

1. Passo 1

Curso(código, nome, carga_horária, total_vagas)

Regra(id_regra, descrição)

Infraestrutura(id_infraestrutura, nome)

Sala(id_sala, capacidade)

DepartamentoAcadêmico(código, nome)

UnidadeEscolar(código, cidade, estado, país, bloco, Localidade_Especifica)

Disciplina(código, num_aulas_semanais, material_recomendado)

Turma(id_turma, capacidade)

Avaliação(id_avaliação, texto, didática)

Mensagem(id_mensagem, timestamp, texto)

AvisoGeral(id_aviso, texto, timestamp)

Usuário(nome, sobrenome, telefone, dataNasc, sexo, senha, e-mail)

Bolsa(id_bolsa, valor_bolsa, descrição, instituição)

Desconto(id_desconto, valor, motivo)

2. Passo 2

Não tem nenhum atributo fraco!

a. 2a)

- Abstração de Agregação Ministrada

Caso 2(a), eu consigo deixar todos os atributos no agregado!

Ministra(id_sala, id_turma, dia, hora_inicio, hora_fim)

☐ id_sala -> FK -> Sala(id_sala)

☐ id_turma -> FK -> Turma(id_turma)

- Abstração Oferecimento

Caso 2(a), eu consigo deixar todos os atributos no agregado!

Oferecimento(código, código_oferecimento, id_turma, periodo)

☐ código -> FK -> Disciplina(código)

☐ id_turma -> FK -> Turma(id_turma)

- Abstração Matrícula

Caso 2(a), eu consigo deixar todos os atributos no agregado!

Matricula(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, status, data_limite, valor_matricula)

☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nome, sobrenome, telefone)

☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)

b. 2b)

- Abstração Curso

Como eu tenho poucos CEEs e nenhum atributo específico, irei para o caso 1G, mais especificamente G4, pois é mutuamente exclusivo.

Curso(código, nome, carga_horária, total_vagas, nível)

- Abstração Usuário

Como usuário é uma abstração, ou seja, usuário em si sozinho não existe, irei utilizar o NE, mais especificamente G7, já que é mutuamente exclusivo.

Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, dataNasc, sexo, senha, e-mail, área_especialização, titulação)

Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, dataNasc, sexo, senha, e-mail)

FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, dataNasc, sexo, senha, e-mail)

3. Passo 3

Não existe nenhum relacionamento 1:1.

4. Passo 4 (1:N)

- tem por padrão

Curso(código, nome, carga_horária, total_vagas, nível, id_sala)

☐ id_sala -> FK -> Sala(id_sala)

- pertence a

Curso(código, nome, carga_horária, total_vagas, nível, id_sala, codigoDepartamento)

☐ id_sala -> FK -> Sala(id_sala)

☐ codigoDepartamento -> FK -> DepartamentoAcadêmico(código)

- chefe

DepartamentoAcadêmico(código, nome, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- Todas as relações “associado”

Curso(código, nome, carga_horária, total_vagas, nível, id_sala, codigoDepartamento, códigoUnidadeEscolar)

☐ id_sala -> FK -> Sala(id_sala)

☐ codigoDepartamento -> FK -> DepartamentoAcadêmico(código)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Disciplina(código, num_aulas_semanais, material_recomendado, códigoUnidadeEscolar)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, dataNasc, sexo, senha, e-mail, área_especialização, titulação, códigoUnidadeEscolar)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, dataNasc, sexo, senha, e-mail, códigoUnidadeEscolar)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

- ministra

Disciplina(código, num_aulas_semanais, material_recomendado, códigoUnidadeEscolar, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

☐ nomeProfessor, sobrenomeProfessor, telefoneProfessor -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- faz avaliação

Avaliação(id_avaliação, texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)

- envia Aviso Geral

AvisoGeral(id_aviso, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- Usuário envia mensagem

Como usuário é só uma abstração, vamos ter a possibilidade de ser um dos seus 3 CEEs:

Mensagem(id_mensagem, timestamp, texto, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- Professor recebe avaliação

Avaliação(id_avaliação, texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- Oferecimento recebe avaliação

Avaliação(id_avaliação, texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, código_oferecimento)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)

- Funcionário Administrativo envia Aviso geral

AvisoGeral(id_aviso, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

5. Passo 5 (N:M)

- Curso pode ter Regra

ContemRegra(id_regra, códigoCurso)

☐ id_regra -> FK -> Regra(id_regra)
☐ códigoCurso -> FK -> Curso(código)

- Curso pode precisar de infraestrutura

PrecisaInfraestrutura(id_infraestrutura, códigoCurso)

☐ id_infraestrutura -> FK -> Infraestrutura(id_infraestrutura)
☐ códigoCurso -> FK -> Curso(código)

- Curso composto por Disciplina

CursoTemDisciplina(códigoCurso, códigoDisciplina)

☐ códigoCurso -> FK -> Curso(código)
☐ códigoDisciplina -> FK -> Disciplina(código)

- Curso é pré-requisito de curso

CursoPreRequisitoCurso(códigoCurso, códigoCursoPreRequisito)

☐ códigoCurso -> FK -> Curso(código)
☐ códigoCursoPreRequisito -> FK -> Curso(código)

- Disciplina é pré-requisito de curso

DisciplinaPreRequisitoCurso(códigoCurso, códigoDisciplina)

☐ códigoCurso -> FK -> Curso(código)
☐ códigoDisciplina -> FK -> Disciplina(código)

- Turma recebe Aviso Geral

TurmaRecebeAviso(id_turma, id_aviso)

- ☐ id_turma -> FK -> Turma(id_turma)
- ☐ id_aviso -> FK -> AvisoGeral(id_aviso)

- Turma recebe Mensagem

TurmaRecebeMensagem(id_turma, id_mensagem)

- ☐ id_turma -> FK -> Turma(id_turma)
- ☐ id_mensagem -> FK -> AvisoGeral(id_mensagem)

- Usuario recebe Mensagem

Como usuário é abstrato, devemos criar a relação para cada um dos seus CEEs:

AlunoRecebeMensagem(nomeAluno, sobrenomeAluno, telefoneAluno, id_mensagem)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

ProfessorRecebeMensagem(nomeProfessor, sobrenomeProfessor, telefoneProfessor, id_mensagem)

- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

FuncionarioRecebeMensagem(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, id_mensagem)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> Funcionario(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

- Matrícula pode possuir Bolsa

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_bolsa)

- ☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)
- ☐ id_bolsa -> FK -> Bolsa(id_bolsa)

- Matrícula pode possuir Desconto

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_desconto)

- ☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)
- ☐ id_desconto -> FK -> Desconto(id_desconto)

6. Passo 6

Não tem nenhum relacionamento de ordem maior do que 2.

7. Atributos compostos e multivalorados

- Notas em Matrícula

Matrícula originalmente tem notas, que é multivalorado. Como eu não sei de antemão quantas notas vão ter, irei criar uma nova relação:

Notas(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, nota)

- ☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)

- Endereço em usuário

Endereço é composto, mas como já sabemos que endereço é rua, cidade, estado e como Usuário é abstrato:

Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, dataNasc, sexo, senha, e-mail, códigoUnidadeEscolar, rua, cidade, estado)

- ☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, dataNasc, sexo, senha, e-mail, área_especialização, titulação, códigoUnidadeEscolar, rua, cidade, estado)

- ☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, dataNasc, sexo, senha, e-mail, rua, cidade, estado)

8. Mapeamento finalizado com as chaves candidatas

Regra(id_regra, descrição)

Infraestrutura(id_infraestrutura, nome)

Sala(id_sala, capacidade)

UnidadeEscolar(código, cidade, estado, país, bloco, Localidade_Especificas)

Turma(id_turma, capacidade)

Bolsa(id_bolsa, valor_bolsa, descrição, instituição)

Desconto(id_desconto, valor, motivo)

Ministra(id_sala, id_turma, dia, hora_inicio, hora_fim)

☐ id_sala -> FK -> Sala(id_sala)

☐ id_turma -> FK -> Turma(id_turma)

Oferecimento(código_oferecimento, código, id_turma, periodo)

☐ código -> FK -> Disciplina(código)

☐ id_turma -> FK -> Turma(id_turma)

Matricula(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, status, data_limite, valor_matricula)

☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nome, sobrenome, telefone)

☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)

DepartamentoAcadêmico(código, nome, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

Curso(código, nome, carga_horária, total_vagas, nível, id_sala, codigoDepartamento, códigoUnidadeEscolar)

☐ id_sala -> FK -> Sala(id_sala)

☐ codigoDepartamento -> FK -> DepartamentoAcadêmico(código)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Disciplina(código, num_aulas_semanais, material_recomendado, códigoUnidadeEscolar, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

- ☐ nomeProfessor, sobrenomeProfessor, telefoneProfessor -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

AvisoGeral(id_avisos, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

Mensagem(id_mensagem, timestamp, texto, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

Avaliação(id_avaliação, texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, código_oferecimento)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)

AvisoGeral(id_avisos, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

ContemRegra(id_regra, códigoCurso)

- ☐ id_regra -> FK -> Regra(id_regra)
- ☐ códigoCurso -> FK -> Curso(código)

PrecisaInfraestrutura(id_infraestrutura, códigoCurso)

- ☐ id_infraestrutura -> FK -> Infraestrutura(id_infraestrutura)
- ☐ códigoCurso -> FK -> Curso(código)

CursoTemDisciplina(códigoCurso, códigoDisciplina)

- ☐ códigoCurso -> FK -> Curso(código)
- ☐ códigoDisciplina -> FK -> Disciplina(código)

CursoPreRequisitoCurso(códigoCurso, códigoCursoPreRequisito)

- ☐ códigoCurso -> FK -> Curso(código)
- ☐ códigoCursoPreRequisito -> FK -> Curso(código)

DisciplinaPreRequisitoCurso(códigoCurso, códigoDisciplina)

- ☐ códigoCurso -> FK -> Curso(código)
- ☐ códigoDisciplina -> FK -> Disciplina(código)

TurmaRecebeAviso(id_turma, id_aviso)

- ☐ id_turma -> FK -> Turma(id_turma)
- ☐ id_aviso -> FK -> AvisoGeral(id_aviso)

TurmaRecebeMensagem(id_turma, id_mensagem)

- ☐ id_turma -> FK -> Turma(id_turma)
- ☐ id_mensagem -> FK -> AvisoGeral(id_mensagem)

AlunoRecebeMensagem(nomeAluno, sobrenomeAluno, telefoneAluno, id_mensagem)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

ProfessorRecebeMensagem(nomeProfessor, sobrenomeProfessor, telefoneProfessor, id_mensagem)

- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

FuncionarioRecebeMensagem(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, id_mensagem)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> Funcionario(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)
- ☐ id_mensagem -> FK -> Mensagem(id_mensagem)

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_bolsa)

- ☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)
- ☐ id_bolsa -> FK -> Bolsa(id_bolsa)

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_desconto)

- ☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)
- ☐ id_desconto -> FK -> Desconto(id_desconto)

Notas(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, nota)

☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)

Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, dataNasc, sexo, senha, e-mail, códigoUnidadeEscolar, rua, cidade, estado)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, dataNasc, sexo, senha, e-mail, área_especialização, titulação, códigoUnidadeEscolar, rua, cidade, estado)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, dataNasc, sexo, senha, e-mail, rua, cidade, estado)