

# Exercício-Programa 1: Base de dados

Caio Calisto Gaede Hirakawa  
Matheus Santos Conceição

11 de Abril de 2019

# Conteúdo

<b>1</b>	<b>Parte I: Entendimento e validação do modelo</b>	<b>2</b>
<b>2</b>	<b>Parte II: Descrição das classes abstratas</b>	<b>3</b>
2.1	Entidades . . . . .	3
2.1.1	Pessoa: . . . . .	3
2.1.2	Aluno: . . . . .	5
2.1.3	Professor: . . . . .	6
2.1.4	Administrador: . . . . .	6
2.1.5	Disciplina: . . . . .	7
2.1.6	Currículo: . . . . .	8
2.1.7	Módulo: . . . . .	9
2.1.8	Trilha: . . . . .	9
2.1.9	Usuário: . . . . .	10
2.1.10	Perfil: . . . . .	11
2.1.11	Serviço: . . . . .	11
2.2	Agregados . . . . .	12
2.2.1	Oferecimento: . . . . .	12
2.3	Relacionamentos . . . . .	12
2.3.1	pe-us: . . . . .	12
2.3.2	us-pf: . . . . .	12
2.3.3	pf-se: . . . . .	13
2.3.4	Cursa: . . . . .	13
2.3.5	Planeja: . . . . .	13
2.3.6	Administra: . . . . .	14
2.3.7	rel_cur_tri: . . . . .	14
2.3.8	tr_mo: . . . . .	14
2.3.9	rel_dis_mod: . . . . .	14
2.3.10	Funcionalidades esperadas . . . . .	14
2.4	Modelo lógico . . . . .	14

# Capítulo 1

## Parte I: Entendimento e validação do modelo

**Validar o modelo** Para criar um banco de dados sobre gestão de grades curriculares dentro da USP foram oferecidos dois modelos conceituais. Nesse capítulo iremos discutir quais serão os benefícios e os malefícios de cada um desses dois modelos conceituais e tentar criar um novo modelo conceitual a partir desses dois modelos.

Alguns pontos que podemos apresentar é que nesse banco de dados precisamos de uma estrutura de persistência, ou seja, os dados somente podem ser acrescentados e uma condição extra que não podemos alterar os dados. Essa falta de modificação de dados faz com que a modelagem do banco de dados fique mais complicada e provocando uma necessidade de um planejamento antecipado, com alguns sistemas que modificam de ano para ano seus dados, como o cursos obrigatórios, onde os cursos necessários podem mudar, os cursos lecionados, onde os cursos disponíveis podem mudar e o novo sistema de trilhas, onde os requerimentos de cursos podem mudar.

Podemos olhar modelo conceitual I, as relações entre os alunos, professores e administradores não tem muita mudança e podem ser facilmente usadas em persistência, porém temos outras relações como alunos e trilhas, alunos e cursos que modificam de ano para ano.

Analizar melhor, para cada relação verificar a aplicação de persistência nessa relação (...)

Explicar as adaptações depois no novo MER(...)

## Capítulo 2

# Parte II: Descrição das classes abstratas

### 2.1 Entidades

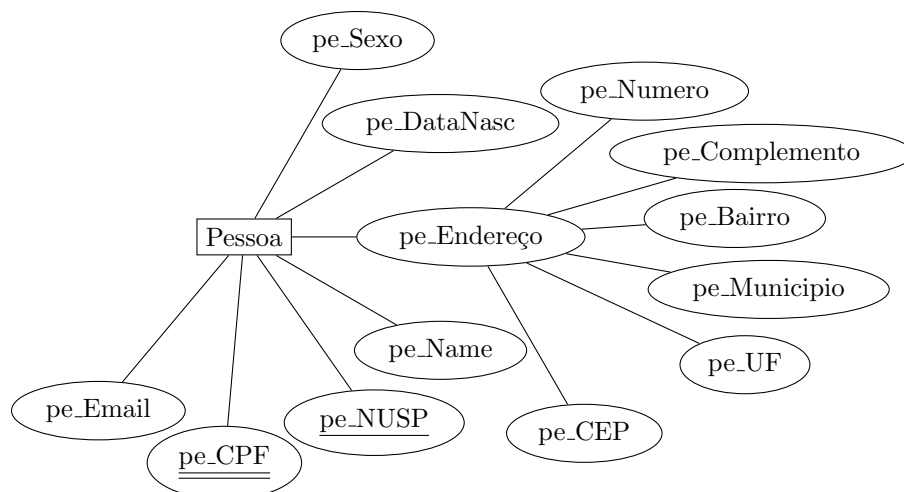
#### 2.1.1 Pessoa:

A entidade regular Pessoa é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada uma das pessoas da USP presentes na nossa base de dados, seja ela Aluno, Professor ou Administrador.

Segundo nosso modelo, cada instância de Pessoa será composta por seu número USP, que será utilizado como chave primária, seu CPF, que será utilizado como chave secundária, seu nome, e-mail, sexo, data de nascimento e endereço, que será um atributo composto de logradouro, número, complemento, bairro, município, unidade federativa e CEP.

O listing abaixo traz um exemplo da estrutura da entidade:

- Número USP (Chave primária)
- CPF (Chave secundária)
- Nome
- E-mail
- Sexo
- Data de Nascimento
- Endereço:
  - Logradouro
  - Número
  - Complemento
  - Bairro
  - Município
  - Unidade Federativa
  - CEP



### Restrições de Domínio:

- **pe\_NUSP:** Chave primária da entidade Pessoa. Sequência de 6 a 9 dígitos, englobando todos os números USP conhecidos e mais um dígito para quando os dígitos disponíveis não forem suficientes. Não pode ser nulo.
- **pe\_CPF:** Chave secundária da entidade Pessoa. Sequência de 11 dígitos que passe pelas regras de validação da Receita Federal:

- verificação do primeiro dígito
- verificação de segundo dígito
- verificação de dígitos iguais

Não pode ser nulo.

- **pe\_Nome:** Sequência de letras maiúsculas e minúsculas, apóstrofe, hífen e espaço. No mínimo de 5 e máximo de 80 caracteres. Não pode ser nulo.
- **pe\_Email:** Sequência de 1 a 80 caracteres alfanuméricos e caracteres especiais

(!#\$%&'\*+,-/=?^\_`{|}~.)

seguidos de um @, seguido de mais uma sequência de 1 a 80 caracteres alfanuméricos e caracteres especiais

(!#\$%&'\*+,-/=?^\_`{|}~.)

. Não pode ser nulo.

- **pe\_DataNascimento:** no formato data dd/mm/aaaa. Não pode ser nulo.
- **pe\_Sexo:** M, F ou N, representando Sexo Masculino, Feminino ou Não declarado respectivamente.
- **pe\_Endereco:**
  - **pe\_Logradouro:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
  - **pe\_Numero:** Sequência de 1 a 10 caracteres alfanuméricos e espaço, com o intuito de permitir escrever "sem número". Não pode ser nulo.
  - **pe\_Complemento:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Pode ser nulo.
  - **pe\_Bairro:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.

- **pe\_Município:** Complemento Logradouro: Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **pe\_UF:** Complemento Logradouro: Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **pe\_CEP:** Sequência de 5 dígitos, seguido de um hífen, seguidos de mais três dígitos. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Pessoa, no formato (NUSP, CPF, Nome, email, sexo, DataNasc, End):

- 7777777; 003.939.708-41; Juca Araujo; juca@usp.br; M; 01/04/1999; Avenida A, 10, null, Butanta, Sao Paulo, SP, 10.200-100
- 7777888; 076.713.684-58; Celia; celia@usp.br; F; 02/03/2000; Avenida B, 11, apto 11, Pinheiros, Sao Paulo, SP, 01.500-250

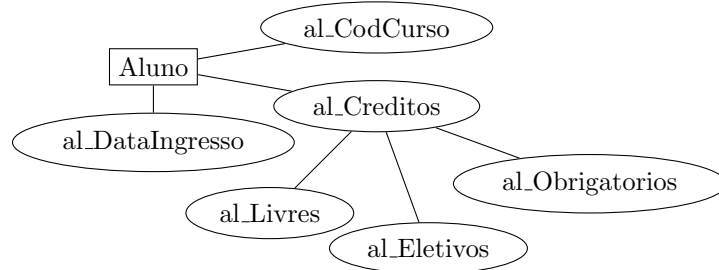
### 2.1.2 Aluno:

A entidade regular Aluno é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos alunos da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Aluno será composta por sua Data de Ingresso, o Código do Curso a qual ele pertence e a quantidade de Créditos Acumulados por ele até o momento da consulta, que será um atributo composto por créditos referentes a disciplinas Obrigatórios, Eletivas e Livres.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Ingresso
- Código do Curso
- Créditos acumulados
  - Obrigatórios
  - Optativos Eletivos
  - Optativos Livres



#### Restrições de Domínio:

- **Data de Ingresso:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **Código do Curso:** Números entre 1 e +infinito. Não pode ser nulo.
- **Créditos acumulados:**
  - **Obrigatórios:** inteiro entre 0 e +infinito. Não pode ser nulo.
  - **Optativos Eletivos:** inteiro entre 0 e +infinito. Não pode ser nulo.
  - **Optativos Livres:** inteiro entre 0 e +infinito. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Aluno, no formato (DataIngr, Cod-Curso, Creditos):

- 01/02/2016; 42; 0, 0, 0
- 03/02/2017; 15; 0, 0, 0

A inserção de um aluno no banco de dados sempre acarretará que ele entre com 0 créditos e, para alterá-lo no decorrer de seu período na USP, um procedure será feito.

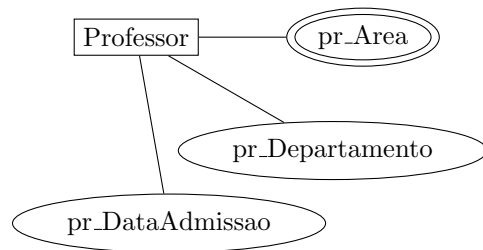
### 2.1.3 Professor:

A entidade regular Professor é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos professores da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Professor será composta por seu Departamento, Data de Admissão e Área de atuação, que será um atributo multivalorado.

O listing abaixo traz um exemplo da estrutura da entidade:

- Departamento
- Data de Admissão
- Área de Atuação



#### Restrições de Domínio:

- **pr\_Departamento:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **pr\_DataAdmissao:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **pr\_Area:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Professor no formato (Depart, Data-Admissao, Area):

- IME; 01/02/1985; Software.
- FMU; 05/03/2000; Neurologia, Clinico Geral
- BIO; 10/01/1992; Zoologia

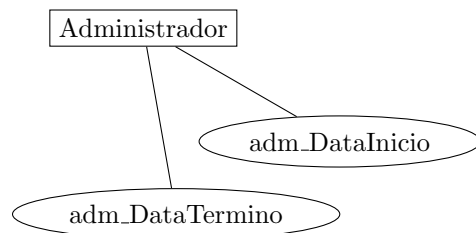
### 2.1.4 Administrador:

A entidade regular Administrador é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos administradores dos cursos da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Administrador será composta por sua Data de Início do mandato e Data de Término, este último podendo ser nulo, indicando que o mandato ainda está em vigência.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Início da Gestão
- Data de Término da Gestão



### Restrições de Domínio:

- **adm\_DataInicio:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **adm\_DataTermino:** Data no formato dd/mm/aaaa.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Administrador no formato (DataInicio, DataTermino):

- 01/01/2010; 31/12/2015.
- 01/01/2016; null.

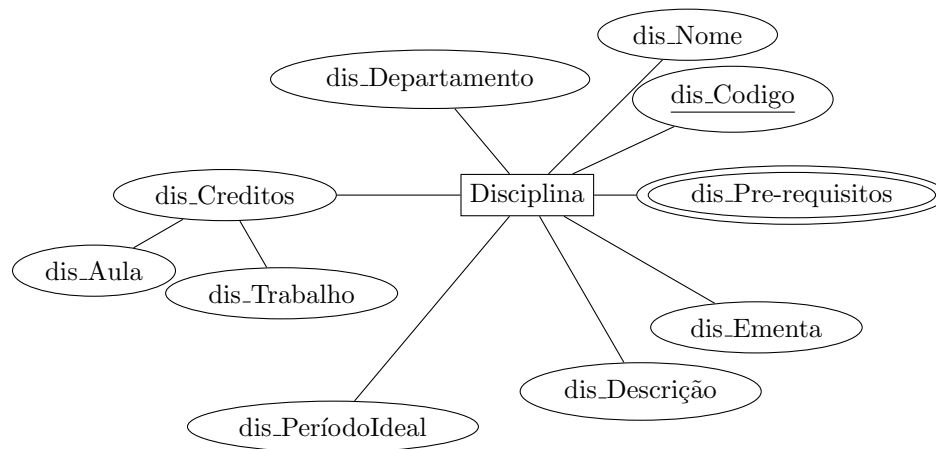
### 2.1.5 Disciplina:

A entidade regular Disciplina é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um das disciplinas da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Disciplina será composta por seu Nome, Código, que será sua chave primária, Departamento a que pertence, Ementa, Descrição do que é lecionado nela, Pré-Requisitos, um atributo multivalorado, Período Ideal e quantidade de créditos que ela fornece.

O listing abaixo traz um exemplo da estrutura da entidade:

- Nome
- Código (Chave Primária)
- Departamento
- Ementa
- Descrição
- Pré-requisitos
- Período ideal
- Créditos:
  - Aula
  - Trabalho





### Restrições de Domínio:

- **dis\_Nome:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **dis\_Codigo:** Sequência de caracteres alfanuméricos. No mínimo de 4 e máximo de 9 caracteres. Não pode ser nulo.
- **dis\_Departamento:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **dis\_Ementa:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **dis\_Descricao:** Sequência de caracteres UTF-8. Mínimo de 4 e máximo de 1000 caracteres. Não pode ser nulo.
- **dis\_PreRequisitos:** Sequencia de caracteres alfanuméricos. No mínimo de 4 de 80 caracteres.
- **dis\_PeriodoIdeal:** Número inteiro entre 1 e 12.
- **dis\_Aula:** Inteiro positivo entre 1 e 50.
- **dis\_Trabalho:** Inteiro positivo entre 1 e 50.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Disciplina no formato (Nome, Código, Depart, Ementa, Descrição, PreRequis, PerIdeal, Cred):

- Introdução a Computação; MAC110; MAC; Variáveis, Condicionais, Loops; Conceitos básicos de Programação; null; 1; 4, 2.
- Analise de Algoritmo; MAC425; MAC; Complexidade, Programação Dinâmica, Gulosos; Analisar custo de tempo e espaço extra em algoritmos clássicos; MAC323, MAC215; 4, 0.

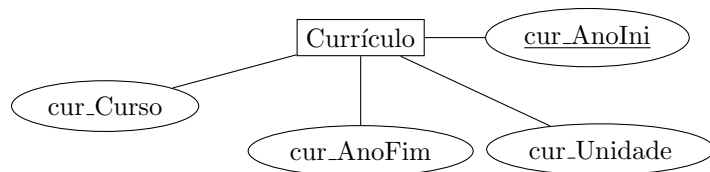
### 2.1.6 Currículo:

A entidade regular Currículo é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um currículos dos cursos da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Currículo será composta por seu Ano de Inicio, Ano de Fim, este último podendo ser nulo, indicando que o currículo ainda está em vigência para um certo Curso, Unidade e Curso a qual o currículo pertence.

O listing abaixo traz um exemplo da estrutura da entidade:

- Ano de Inicio (Chave primária)
- Ano de Fim
- Unidade
- Curso



### Restrições de Domínio:

- **cur\_AnoIni:** Número inteiro de 4 dígitos entre 1500 e +infinito.
- **cur\_AnoFim:** Número inteiro de 4 dígitos entre 1500 e +infinito.
- **cur\_Curso:** Sequência de caracteres. No mínimo de 2 até 80 caracteres.
- **cur\_Unidade:** Sequência de caracteres. No mínimo de 2 até 80 caracteres.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Currículo no formato (AnoInicio, AnoFim, Curso, Unidade):

- 01/01/2000; 31/12/2016; 42; IME.
- 01/01/2017; null; 42; IME

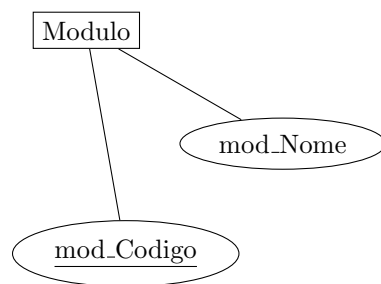
### 2.1.7 Módulo:

A entidade regular Módulo é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos módulos existentes dos cursos da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Módulo será composta por seu Nome e Código, este sendo sua chave primária.

O listing abaixo traz um exemplo da estrutura da entidade:

- Nome
- Código (Chave primária)



### Restrições de Domínio:

- **mod\_Nome:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **mod\_Codigo:** Sequência de caracteres numéricos. No mínimo de 1 e máximo de 9 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Módulo no formato (Nome, Codigo):

- Matematica Discreta I; 1.
- Algoritmos 2; 5.

### 2.1.8 Trilha:

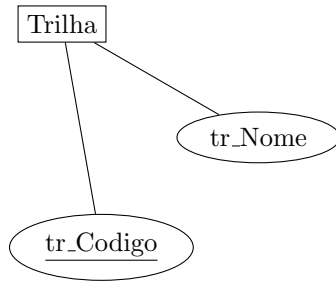
A entidade regular Trilha é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada uma das trilhas dos cursos da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Trilha será composta por seu Nome e Código, este último sendo sua chave primária.

O listing abaixo traz um exemplo da estrutura da entidade:

- Nome

- Código (Chave primária)



#### Restrições de Domínio:

- **tr\_Nome:** Sequência de caracteres alfanuméricos, apóstrofe, hífen e espaço. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **tr\_Codigo:** Sequência de caracteres alfanuméricos. No mínimo de 4 e máximo de 9 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Trilha no formato (Nome, Codigo):

- Inteligência Artificial; 10.
- Teoria da Computação; 3.

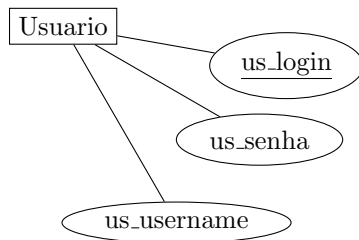
#### 2.1.9 Usuário:

A entidade regular Usuário é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos Usuários da nossa plataforma presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Usuário será composta por seu Login, este sendo sua chave primária, sua Senha e seu Username.

O listing abaixo traz um exemplo da estrutura da entidade:

- Login (Chave primária)
- Senha
- Username



#### Restrições de Domínio:

- **us\_login:** Sequência de caracteres alfanuméricos. No mínimo de 6 até 80 caracteres. Não pode ser nulo.
- **us\_senha:** Sequência de caracteres alfanuméricos. No mínimo de 6 até 80 caracteres. Não pode ser nulo.
- **us\_username:** Sequência de caracteres alfanuméricos, separados ou não pode espaço. No mínimo de 6 até 80 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Usuário no formato (Login, Senha, Username):

- anabeatriz2007; 123456; Ana Bia.
- marcelinho2000; 102030; Marcelo Silva

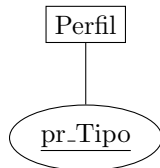
### 2.1.10 Perfil:

A entidade regular Perfil é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos Perfis da nossa plataforma presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Perfil será composta apenas por seu Tipo, que será sua chave primária.

O listing abaixo traz um exemplo da estrutura da entidade:

- Tipo



#### Restrições de Domínio:

- **pr\_Tipo:** Sequência de caracteres alfanuméricos. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis entradas a serem inseridas em Perfil no formato (Tipo):

- Visitante
- Aluno Monitor.

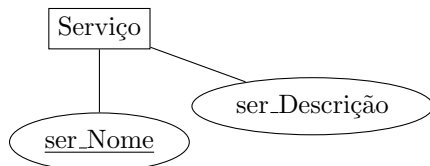
### 2.1.11 Serviço:

A entidade regular Serviço é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos Serviços presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Serviço será composta por seu Nome, que será sua chave primária, e uma Descrição de cada um deles.

O listing abaixo traz um exemplo da estrutura da entidade:

- Nome (Chave primário)
- Descrição



#### Restrições de Domínio:

- **ser\_Nome:** Sequência de caracteres alfanuméricos. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.
- **ser\_Descrição:** Sequência de caracteres alfanuméricos. No mínimo de 4 e máximo de 80 caracteres. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Serviços no formato (Nome, Descrição):

- Alteração; Alterar atributos de entidades e relacionamentos do Banco de Dados
- Inserção; Inserir dados no Banco de Dados, como novos alunos e novas disciplinas.

## 2.2 Agregados

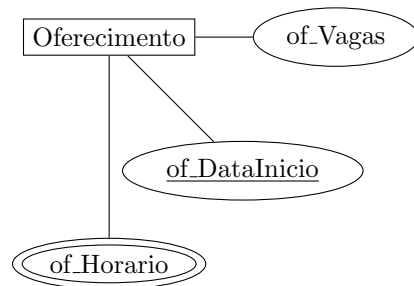
### 2.2.1 Oferecimento:

O agregado Oferecimento é um agregado chave para nosso modelo e é nele que armazenaremos todos os dados das instâncias de cada um dos Oferecimentos presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Oferecimento será composta por sua Data de Início, que será sua chave primária, seu horário, atributo multivalorado, e a quantidade máxima de Vagas que o oferecimento suporta.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Início (Chave Primária)
- Horário
- Vagas



#### Restrições de Domínio:

- **ser\_DataInicio:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **ser\_Horario:** Horário no formato "ddd hh:mm - hh:mm", no qual ddd = dia da semana, hh = hora (00-23) e mm=minutos (00-59). Não pode ser nulo.
- **ser\_Vagas:** Número inteiro entre 1 e +infinito. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Oferecimento no formato (DataInicio, Horario, Vagas):

- 01/02/2019; TER 08:00 - 09:40, QUI 10:00 - 11:40; 60.
- 01/08/2019; SEG 10:00 - 11:40, QUA 08:00 - 09:40; 40.

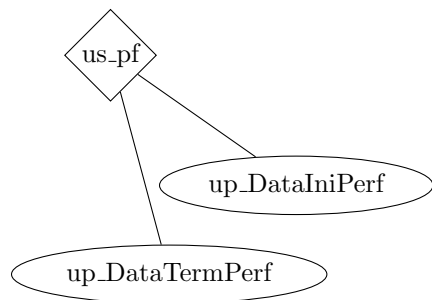
## 2.3 Relacionamentos

### 2.3.1 pe\_us:

Não identificamos a necessidade deste relacionamento conter atributos, pois só importa o mapeamento entre as entidades Pessoa e Usuário.

### 2.3.2 us\_pf:

- Data de início
- Data de término



#### Restrições de Domínio:

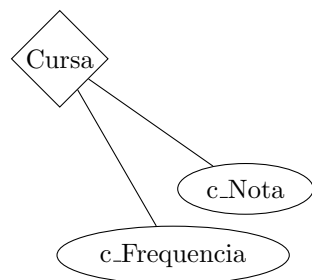
- **up\_DataIniPerf:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **up\_DataTermPerf:** Data no formato dd/mm/aaaa. Não pode ser nulo.

#### 2.3.3 pf\_se:

Não identificamos a necessidade deste relacionamento conter atributos, pois só importa o mapeamento entre as entidades Perfil e Serviço.

#### 2.3.4 Cursa:

- Frequência
- Nota

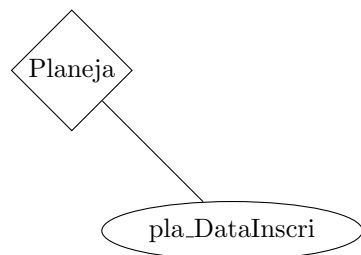


#### Restrições de Domínio:

- **c\_Frequencia:** Número entre 0 e 100.
- **c\_Nota:** Número entre 0 e 10.

#### 2.3.5 Planeja:

- Data de Inscrição

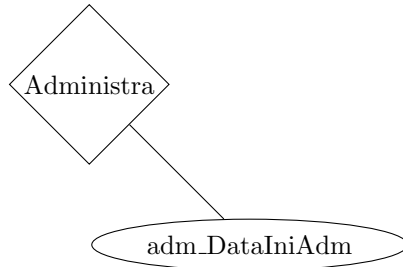


### Restrições de Domínio:

- **pla\_DataInscri:** Data no formato dd/mm/aaaa. Não pode ser nulo.

### 2.3.6 Administra:

- Data de Início da Administração



### Restrições de Domínio:

- **adm\_DataIniAdm:** Data no formato dd/mm/aaaa. Não pode ser nulo.

### 2.3.7 rel\_cur\_tri:

Não identificamos a necessidade deste relacionamento conter atributos, porque sua definição já está contida na relação Currículo.

### 2.3.8 tr\_mo:

Não identificamos a necessidade deste relacionamento conter atributos, pois só importa o mapeamento entre as entidades Trilha e Módulo.

### 2.3.9 rel\_dis\_mod:

Não identificamos a necessidade deste relacionamento conter atributos, pois só importa o mapeamento entre as entidades Disciplina e Módulo.

### 2.3.10 Funcionalidades esperadas

fazer isso no final

- Permitir ao aluno planejar um cronograma de disciplinas segundo uma trilha escolhida;
- Data de término

## 2.4 Modelo lógico

Já temos o modelo lógico no papel, falta criar a imagem// vamos usar um software que o César recomendou