENCES EXERCICIO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT

```

);

```
CREATE TABLE PROFESSOR(  
    CPF VARCHAR(14) NOT NULL UNIQUE,  
    Sexo CHAR NOT NULL,  
    Nome VARCHAR(100) NOT NULL,  
   Codigo INT,  
    DataPagamento DATE NOT NULL,  
    Salario REAL NOT NULL,  
    CONSTRAINT ProfessorPK PRIMARY KEY (Codigo),  
    CHECK(Salario>0)  
);
```

```
CREATE TABLE EspecialidadePROFESSOR(  
    CodigoProfessor INT,  
    Especialidade VARCHAR(100),  
    CONSTRAINT EspecialidadeProfessorPK PRIMARY KEY (CodigoProfessor,  
    Especialidade),  
    CONSTRAINT EspecialidadeProfessorFK FOREIGN KEY (CodigoProfessor)  
        REFERENCES PROFESSOR(Codigo)  
    ON UPDATE CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE PROFESSOR_EXERCICIO(  
    CodigoProfessor INT,  
    CodigoTreino INT,  
    CodigoExercicio INT,  
    CONSTRAINT Professor_ExercicioPK PRIMARY KEY (CodigoProfessor,  
    CodigoTreino, CodigoExercicio),  
    CONSTRAINT Professor_ExercicioFK FOREIGN KEY (CodigoProfessor)  
        REFERENCES PROFESSOR(Codigo)
```

```

ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT Professor_ExercicioFK2 FOREIGN KEY (CodigoTreino)
    REFERENCES TREINO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT Professor_ExercicioFK3 FOREIGN KEY (CodigoExercicio)
    REFERENCES EXERCICIO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE GERENTE(
    Codigo INT,
    CPF VARCHAR(14) NOT NULL UNIQUE,
    Sexo CHAR NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    DataPagamento DATE NOT NULL,
    LucroMensal REAL NOT NULL,
    CONSTRAINT GerentePK PRIMARY KEY (Codigo)
);

```

```

CREATE TABLE DESPESA(
    Codigo INT,
    Descrição VARCHAR(100) NOT NULL,
    Valor REAL NOT NULL,
    Data DATE NOT NULL,
    CONSTRAINT DespesaPK PRIMARY KEY (Codigo)
);

```

```

CREATE TABLE GERENTE_Gerencia_DESPESA(
    CodigoGerente INT,
    CodigoDespesa INT,
    CONSTRAINT GERENTE_Gerencia_DESPESAPK PRIMARY KEY

```

```

(CodigoGerente, CodigoDespesa),
CONSTRAINT Gerente_Gerencia_DespesaFK FOREIGN KEY (CodigoGerente)
    REFERENCES GERENTE(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT Gerente_Gerencia_DespesaFK2 FOREIGN KEY
(CodigoDespesa)
    REFERENCES DESPESA(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE COMPRA(
    Codigo INT,
    CodigoAluno INT,
    CNPJouCPF VARCHAR(20) NOT NULL UNIQUE,
    Descrição VARCHAR(100) NOT NULL,
    Valor REAL NOT NULL,
    Data DATE NOT NULL,
    CONSTRAINT CompraPK PRIMARY KEY (Codigo),
    CONSTRAINT CompraFK FOREIGN KEY (CodigoAluno)
        REFERENCES ALUNO(Codigo)
ON UPDATE CASCADE ON DELETE RESTRICT,
CHECK(Valor>0)
);

```

```

CREATE TABLE COMPRAaPRAZO(
    Codigo INT,
    ValorParcela REAL NOT NULL,
    Quitada BOOLEAN NOT NULL,
    Parcelas INT NOT NULL,
    CONSTRAINT CompraAprazoPK PRIMARY KEY (Codigo),
CHECK(ValorParcela>0),

```

```

        CHECK(Parcelas>0 AND Parcelas<=5)
    );

CREATE TABLE COMPRAaVISTA(
    Codigo INT,
    Desconto REAL NOT NULL,
    CONSTRAINT CompraAvistaPK PRIMARY KEY (Codigo),
    CHECK(Desconto>0)
);

CREATE TABLE GERENTE_Realiza_COMPRA(
    CodigoGerente INT,
    CodigoCompra INT,
    Data DATE NOT NULL,
    CONSTRAINT Gerente_Realiza_CompraPK PRIMARY KEY (CodigoGerente,
    CodigoCompra),
    CONSTRAINT Gerente_Realiza_CompraFK FOREIGN KEY (CodigoGerente)
        REFERENCES GERENTE(Codigo)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT Gerente_Realiza_CompraFK2 FOREIGN KEY (CodigoCompra)
        REFERENCES COMPRA(Codigo)
        ON UPDATE CASCADE ON DELETE RESTRICT
);

CREATE TABLE PRODUTO(
    Codigo INT,
    Preço REAL NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    Marca VARCHAR(50) NOT NULL,
    Quantidade INT NOT NULL,
    CONSTRAINT PRODUTOPK PRIMARY KEY (Codigo),

```

```

        CHECK(Preço>0),
        CHECK(Quantidade>0)
    );

CREATE TABLE COMPRA_Contem_PRODUTO(
    CodigoCompra INT,
    CodigoProduto INT,
    CONSTRAINT Compra_Contem_ProdutoPK PRIMARY KEY (CodigoCompra,
    CodigoProduto),
    CONSTRAINT Compra_Contem_ProdutoFK FOREIGN KEY (CodigoCompra)
        REFERENCES COMPRA(Codigo)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT Compra_Contem_ProdutoFK2 FOREIGN KEY (CodigoProduto)
        REFERENCES Produto(Codigo)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE MEDIDAS(
    Codigo INT,
    Abdome REAL NOT NULL,
    CoxaDireita REAL NOT NULL,
    CoxaEsquerda REAL NOT NULL,
    BraçoDireito REAL NOT NULL,
    BraçoEsquerdo REAL NOT NULL,
    Peitoral REAL NOT NULL,
    AntebraçoDireito REAL NOT NULL,
    AntebraçoEsquerdo REAL NOT NULL,
    Deltoides REAL NOT NULL,
    PanturrilhaEsquerda REAL NOT NULL,
    PanturrilhaDireita REAL NOT NULL,
    CONSTRAINT MEDIDASPK PRIMARY KEY (Codigo)
);

```

);

```
CREATE TABLE PROFESSOR_ALUNO(  
    CodigoProfessor INT,  
    CodigoMedidas INT,  
    CodigoAluno INT,  
    Data DATE NOT NULL,  
    CONSTRAINT Professor_AlunoPK PRIMARY KEY (CodigoProfessor,  
    CodigoAluno, CodigoMedidas),  
    CONSTRAINT Professor_AlunoFK FOREIGN KEY (CodigoProfessor)  
        REFERENCES PROFESSOR(Codigo)  
    ON UPDATE CASCADE ON DELETE RESTRICT,  
    CONSTRAINT Professor_AlunoFK2 FOREIGN KEY (CodigoAluno)  
        REFERENCES ALUNO(Codigo)  
    ON UPDATE CASCADE ON DELETE RESTRICT,  
    CONSTRAINT Professor_AlunoFK3 FOREIGN KEY (CodigoMedidas)  
        REFERENCES MEDIDAS(Codigo)  
    ON UPDATE CASCADE ON DELETE RESTRICT  
);
```

### 3.4 Scripts de Inserção SQL

#### Relação ALUNO

```
INSERT INTO ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo)
```

```
VALUES (11111, 'João de Sousa', 'M', 20, 80.8, 1.70, '20%', 27.7, '(83) 99834-0672', 'Acima do Peso', 'Emagrecimento');
```

```
INSERT INTO ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo)
```

```
VALUES (11112, 'Pedro da Silva', 'M', 25, 75.3, 1.65, '18%', 27.5, '(83) 99939-4097', 'Acima do Peso', 'Emagrecimento');
```

```
INSERT INTO ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo)
```

```
VALUES (11113, 'Maria do Carmo Pereira', 'F', 19, 58.7, 1.62, '15%', 22.1, '(83) 99834-0672', 'Peso Normal', 'Hipertrofia Muscular');
```

```
INSERT INTO ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo)
```

```
VALUES (11114, 'Francisco Carlos de Araújo', 'M', 28, 85.4, 1.74, '26%', 28.1, '(83) 99476-9835', 'Acima do Peso', 'Emagrecimento');
```

```
INSERT INTO ALUNO (Codigo, Nome, Sexo, Idade, Peso, Altura, BF, IMC, Telefone, EstadoSaude, Objetivo)
```

```
VALUES (11115, 'Joana Pereira de Sousa', 'F', 18, 56.2, 1.65, '12%', 20.6, '(83) 99855-4568', 'Peso Normal', 'Hipertrofia Muscular');
```

#### Relação MATRICULA

```
INSERT INTO MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento)
```

```
VALUES (22222, 11111, '02/01/2017', '02/03/2017');
```

```
INSERT INTO MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento)
```

```
VALUES (22223, 11112, '05/01/2017', null);
```

```
INSERT INTO MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento)
```



```
VALUES (22224, 11113, '07/01/2017', null);
INSERT INTO MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento)
VALUES (22225, 11114, '10/01/2017', null);
INSERT INTO MATRICULA (Codigo, CodigoAluno, DataAbertura, DataTrancamento)
VALUES (22226, 11115, '15/01/2017', null);
```

### **Relação MENSALIDADE**

```
INSERT INTO MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo,
DataPagamento)
VALUES (33333, 11111, 50, '15/02/2017', '02/02/2017');
INSERT INTO MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo,
DataPagamento)
VALUES (33334, 11112, 50, '05/03/2017', '05/03/2017');
INSERT INTO MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo,
DataPagamento)
VALUES (33335, 11113, 50, '09/03/2017', '07/03/2017');
INSERT INTO MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo,
DataPagamento)
VALUES (33336, 11114, 50, '18/03/2017', '10/03/2017');
INSERT INTO MENSALIDADE (Codigo, CodigoAluno, Valor, DataRecibo,
DataPagamento)
VALUES (33337, 11115, 50, '15/03/2017', '15/03/2017');
```

### **Relação TREINO**

```
INSERT INTO TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga,
DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao)
VALUES (44444, 'ABC', '30s', 'Moderada', '1m', 2, null);
INSERT INTO TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga,
DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao)
VALUES (44445, 'A', '30s', 'Leve', '1m', 1, null);
```

```

INSERT INTO TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga,
DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao)
VALUES (44446, 'ABCD', '30s', 'Pesada', '1m', 3, null);
INSERT INTO TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga,
DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao)
VALUES (44447, 'CDE', '30s', 'Moderada', '1m', 2, null);
INSERT INTO TREINO (Codigo, Tipo, DescansoEntreSeries, UsoDeCarga,
DescansoEntreRepeticoes, DescansoEntreCiclo, Descricao)
VALUES (44448, 'ABCDE', '45s', 'Pesada', '2m', 4, null);

```

### **Relação ALUNO\_Tem\_TREINO**

```

INSERT INTO ALUNO_Tem_TREINO (CodigoAluno, CodigoTreino, Duração,
DataInicio)
VALUES (11111, 44444, '01:30', '02/02/2017');
INSERT INTO ALUNO_Tem_TREINO (CodigoAluno, CodigoTreino, Duração,
DataInicio)
VALUES (11112, 44445, '01:00', '05/03/2017');
INSERT INTO ALUNO_Tem_TREINO (CodigoAluno, CodigoTreino, Duração,
DataInicio)
VALUES (11113, 44446, '01:00', '07/04/2017');
INSERT INTO ALUNO_Tem_TREINO (CodigoAluno, CodigoTreino, Duração,
DataInicio)
VALUES (11114, 44447, '01:00', '10/03/2017');
INSERT INTO ALUNO_Tem_TREINO (CodigoAluno, CodigoTreino, Duração,
DataInicio)
VALUES (11115, 44448, '01:30', '15/03/2017');

```

### **Relação EXERCICIO**

```

INSERT INTO EXERCICIO (Codigo, Tipo, Series, Repeticoes, Nome, Musculo,
DropSet)
VALUES (55555, 'A', 5, 10, 'Rosca Martelo', 'Bíceps', true);

```

```

INSERT INTO EXERCICIO (Codigo, Tipo, Series, Repetições, Nome, Musculo,
DropSet)
VALUES (55556, 'B', 5, 15, 'Rosca Direta', 'Bíceps', true);
INSERT INTO EXERCICIO (Codigo, Tipo, Series, Repetições, Nome, Musculo,
DropSet)
VALUES (55557, 'C', 5, 15, 'Agachamento com Barra', 'Coxas', false);
INSERT INTO EXERCICIO (Codigo, Tipo, Series, Repetições, Nome, Musculo,
DropSet)
VALUES (55558, 'D', 5, 10, 'Abdominal na Máquina', 'Abdome', true);
INSERT INTO EXERCICIO (Codigo, Tipo, Series, Repetições, Nome, Musculo,
DropSet)
VALUES (55559, 'E', 5, 10, 'Panturrilha Sentado', 'Panturrilhas', false);

```

#### **Relação TREINO\_Tem\_EXERCICIO**

```

INSERT INTO TREINO_Tem_EXERCICIO (CodigoTreino, CodigoExercicio)
VALUES (44444, 55555);
INSERT INTO TREINO_Tem_EXERCICIO (CodigoTreino, CodigoExercicio)
VALUES (44445, 55556);
INSERT INTO TREINO_Tem_EXERCICIO (CodigoTreino, CodigoExercicio)
VALUES (44446, 55557);
INSERT INTO TREINO_Tem_EXERCICIO (CodigoTreino, CodigoExercicio)
VALUES (44447, 55558);
INSERT INTO TREINO_Tem_EXERCICIO (CodigoTreino, CodigoExercicio)
VALUES (44448, 55559);

```

#### **Relação PROFESSOR**

```

INSERT INTO PROFESSOR (CPF, Sexo, Nome, Codigo, DataPagamento, Salario)
VALUES ('111.111.111-01', 'M', 'José Ricardo Oliveira', 66666, '01/04/2017', 1000);
INSERT INTO PROFESSOR (CPF, Sexo, Nome, Codigo, DataPagamento, Salario)
VALUES ('222.222.333-02', 'M', 'André de Sousa Dias', 66667, '02/04/2017', 1000);
INSERT INTO PROFESSOR (CPF, Sexo, Nome, Codigo, DataPagamento, Salario)

```

```
VALUES ('333.333.333-03', 'M', 'Marcos Rogerio da Costa', 66668, '03/04/2017', 1000);
INSERT INTO PROFESSOR (CPF, Sexo, Nome, Codigo, DataPagamento, Salario)
VALUES ('444.444.444-04', 'M', 'Alexandre da Silva Cartaxo', 66669, '04/04/2017',
1000);
INSERT INTO PROFESSOR (CPF, Sexo, Nome, Codigo, DataPagamento, Salario)
VALUES ('555.555.555-05', 'F', 'Andréia Rodrigues de Oliveira', 66610, '05/04/2017',
1500);
```

### **Relação EspecialidadePROFESSOR**

```
INSERT INTO EspecialidadePROFESSOR (CodigoProfessor, Especialidade)
VALUES (66666, 'Educador Físico');
INSERT INTO EspecialidadePROFESSOR (CodigoProfessor, Especialidade)
VALUES (66667, 'Educador Físico');
INSERT INTO EspecialidadePROFESSOR (CodigoProfessor, Especialidade)
VALUES (66668, 'Educador Físico');
INSERT INTO EspecialidadePROFESSOR (CodigoProfessor, Especialidade)
VALUES (66669, 'Educador Físico');
INSERT INTO EspecialidadePROFESSOR (CodigoProfessor, Especialidade)
VALUES (66610, 'Personal Trainer');
```

### **Relação PROFESSOR\_EXERCICIO**

```
INSERT INTO PROFESSOR_EXERCICIO (CodigoProfessor, CodigoTreino,
CodigoExercicio)
VALUES (66666, 44444, 55555);
INSERT INTO PROFESSOR_EXERCICIO (CodigoProfessor, CodigoTreino,
CodigoExercicio)
VALUES (66667, 44445, 55556);
INSERT INTO PROFESSOR_EXERCICIO (CodigoProfessor, CodigoTreino,
CodigoExercicio)
VALUES (66668, 44446, 55557);
```

```

INSERT INTO PROFESSOR_EXERCICIO (CodigoProfessor, CodigoTreino,
CodigoExercicio)
VALUES (66669, 44447, 55558);
INSERT INTO PROFESSOR_EXERCICIO (CodigoProfessor, CodigoTreino,
CodigoExercicio)
VALUES (66610, 44448, 55559);

```

### **Relação GERENTE**

```

INSERT INTO GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal)
VALUES (77777, '666.666.666-06', 'M', 'Felipe Augusto Vilela', '01/04/2017', 10000);
INSERT INTO GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal)
VALUES (77778, '777.777.777-07', 'M', 'Sérgio Brito Ferrari', '02/04/2017', 10000);
INSERT INTO GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal)
VALUES (77779, '888.888.888-08', 'F', 'Valesca Maria da Costa', '03/04/2017', 10500);
INSERT INTO GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal)
VALUES (77710, '999.999.999-09', 'M', 'Cristovão de Lima', '04/04/2017', 10000);
INSERT INTO GERENTE (Codigo, CPF, Sexo, Nome, DataPagamento, LucroMensal)
VALUES (77711, '101.101.101-10', 'M', 'Fernando Pereira Belém', '05/04/2017', 10000);

```

### **Relação DESPESA**

```

INSERT INTO DESPESA (Codigo, Descrição, Valor, Data)
VALUES (88888, 'Água, Energia', 500, '01/04/2017');
INSERT INTO DESPESA (Codigo, Descrição, Valor, Data)
VALUES (88889, 'Água, Energia', 600, '02/04/2017');
INSERT INTO DESPESA (Codigo, Descrição, Valor, Data)
VALUES (88810, 'Água, Energia', 400, '03/04/2017');
INSERT INTO DESPESA (Codigo, Descrição, Valor, Data)
VALUES (88811, 'Água, Energia', 700, '04/04/2017');
INSERT INTO DESPESA (Codigo, Descrição, Valor, Data)
VALUES (88812, 'Água, Energia', 200, '05/04/2017');

```

### **Relação GERENTE\_Gerencia\_DESPESA**

```
INSERT INTO GERENTE_Gerencia_DESPESA (CodigoGerente, CodigoDespesa)
VALUES (77777, 88888);
INSERT INTO GERENTE_Gerencia_DESPESA (CodigoGerente, CodigoDespesa)
VALUES (77778, 88889);
INSERT INTO GERENTE_Gerencia_DESPESA (CodigoGerente, CodigoDespesa)
VALUES (77779, 88810);
INSERT INTO GERENTE_Gerencia_DESPESA (CodigoGerente, CodigoDespesa)
VALUES (77710, 88811);
INSERT INTO GERENTE_Gerencia_DESPESA (CodigoGerente, CodigoDespesa)
VALUES (77711, 88812);
```

### **Relação COMPRA**

```
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99999, 11111, '666.666.666-06', 'Produto Emagrecedor', 200, '01/03/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99910, 11112, '777.777.777-07', 'Produto Emagrecedor', 300, '02/03/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99911, 11113, '888.888.888-08', 'Produto Energético', 150, '05/03/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99912, 11114, '999.999.999-09', 'Produto Termogênico', 180, '08/03/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99913, 11115, '101.101.101-10', 'Produto Proteico', 450, '17/03/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99914, null, '11.111.111/1111-11', 'Produto Proteico', 2000, '19/02/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99915, null, '22.222.222/2222-22', 'Produto Emagrecedor', 1500,
'25/02/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
VALUES (99916, null, '33.333.333/3333-33', 'Produto Energético', 1000, '01/04/2017');
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)
```

```
VALUES (99917, null, '44.444.444/4444-44', 'Produto Termogênico', 800, '03/04/2017');  
INSERT INTO COMPRA (Codigo, CodigoAluno, CNPJouCPF, Descrição, Valor, Data)  
VALUES (99918, null, '55.555.555/5555-55', 'Produto Proteico', 2500, '06/04/2017');
```

### **Relação COMPRAaPRAZO**

```
INSERT INTO COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas)  
VALUES (99999, 50, false, 4);  
INSERT INTO COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas)  
VALUES (99912, 45, false, 4);  
INSERT INTO COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas)  
VALUES (99913, 90, false, 5);  
INSERT INTO COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas)  
VALUES (99914, 400, false, 5);  
INSERT INTO COMPRAaPRAZO (Codigo, ValorParcela, Quitada, Parcelas)  
VALUES (99915, 375, false, 4);
```

### **Relação COMPRAaVISTA**

```
INSERT INTO COMPRAaVISTA (Codigo, Desconto)  
VALUES (99910, 10);  
INSERT INTO COMPRAaVISTA (Codigo, Desconto)  
VALUES (99911, 10);  
INSERT INTO COMPRAaVISTA (Codigo, Desconto)  
VALUES (99916, 20);  
INSERT INTO COMPRAaVISTA (Codigo, Desconto)  
VALUES (99917, 10);  
INSERT INTO COMPRAaVISTA (Codigo, Desconto)  
VALUES (99918, 30);
```

### **Relação GERENTE\_Realiza\_COMPRA**

```
INSERT INTO GERENTE_Realiza_COMPRA (CodigoGerente, CodigoCompra, Data)  
VALUES (77777, 99914, '19/02/2017');
```

```

INSERT INTO GERENTE_Realiza_COMPRA (CodigoGerente, CodigoCompra, Data)
VALUES (77778, 99915, '25/02/2017');
INSERT INTO GERENTE_Realiza_COMPRA (CodigoGerente, CodigoCompra, Data)
VALUES (77779, 99916, '01/04/2017');
INSERT INTO GERENTE_Realiza_COMPRA (CodigoGerente, CodigoCompra, Data)
VALUES (77710, 99917, '03/04/2017');
INSERT INTO GERENTE_Realiza_COMPRA (CodigoGerente, CodigoCompra, Data)
VALUES (77711, 99918, '06/04/2017');

```

### **Relação PRODUTO**

```

INSERT INTO PRODUTO (Codigo, Preço, Nome, Marca, Quantidade)
VALUES (10101, 450, 'Whey Protein', 'Optimum Nutrition', 1);
INSERT INTO PRODUTO (Codigo, Preço, Nome, Marca, Quantidade)
VALUES (20202, 300, 'BCAA Recovery', 'Athletica Nutrition', 1);
INSERT INTO PRODUTO (Codigo, Preço, Nome, Marca, Quantidade)
VALUES (30303, 200, 'Cactinea', 'Nutreo', 1);
INSERT INTO PRODUTO (Codigo, Preço, Nome, Marca, Quantidade)
VALUES (40404, 150, 'Turbo Force', 'Uni Flora', 1);
INSERT INTO PRODUTO (Codigo, Preço, Nome, Marca, Quantidade)
VALUES (50505, 180, 'Ultimate Fire Black', 'Max Titanium', 1);

```

### **Relação COMPRA\_Contem\_PRODUTO**

```

INSERT INTO COMPRA_Contem_PRODUTO (CodigoCompra, CodigoProduto)
VALUES (99999, 10101);
INSERT INTO COMPRA_Contem_PRODUTO (CodigoCompra, CodigoProduto)
VALUES (99910, 20202);
INSERT INTO COMPRA_Contem_PRODUTO (CodigoCompra, CodigoProduto)
VALUES (99911, 30303);
INSERT INTO COMPRA_Contem_PRODUTO (CodigoCompra, CodigoProduto)
VALUES (99912, 40404);
INSERT INTO COMPRA_Contem_PRODUTO (CodigoCompra, CodigoProduto)

```



VALUES (99913, 50505);

### **Relação MEDIDAS**

INSERT INTO MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita)

VALUES (12345, 90, 54, 53, 36, 38, 100, 25, 26, 50, 40, 38);

INSERT INTO MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita)

VALUES (67891, 85, 52, 50, 37, 36, 97, 24, 23, 60, 39, 38);

INSERT INTO MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita)

VALUES (10111, 60, 52, 51, 24, 25, 88, 22, 23, 40, 36, 35);

INSERT INTO MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita)

VALUES (12131, 95, 55, 54, 39, 38, 102, 29, 28, 52, 33, 34);

INSERT INTO MEDIDAS (Codigo, Abdome, CoxaDireita, CoxaEsquerda, BraçoDireito, BraçoEsquerdo, Peitoral, AntebraçoDireito, AntebraçoEsquerdo, Deltoides, PanturrilhaEsquerda, PanturrilhaDireita)

VALUES (14151, 62, 51, 52, 23, 24, 86, 24, 25, 43, 32, 31);

### **Relação PROFESSOR\_ALUNO**

INSERT INTO PROFESSOR\_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno, Data)

VALUES (66666, 12345, 11111, '01/04/2017');

INSERT INTO PROFESSOR\_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno, Data)

VALUES (66667, 67891, 11112, '02/04/2017');

```
INSERT INTO PROFESSOR_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno,  
Data)  
VALUES (66668, 10111, 11113, '03/04/2017');  
INSERT INTO PROFESSOR_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno,  
Data)  
VALUES (66669, 12131, 11114, '04/04/2017');  
INSERT INTO PROFESSOR_ALUNO (CodigoProfessor, CodigoMedidas, CodigoAluno,  
Data)  
VALUES (66610, 14151, 11115, '05/04/2017');
```