

O Curso vai ter que ter diversos relacionamentos, como também identificação própria, sendo assim, decidimos criar o Conjunto Entidade Curso que tem como identificador próprio o código (PK) e como atributos: nome, carga horária, total de vagas.

Um Curso tem uma sala padrão, e, como Sala é relacionada tanto com curso, quanto com, posteriormente, Turma, optamos por criar o CE Sala, que tem como chave primária um id e a capacidade total da sala. Para relacionar a Sala com a sala padrão do Curso criamos o relacionamento “tem por padrão” de Cardinalidade 1:N, para indicar que um curso tem uma sala por padrão enquanto uma sala pode ser padrão de vários cursos.

Para a classificação do curso optamos por uma abstração de generalização com exclusão mútua e especialização total, sendo assim, ele precisa ser obrigatoriamente um tipo de curso e somente um. Essa abordagem evita o uso de atributos multivalorados, que poderiam complicar a estrutura do modelo.

A entidade Regra (CE Regra) foi definida como um conjunto entidade para que cada regra possa ser identificada de forma única. Com isso, é possível associar uma ou mais regras a diferentes cursos, e vice-versa. Essa entidade possui como chave primária o campo `Id_regra` e conta com um atributo adicional de descrição.

Para modelar a relação entre Curso e Regra, foi criado o conjunto de relacionamento chamado ‘pode ter’ (CR ‘pode ter’), conectando ambas as entidades. Esse relacionamento reflete uma associação muitos-para-muitos (N:M), permitindo que um curso esteja vinculado a diversas regras e que uma mesma regra seja aplicada a múltiplos cursos.

A entidade Infraestrutura foi criada como um conjunto entidade (CE) para possibilitar sua identificação única e permitir, por meio de um relacionamento, que um curso esteja associado a diversas infraestruturas e que uma mesma infraestrutura possa estar vinculada a vários cursos. O conjunto de relacionamento ‘pode precisar’ conecta o CE Curso ao CE Infraestrutura, indicando que um curso pode necessitar de múltiplas infraestruturas e que cada infraestrutura pode ser requerida por diferentes cursos, caracterizando assim uma cardinalidade N:M.

Criamos o Departamento Acadêmico como CE por causa da sua necessidade de identificação própria e possui como chave primária o código e inclui o atributo nome. O conjunto de relacionamento “pertence a” conecta o CE Curso ao CE Departamento Acadêmico, indicando que cada curso está vinculado a um único departamento, enquanto um departamento pode estar associado a diversos cursos. Dessa forma, a cardinalidade desse relacionamento é 1:N.

Disciplina é um Conjunto Entidade, pela sua necessidade de relacionamentos e identificação própria. Ele contém como chave primária o atributo código e os atributos de número de aulas semanais e o material recomendado.

Para fazer com que um Curso seja composto por disciplinas, criamos o Conjunto Relacionamento “composto por” que tem como finalidade mostrar que um curso é composto por diversas disciplinas, enquanto uma disciplina pode ser composta por vários cursos. Sendo assim, sua cardinalidade é N:M.

Para implementar a parte de Pré-Requisitos criamos diversos relacionamentos, sendo eles, o relacionamento “disciplina pré requisito”, que ocorre entre Curso e Disciplina com cardinalidade N:M, indicando que uma disciplina pode ser pré-requisito para vários cursos e um Curso pode ter várias disciplinas como pré-requisitos.

Da mesma forma criamos o auto relacionamento “curso pré requisito”, com a entidade Curso de cardinalidade N:M, que indica que um curso pode ser pré-requisito para vários cursos e que um curso pode ter como pré-requisito, vários cursos.

Usuário deve ser um conjunto Entidade, por causa de identificação própria e pelo fato de o mesmo poder possuir vários tipos, levando a necessidade de fazer uma abstração de generalização. O Conjunto Entidade Usuário tem como chave primária os atributos: nome, sobrenome, telefone. Além desses atributos, este CE também contém os atributos data de nascimento, endereço, o qual é um atributo composto, sexo, senha e email.

Fizemos uma abstração de generalização, com exclusão mútua e especialização total, através do atributo ‘tipo’ que irá definir se o Usuário será um Aluno, Funcionário Administrativo ou Professor. Sendo assim, cada usuário deve ser pelo menos um tipo, mas não pode ser mais de um.

Apesar de nenhum dos CEEs terem chave própria, a abstração de generalização foi feita, pois o professor contém alguns atributos próprios, além de que cada CEE faz relacionamentos com outros CE da base de dados, ou seja, existem relacionamentos que o professor faz que o aluno não faz.

O CE Aluno é um CEE do CEG Usuário. Ele não contém nenhum atributo próprio, sendo assim todos seus atributos são os mesmos do CE Usuário. Ele existe devido ao fato que o CE Aluno contém relacionamentos específicos que nem todo Usuário deve ter.

O CE Funcionário Administrativo é um CEE do CEG Usuário. Ele não contém nenhum atributo próprio, sendo assim todos seus atributos são os mesmos do CE Usuário. Ele existe devido ao fato que o CE Funcionário Administrativo contém relacionamentos específicos que nem todo Usuário deve ter.

O CE Professor é um CEE do CEG Usuário. Apesar de não conter chave própria, ele contém os atributos de área de especialização e titulação.

Cada disciplina é de responsabilidade de um ou vários professores, enquanto os professores ministram a disciplina, sendo assim, criamos o relacionamento “ministra” entre Professor e Disciplina, que tem cardinalidade 1:N.

Segundo a descrição do projeto temos: “Cada matrícula ... a uma oferta de disciplina específica”, sendo assim, criamos o Conjunto Relacionamento “oferece” que envolve o CE Disciplina e o CE Turma. Ele serve para indicar que o CE Turma é o oferecimento de uma disciplina. Cada turma vai conter um id, que será sua chave primária, junto da sua capacidade.

Devido ao fato da matrícula ser em relação a uma oferta de disciplina específica queremos que este oferecimento possa ser identificado unicamente por um código, e não somente pelo conjunto {disciplina, turma}, sendo assim optamos por fazer uma abstração de agregação no CR “oferece”, sendo esta feita pelo primeiro caso, ou seja, *“É necessário identificar cada instância do conjunto de relacionamentos (o CR precisa ter chave própria)”*.

A agregação de “oferece” que tem o nome “Oferecimento” tem como chave primária o código do oferecimento e um outro atributo que indica o período no qual está acontecendo o oferecimento.

Uma turma é um oferecimento de uma disciplina, logo a turma só pode ter 1 disciplina enquanto uma disciplina pode ter várias turmas, levando assim a cardinalidade 1:N.

Para representar que uma turma pode usar diversas salas, tivemos a ideia de criar o relacionamento “usa a”, porém, se ele for simplesmente um relacionamento comum, “usa a”, que seria de cardinalidade N:M, para indicar que uma turma pode usar diversas salas e que uma sala pode ser usada por diversas turmas, só seria possível ter uma única combinação de horário para um certo dia de uma turma em uma mesma sala, o que não é ideal, já que é possível a mesma turma, ter aula, na mesma sala, em horários diferentes em dias diferentes. Sendo assim, criamos uma abstração de agregação com as chaves adicionais: dia, hora\_inicio e hora\_fim.

A matrícula deve ser feita entre Aluno e Oferecimento, sendo assim, a matrícula será entre CE Aluno e o CEA Oferecimento. Não fizemos a matrícula ser no CE Turma, pois o texto deixa claro que a matrícula é sobre oferecimento, enquanto uma Turma é só parte de um oferecimento.

Um aluno pode se matricular em vários oferecimentos de disciplinas, sendo assim, se fizéssemos a relação “matricular” entre Aluno e Oferecimento com cardinalidade 1:N, íamos ter que para um dado Oferecimento só teríamos um aluno matriculado, o que não é o que queremos. Logo, para resolver isso, vamos fazer uma abstração de agregação tendo como chave adicional data. Sendo assim, aluno, data, códigoOferecimento juntos definem uma matrícula, enquanto, se eu pegar o conjunto, aluno, data, eu consigo descobrir quais são todos os oferecimentos, logo, um aluno pode se matricular em vários oferecimentos de disciplinas enquanto para uma dado oferecimento e aluno só existe uma matrícula.

Uma matrícula vai ter como chave adicional data pelo motivo explicado acima, como também um status, um atributo multivalorado, notas, uma data limite até quando pode ser confirmada ou cancelada e o valor da matrícula sem levar em consideração possíveis descontos.

Para representar que uma matrícula pode ter uma bolsa, vimos a necessidade de criar o CE Bolsa, pois caso Bolsa fosse um atributo não seríamos capaz de adicionar informações a mais como descrição e instituição, o que iria fazer com que uma bolsa fosse exatamente igual a um desconto. A bolsa será representada por um id e terá um valor, descrição e instituição. O relacionamento entre Bolsa e Matrícula é dado pela relação “pode possuir” com cardinalidade M:N, o que indica que uma Bolsa pode ser de várias Matrículas e que uma Matrícula pode ter várias Bolsas.

Seguindo o mesmo raciocínio da Bolsa criamos o CE Desconto, sendo que ele será identificado por um id, terá um valor, um motivo e é ligado a Matrícula pela relação “pode possuir” de cardinalidade N:M.

Como a escola mantém o registro de Departamento Acadêmico e o mesmo deve ser identificado unicamente por um código, criamos o CE Departamento Acadêmico que tem como chave primária o código e o atributo nome.

Cada departamento tem um professor chefe, logo, criamos a relação “chefe” entre Departamento e Professor com cardinalidade 1:N, indicando que um departamento pode ter somente um Professor chefe, enquanto um professor pode ser chefe de vários departamentos.

Uma Unidade Escolar é um CE, pois devemos ser capazes de identificar unicamente quem ela é e quais relações ela tem. A Unidade Escolar será definida por um código e terá os atributos cidade, estado, país. Para definir o local da Unidade Escolar criamos o atributo bloco, que pode ser nulo, onde não sendo, ele representa o bloco da Unidade, e criamos o atributo Localidade Específica, que também pode ser nula, onde não sendo, define um local específico, como por exemplo “Campos 2”;

Tanto curso, quanto disciplina, aluno e professor devem estar associados a Unidade Escolar, Sendo assim, criamos para cada par, o relacionamento “associado”, de cardinalidade 1:N, para representar que um curso é de uma Unidade Escolar, enquanto a Unidade Escolar pode ter vários cursos, assim por diante para disciplina, aluno e professor. A escolha de não fazer a relação entre Unidade Escolar e Usuário, invés de aluno e professor foi porque o texto deixa de fora a associação entre Unidade Escolar e Funcionário Administrativo.

Como uma mensagem pode ser enviada por vários tipos de Usuários e pode ser recebida unicamente por um Usuário ou para uma turma toda, houve uma necessidade de ter vários relacionamentos, logo, decidimos criar o CE Mensagem. A mensagem é identificada unicamente por um id, porém, ela contém um texto e o timestamp de quando ela foi enviada.

Para representar que um Usuário pode enviar uma mensagem, criamos o relacionamento “envia”, de cardinalidade 1:N, indicando que o usuário pode enviar várias mensagens enquanto cada mensagem que foi enviada é relacionada a exatamente o Usuário que a enviou.

Para representar que um Usuário pode receber mensagens criamos o relacionamento “destina\_usuario” de cardinalidade N:M, indicando que uma mensagem pode ser enviada para vários usuários e um usuário pode receber várias mensagens. Seguindo a mesma lógica criamos o relacionamento “destina\_turma”, também de cardinalidade N:M, porém agora, para que a turma possa receber mensagens.

Para representar o Aviso, criamos um CE que contém os mesmos tipos de atributos que mensagem. Fizemos isso, pois Aviso têm relações específicas que Mensagem não tem e vice-versa, sendo assim, preferimos não fazer a generalização, o que, na hora do mapeamento, resultaria em atributos desnecessários em Aviso. Aviso tem texto, timestamp e um id, porém ele só pode ser mandado pelo Funcionário Administrativo. Sendo assim, fizemos o relacionamento “envia”, com Funcionário Administrativo para mostrar que ele é capaz de enviar avisos, com cardinalidade 1:N, ou seja, o Funcionário pode mandar vários avisos, mas cada aviso em específico está relacionado a um Funcionário.

Para representar que uma turma pode receber um Aviso Geral criamos o relacionamento “recebe” com Turma de cardinalidade 1:N, ou seja, uma turma pode receber vários avisos, mas cada aviso vai para uma turma.

Como a Avaliação é feita por um Aluno e quem a recebe pode ser o Professor ou uma disciplina no fim do seu período, ou seja, o oferecimento da disciplina, optamos por criar o CE Avaliação. Este CE tem como chave primária um id, junto de um texto, uma avaliação para a didática, uma avaliação do material de apoio, uma avaliação da relevância e uma avaliação de infraestrutura.

Para fazer com que um Aluno possa fazer uma avaliação criamos o relacionamento “faz”, de cardinalidade 1:N, indicando que um Aluno pode fazer várias avaliações e cada avaliação é referente a um Aluno.

Para fazer com que o Professor e o Oferecimento recebam uma avaliação, criamos, para cada um, o relacionamento “recebe”, de cardinalidade 1:N, indicando que um professor pode receber várias avaliações e que uma avaliação vai para um Professor, e a mesma lógica segue para Oferecimento, que também tem cardinalidade 1:N.