

Exercício-Programa 1: Base de dados

Bruna Meneguzzi
Caio Calisto Gaede Hirakawa
Eduardo Lima
Jiang Zhi
Matheus Santos Conceição

13 de Abril de 2019

Conteúdo

1	Parte I: Entendimento e validação do modelo	2
2	Parte II: Modelo Conceitual e suas classes abstratas	3
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:	14
2.3.6	Administra:	14
2.3.7	rel_cur_tri:	15
2.3.8	tr_mo:	15
2.3.9	rel_dis_mod:	15
2.3.10	Funcionalidades esperadas	15
2.4	DER-X	16
3	Parte III: Modelo lógico e Físico	17
3.1	Modelo lógico	17
3.2	Modelo físico	17

Capítulo 1

Parte I: Entendimento e validação do modelo

Nesse exercício-programa iremos modelar o banco de dados para um sistema de gestão de grades curriculares da USP, a partir dos dois modelos conceituais oferecidos. No capítulo atual 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 deles.

Alguns pontos que sabemos é que nesse banco de dados utilizamos uma estrutura de persistência, ou seja, os dados somente podem ser acrescentados e além disso não podem ser modificados. Essa falta de variabilidade de dados faz com que a modelagem do banco de dados fique mais complicada, criando uma necessidade de um planejamento antecipado para alguns sistemas que modificam de ano para ano seus dados, como os cursos obrigatórios, onde os cursos necessários podem mudar, os cursos lecionados, onde os cursos disponíveis podem mudar, o novo sistema de trilhas, onde os requerimentos para cumprir a trilha podem mudar.

Para os cursos obrigatórios e o novo sistema de trilhas podemos usar um sistema externo para colocar as matérias feitas e verificar quais são os requisitos que precisam ser cumpridos. Após a verificação podemos marcar no banco de dados quais foram as matérias cumpridas e se terminamos os cursos obrigatórios e/ou completamos uma trilha, evitando criar um banco de dados para marcar os requisitos da trilha e das obrigatórias de cada ano dentro do banco. Mas devido a ausência desse sistema externo, precisamos criar um módulo para colocar os currículos.

O sistema de cursos lecionados pode ser feito facilmente, pois acrescentamos os dados ao decorrer das aulas lecionadas pelos professor sobre a disciplina.

Inicialmente foi proposto o modelo MER-X 1, onde vemos que existe um trecho onde definimos a grade como obrigatória ou optativa, essa trecho pode ser transformado em informação da entidade Disciplina. Como mostrado no modelo MER-X 2 onde o banco de dados fica mais compacto.

Assim, utilizaremos o modelo MER-X 2 com algumas modificações para a modelagem do nosso banco de dados.

Capítulo 2

Parte II: Modelo Conceitual e suas classes abstratas

2.1 Entidades

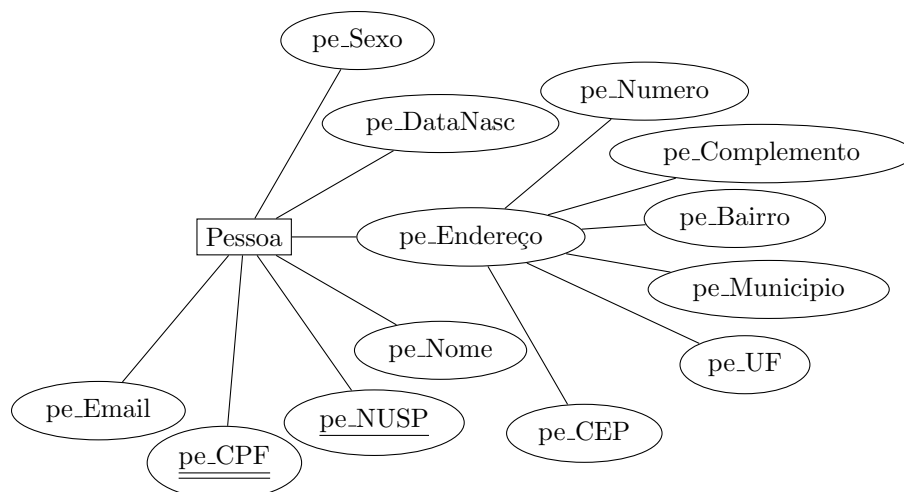
2.1.1 Pessoa:

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_DataNasc:** 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, Endereco):

- 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

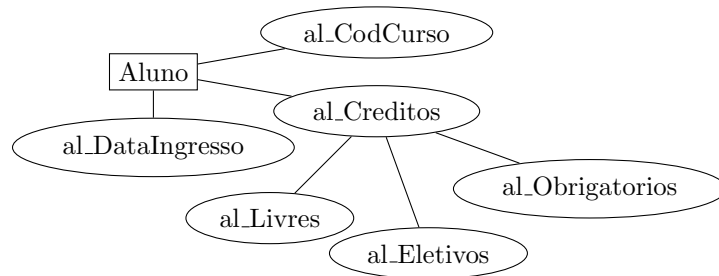
2.1.2 Aluno:

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.

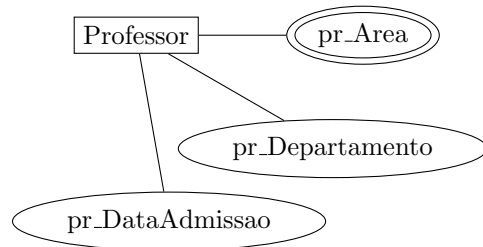
2.1.3 Professor:

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 (Departamento, DataAdmissao, Area):

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

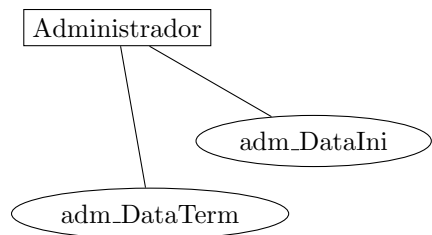
2.1.4 Administrador:

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 (DataIni, DataTerm):

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

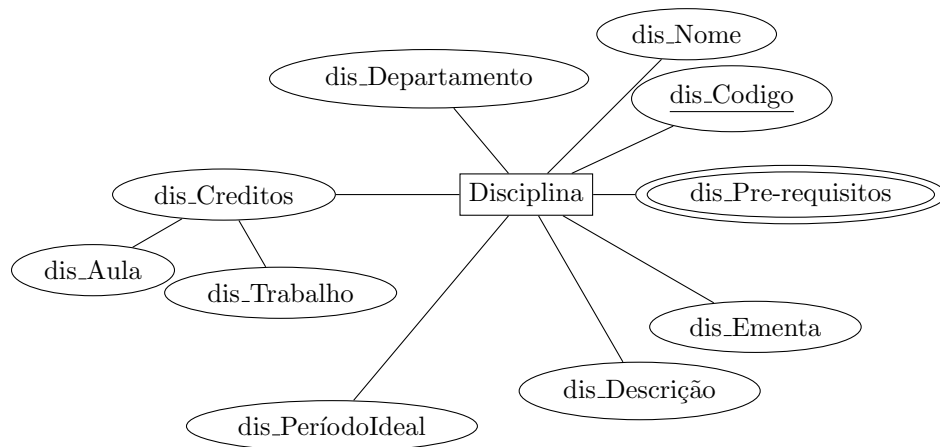
2.1.5 Disciplina:

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_Depart:** 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_PreRequis:** Sequência de caracteres alfanuméricos. No mínimo de 4 de 80 caracteres.
- **dis_PerIdeal:** 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, Codigo, Depart, Ementa, Descricao, PreRequis, PerIdeal, Aula, Trabalho):

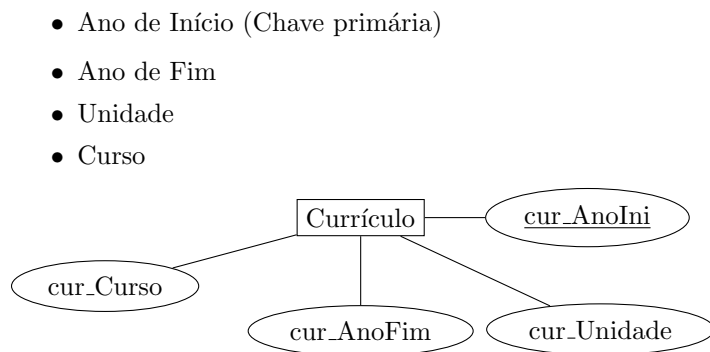
- Introdução a Computação; MAC110; MAC; Variáveis, Condicionais, Loops; Conceitos básicos de Programação; null; 1; 4, 2.
- Analise de Algoritmo; MAC425; MAC; Complexidade, Programação Dinâmica, Gulosos; Analisar custo de tempo e espaço extra em algoritmos clássicos; MAC323, MAC215; 4, 0.

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:



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 (AnoIni, AnoFim, Curso, Unidade):

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

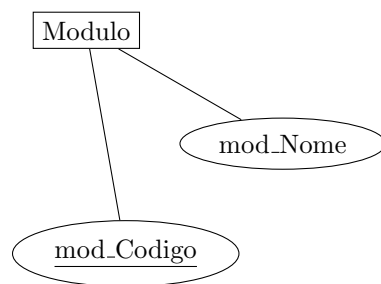
2.1.7 Módulo:

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:

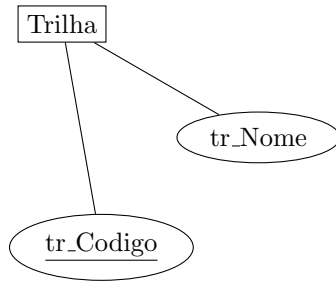
A entidade regular Trilha é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada uma das trilhas dos cursos da USP presentes na nossa base de dados.

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, Código):

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

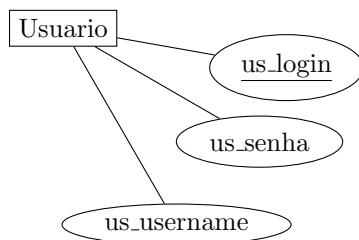
2.1.9 Usuário:

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

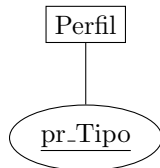
2.1.10 Perfil:

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.

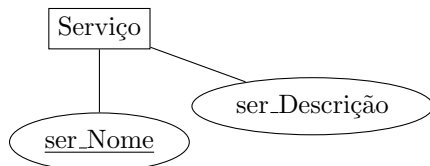
2.1.11 Serviço:

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

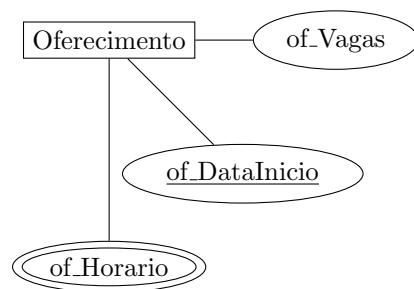
2.2.1 Oferecimento:

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.

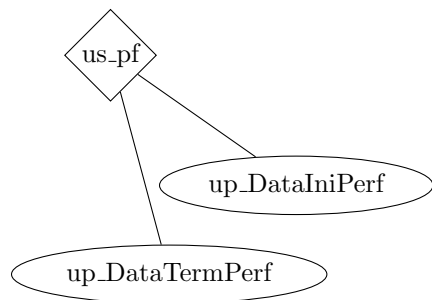
2.3.2 us_pf:

O relacionamento us_pf é um relacionamento chave para nosso modelo e é nele que armazenaremos todos os dados das instâncias de cada usuário com o tipo de perfil que ele possui.

Segundo nosso modelo, cada instância de us_pf será composta pela Data de início de quando um usuário adquiriu aquele perfil e sua data de término.

O listing abaixo traz um exemplo da estrutura da entidade:

- 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.

O listing traz um exemplo de possíveis tuplas a serem inseridas em **us_pf** no formato (DataIniPerf, DataTermPerf):

- 01/02/2019; 30/06/2019.
- 01/08/2019; 31/12/2019.

2.3.3 pf_se:

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

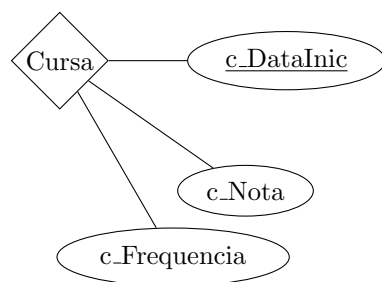
2.3.4 Cursa:

O relacionamento Curso é um relacionamento chave para nosso modelo e é nele que armazenaremos todos os dados das instâncias de cada um dos alunos da USP que cursarem alguma disciplina presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Cursa será composta pela Data de quando o Aluno cursou a Disciplina, que será a chave primária desse relacionamento, sua Frequência e Nota.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Início (Chave primária)
- Frequência
- Nota



Restrições de Domínio:

- **c_DataIni:** Data no formato dd/mm/aaaa. Não pode ser nulo.
- **c_Frequencia:** Número entre 0 e 100.
- **c_Nota:** Número entre 0 e 10.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Cursa no formato (DataIni, Frequência, Nota):

- 01/02/2019; null; null.
- 01/08/2019; null; null.

Na inserção desse relacionamento no Banco de Dados, todo aluno começará com valores nulos de frequência de nota, implicando que ele está cursando aquela matéria. No momento em que ele realizou a matéria, um procedure será feito para alterar sua nota e frequência.

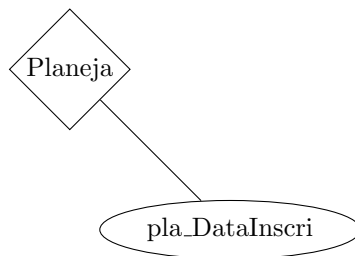
2.3.5 Planeja:

O relacionamento Planeja é um relacionamento chave para nosso modelo e é nele que armazenaremos todos os dados das instâncias de cada um dos alunos da USP e seu planejamento em cursar alguma disciplina presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Planeja será composta pela Data de quando o Aluno pretende cursar uma disciplina.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Inscrição



Restrições de Domínio:

- **pla_DataInscri:** Data no formato dd/mm/aaaa. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Planeja no formato (DataInscri):

- 01/02/2019.
- 01/08/2019.

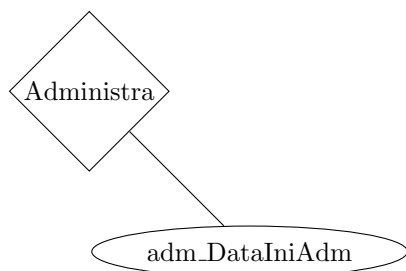
2.3.6 Administra:

O relacionamento Administra é um relacionamento chave para nosso modelo e é nele que armazenaremos todos os dados das instâncias de Administrador da USP e o início do seu mandato.

Segundo nosso modelo, cada instância de Administra será composta pela Data de quando o Administrador começou a ser responsável por um curso.

O listing abaixo traz um exemplo da estrutura da entidade:

- Data de Início da Administração



Restrições de Domínio:

- **adm_DataIniAdm:** Data no formato dd/mm/aaaa. Não pode ser nulo.

O listing traz um exemplo de possíveis tuplas a serem inseridas em Administra no formato (DataIniADM):

- 01/02/2019.
- 01/08/2019.

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

- Permitir ao aluno planejar um cronograma de disciplinas segundo uma trilha escolhida;
- Permitir o professor ao controle das disciplinas e dos eventuais oferecimentos dessas disciplinas;
- Permitir a administradores a gestão de grades curriculares;

2.4 DER-X

Como a modelagem do DER-X ficou muito extensa tivemos uma dificuldade de colocar essa modelagem dentro do pdf, mantendo as informações de forma legível.

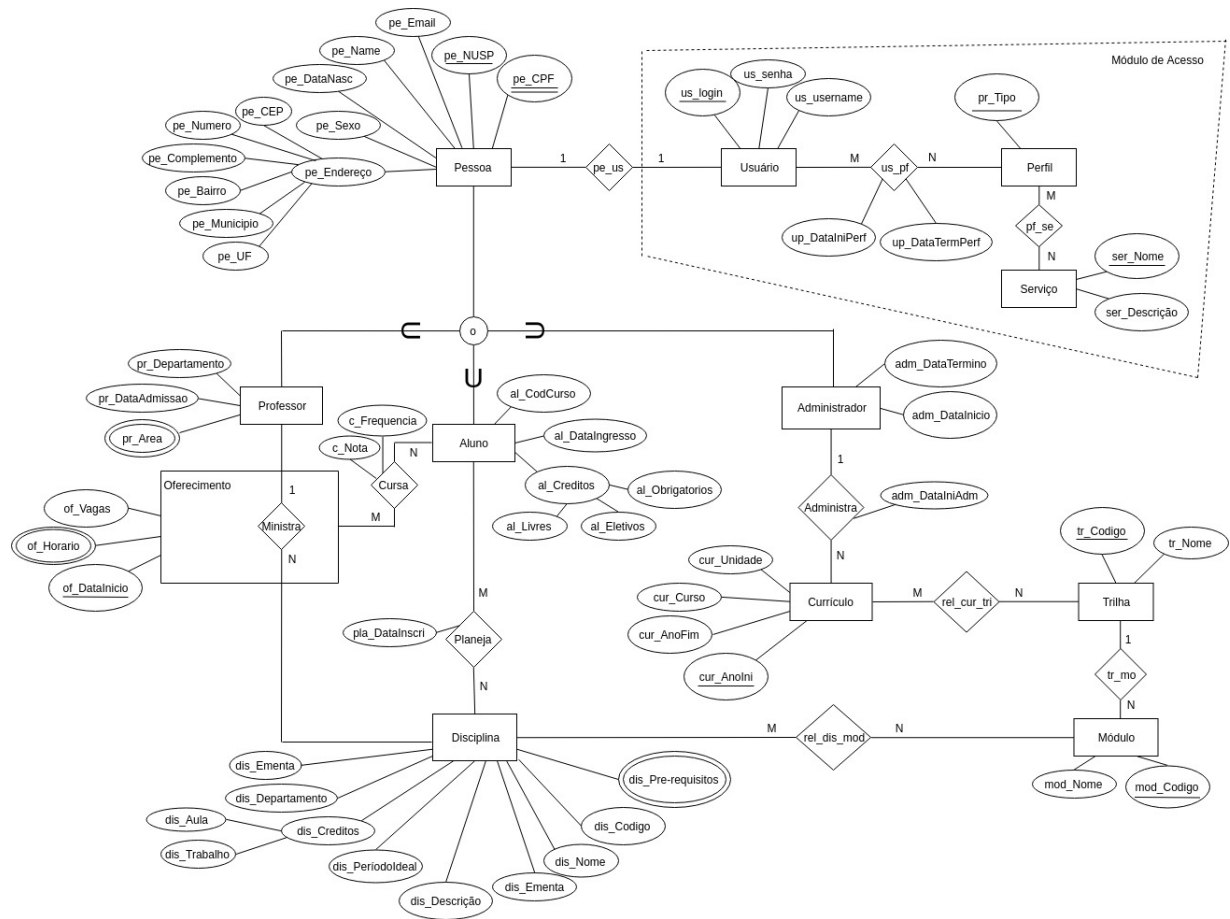


Figura 2.1: DER-X

Essa imagem, pode ser encontrada também na pasta imagens dentro do arquivo DER_EP1.jpg

Capítulo 3

Parte III: Modelo lógico e Físico

3.1 Modelo lógico

O modelo conceitual pode ser transformado em lógico de maneira fácil, agora só precisamos saber quais são as chaves primárias e as chaves estrangeiras.

Para as chaves primárias podemos definir diretamente do modelo conceitual.

Para as chaves estrangeiras, precisamos observar para as relações, para as especializações, para a multivaloração.

Quando temos relações podemos criar uma chave estrangeira da tabela de relações para suas respectivas chaves primárias das entidades relacionadas.

Quando temos especialização, como aluno, professor e administrador, criamos a ligação dessas chave estrangeira dessas entidades para a entidade pessoal.

Quando temos uma multivaloração, criamos a ligação desses a entidade que contém esses múltiplos valores para a entidade principal que queremos achar.

Após finalizar esse processo, conseguimos um modelo lógico, mas esse modelo lógico ficou muito extenso e criamos um pdf que pode ser encontrado na pasta pdf no arquivo ModeloLogico.pdf.

3.2 Modelo físico

Nessa parte convertemos o modelo lógico em modelo físico, a partir das informações dentro das entidades e das suas chaves primárias e suas chaves estrangeiras criada no modelo lógico.

Algumas verificações de consistência de dados podem ser muito difícil de fazer internamente no banco de dados como saber se o e-mail é válido, então podemos fazer um checador de consistência externa no momento de inserir os dados e de periodos e periodos de tempo podemos checar a consistência interna do banco de dados.