

## MAC0350 - EP1

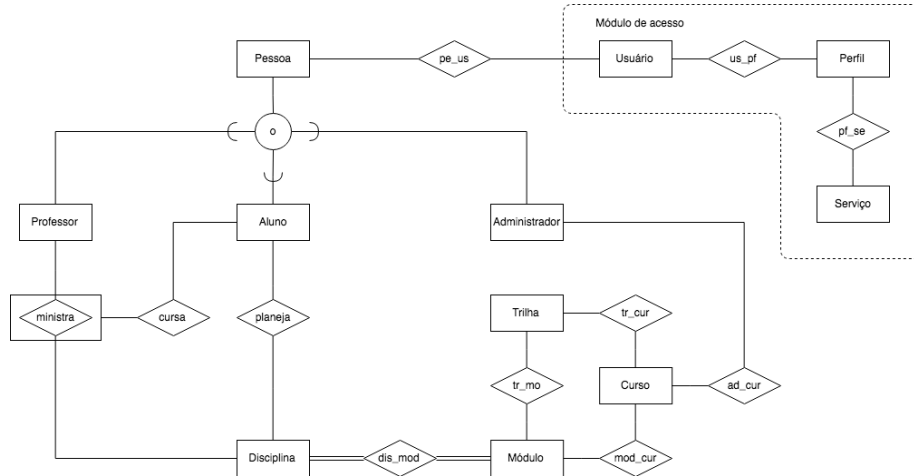
André Ferrari Moukarzel  
9298169

Arthur Vieira Barbosa  
6482041

Gabriel Sarti Massukado  
10284177

Matheus Lima Cunha  
10297755

# 1 Validação do Modelo



Alteramos o modelo disponível da seguinte forma. A principal alteração é quanto aos módulos/trilhas. Adicionamos uma entidade Curso com o objetivo de tornar o modelo mais similar ao sistema real e diminuir o número de informações redundantes na base.

Da forma como entendemos a versão anterior do modelo, poderiam existir, por exemplo, as Trilhas "BCC" e "BCC com especialização em IA", e é óbvio que elas apresentariam majoritariamente as mesmas disciplinas.

Com a abstração extra da entidade Curso, teríamos o Curso "BCC" e a Trilha "IA", com o Curso contendo todas as matérias base (Módulo de matérias obrigatórias, eletivas, livres, etc..) e a Trilha apresentaria apenas os módulos referentes a IA (Sistemas, Introdução a IA, etc...).

## 2 Descrições e Restrições

### 2.1 Generalização: Pessoa

A entidade "Pessoa" generaliza de forma sobreponível as entidades regulares "Professor", "Aluno" e "Administradores".

A pessoa possui CPF e nome como atributos, sendo que CPF é chave primária.

#### 2.1.1 Restrições

- pes.cpf: Número de Cadastro de Pessoa Física da pessoa. Deve ser uma sequência de 11 dígitos constituindo uma sequência válida seguinte as regras da Receita Federal. Não pode ser NULL.

- **pes\_name:** Nome completo do individuo conforme escrito em sua certidão de nascimento. É composto de letras, podendo conter no máximo 200 caracteres.

## 2.2 Entidade Regular: Professor

O professor é uma entidade especializada de pessoa onde armazenaremos todos os dados das instâncias de cada um dos professores da USP presentes na base de dados.

Um professor possui os atributos NUSP e Departamento, sendo que NUSP é chave primária.

### 2.2.1 Restrições

- **prof\_nusp:** Número de identificação da USP do professor. É composta de até 9 dígitos e, por ser a chave primária, não pode ser nulo.
- **prof\_dept:** Identificação de qual o departamento o professor pertence. Os valores que pode assumir são: MAT, MAC, MAE ou MAP.

## 2.3 Entidade Regular: Aluno

O aluno é uma entidade especializada de pessoa onde armazenaremos todos os dados das instâncias de cada um dos alunos da USP presentes na base de dados.

Uma instância de aluno possui o atributo NUSP, que é chave primária.

### 2.3.1 Restrições

- **al\_nusp:** Número de identificação da USP do professor. É composta de até 9 dígitos e, por ser a chave primária, não pode ser nulo.

## 2.4 Entidade Regular: Administrador

O administrador é uma entidade especializada de pessoa em que armazenaremos todos os dados das instancias de cada um dos administradores na base de dados.

Uma instância de administrador possui os atributos Data de Início e Data de Fim, representando o período de tempo no qual uma Pessoa foi Administrador, e-mail (que é uma chave secundária) e seu CPF, que é a chave primária.

### 2.4.1 Restrições

- **adm\_dat\_in:** Data não nula, representando o início da gestão do administrador.
- **adm\_dat\_out:** Data representando o fim da gestão do administrador.

- `adm_email`: E-mail de até 40 caracteres, contendo um único "@" em uma posição que não seja do primeiro nem último carácter. Não pode ser NULL.
- `adm_cpf`: Número de Cadastro de Pessoa Física da pessoa. Deve ser uma sequência de 11 dígitos constituindo uma sequência válida seguinte as regras da Receita Federal. Não pode ser NULL.

## 2.5 Entidade Regular: Disciplina

A disciplina é uma entidade em que armazenaremos todos os dados das instâncias de cada uma das disciplinas na base de dados.

Uma disciplina possui um nome, um código, um número de créditos trabalho, um número de créditos aula e período ideal, sendo que o código da matéria é sua chave primária.

### 2.5.1 Restrições

- `dis_name`: Nome por extenso da disciplina. É uma sequência de até 80 caracteres.
- `dis_cod`: O código da matéria, contendo 3 letras seguidas de 4 dígitos numéricos. Não pode ser NULL.
- `dis_work_creds`: Número de créditos trabalho concedidos pela disciplina. Deve ser um número maior ou igual a zero.
- `dis_class_creds`: Número de créditos aula concedidos pela disciplina. Deve ser um número maior ou igual a zero.
- `dis_ideal_period`: O período ideal para essa disciplina ser cursada. Deve ser um número maior que zero e menor que 11, dado que nenhum curso da USP tem duração de mais de 10 semestres.

## 2.6 Entidade Regular: Curso

Um curso é uma entidade que representa um curso de graduação, licenciatura, etc... O BCC, por exemplo, seria um curso.

Um curso possui os atributos nome e código, onde o código é a chave primária. Com essas atribuições, o BCC de 2014 seria considerado um Curso diferente do BCC atual. Isso é intencional, visto que ambos os cursos tem requerimentos diferentes para sua conclusão.

### 2.6.1 Restrições

- `cur_name`: Nome por extenso do Curso. É uma sequência de até 60 caracteres alfanuméricos. O nome do BCC por extenso possui 36 caracteres e é um nome de curso grande, então acreditamos que nenhum curso ultrapasse 60 caracteres.

- `cur_code`: Número inteiro. Não pode ser nulo, visto que é a chave primária.

## 2.7 Entidade Regular: Trilha

A trilha é uma entidade regular que representa uma especialização de um curso, como, por exemplo, a trilha em Sistemas de Software do BCC. Uma trilha possui um atributo nome, que é sua chave primária.

### 2.7.1 Restrições

- `tri_name`: Nome por extenso da trilha. É uma sequência de até 80 caracteres alfanuméricos. Não pode ser nulo, por ser a chave primária.

## 2.8 Entidade Regular: Módulo

Um módulo é um conjunto de Disciplinas específicas. Um módulo pode ser, por exemplo, o módulo de matérias obrigatórias de um curso, ou o módulo de "Sistemas" da trilha de IA.

Cada módulo possui um nome, o número de créditos a serem cumpridos e um código, sendo o código sua chave primária.

- `mod_name`: Nome por extenso do módulo. É uma sequência de até 40 caracteres.
- `mod_cred_min`: Número de mínimo de créditos que devem ser concluídos para completar o módulo. Deve ser um número maior que zero e menor que o total de créditos oferecidos por todas as Disciplinas presentes no módulo.
- `mod_code`: Número inteiro. Não pode ser nulo, visto que é a chave primária. Com a mudança do que os Módulos representam, se torna possível que nomes de módulos se repitam (por exemplo, muitos Cursos podem ter um Módulo "Obrigatórias"), então criamos um código para diferenciar os Módulos.

## 2.9 Entidade Regular: Usuário

O usuário é uma entidade regular em que armazenamos cada uma das instâncias dos usuários do serviço fornecido pelo nosso banco de dados.

Cada usuário tem o seu login, uma string única que representa seu nome virtual, uma senha para acessar sua conta.

### 2.9.1 Restrições

- `user_login`: String de até 20 caracteres, tendo que necessariamente começar com uma letra. Não pode ser nulo e é único, pois é o atributo chave do usuário.

- `user_password`: String de até 20 caracteres, com um mínimo de 6 caracteres. Deve conter pelo menos uma letra maiúscula, uma minúscula, um dígito e um carácter especial.

## 2.10 Entidade Regular: Perfil

O perfil representa os "cargos" de um usuário, isso é, as funções que ele pode realizar no site. Exemplos de perfis são "Pessoa" (pode olhar as trilhas), "Aluno" (pode planejar suas disciplinas) e "Administrador" (pode modificar os módulos das trilhas). Cada perfil possui um nome e (opcionalmente) uma descrição.

- `perf_name`: String de até 20 caracteres, tendo que necessariamente começar com uma letra e não pode ser nulo. É o nome do perfil e é sua chave primária.
- `perf_desc`: String de até 100 caracteres. Contém uma descrição do perfil, incluindo suas permissões.

## 2.11 Entidade Regular: Serviço

Serviços abstraem as capacidades e permissões que os usuários tem no banco de dados. Exemplos de serviços são a leitura dos dados das disciplinas, modificar as informações de uma trilha, deletar uma disciplina e similares. Basicamente definem as permissões para realizar um CRUD no banco de dados. Um serviço possui um código, que distingue uma permissão de outra, uma descrição do serviço, o conjunto de tabelas relacionadas ao serviço e que permissões ele possui nessas tabelas.

- `serv_code`: Sequencia de 4 dígitos que define o serviço. Não pode ser nulo pois é o atributo chave do serviço.
- `serv_desc`: String de até 50 dígitos descrevendo as permissões que esse serviço implica.
- `serv_tables`: Atributo multi-valorado contendo as tabelas relacionadas a esse serviço.
- `serv_permissions`: Atributo multi-valorado, sendo um ou mais valores do conjunto Create, Read, Update, Delete, que define o que o usuário pode realizar com as tabelas descritas no atributo Tabelas.

## 2.12 Agregação: Oferecimento

Um oferecimento é uma agregação das entidades "Professor" e "Disciplina" e da relação "Ministra" em que armazenamos as instâncias de oferecimentos de certas disciplinas em um semestre.

Uma instância de oferecimento tem como atributos turma e múltiplas datas/horários de início e fim, representando os horários da semana em que as aulas ocorrem.

### 2.12.1 Restrições

- `of.class`: Inteiro que indica um grupo de alunos que está cursando o oferecimento. Não pode ser NULL.
- `of.time`: Lista de tuplas que indica em que horários as aulas do oferecimento ocorrem. Cada tupla contém um *datetime* que representa o horário de início da aula, um *datetime* que representa o horário de término da aula e um inteiro no intervalo  $[1, 7]$  que representa o dia da semana. Nenhum desses valores pode ser NULL e o horário deve possuir ao menos uma tupla (Ter ao menos uma tupla é necessário porque o modelo de aulas atual da USP sempre requer um horário. Na prática, há várias matérias sem aulas presenciais, então não seria necessário ter qualquer tupla nesses casos).

### 2.13 Relação: `pe_us`

Relação que representa a posse de uma conta por uma pessoa. Relação 1 para 1 entre as entidades "Pessoa" e "Usuário".

### 2.14 Relação: `us_pf`

Relação que abstrai os perfis possuídos por um usuário. É uma relação com cardinalidade M:N, uma vez que cada pessoa pode possuir vários perfis e cada perfil pode pertencer a mais de uma pessoa. Cada relação possui uma data de adição e uma data de remoção.

- `us_pf.date.in`: Data de Adição. Data que mostra a data que determinado usuário se juntou a determinado perfil. Não pode ser nula.
- `us_pf.date.out`: Data de remoção. Data que mostra até quando o usuário pertence a determinado perfil. Útil para quando queremos dar acesso temporário a um usuário, como monitores que têm permissões atreladas a seu período de monitoria. Caso seja nula, o usuário pertence a aquele perfil até que seja removido.

### 2.15 Relação: `pf_se`

Relação que liga um perfil a um serviço, representando os poderes do perfil em relação ao banco de dados. Um perfil pode ter vários serviços, e um serviço estar presente em vários perfis, configurando uma relação de cardinalidade M:N.

### 2.16 Relação: `ministra`

Relação que representa a ministração de uma disciplina por um professor. Um professor pode ministrar múltiplas disciplinas, enquanto uma disciplina pode ser ministrada por apenas múltiplos professores, configurando uma relação M:N. Uma ministração possui os atributos semestre e ano, indicando o período em que a ministração ocorre.

### 2.16.1 Restrições

- `minist_semester`: Inteiro que indica o semestre em que ocorre o oferecimento. Deve ser "1" ou "2".
- `minist_year`: Inteiro que indica o ano em que ocorre o oferecimento. Deve ser um inteiro maior que 1912 (data de fundação da USP, caso queiram adicionar registros de ministrações passadas por algum motivo louco).

## 2.17 Relação: `curso`

Relação que representa um aluno matriculado em um oferecimento de uma disciplina. Um aluno pode cursar múltiplos oferecimentos e um mesmo oferecimento pode ser cursado por múltiplos alunos, configurando uma relação N:M. Esta relação possui os atributos `nota` e `presenca`, necessários para determinar se um aluno foi aprovado. Podem ser nulos inicialmente, pois seus valores só costumam ser atribuídos ao final do curso.

### 2.17.1 Restrições

- `curso_grade`: Nota de um aluno em um oferecimento. A nota é um valor de ponto flutuante maior ou igual a 0.0 e menor ou igual a 10.0.
- `curso_presence`: Presença total de um aluno em um oferecimento que cursa. É um valor de ponto flutuante maior ou igual a 0.0 e menor ou igual a 1.0.

## 2.18 Relação: `planeja`

Relação que representa um aluno com pedido de participação em uma disciplina. Um aluno pode planejar cursar múltiplas disciplinas e uma mesma disciplina pode ser alvo de planejamento de múltiplos alunos, configurando uma relação N:M.

Esta relação possui os atributos `semestre` e `ano` em que o aluno planeja cursar a disciplina. Com isso, teoricamente um aluno poderia planejar seu curso inteiro semestralmente.

### 2.18.1 Restrições

- `plan_semester`: Inteiro que indica o semestre planejado. Deve ser "1" ou "2".
- `plan_year`: Inteiro que indica o ano planejado. Deve ser um inteiro maior ou igual ao ano no momento do planejamento.

## 2.19 Relação: `tr_mod`

Relação que representa a inclusão de um módulo em uma trilha. Um módulo só pode ser parte de uma trilha, entretanto uma trilha pode conter N módulos,



sendo uma relação de cardinalidade 1:N. Uma trilha sempre terá ao menos um módulo.

A relação `tr_mod` possui o atributo Obrigatório, que indica se é obrigatória a conclusão do módulo em questão para a conclusão da trilha.

#### 2.19.1 Restrições

- `tr_mod_mandatory`: Valor binário não nulo. Caso verdadeiro, a conclusão do módulo é obrigatória para a conclusão da trilha em questão.

### 2.20 Relação: `dis_mod`

Relação que representa a inclusão de uma disciplina em um módulo. É uma relação total M:N, pois uma disciplina pode estar em múltiplos módulos, e um módulo pode possuir múltiplas disciplinas. Além disso, toda Disciplina pertence a ao menos um módulo, e todo módulo possui ao menos uma disciplina.

### 2.21 Relação: `tr_cur`

Relação que representa a presença de uma Trilha em um Curso. Uma Trilha só pode pertencer a um Curso, enquanto um Curso pode ter múltiplas trilhas, configurando uma relação de cardinalidade 1:N. Toda Trilha está relacionada a um Curso, porém nem todo Curso está relacionado a Trilhas.

### 2.22 Relação: `mod_cur`

Relação que representa a presença de um Módulo em um Curso. Todo Curso possui ao menos um Módulo, enquanto nem todo Módulo pertence a um Curso (Módulos podem ser exclusivos de Trilhas). Um módulo pode estar em múltiplos Cursos (como um Módulo com as matérias comuns dos cursos de Engenharia), e um Curso pode possuir múltiplos Módulos (Disciplinas obrigatórias, eletivas, livres, etc...), configurando uma relação de cardinalidade M:N.

### 2.23 Relação: `ad_cur`

Relação equivalente à removida "Administra", representando o poder de um Administrador de administrar um Curso. Um administrador pode administrar múltiplos cursos, porém um curso só poderia ser administrado por um administrador, configurando uma relação de cardinalidade N:1.

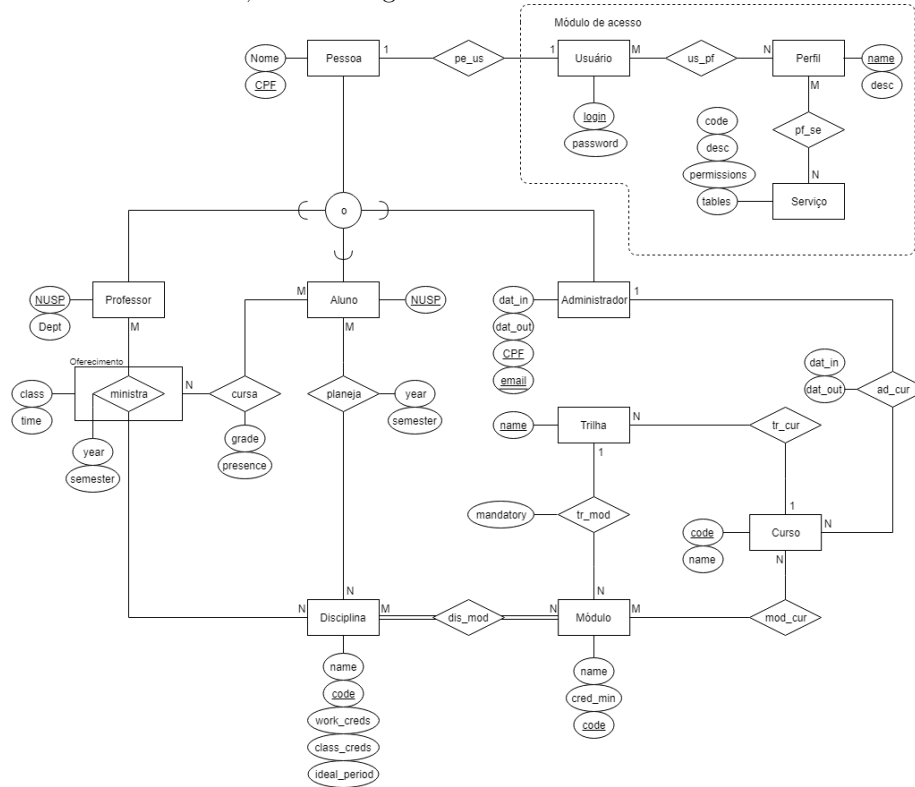
Esta relação possui os atributos Data de Início e Data de Fim, representando o período de tempo no qual o administrador tem o direito de administrar o Curso.

#### 2.23.1 Restrições

- `ad_cur_date.in`: Data de Início. Deve ser uma data não nula.

- ad\_cur\_date\_out: Data de Fim/Saída. Deve ser uma data, mas pode ser nulo.

Com tudo isso, temos o seguinte DER:



### 3 Modelo Lógico