

UNIVERSIDADE DE SÃO PAULO
BACHARELADO EM ESTATISTICA

BARBARA BENVENUTO IZIDÓRIO (9793381)
GUILHERME NAVARRO (8943160)
LEONARDO RAMACHOTI NORONHA (9793436)

Documentação Projeto
PALESTRUSP

SÃO PAULO
2018

1 Introdução

O nosso trabalho tem como objetivo de aplicar os conhecimentos obtidos durante o semestre relacionado a disciplina de Banco de Dados, em nosso projeto nós criamos um aplicativo que gerencia palestras que são realizadas na USP, utilizando ferramentas aprendidas em aula como modelo de banco de dados relacional, mapeamento de banco de dados, modelo entidade-relacionamento estendido, e implementação de banco de dados com o modelo físico utilizando a linguagem SQL, além de consultas, onde também foi utilizado Stored Procedures e Triggers utilizando a ferramenta PostgreSQL (pgAdmin III).

2 Análise de Requisitos do Banco de Dados

O PalestrUSP é o mais novo aplicativo gerenciador de palestras realizadas na USP. As pessoas podem assistir palestras e opinar sobre elas, compartilhar conhecimento e ficar por dentro das novidades.

Esta plataforma possui um banco de dados em que são cadastrados usuários definidos como alunos e professores, tanto da Universidade de São Paulo, como de outras universidades, e convidados. Os usuários terão login único, e-mail, telefone, data de nascimento e, a partir desta, sua idade será calculada (por se tratar de um aplicativo de ensino superior, apenas usuários com mais de 16 anos serão autorizados).

Caso o usuário não seja professor ou aluno, ele terá um número de convidado e será solicitada sua área de trabalho. Os alunos da USP serão identificados por seu número USP, enquanto aos alunos de fora será atribuído um número identificador de aluno convidado e será registrado o nome de sua instituição de ensino. Aos professores, será solicitada a área de atuação. Os professores de fora terão número de identificação e o nome da instituição à qual estão associados; já os da USP serão identificados por seu número USP.

O usuário pode ser um palestrante, associado a um número identificador, e deverá ter sua disponibilidade de horários e as línguas nas quais possui fluência. As palestras físicas, realizadas pelos palestrantes, deverão ser marcadas previamente, respeitando data e horário determinados, tendo um título; e também receberão um número identificador. O idioma e a duração também deverão ser especificados.

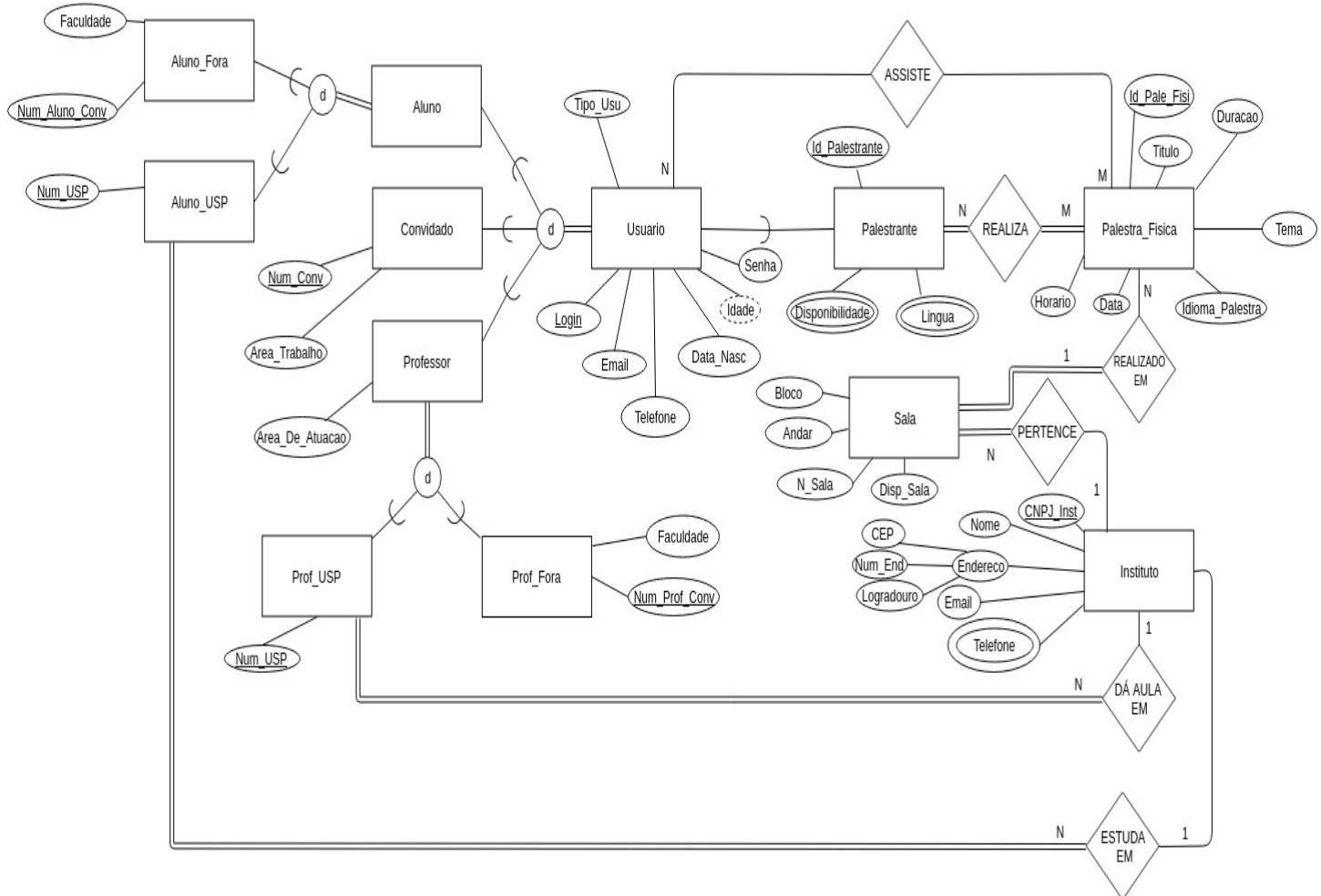
As palestras físicas serão realizada em uma sala reservada em um determinado instituto de acordo com sua disponibilidade. Serão informados o bloco, o andar e a identificação da sala (podendo ser o número, nome ou ambos). Os institutos possuem nome, CNPJ, e-mail, telefone(s), endereço (contendo CEP, número e logradouro). Os professores realizam aulas assistidas por alunos nos institutos. Os usuários assistem às palestras físicas.



Figura 1: Logo do aplicativo

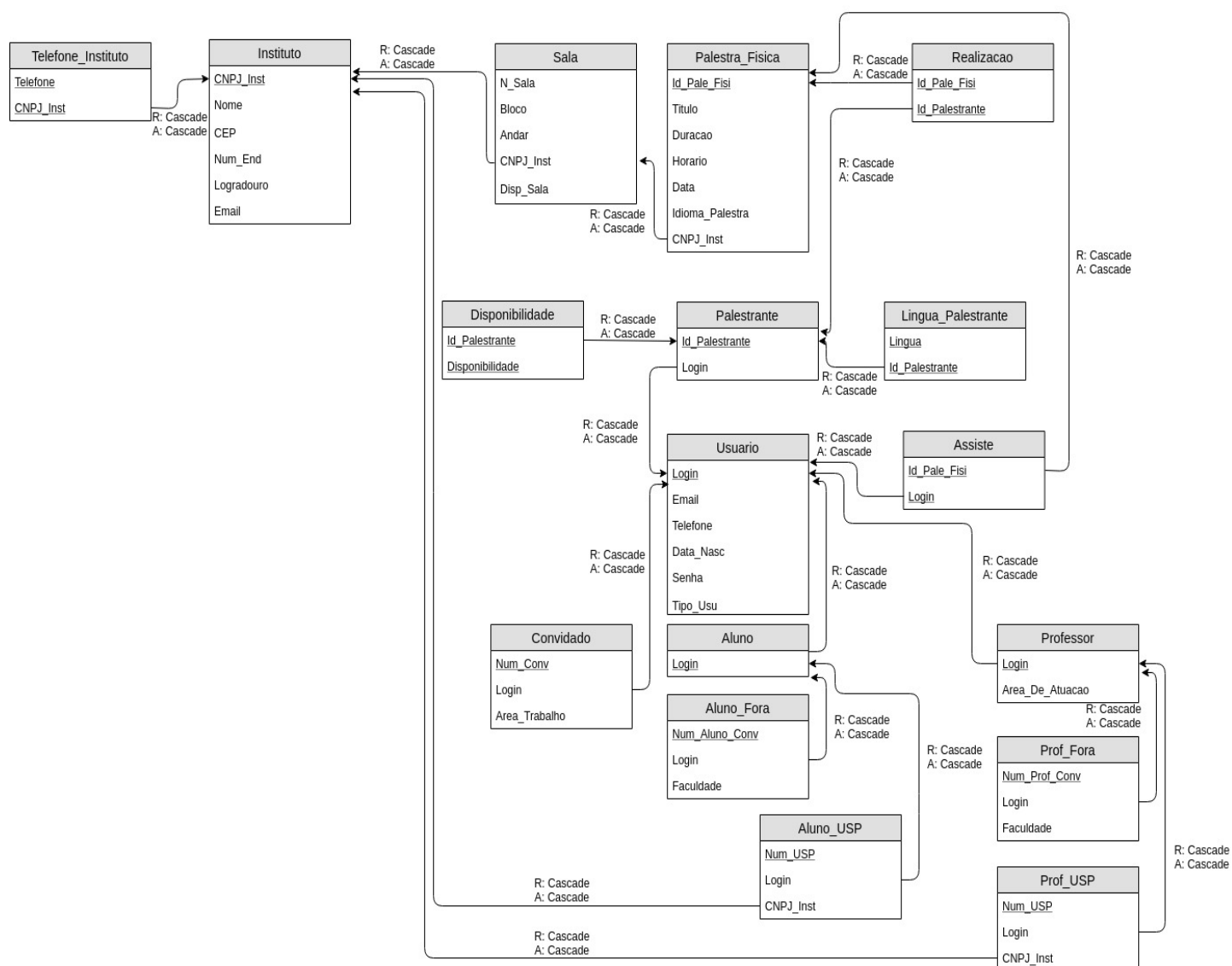
3 Modelo Conceitual do PalestrUSP

Baseado na análise de requisitos, e com recursos que desenvolvemos em sala de aula desenvolvemos o modelo conceitual de nosso projeto PalestrUSP, utilizando a ferramenta do Google que se chama Draw.IO versão 9.4.6, empregando o uso do modelo entidade-relacionamento estendido para modelo de banco de dados relacional.



4 Modelo Lógico do PalestrUSP

Assim como na criação do modelo conceitual, utilizando de recursos desenvolvidos em sala de aula desenvolvemos o modelo lógico, também utilizando a ferramenta do Google chamada Draw.IO versão 9.4.6, com todo detalhamento em suas chaves primárias e estrangeiras e com o devidos cuidado em sua atualização e remoção, empregando o mapeamento modelo entidade-relacionamento estendido para modelo de banco de dados Relacional.



5 Modelo Físico do PalestrUSP

Para o Modelo físico de nosso projeto PalestrUSP, utilizando os modelos conceitual lógico e físico desenvolvemos o modelo físico através do PostgreSQL (pgAdmin III) Versão 1.22.2, onde foi desenvolvida toda a implementação do PalestrUSP utilizando a linguagem SQL.

```
-----
---- CRIACAO DAS TABELAS DO PALESTRUSP ----
-----

-- Criação de Domínios

CREATE DOMAIN TIPO_CNPJ AS CHAR(14); -- Domínio para CNPJ
CREATE DOMAIN TIPO_TEL AS CHAR(11); -- Domínio para Telefone
CREATE DOMAIN TIPO_CEP AS CHAR(8); -- Domínio para CEP

-- cria a tabela Usuário

-- Tabela que contém informações de cadastro de um usuário do aplicativo
-- onde Login é chave primária.

CREATE TABLE USUARIO(
Login VARCHAR(30) PRIMARY KEY,
Email VARCHAR(50) NOT NULL,
Telefone TIPO_TEL NOT NULL,
Data_Nasc DATE NOT NULL,
Senha VARCHAR(30) NOT NULL,
Tipo_Usu VARCHAR(30) NOT NULL
);

-- cria a tabela Instituto

-- Tabela que contem informações dos institutos que oferecem palestras
-- onde o CNPJ do Instituto é a chave primária.

CREATE TABLE INSTITUTO(
CNPJ_Inst TIPO_CNPJ PRIMARY KEY,
Nome VARCHAR(70) NOT NULL,
CEP TIPO_CEP NOT NULL,
Num_End INT NOT NULL,
Logradouro VARCHAR(100),
Email VARCHAR(50) NOT NULL
);

-- cria a tabela Aluno

-- Tabela aluno se trata de todos os alunos cadastrados no aplicativo,
-- onde Login é chave primária e chave estrangeira que
-- referencia a tabela Usuário.
```

```

CREATE TABLE ALUNO(
Login VARCHAR(30) PRIMARY KEY,
FOREIGN KEY (Login) REFERENCES USUARIO (Login)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela Aluno fora

-- Tabela de aluno fora onde é destinada aos alunos que vem de fora da USP,
-- onde o mesmo recebe um número de identificação que é sua
-- chave primária e Login é a chave estrangeira que referencia a tabela Usuário.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE ALUNO_FORA(
Num_Aluno_Conv VARCHAR(10) PRIMARY KEY,
Login VARCHAR(30) NOT NULL,
Faculdade VARCHAR(50) NOT NULL,
FOREIGN KEY (Login) REFERENCES ALUNO (Login)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela Aluno USP

-- Tabela de aluno USP onde é destinada aos alunos USP, onde o mesmo
-- tem seu número USP como identificação que é sua chave primária e Login
-- é a chave estrangeira que referencia a tabela Aluno e também tem CNPJ como
-- chave estrangeira que referencia a tabela Instituto.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE ALUNO_USP(
Num_USP VARCHAR(10) PRIMARY KEY,
Login VARCHAR(30) UNIQUE NOT NULL,
CNPJ_Inst TIPO_CNPJ,
FOREIGN KEY (Login) REFERENCES ALUNO (Login)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (CNPJ_Inst) REFERENCES INSTITUTO (CNPJ_Inst)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela Convidado

-- Tabela destinada aos Convidados das palestras onde o mesmo
-- recebe um número de identificação que é sua chave primária
-- e Login é a chave estrangeira que referencia a tabela Usuário.
-- Remoção: Cascade

```

-- Alteração: Cascade

```
CREATE TABLE CONVIDADO(  
Num_Conv VARCHAR(10) PRIMARY KEY,  
Login VARCHAR(30) UNIQUE NOT NULL,  
Area_Trabalho VARCHAR(50) NOT NULL,  
FOREIGN KEY (Login) REFERENCES USUARIO (Login)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

-- cria a tabela Professor

-- Tabela Professor se trata de todos os Professores cadastrados no aplicativo
-- onde Login é chave primária e chave estrangeira que
-- referencia a tabela Usuário.
-- Remoção: Cascade
-- Alteração: Cascade

```
CREATE TABLE PROFESSOR(  
Login VARCHAR(30) PRIMARY KEY,  
Area_De_Atuação VARCHAR(30) NOT NULL,  
FOREIGN KEY (Login) REFERENCES USUARIO (Login)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

-- cria a tabela professor de fora

-- Tabela de Professor Fora onde é destinada aos professores que vem de
-- fora da USP, onde o mesmo recebe um número de identificação que é sua
-- chave primária e Login é a chave estrangeira que referencia
-- a tabela Professor.
-- Remoção: Cascade
-- Alteração: Cascade

```
CREATE TABLE PROF_FORA(  
Num_Prof_Conv VARCHAR(10) PRIMARY KEY,  
Login VARCHAR(30) UNIQUE NOT NULL,  
Faculdade VARCHAR(50) NOT NULL,  
FOREIGN KEY (Login) REFERENCES PROFESSOR (Login)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

-- cria a tabela professor USP

-- Tabela Professor USP onde é destinada aos professores da USP, onde o mesmo
-- tem seu número USP como identificação que é sua chave primária e Login
-- e a chave estrangeira que referencia a tabela Professor e também tem CNPJ
-- como chave estrangeira que referencia a tabela Instituto.


```

-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE PROF_USP(
Num_USP VARCHAR(10) PRIMARY KEY,
Login VARCHAR(30) UNIQUE NOT NULL,
CNPJ_Inst TIPO_CNPJ NOT NULL,
FOREIGN KEY (Login) REFERENCES PROFESSOR (Login)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (CNPJ_Inst) REFERENCES INSTITUTO (CNPJ_Inst)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela palestrante

-- Tabela destinada ao professor palestrante que recebe um número de
-- identificação que é sua chave primária
-- e Login é a chave estrangeira que referencia a tabela Usuário.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE PALESTRANTE(
Id_Palestrante VARCHAR(10) PRIMARY KEY,
Login VARCHAR(30) UNIQUE NOT NULL,
FOREIGN KEY (Login) REFERENCES USUARIO (Login)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- Cria a tabela de disponibilidades do palestrante

-- Tabela que indica a disponibilidade dos palestrantes onde
-- Disponibilidade e Identificação Palestrante
-- é uma chave primária composta e Identificação Palestrante é
-- chave estrangeira que referencia a tabela Palestrante.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE DISPONIBILIDADE(
Disponibilidade DATE NOT NULL,
Id_Palestrante VARCHAR(10) NOT NULL,
PRIMARY KEY (Disponibilidade,Id_Palestrante),
FOREIGN KEY (Id_Palestrante) REFERENCES PALESTRANTE (Id_Palestrante)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela lingua do palestrante

-- Tabela que se trata das linguas disponíveis dos palestrantes onde Lingua e

```

```

-- Identificação do Palestrante é uma chave primária composta
-- e também Identificação do Palestrante é chave estrangeira que
-- referencia a tabela Palestrante.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE LINGUA_PALESTRANTE(
Lingua VARCHAR(30) NOT NULL,
Id_Palestrante VARCHAR(10) NOT NULL,
PRIMARY KEY (Lingua,Id_Palestrante),
FOREIGN KEY (Id_Palestrante) REFERENCES PALESTRANTE (Id_Palestrante)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela sala

-- Tabela que se trata das salas onde é possível realizar as
-- palestras e tem como chave primária
-- e chave estrangeira o CNPJ do Instituto que referencia a tabela Instituto.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE SALA(
N_Sala INT NOT NULL,
Bloco CHAR(1) NOT NULL,
Andar INT NOT NULL,
DISP_Sala BOOLEAN DEFAULT TRUE,
CNPJ_Inst TIPO_CNPJ PRIMARY KEY,
FOREIGN KEY (CNPJ_Inst) REFERENCES INSTITUTO (CNPJ_Inst)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela palestra fisica

-- Tabela que contém informações das palestras, onde
-- Identificação da Palestra Física é chave primária e CNPJ é
-- chave estrangeira que referencia a tabela Sala.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE PALESTRA_FISICA(
Id_Pale_Fisi VARCHAR(10) PRIMARY KEY,
Titulo VARCHAR(100) NOT NULL,
Duracao TIME NOT NULL,
Horario TIME NOT NULL,
Data DATE NOT NULL,
Idioma VARCHAR(30) NOT NULL,
CNPJ_Inst TIPO_CNPJ NOT NULL,

```

```

FOREIGN KEY (CNPJ_Inst) REFERENCES SALA (CNPJ_Inst)
ON DELETE CASCADE ON UPDATE CASCADE
);

-- cria a tabela assiste

-- Tabela que se trata dos usuários do aplicativo que assistem as palestras
-- e tem Identificação do Palestrante e Login como chave primária composta e
-- Identificação do Palestrante como chave estrangeira que referencia
-- a tabela Palestrante e também Login que referencia a tabela Usuario.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE ASSISTE(
Id_Pale_Fisi VARCHAR(10) NOT NULL,
Login VARCHAR(30) NOT NULL,
PRIMARY KEY (Id_Pale_Fisi, Login),
FOREIGN KEY (Id_Pale_Fisi) REFERENCES PALESTRA_FISICA (Id_Pale_Fisi)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Login) REFERENCES USUARIO (Login)
ON DELETE CASCADE ON UPDATE CASCADE

);

-- cria a tabela realização da palestra

-- Tabela que se trata da realização das palestras e tem Identificação da
-- Palestra Física, Identificação do Palestrante como chave primária composta e
-- Identificação da Palestra Física e Identificação do Palestrante são chaves
-- estrangeiras que referencia as tabelas Palestra Física e
-- Palestrante respectivamente.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE REALIZACAO(
Id_Pale_Fisi VARCHAR(10) NOT NULL,
Id_Palestrante VARCHAR(10) NOT NULL,
PRIMARY KEY (Id_Pale_Fisi, Id_Palestrante),
FOREIGN KEY (Id_Pale_Fisi) REFERENCES PALESTRA_FISICA (Id_Pale_Fisi)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Id_Palestrante) REFERENCES PALESTRANTE (Id_Palestrante)
ON DELETE CASCADE ON UPDATE CASCADE

);

-- cria a tabela Telefone do Instituto

-- Tabela com os telefones dos institutos onde Telefone e CNPJ é uma

```

```
-- chave primária composta e CNPJ é chave estrangeira que
-- referencia a tabela Instituto.
-- Remoção: Cascade
-- Alteração: Cascade

CREATE TABLE TELEFONE_INSTITUTO(
Telefone CHAR(10) NOT NULL,
CNPJ_Inst TIPO_CNPJ UNIQUE NOT NULL,
PRIMARY KEY (Telefone, CNPJ_Inst),
FOREIGN KEY (CNPJ_Inst) REFERENCES INSTITUTO (CNPJ_Inst)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

Referências

- [1] Sistemas de Bancos de Dados (6a edição), Elmasri e Navathe. Pearson, 2010
- [2] Comandos básicos em SQL - INSERT, UPDATE, DELETE e SELECT *Disponível em:* <https://www.devmedia.com.br/comandos-basicos-em-sql-insert-update-delete-e-select/37170>. acesso em **26/11/2018**.
- [3] HEUSER, Carlos Alberto. Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS. Bookman Editora, 2009. *Disponível em:* http://www.fernandozaidan.com.br/pitgrad/Diversos/Livros_Disciplinas/Projeto_de_Banco_de_Dados_-_Carlos_Alberto_Heuser.pdf Acesso em **10/11/2018**.
- [4] Notas de Aula / Slides da disciplina do segundo semestre de 2018.