	Ano Letivo: 20251	Turno: Noturno	Local: A definir
	Curso: ENGENHARIA DA COMPUTAÇÃO		Turma: 3ª Fase
Disciplina: BANCO DE DADOS II		Professor(a): JORGE LUIZ DA SILVA	

TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM

Projeto de banco de dados para um sistema de << Insira o nome do sistema aqui >>

João Victor da Rosa Falcão + <https://github.com/Tecleino>

Thalysson Dos Santos + <https://github.com/thalysson98>

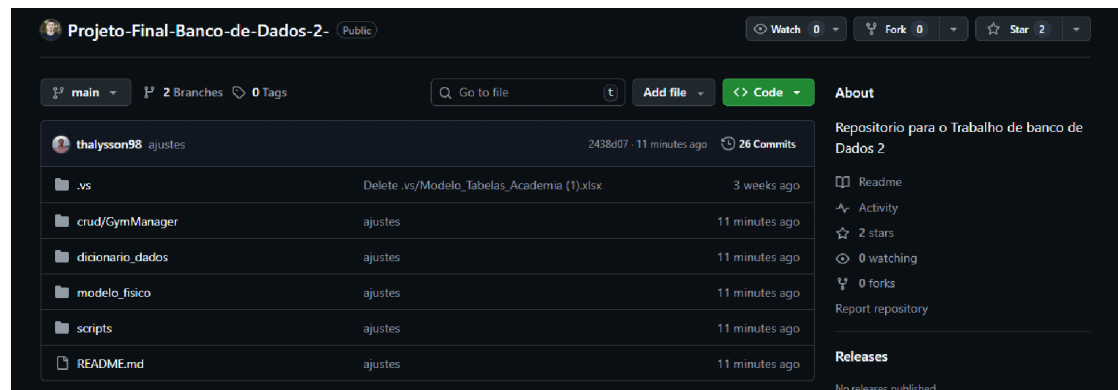
Leticia Vociekoski + <https://github.com/leticia-vociekoski>

Luiz Coral + <https://github.com/Luizcoral>

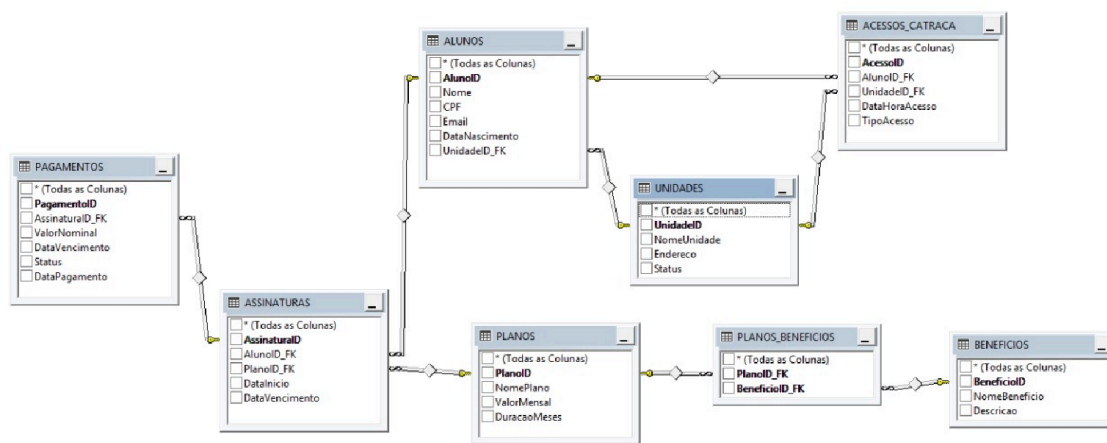
Criciúma, 30/11/2025

URL do projeto no GitHub

<https://github.com/Luizcoral/Projeto-Final-Banco-de-Dados-2->



Modelo ER Físico



Dicionário de Dados

Tabela	BENEFICIOS					
Descrição	Armazena os benefícios extras que podem estar associados a um plano.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
BeneficioID	int	1 – 2.147.483.647	NOT NULL	X		Identificador único do benefício
NomeBeneficio	varchar(200)	1 - 200	NOT NULL			Nome do benefício
Descricao	varchar(255)	1 - 255	NULL			Descrição detalhada do benefício
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Beneficios	X		X	BeneficioID		

Tabela	PLANOS
Descrição	Armazena os diferentes planos oferecidos pela academia.

Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
PlanoID	int	1 – 2.147.483.647	NOT NULL	X		Identificador único do benefício
NomeBeneficio	varchar(100)	1 - 100	NOT NULL			Nome do plano
ValorMensal	decimal(10,2)	>0	NOT NULL			Valor mensal do plano
DuracaoMeses	int	>0	NOT NULL			Duração do plano em meses
Status	char(1)	A/I	NOT NULL, DEFAULT 'A'			Status ativo ou inativo
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Planos	X		X	PlanoID		
UQ_NomePlano		X	X	NomePlano		

Tabela	PLANOS_BENEFICIOS					
Descrição	Tabela associativa que liga planos aos benefícios inclusos.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
PlanoID_FK	int	-	NOT NULL	x	x	Referência ao plano
BeneficioID_FK	int	-	NOT NULL	x	x	Referência ao benefício
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_PlanosBeneficios	X		X	PlanoID_FK, BeneficioID_FK		

Tabela	UNIDADES
---------------	----------

Descrição	Armazena informações sobre cada unidade física da academia.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
UnidadeID	int	1 – 2.147.483.647	NOT NULL	X		Identificador único da unidade
NomeUnidade	varchar(200)	1 - 150	NOT NULL			Nome da unidade
Endereco	varchar(255)	1 - 255	NOT NULL			Endereço da unidade
Status	char(1)	A/I	NOT NULL, DEFAULT 'A'			Status ativo/inativo
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Unidades	X		X	UnidadeID		
UQ_NomeUnidade		X	X	NomeUnidade		

Tabela	ALUNOS					
Descrição	Armazena dados dos alunos cadastrados.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
AlunoID	int		NOT NULL	X		Identificador único do aluno
Nome	varchar(150)	1 - 150	NOT NULL			Nome completo
CPF	varchar(11)	11	NOT NULL			CPF único
Email	varchar(100)	5 - 100	NOT NULL			Email do aluno
DataNascimento	date		NOT NULL			Data de nascimento
DataCadastro	datetime		NOT NULL DEFAULT GETDATE()			Data de criação do cadastro
UnidadeID_FK	int		NOT NULL		X	Unidade onde o aluno está matriculado

Índice				
Nome do índice	Clustered	NonClustered	Unique	Colunas
PK_Alunos	X		X	AlunoID
UQ_CPF		X	X	CPF
UQ_Email		X	X	Email
IDX_UnidadeAluno		X		UnidadeID_FK

Tabela	ASSINATURAS					
Descrição	Representa a contratação de um plano por um aluno.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
AssinaturalID	int		NOT NULL	X		Identificador da assinatura
AlunoID_FK	int		NOT NULL		X	Aluno contratante
PlanoID_FK	int		NOT NULL		X	Plano contratado
DataInicio	date		NOT NULL			Data de início da assinatura
DataVencimento	date		NOT NULL			Próximo vencimento
Status	char(1)	A/C/V	NOT NULL DEFAULT 'A'			Status: Ativa / Cancelada / Vencida
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Assinaturas	X		X	AssinaturalID		
IDX_Assinatura_Aluno		X		AlunoID_FK		
IDX_Assinatura_Plano		X		PlanoID_FK		
IDX_Assinatura_Status		X		Status		

Tabela	PAGAMENTOS					
Descrição	Armazena os pagamentos gerados a partir das assinaturas.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
PagamentoID	int		NOT NULL	X		Identificador do pagamento
AssinaturalID_FK	int		NOT NULL		X	Assinatura vinculada
ValorNominal	decimal(10,2)	>0	NOT NULL			Valor original da mensalidade
DataVencimento	date		NOT NULL			Data de vencimento
DataPagamento	datetime		NULL			Data em que o pagamento foi efetuado
Status	char(1)	P/G/C	NOT NULL DEFAULT 'P'			Pago / Gerado / Cancelado
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Pagamentos	X		X	PagamentoID		
IDX_Pag_Assinatura		X		AssinaturalID_FK		
IDX_Pag_Status		X		Status		
IDX_Pag_Vencimento		X		DataVencimento		

Tabela	ACESSOS_CATRACA					
Descrição	Representa a contratação de um plano por um aluno.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	P K	F K	Descrição
AcessoID	bigint		NOT NULL	X		Identificador do acesso

AlunoID_FK	int		NOT NULL		X	Aluno que realizou o acesso
UnidadeID_FK	int		NOT NULL			Unidade onde ocorreu o acesso
DataHoraAcesso	datetime		NOT NULL DEFAULT GETDATE()			Data e hora do registro
TipoAcesso	char(1)		NULL			Entrada ou Saída
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
PK_Acessos	X		X	AcessoID		
IDX_Acesso_Alun o		X		AlunoID_FK		
IDX_Acesso_Uni dade		X		UnidadeID_FK		
IDX_TipoAcesso		X		TipoAcesso		
IDX_DataHoraAc esso		X		DataHoraAcesso		

Script dos comandos DDL para criação do Banco de dados (mínimo 8 tabelas)

-- 0. CRIAÇÃO DO BANCO

```
IF DB_ID('AcademiaDB') IS NOT NULL
    DROP DATABASE AcademiaDB;
GO
```

```
CREATE DATABASE AcademiaDB;
GO
USE AcademiaDB;
GO
```


1. CRIAÇÃO DAS TABELAS

-- TABELA: UNIDADES

```
CREATE TABLE UNIDADES (  
  UnidadeID INT IDENTITY(1,1) PRIMARY KEY,  
  NomeUnidade VARCHAR(100) NOT NULL,  
  Endereco VARCHAR(150),  
  Status CHAR(1) DEFAULT 'A'  
);
```

-- TABELA: ALUNOS

```
CREATE TABLE ALUNOS (  
  AlunoID INT IDENTITY(1,1) PRIMARY KEY,  
  Nome VARCHAR(100),  
  CPF VARCHAR(11) UNIQUE,  
  Email VARCHAR(100),  
  DataNascimento DATE,  
  UnidadeID_FK INT,  
  FOREIGN KEY (UnidadeID_FK) REFERENCES UNIDADES(UnidadeID)  
);
```

-- TABELA: PLANOS

```
CREATE TABLE PLANOS (  
  PlanoID INT IDENTITY(1,1) PRIMARY KEY,  
  NomePlano VARCHAR(50),  
  ValorMensal DECIMAL(10,2),  
  DuracaoMeses INT  
);
```

-- TABELA: BENEFICIOS

```
CREATE TABLE BENEFICIOS (  
    BeneficioID INT IDENTITY(1,1) PRIMARY KEY,  
    NomeBeneficio VARCHAR(100),  
    Descricao VARCHAR(200)  
);
```

-- TABELA: PLANOS_BENEFICIOS (N:N)

```
CREATE TABLE PLANOS_BENEFICIOS (  
    PlanoID_FK INT,  
    BeneficioID_FK INT,  
    PRIMARY KEY (PlanoID_FK, BeneficioID_FK),  
    FOREIGN KEY (PlanoID_FK) REFERENCES PLANOS(PlanoID),  
    FOREIGN KEY (BeneficioID_FK) REFERENCES BENEFICIOS(BeneficioID)  
);
```

-- TABELA: ASSINATURAS

```
CREATE TABLE ASSINATURAS (  
    AssinaturaID INT IDENTITY(1,1) PRIMARY KEY,  
    AlunoID_FK INT,  
    PlanoID_FK INT,  
    DataInicio DATE,  
    DataVencimento DATE,  
    FOREIGN KEY (AlunoID_FK) REFERENCES ALUNOS(AlunoID),  
    FOREIGN KEY (PlanoID_FK) REFERENCES PLANOS(PlanoID)  
);
```

-- TABELA: PAGAMENTOS

```
CREATE TABLE PAGAMENTOS (  
    PagamentoID INT IDENTITY(1,1) PRIMARY KEY,  
    AssinaturaID_FK INT,  
    ValorNominal DECIMAL(10,2),
```

```
DataVencimento DATE,  
Status CHAR(1), -- P=Pago, G=Em atraso  
FOREIGN KEY (AssinaturaID_FK) REFERENCES ASSINATURAS(AssinaturaID)  
);
```

-- TABELA: ACESSOS_CATRACA

```
CREATE TABLE ACESSOS_CATRACA (  
  AcessoID INT IDENTITY(1,1) PRIMARY KEY,  
  AlunoID_FK INT,  
  UnidadeID_FK INT,  
  DataHoraAcesso DATETIME DEFAULT GETDATE(),  
  TipoAcesso CHAR(1), -- E=Entrada, S=Saída  
  FOREIGN KEY (AlunoID_FK) REFERENCES ALUNOS(AlunoID),  
  FOREIGN KEY (UnidadeID_FK) REFERENCES UNIDADES(UnidadeID)  
);
```

Script que popula as tabelas do Banco de dados (mínimo 10 itens principais)

2. POPULAÇÃO DO BANCO (INSERTS)

-- 1. UNIDADES

```
INSERT INTO UNIDADES (NomeUnidade, Endereco, Status) VALUES  
('Unidade Centro', 'Rua Central, 100', 'A'),  
('Unidade Norte', 'Avenida Norte, 200', 'A'),  
('Unidade Sul', 'Rua Sul, 300', 'A'),  
('Unidade Leste', 'Rua Leste, 400', 'A'),  
('Unidade Oeste', 'Rua Oeste, 500', 'A'),  
('Unidade Premium', 'Avenida Luxo, 600', 'A'),  
('Unidade Fit', 'Rua Acadêmica, 700', 'A'),  
('Unidade Strong', 'Rua Musculação, 800', 'A'),  
('Unidade Power', 'Avenida Esportes, 900', 'A'),  
('Unidade Prime', 'Rua Prime, 1000', 'A');
```

-- 2. ALUNOS

INSERT INTO ALUNOS (Nome, CPF, Email, DataNascimento, UnidadeID_FK) VALUES

('João Silva', '12345678901', 'joao@email.com', '2000-05-10', 1),
('Maria Costa', '23456789012', 'maria@email.com', '1999-08-21', 1),
('Pedro Souza', '34567890123', 'pedro@email.com', '1998-03-11', 2),
('Ana Lima', '45678901234', 'ana@email.com', '2001-01-15', 2),
('Rafael Dias', '56789012345', 'rafael@email.com', '2002-06-01', 3),
('Camila Rocha', '67890123456', 'camila@email.com', '1997-07-27', 3),
('Lucas Alves', '78901234567', 'lucas@email.com', '1995-04-02', 4),
('Beatriz Melo', '89012345678', 'bia@email.com', '1996-12-19', 5),
('Julia Nunes', '90123456789', 'julia@email.com', '2000-10-09', 6),
('Carlos Peres', '01234567890', 'carlos@email.com', '1999-02-14', 7);

-- 3. PLANOS

INSERT INTO PLANOS (NomePlano, ValorMensal, DuracaoMeses) VALUES

('Básico', 99.90, 1),
('Intermediário', 149.90, 1),
('Premium', 199.90, 1),
('Musculação', 120.00, 3),
('CrossFit', 220.00, 1),
('Funcional', 130.00, 1),
('Bike Indoor', 110.00, 1),
('Natação', 160.00, 1),
('Boxe', 140.00, 1),
('Completo', 299.90, 1);

-- 4. BENEFICIOS

INSERT INTO BENEFICIOS (NomeBeneficio, Descricao) VALUES

('Acesso Total', 'Acesso a todas as áreas da academia'),
('Personal Trainer', 'Sessões com personal'),

```
('Nutricionista', 'Consultas mensais'),  
( 'Aulas Coletivas', 'Participação ilimitada'),  
( 'Piscina', 'Acesso à piscina'),  
( 'Sauna', 'Acesso à sauna'),  
( 'Estacionamento', 'Vaga exclusiva'),  
( 'Massagem', 'Sessão mensal'),  
( 'CrossZone', 'Área de crossfit'),  
( 'Café Fit', 'Café saudável à vontade');
```

-- 5. PLANOS_BENEFICIOS

```
INSERT INTO PLANOS_BENEFICIOS VALUES  
(1,1),(1,4),  
(2,1),(2,4),(2,7),  
(3,1),(3,2),(3,3),(3,4),(3,5),  
(10,1),(10,2),(10,3),(10,4),(10,5),(10,6),(10,7),(10,8);
```

-- 6. ASSINATURAS

```
INSERT INTO ASSINATURAS (AlunoID_FK, PlanoID_FK, DataInicio, DataVencimento) VALUES  
(1,1,'2025-01-01','2025-02-01'),  
(2,2,'2025-01-02','2025-02-02'),  
(3,3,'2025-01-03','2025-02-03'),  
(4,4,'2025-01-04','2025-04-04'),  
(5,5,'2025-01-05','2025-02-05'),  
(6,6,'2025-01-06','2025-02-06'),  
(7,7,'2025-01-07','2025-02-07'),  
(8,8,'2025-01-08','2025-02-08'),  
(9,9,'2025-01-09','2025-02-09'),  
(10,10,'2025-01-10','2025-02-10');
```

-- 7. PAGAMENTOS

```
INSERT INTO PAGAMENTOS (AssinaturaID_FK, ValorNominal, DataVencimento, Status) VALUES
```

(1, 99.90, '2025-02-01', 'P'),
(2, 149.90, '2025-02-02', 'P'),
(3, 199.90, '2025-02-03', 'G'),
(4, 120.00, '2025-04-04', 'G'),
(5, 220.00, '2

Principais consultas mapeadas baseadas em regras de negócio (mínimo 6)

-- 1. Listar alunos com suas unidades

```
SELECT A.Nome, U.NomeUnidade  
FROM ALUNOS A  
JOIN UNIDADES U ON A.UnidadeID_FK = U.UnidadeID;
```

-- 2. Ver assinaturas e seus respectivos planos

```
SELECT A.AssinaturaID, AL.Nome, P.NomePlano, A.DataInicio, A.DataVencimento  
FROM ASSINATURAS A  
JOIN ALUNOS AL ON A.AlunoID_FK = AL.AlunoID  
JOIN PLANOS P ON A.PlanoID_FK = P.PlanoID;
```

-- 3. Listar pagamentos em atraso

```
SELECT *  
FROM PAGAMENTOS  
WHERE Status = 'G';
```

-- 4. Listar benefícios por plano

```
SELECT P.NomePlano, B.NomeBeneficio  
FROM PLANOS_BENEFICIOS PB  
JOIN PLANOS P ON P.PlanoID = PB.PlanoID_FK  
JOIN BENEFICIOS B ON B.BeneficioID = PB.BeneficioID_FK;
```

-- 5. Consultar acessos por aluno

```
SELECT A.Nome, AC.DataHoraAcesso, AC.TipoAcesso  
FROM ACESSOS_CATRACA AC  
JOIN ALUNOS A ON A.AlunoID = AC.AlunoID_FK;
```

```
-- 6. Ver alunos que têm pagamentos atrasados
SELECT DISTINCT AL.Nome
FROM ALUNOS AL
JOIN ASSINATURAS ASI ON ASI.AlunoID_FK = AL.AlunoID
JOIN PAGAMENTOS P ON P.AssinaturalID_FK = ASI.AssinaturalID
WHERE P.Status = 'G';
```