

PROJETO DE BANCO DE DADOS

MINIMUNDO: UMA ONG

1. INTRODUÇÃO

O objetivo deste documento é conseguir entender e integrar todas as etapas de modelagem de uma base de dados num único projeto. Imagine que você vá desenvolver um sistema de informação que necessite de um banco de dados. Depois de algumas entrevistas com o usuário (levantamento de requisitos), você decide começar a modelar a base de dados. O problema apresentado pelo usuário está descrito a seguir.

2. PROBLEMA

Uma ONG, sem fins lucrativos, deseja informatizar o seu processo de trabalho. Existem algumas pessoas que colaboram com a ONG e é necessário que se consiga identificar quem são essas pessoas e como elas contribuem.

As pessoas envolvidas (voluntários) podem ser de dois tipos: uma, é pessoa física e doa seu tempo para desenvolver algum projeto para a entidade e outra, é pessoa jurídica, e faz as doações em dinheiro para a entidade.

Quando uma pessoa se inscreve para ajudar a ONG, é necessário armazenar o código do voluntário, nome do voluntário e telefone do voluntário. Se esta pessoa for uma pessoa física, deve-se armazenar também o CPF da pessoa. Se ela for uma pessoa jurídica, é necessário saber o CNPJ da empresa. Para todas as pessoas, deve-se cadastrar o telefone e o nome de pelo menos duas outras pessoas para contato.

Existem dois tipos de colaboração que se pode fazer à ONG: doações e participação em projetos.

A participação em projetos é feita por pessoas que doam seu tempo para participar de um projeto, que pode ter várias pessoas envolvidas. Quando um projeto termina, as pessoas envolvidas podem participar de um novo. Um projeto terá sempre um coordenador, que é uma pessoa física cadastrada. Além disso, sobre cada projeto é necessário saber: o número dele (que é único), nome, data de início e data prevista para o fim do projeto.

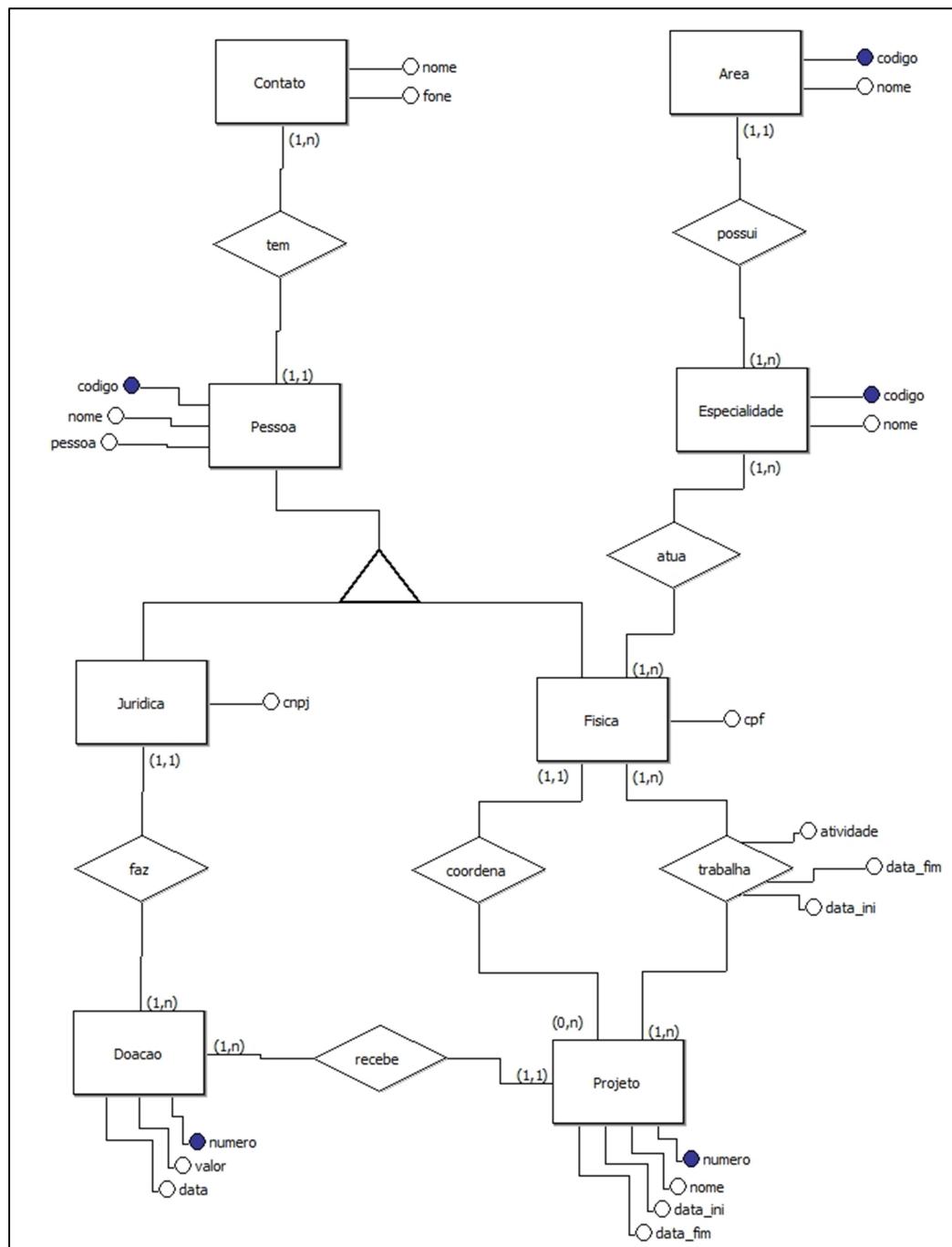
Um projeto pode durar até um ano e as pessoas que trabalham nele, exceto o coordenador, não precisam estar envolvidas no processo durante toda a sua existência. As pessoas podem trabalhar apenas alguns dias ou apenas algumas semanas no projeto. Por isso, é necessário cadastrar quando a pessoa começou a trabalhar, quando sua participação foi encerrada e qual atividade ele desenvolveu durante esse tempo.

Para cada pessoa física, deve-se cadastrar a sua especialidade (por exemplo: informática, enfermagem, artes, dança, etc.). Existem várias especialidades previamente cadastradas, mas deve ser possível cadastrar uma especialidade nova. As especialidades pertencem a áreas, que também são pré-cadastradas (por exemplo: área de saúde, área de tecnologia, etc.). É importante esse cadastro de especialidades e áreas para que se possa procurar na base de dados por uma pessoa que tenha habilidade para trabalhar em um projeto e convidá-la a participar dele.

As pessoas jurídicas fazem doações em dinheiro. Essas doações são encaminhadas especificamente para um projeto, e este projeto pode receber várias doações. Quando uma doação é realizada, é necessário saber o número dessa doação, o valor da doação e a data em que ela foi realizada.

3. DIAGRAMA ENTIDADE RELACIONAMENTO

Com base nas informações obtidas, é possível fazer o modelo conceitual da base de dados. Se houver ambiguidade de interpretação nas informações é necessário conversar novamente com o usuário para esclarecer a situação. O Diagrama de ER para o problema da ONG é apresentado na figura a seguir:



Duas observações importantes sobre o nosso modelo:

- Os atributos nome_contato e fone_contato poderiam ter sido representados como atributos multivalorados na entidade “Pessoa”. Mas como sabemos que atributos multivalorados vão virar tabela no modelo relacional, podemos representá-los no modelo de ER como uma entidade fraca.
- Existem dois relacionamentos entre “Projeto” e “Pessoa Física”. Um deles (“coordena”) para relacionar a pessoa que é coordenadora do projeto e outro (“trabalha”) para representar as pessoas que estão envolvidas com o projeto.

O próximo passo depois de criado o Diagrama de ER é fazer a descrição do modelo relacional da base de dados.

4. DESCRIÇÃO DO MODELO RELACIONAL

A descrição do modelo relacional consiste basicamente em definir o que vira e o que não vira tabela, usando as regrinhas apresentadas nos capítulos 4 e 5. Lembre-se de que todas as entidades e todos os relacionamentos N:N tornam-se tabelas no modelo relacional.

No nosso exemplo, temos uma especialização e vamos optar por criar tabelas para a entidade pai e para as entidades filhas. Essa decisão foi tomada, considerando que tanto as entidades filhas quanto a entidade pai possuem relacionamentos específicos com outras entidades do modelo. Além disso, a abordagem escolhida evita valores nulos e redundantes.

Sendo assim, teremos 10 tabelas no nosso modelo relacional. A descrição do modelo relacional para o nosso problema é apresentada na figura 6.2. Os tipos de dados para cada atributo serão apresentados apenas no Diagrama do Modelo Relacional. As chaves estrangeiras são decorrentes do tipo de relacionamento entre as entidades.

tbPessoa(cod_pessoa, nome_pessoa, fone_pessoa)

tbContato(cod_pessoa, nome_contato, fone_contato)
cod_pessoa referencia tbPessoa

tbArea(cod_area, nome_area)

tbEspecialidade(cod_especialidade, nome_especialidade, cod_area)
cod_area referencia tbArea

tbPFisica(CPF_pessoa, cod_pessoa)
cod_pessoa referencia tbPessoa

tbPFisicaEspecialidade(cod_especialidade, CPF_pessoa)
cod_especialidade referencia tbEspecialidade, CPF_pessoa referencia tbPFisica

tbProjeto(numero_projeto, nome_projeto, data_inicio, data_fim, CPF_coordenador)
CPF_coordenador referencia tbPFisica

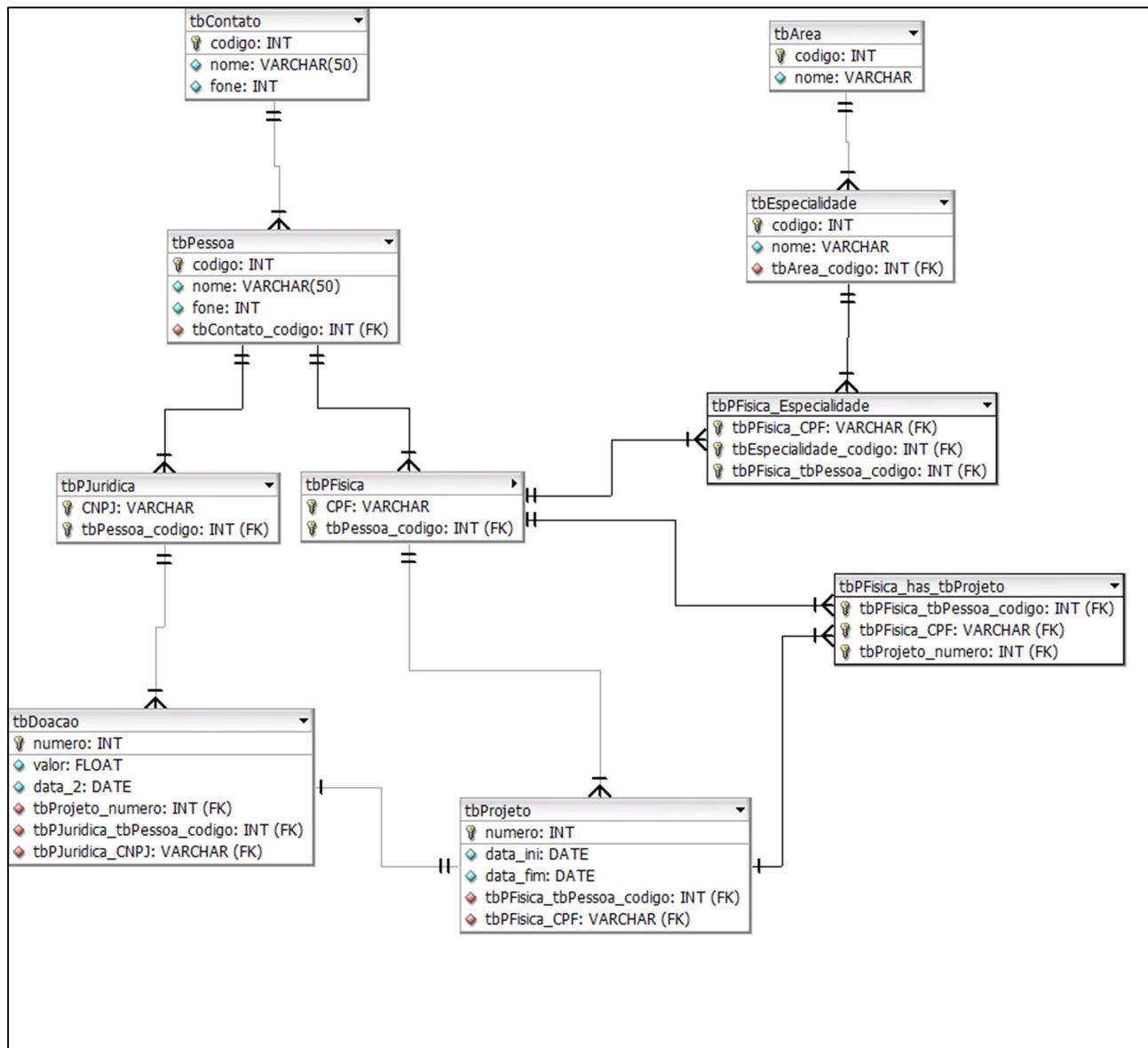
tbPFisicaProjeto(CPF_pessoa, numero_projeto, data_inicio, atividade, data_fim)
CPF_pessoa referencia tbPFisica, numero_projeto referencia tbProjeto

*tbPJuridica(CNPJ_pessoa, cod_pessoa)
cod_pessoa referencia tbPessoa*

*tbDoacao(numero_doacao, valor_doacao, data_doacao, numero_projeto, CNPJ_pessoa)
numero_projeto referencia tbProjeto, CNPJ_pessoa referencia tbPJuridica*

5. DIAGRAMA DO MODELO RELACIONAL

Uma vez que a descrição do modelo relacional foi feita, fica fácil desenhar o diagrama do modelo relacional.



6. DICIONÁRIO DE DADOS

Após conferir se o modelo está normalizado, a próxima etapa, antes de entregar para o desenvolvedor os modelos da base de dados, é escrever o Dicionário de Dados para a base, como será mostrado a seguir. Lembre-se de que o Dicionário de Dados deve ser feito para todas as tabelas do modelo relacional.

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
cod_pessoa	Armazena o código do voluntário	Inteiro	—	Não	—	PK	—	Não
nome_pessoa	Armazena o nome do voluntário	Caracter	50	Não	—	—	—	Não
fone_pessoa	Armazena o telefone do voluntário	Caracter	10	Sim	—	—	—	Não

Tabela tbPessoa

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
cod_pessoa	Armazena o código do voluntário	Inteiro	—	Não	—	PK, FK que referencia tbPessoa	—	Não
nome_contato	Armazena o nome da pessoa de contato	Caracter	50	Não	—	PK	—	Não
fone_contato	Armazena o telefone do contato	Caracter	10	Não	—	—	—	Não

Tabela tbContato

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
CPF_pessoa	Armazena o CPF de uma pessoa física	Caracter	11	Não	—	PK	—	Não
cod_pessoa	Armazena o código do voluntário	Inteiro	—	Não	—	FK que referencia tbPessoa	—	Não

Tabela tbPFisica

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
cod_area	Armazena o código da área	Inteiro	—	Não	—	PK	—	Não
nome_area	Armazena o nome da área	Caracter	50	Não	—	—	—	Não

Tabela tbArea

Projeto Exemplo da Profa. Elaini Simoni Angelotti

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
cod_especialidade	Armazena o código de uma especialidade	Inteiro	—	Não	—	PK	—	Não
nome_especialidade	Armazena o nome da especialidade	Caracter	50	Não	—	—	—	Não
cod_area	Armazena o código da área de uma especialidade	Inteiro	—	Não	—	FK que referencia tbArea	—	Não

Tabela tbEspecialidade

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
cod_especialidade	Armazena o código de uma especialidade	Inteiro	—	Não	—	PK, FK que referencia tbEspecialidade	—	Não
CPF_pessoa	Armazena o CPF de uma pessoa física	Caracter	11	Não	—	PK, FK que referencia tbPFFisica	—	Não

Tabela tbPFFisicaEspecialidade

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
numero_projeto	Armazena o número do projeto	Inteiro	—	Não	—	PK	—	Não
nome_projeto	Armazena o nome do projeto	Caracter	50	Não	—	—	—	Não
data_inicio	Armazena a data em que o projeto iniciou	Data	—	Não	—	—	—	Não
data_fim	Armazena a data em que o projeto terminou	Data	—	Não	—	—	—	Não
CPF_pessoa	Armazena o CPF do voluntário	Caracter	11	Não	—	FK que referencia tbPFFisica	—	Não

Tabela tbProjeto

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
numero_projeto	Armazena o número do projeto	Inteiro	—	Não	—	PK, FK que referencia tbProjeto	—	Não
CPF_pessoa	Armazena o CPF do voluntário	Caracter	11	Não	—	PK, FK que referencia tbPFisica	—	Não
data_inicio	Armazena a data em que a pessoa começou a trabalhar no projeto	Data	—	Não	—	PK	—	Não
data_fim	Armazena a data em que a pessoa deixou de trabalhar no projeto	Data	—	Não	—	—	—	Não
atividade	Armazena a atividade que uma pessoa desenvolveu no projeto	Caracter	100	Não	—	—	—	Não

Tabela tbPFisicaProjeto

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
CNPJ_pessoa	Armazena o CNPJ de uma pessoa jurídica	Caracter	13	Não	—	PK	—	Não
cod_pessoa	Armazena o código do voluntário	Inteiro	—	Não	—	FK que referencia tbPessoa	—	Não

Tabela tbPJuridica

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
numero_doacao	Armazena o número da doação	Inteiro	—	Não	—	PK	—	Não
valor_doacao	Armazena o valor da doação	Real	—	Não	—	—	—	Não
data_doacao	Armazena a data em que a doação foi feita	Data	—	Não	—	—	—	Não
CNPJ_pessoa	Armazena o CNPJ da empresa que fez a doação	Caracter	13	Não	—	FK que referencia tbPJuridica	—	Não
numero_projeto	Armazena o número do projeto que recebeu a doação	Inteiro	—	Não	—	FK que referencia tbProjeto	—	Não

Tabela tbDoacao

7. SCRIPT DO BANCO EM SQL

```
CREATE TABLE tbArea (
    codigo INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR NULL,
    PRIMARY KEY(codigo)
);

CREATE TABLE tbContato (
    codigo INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(50) NULL,
    fone INT NULL,
    PRIMARY KEY(codigo)
);

CREATE TABLE tbDoacao (
    numero INT NOT NULL AUTO_INCREMENT,
    valor FLOAT NULL,
    data_2 DATE NULL,
    tbProjeto_numero INT NOT NULL,
    tbPJuridica_tbPessoa_codigo INT NOT NULL,
    tbPJuridica_CNPJ VARCHAR NOT NULL,
    PRIMARY KEY(numero),
    INDEX tbDoacao_FKIndex1(tbPJuridica_CNPJ, tbPJuridica_tbPessoa_codigo),
    INDEX tbDoacao_FKIndex2(tbProjeto_numero)
);

CREATE TABLE tbEspecialidade (
    codigo INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR NULL,
    tbArea_codigo INT NOT NULL,
    PRIMARY KEY(codigo),
    INDEX tbEspecialidade_FKIndex1(tbArea_codigo)
);

CREATE TABLE tbPessoa (
    codigo INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(50) NULL,
    fone INT NULL,
    tbContato_codigo INT NOT NULL,
    PRIMARY KEY(codigo),
    INDEX tbPessoa_FKIndex1(tbContato_codigo)
);
```

```
CREATE TABLE tbPFisica (
    CPF VARCHAR NOT NULL AUTO_INCREMENT,
    tbPessoa_codigo INT NOT NULL,
    PRIMARY KEY(CPF, tbPessoa_codigo),
    INDEX tbPFisica_FKIndex1(tbPessoa_codigo)
);

CREATE TABLE tbPFisica_Especialidade (
    tbPFisica_CPF VARCHAR NOT NULL,
    tbEspecialidade_codigo INT NOT NULL,
    tbPFisica_tbPessoa_codigo INT NOT NULL,
    PRIMARY KEY(tbPFisica_CPF, tbEspecialidade_codigo, tbPFisica_tbPessoa_codigo),
    INDEX tbEspecialidade_has_tbPFisica_FKIndex1(tbEspecialidade_codigo),
    INDEX tbPFisica_Especialidade_FKIndex2(tbPFisica_CPF, tbPFisica_tbPessoa_codigo)
);

CREATE TABLE tbPFisica_has_tbProjeto (
    tbPFisica_tbPessoa_codigo INT NOT NULL,
    tbPFisica_CPF VARCHAR NOT NULL,
    tbProjeto_numero INT NOT NULL,
    PRIMARY KEY(tbPFisica_tbPessoa_codigo, tbPFisica_CPF, tbProjeto_numero),
    INDEX tbPFisica_has_tbProjeto_FKIndex1(tbPFisica_CPF, tbPFisica_tbPessoa_codigo),
    INDEX tbPFisica_has_tbProjeto_FKIndex2(tbProjeto_numero)
);

CREATE TABLE tbPJuridica (
    CNPJ VARCHAR NOT NULL AUTO_INCREMENT,
    tbPessoa_codigo INT NOT NULL,
    PRIMARY KEY(CNPJ, tbPessoa_codigo),
    INDEX tbPJuridica_FKIndex1(tbPessoa_codigo)
);

CREATE TABLE tbProjeto (
    numero INT NOT NULL AUTO_INCREMENT,
    data_ini DATE NULL,
    data_fim DATE NULL,
    tbPFisica_tbPessoa_codigo INT NOT NULL,
    tbPFisica_CPF VARCHAR NOT NULL,
    PRIMARY KEY(numero),
    INDEX tbProjeto_FKIndex1(tbPFisica_CPF, tbPFisica_tbPessoa_codigo)
);
```