

Autores:

Thiago Silva id: 201705644

Alisson Barros id: 201705580

Em uma rua na cidade de Goiânia localiza-se um condomínio chamado Terra do Nunca. O condomínio possui código, nome, CNPJ, endereço, tipo de unidade, dia de vencimento, senha de acesso, e um total de unidades que estão em N blocos dentro do condomínio.

Um bloco possui código, nome, e uma referência ao condomínio em que ele se localiza. Além disso, um bloco possui N unidades associadas à ele.

Uma unidade possui id, nome da unidade, morador, proprietário, telefone, data de vencimento, uma referência ao condomínio e uma ao bloco em que se localiza.

No condomínio Terra do Nunca, são emitidos boletos referentes a cada uma das suas unidades. Um boleto possui id, status, mês, ano, data de vencimento/pagamento, valor, juros, descontos, e uma referência ao condomínio e uma à unidade referentes ao mesmo.

No condomínio em questão, há um síndico responsável pela gestão do mesmo. Um síndico possui id, nome do síndico/subsíndico, conselheiro, data de nascimento, CPF, data inicial, data final e uma referência ao condomínio gerenciado por ele.

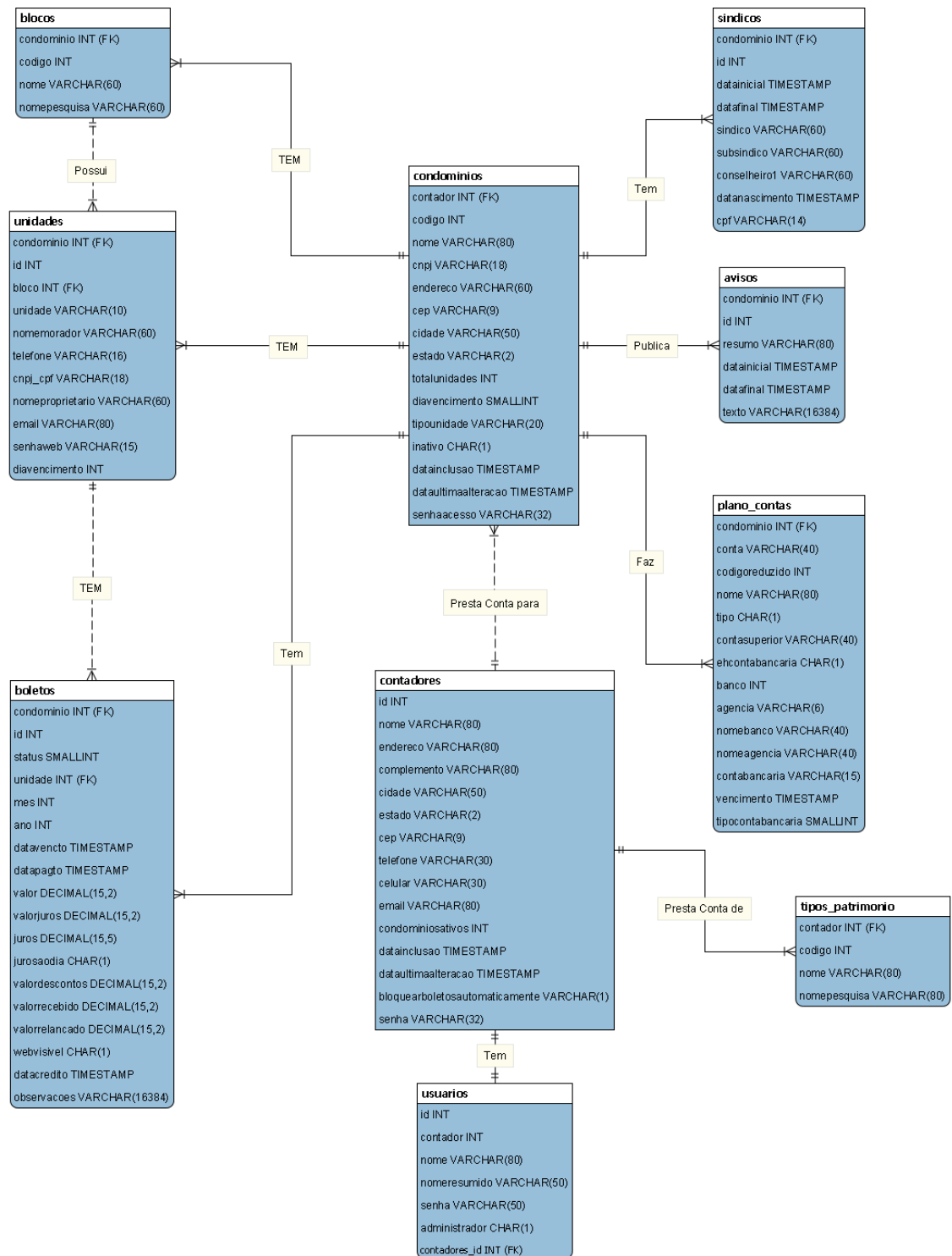
O condomínio Terra do Nunca emite avisos aos seus moradores. Um aviso conta com um id, resumo, data inicial/final, um texto e uma referência ao condomínio.

Além disso, o condomínio conta com um plano de contas. Um plano de contas possui conta, código, nome da conta, tipo, nome do banco, agencia, nome da agência, conta bancária, vencimento, tipo da conta bancária e uma referência ao condomínio.

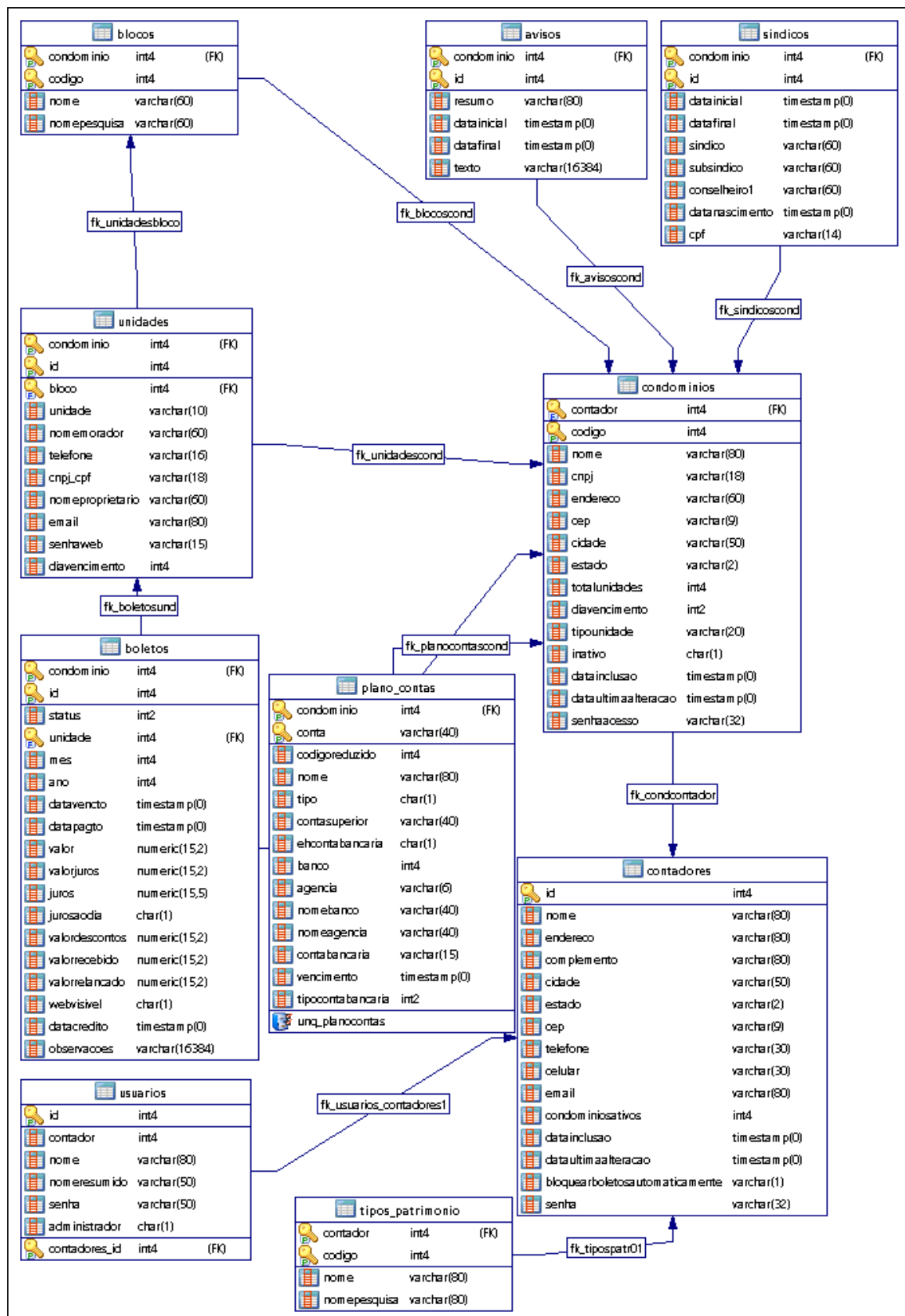
No condomínio Terra do Nunca, um contador é responsável pela prestação de contas dos patrimônios do condomínio que podem admitir vários tipos. Os tipos de patrimônio possuem código, nome, um nome para pesquisa e uma referência ao contador. Um contador possui id, nome, endereço, complemento do endereço, CEP, telefone, e-mail, data de inclusão, data da última alteração, senha, e uma referência ao condomínio ativo.

Os usuários do condomínio fazem uso dos serviços de prestação de contas do contador. Um usuário possui id, nome, senha, administrador e uma referência ao contador.

Modelo logico, utilizando a notação clássica Cross Foot(IE)



Modelo Logico



Lista de consultas

```
select distinct u.nome
from usuarios u inner join condominios c
on c.contador = u.contador
where not exists
    (select *
     from avisos a
     where a.condominio = c.codigo and a.resumo = null)
```

```
select distinct s.sindico
from syndicos s inner join condominios c
on s.id != c.codigo
where exists
    (select *
     from blocos b
     where b.condominio = s.condominio and b.codigo > c.codigo)
```

```
select distinct s.sindico, c.nome
from syndicos s inner join condominios c
on s.id != c.codigo
where exists
    (select *
     from blocos b
     where b.condominio = s.condominio and
     exists
        (select *
         from contadores con
         where c.contador = con.id)
    )
select count(*)
```

```
from blocos b
inner join condominios c
on b.condominio = c.codigo and c.cidade = 'Goiania'
```

```
select count(*)
from condominios c
where exists
(select *
from contadores con
where c.contador = con.id and con.id = 1)
SELECT COUNT(b.codigo), b.condominio
FROM blocos b
GROUP BY condominio
HAVING COUNT(b.codigo) > 5
```

DDL do banco

```
/* SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0; */
/* SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
*/
/* SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES'; */

-----

-- Schema mydb

-----

-----

-- Schema public

-----

DROP SCHEMA IF EXISTS public ;
```

```
-----  
-- Schema public  
-----
```

```
CREATE SCHEMA IF NOT EXISTS public ;
```

```
-----  
-- Table `public`.`contadores`  
-----
```

```
DROP TABLE IF EXISTS public.contadores ;
```

```
CREATE TABLE IF NOT EXISTS public.contadores (  
  id INT NOT NULL,  
  nome VARCHAR(80) NULL DEFAULT "",  
  endereco VARCHAR(80) NULL DEFAULT "",  
  complemento VARCHAR(80) NULL DEFAULT "",  
  cidade VARCHAR(50) NULL DEFAULT "",  
  estado VARCHAR(2) NULL DEFAULT "",  
  cep VARCHAR(9) NULL DEFAULT "",  
  telefone VARCHAR(30) NULL DEFAULT "",  
  celular VARCHAR(30) NULL DEFAULT "",  
  email VARCHAR(80) NULL DEFAULT "",  
  condominiosativos INT NULL DEFAULT 0,  
  datainclusao TIMESTAMP(0) NULL DEFAULT NULL,  
  dataultimaalteracao TIMESTAMP(0) NULL DEFAULT NULL,  
  bloquearboletosautomaticamente VARCHAR(1) NULL DEFAULT 'N',  
  senha VARCHAR(32) NULL DEFAULT "",  
  PRIMARY KEY (id));
```

DROP TABLE IF EXISTS public.condominios ;

```
CREATE TABLE IF NOT EXISTS public.condominios (  
    contador INT NOT NULL,  
    codigo INT NOT NULL,  
    nome VARCHAR(80) NULL DEFAULT "",  
    cnpj VARCHAR(18) NULL DEFAULT "",  
    endereco VARCHAR(60) NULL DEFAULT "",  
    cep VARCHAR(9) NULL DEFAULT "",  
    cidade VARCHAR(50) NULL DEFAULT "",  
    estado VARCHAR(2) NULL DEFAULT "",  
    totalunidades INT NULL DEFAULT NULL,  
    diavencimento SMALLINT NULL DEFAULT NULL,  
    tipounidade VARCHAR(20) NULL DEFAULT NULL,  
    inativo CHAR(1) NULL DEFAULT 'N',  
    datainclusao TIMESTAMP(0) NULL DEFAULT NULL,  
    dataultimaalteracao TIMESTAMP(0) NULL DEFAULT NULL,  
    senhaacesso VARCHAR(32) NULL DEFAULT NULL,  
    PRIMARY KEY (codigo),  
    CONSTRAINT fk_condcontador  
        FOREIGN KEY (contador)  
        REFERENCES public.contadores (id)  
    ON DELETE CASCADE);
```

-- Table `public`.`avisos`

DROP TABLE IF EXISTS public.avisos ;

CREATE TABLE IF NOT EXISTS public.avisos (
condominio INT NOT NULL,
id INT NOT NULL,
resumo VARCHAR(80) NULL DEFAULT NULL,
datainicial TIMESTAMP(0) NULL DEFAULT NULL,
datafinal TIMESTAMP(0) NULL DEFAULT NULL,
texto VARCHAR(16384) NULL DEFAULT NULL,
PRIMARY KEY (condominio, id),
CONSTRAINT fk_avisoscond
FOREIGN KEY (condominio)
REFERENCES public.condominios (codigo)
ON DELETE CASCADE);

-- Table `public`.`blocos`

DROP TABLE IF EXISTS public.blocos ;

CREATE TABLE IF NOT EXISTS public.blocos (
condominio INT NOT NULL,
codigo INT NOT NULL,
nome VARCHAR(60) NULL DEFAULT "",
nomepesquisa VARCHAR(60) NULL DEFAULT NULL,

```
PRIMARY KEY (condominio, codigo),  
CONSTRAINT fk_blocoscond  
FOREIGN KEY (condominio)  
REFERENCES public.condominios (codigo)  
ON DELETE CASCADE);
```

```
--  
-----  
-- Table `public`.`unidades`  
-----  
--
```

```
DROP TABLE IF EXISTS public.unidades ;
```

```
CREATE TABLE IF NOT EXISTS public.unidades (  
condominio INT NOT NULL,  
id INT NOT NULL,  
bloco INT NOT NULL,  
unidade VARCHAR(10) NOT NULL,  
nomemorador VARCHAR(60) NULL DEFAULT "",  
telefone VARCHAR(16) NULL DEFAULT "",  
cnpj_cpf VARCHAR(18) NULL DEFAULT "",  
nomeproprietario VARCHAR(60) NULL DEFAULT "",  
email VARCHAR(80) NULL DEFAULT "",  
senhaweb VARCHAR(15) NULL DEFAULT "",  
diavencimento INT NULL DEFAULT NULL,  
PRIMARY KEY (condominio, id),  
CONSTRAINT fk_unidadesbloco  
FOREIGN KEY (condominio , bloco)  
REFERENCES public.blocos (condominio , codigo)  
ON DELETE CASCADE
```

```
ON UPDATE CASCADE,  
CONSTRAINT fk_unidadescond  
FOREIGN KEY (condominio)  
REFERENCES public.condominios (codigo)  
ON DELETE CASCADE);
```

```
-----  
-- Table `public`.`boletos`  
-----
```

```
DROP TABLE IF EXISTS public.boletos ;
```

```
CREATE TABLE IF NOT EXISTS public.boletos (  
condominio INT NOT NULL,  
id INT NOT NULL,  
status SMALLINT NULL DEFAULT 1,  
unidade INT NOT NULL,  
mes INT NOT NULL,  
ano INT NOT NULL,  
datavencdo TIMESTAMP(0) NULL DEFAULT NULL,  
datapagto TIMESTAMP(0) NULL DEFAULT NULL,  
valor DECIMAL(15,2) NULL DEFAULT 0,  
valorjuros DECIMAL(15,2) NULL DEFAULT 0,  
juros DECIMAL(15,5) NULL DEFAULT NULL,  
jurosaodia CHAR(1) NULL DEFAULT NULL,  
valordescontos DECIMAL(15,2) NULL DEFAULT 0,  
valorrecebido DECIMAL(15,2) NULL DEFAULT 0,  
valorrelancado DECIMAL(15,2) NULL DEFAULT 0,  
webvisivel CHAR(1) NULL DEFAULT 'N',
```

```
datacredito TIMESTAMP(0) NULL DEFAULT NULL,  
observacoes VARCHAR(16384) NULL DEFAULT NULL,  
PRIMARY KEY (condominio, id),  
CONSTRAINT fk_boletoscond  
    FOREIGN KEY (condominio)  
    REFERENCES public.condominios (codigo)  
    ON DELETE CASCADE,  
CONSTRAINT fk_boletosund  
    FOREIGN KEY (condominio , unidade)  
    REFERENCES public.unidades (condominio , id)  
    ON DELETE CASCADE);
```

```
-----  
-- Table `public`.`plano_contas`  
-----
```

```
DROP TABLE IF EXISTS public.plano_contas ;
```

```
CREATE TABLE IF NOT EXISTS public.plano_contas (  
    condominio INT NOT NULL,  
    conta VARCHAR(40) NOT NULL,  
    codigoreduzido INT NOT NULL,  
    nome VARCHAR(80) NULL DEFAULT "",  
    tipo CHAR(1) NULL DEFAULT 'S',  
    contasuperior VARCHAR(40) NULL DEFAULT NULL,  
    ehcontabancaria CHAR(1) NULL DEFAULT 'N',  
    banco INT NULL DEFAULT NULL,  
    agencia VARCHAR(6) NULL DEFAULT "",  
    nomebanco VARCHAR(40) NULL DEFAULT "",
```

```
nomeagencia VARCHAR(40) NULL DEFAULT "",
contabancaria VARCHAR(15) NULL DEFAULT "",
vencimento TIMESTAMP(0) NULL DEFAULT NULL,
tipocontabancaria SMALLINT NULL DEFAULT 2,
PRIMARY KEY (condominio, conta),
CONSTRAINT fk_planocontascond
FOREIGN KEY (condominio)
REFERENCES public.condominios (codigo)
ON DELETE CASCADE);
```

```
CREATE UNIQUE INDEX unq_planocontas ON public.plano_contas (codigoreduzido ASC);
```

```
-----
-- Table `public`.`sindicos`
-----
```

```
DROP TABLE IF EXISTS public.sindicos ;
```

```
CREATE TABLE IF NOT EXISTS public.sindicos (
condominio INT NOT NULL,
id INT NOT NULL,
datainicial TIMESTAMP(0) NOT NULL,
datafinal TIMESTAMP(0) NOT NULL,
sindico VARCHAR(60) NULL DEFAULT NULL,
subsindico VARCHAR(60) NULL DEFAULT NULL,
conselheiro1 VARCHAR(60) NULL DEFAULT NULL,
datanascimento TIMESTAMP(0) NULL DEFAULT NULL,
cpf VARCHAR(14) NULL DEFAULT NULL,
```

```
PRIMARY KEY (condominio, id),  
CONSTRAINT fk_sindicoscnd  
FOREIGN KEY (condominio)  
REFERENCES public.condominios (codigo)  
ON DELETE CASCADE);
```

```
--  
-----  
-- Table `public`.`tipos_patrimonio`  
-----  
--
```

```
DROP TABLE IF EXISTS public.tipos_patrimonio ;
```

```
CREATE TABLE IF NOT EXISTS public.tipos_patrimonio (  
    contador INT NOT NULL,  
    codigo INT NOT NULL,  
    nome VARCHAR(80) NULL DEFAULT NULL,  
    nomepesquisa VARCHAR(80) NULL DEFAULT NULL,  
    PRIMARY KEY (contador, codigo),  
    CONSTRAINT fk_tipospatr01  
    FOREIGN KEY (contador)  
    REFERENCES public.contadores (id)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

```
--  
-----  
-- Table `public`.`usuarios`  
-----  
--
```

```
DROP TABLE IF EXISTS public.usuarios ;
```

```
CREATE TABLE IF NOT EXISTS public.usuarios (  
    id INT NOT NULL,  
    contador INT NOT NULL,  
    nome VARCHAR(80) NULL DEFAULT NULL,  
    nomeresumido VARCHAR(50) NULL DEFAULT NULL,  
    senha VARCHAR(50) NULL DEFAULT NULL,  
    administrador CHAR(1) NULL DEFAULT NULL,  
    contadores_id INT NOT NULL,  
    PRIMARY KEY (id, contadores_id),  
    CONSTRAINT fk_usuarios_contadores1  
        FOREIGN KEY (contadores_id)  
        REFERENCES public.contadores (id)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION);
```

```
/* SET SQL_MODE=@OLD_SQL_MODE; */
```

```
/* SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS; */
```

```
/* SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS; */
```

Na pasta possui os arquivos para popular o banco, não foi adicionado aqui pois são muito extensos.