

PUC-RIO

RESPUC

Solução em Banco de Dados



PUC
RIO

Disciplina: INF1383 - Banco de Dados

Professor: Sérgio Lifschitz

Grupo 3: Fernando Homem da Costa - 1211971

Juliana Dana Zilberberg - 1410899

Pedro Moraes - 1321234

Thiago Nicácio - 1411188

Turma: 3WA

21 de Outubro de 2016

<u>Introdução</u>	1
<u>Empresa e a Motivação</u>	1
<u>Objetivo do Projeto</u>	1
<u>Diagrama E-R (Modelagem Conceitual)</u>	2
<u>Restrições de Integridade Semântica</u>	2
<u>Esquema Lógico Relacional</u>	3
<u>DDL Esquema Lógico</u>	3
<u>Avaliação Sobre a Qualidade do Projeto Lógico</u>	4
<u>Consultas em Álgebra Relacional</u>	4
<u>Consultas em SQL</u>	4
<u>Visão, Função, Procedimento e Trigger</u>	5
<u>Índices</u>	5

Introdução

Empresa e a Motivação

A RESPUC (Rede de Empreendimentos Sociais da PUC-Rio), tem como objetivo reunir os projetos criados, desenvolvidos e mantidos na universidade, ou por membros de sua comunidade.

Considerando que as atividades desenvolvidas em empreendimentos sociais podem potencializar a reflexão e o senso crítico e a responsabilidade social, incorporando valores morais e princípios éticos e, portanto, proporcionar a formação de um profissional cidadão, a Vice-Reitoria para Assuntos Comunitários (VRC) instituiu o Programa RESPUC Ação Comunitária, tendo como finalidade precípua o desenvolvimento pessoal, ético, cívico e social, por meio do serviço voluntário em prol da inclusão social.

Localizada na PUC-Rio, Rio de Janeiro, sua missão é trocar experiências e conhecimentos, potencializar o impacto social gerado pelos empreendimentos, estabelecendo um canal de diálogo com a sociedade e incentivar a participação social, junto à solidariedade comunitária.

Atualmente, os membros desta rede utilizam uma planilha em Excel para controlar os formulários enviados e realizar o planejamento da RESPUC. Porém, enfrentam dificuldades em armazenar o volume de informações que se atualizam constantemente, além de não aplicar várias regras relevantes. Por isso, desejamos propor uma solução em banco de dados, ou seja, um sistema para administrar e acompanhar os alunos que se inscrevem para ser voluntários das instituições.

Os alunos da PUC-Rio interessados no programa da RESPUC devem se cadastrar indicando a instituição de interesse, para participar do processo seletivo. Vale lembrar que o processo seletivo está aberto apenas para alunos matriculados da PUC-Rio.

O cadastro consiste no aluno preencher um formulário em que forneça suas informações básicas como rua, complemento, bairro, cidade, UF, CEP, telefone fixo, celular, além de nome, CPF, número de identidade, órgão emissor da identidade, matrícula, gênero, período atual do curso, data de nascimento, idioma que domina, email e o nome do curso em que está atualmente matriculado.

Ainda sobre o formulário, o aluno obrigatoriamente deve estar matriculado em um curso de graduação, pós ou curso de extensão da PUC-Rio e sobre esse curso deseja-se saber o nome do curso, nome do coordenador responsável, departamento do curso, número total de alunos inscritos no curso. Um aluno tem que estar matriculado somente em um curso e em um curso pode ter nenhum ou muitos alunos.

Todo aluno que quiser se inscrever em um projeto gerenciado pela RESPUC precisa

realizar uma inscrição. Ele deve indicar em qual projeto está interessado em se inscrever e informar como conheceu esse projeto, seus dias e horários disponíveis para se dedicar. É necessário saber também o status da inscrição e a data da inscrição do aluno. O projeto aceita a inscrição de todos os interessados até que haja uma capacidade máxima das instalações físicas do local.

Um aluno pode não estar vinculado em nenhum ou quantos projetos quiser. Cada projeto pode se relacionar com nenhum ou vários alunos.

Sobre o projeto, é necessário saber o nome, data de início, quantidade de pessoas que estão atuando, as vagas disponíveis, informações sobre em que consiste o projeto, a área em que o projeto atua, como por exemplo: saúde, educação, esporte; o público alvo, que representa à quem está destinado, e o status, para que o voluntário saiba o andamento do processo seletivo do projeto. Também é possível saber a data de final de um projeto.

Um projeto tem que pertencer a somente uma instituição, ou seja, não pode estar vinculado em nenhuma ou quantas instituições quiser. Cada instituição pode estar vinculada com nenhum ou vários projetos.

A instituição é composta pelo Nome Fantasia, Razão social (representa o nome comercial), ano de fundação, site, vínculo com a PUC, quantidade de membros da instituição e um responsável de relações públicas, email responsável de relações públicas. Além disso, contém suas informações como rua, complemento, bairro, cidade, UF, CEP, telefone fixo, celular e e-mail.

Sobre o gerente, é necessário saber o nome, CPF e telefone para contato.

Um gerente participa de somente uma instituição, ou seja, não pode estar vinculado em nenhum ou quantas instituições quiser. Cada instituição pode se relacionar com nenhum ou vários gerentes.

Um gerente pode atuar em um ou mais projetos, ou seja, ele tem que estar gerenciando pelo menos um ou quantos projetos quiser. Cada projeto tem que se relacionar com pelo menos um ou vários gerentes.

Dessa forma, as informações fornecidas pela RESPUC juntamente com seus formulários, possibilitaram ao grupo a fazer um estudo detalhado sobre o processo de inscrição e seus participantes. Assim, o nosso grupo propõe essa solução ao RESPUC.

Objetivo do Projeto

Nosso objetivo é a implementação de um banco de dados que irá fazer com que essas duas gestões, Aluno e Projeto, não percam informações cruciais para consultas futuras.

Com isso, irá facilitar o dia-a-dia dos membros, permitindo a rápida e precisa obtenção de informações, relatório mais detalhados e um melhor gerenciamento da empresa.

Para conseguir ter um bom controle dos dados, é necessário um breve planejamento, que será apresentado a seguir.

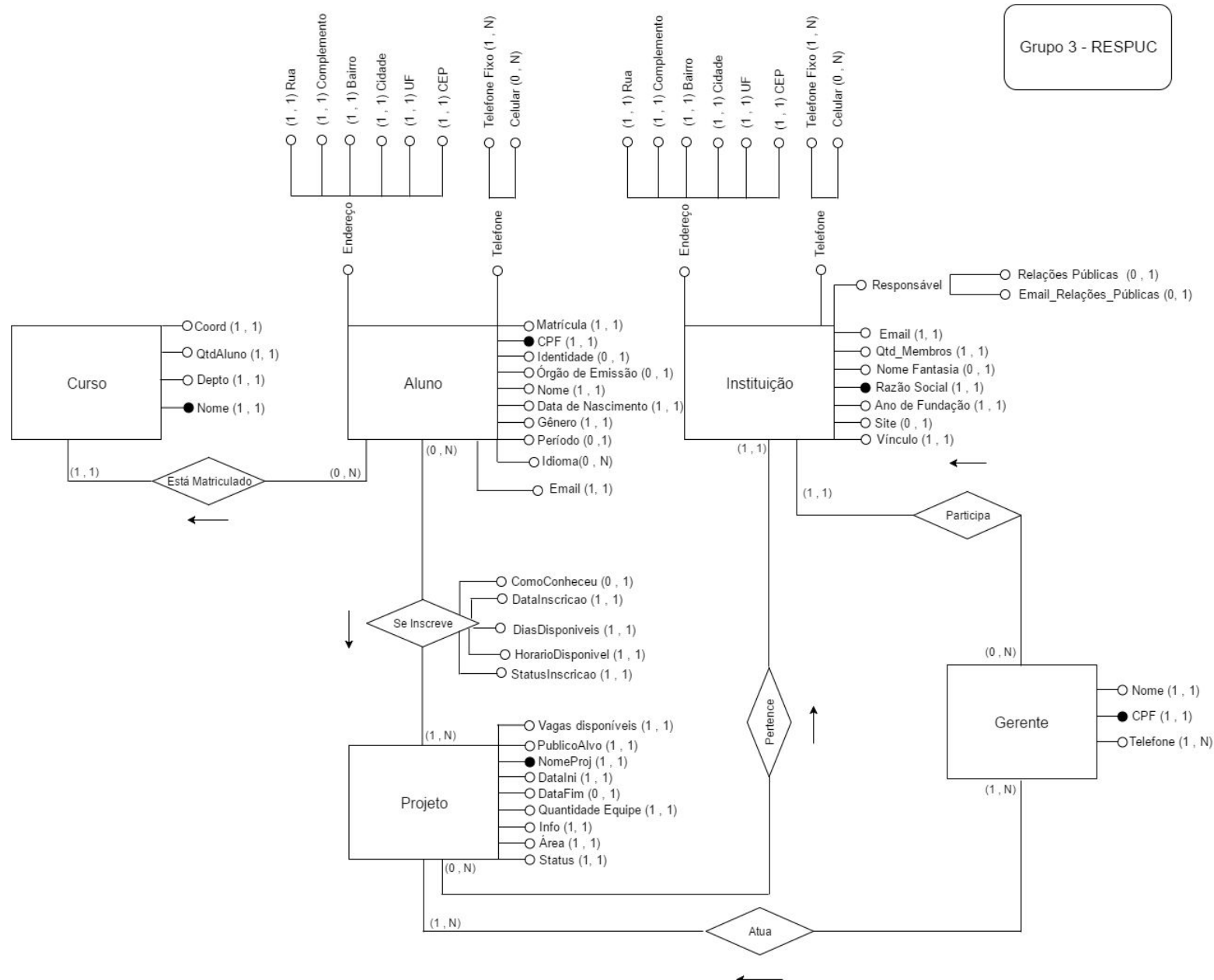
Diagrama E-R (Modelagem Conceitual)

Primeiramente, como parte de todo projeto de banco de dados, foi necessário a sua descrição de forma independente de implementação em um Sistema de Gerenciamento de Banco de Dados (SGBD).

Por isso, o modelo conceitual abaixo, mostra quais dados vão aparecer no banco de dados, porém não demonstra como estes dados serão armazenados e manipulados a nível SGBD.

Para retratar o modelo conceitual, utilizamos a abordagem entidade-relacionamento(ER), onde descrevemos através de um diagrama entidade-relacionamento(DER).

Grupo 3 - RESPUC



Descrição de Entidades, Relacionamentos e Atributos

1. Entidade 'Aluno': Representa os alunos da Pontifícia Universidade Católica do Rio de Janeiro que irão se inscrever nos projetos disponibilizados pela Rede de Empreendimentos Sociais da PUC-Rio (RESPUC). Além disso, possui seus próprios atributos, como:

Atributos:

- a. 'Nome (1 , 1)': Nome completo do aluno representado por uma cadeia de caracteres (String) de no máximo 50 caracteres;
- b. 'CPF (1 , 1)': Cadastro de pessoas físicas válido e representado por uma combinação de 14 caracteres seguindo o seguinte modelo: xxx.xxx.xxx-xx, que também o identifica na inscrição;
- c. 'Identidade (0 , 1)'(Registro Geral): O número de identificação da carteira de identidade é representado por uma combinação de 15 caracteres, seguindo o seguinte modelo: xxxxxxxxxxxxx.
- d. 'Órgão de Emissão (0 , 1)' (Órgão Expedidor): Órgão emissor responsável pela emissão da carteira de identidade, representado por uma cadeia de caracteres (String) de no máximo 5 caracteres. Lembrando que consideramos o par identidade, número de identificação junto com órgão de emissão, dessa forma impossibilitando que haja mais de um número por órgão de emissão para cada pessoa ;
- e. 'Matrícula (1 , 1)': Matrícula do aluno da PUC-Rio formada por 7 números(pertencentes ao conjunto dos naturais);
- f. 'Gênero (1 , 1)': É a identidade social relacionada ao sexo de um indivíduo, por exemplo: masculino, feminino e outro. É representado por uma cadeia de caracteres(String) de no máximo 20 caracteres ;
- g. 'Período (0 , 1)': Número que representa o período atual do aluno na PUC-Rio formada por 1 número(pertencente ao conjunto dos naturais);
- h. 'Data de Nascimento (1 , 1)': É uma cadeia de caracteres(String) com combinação de no máximo 10 caracteres, seguindo o seguinte modelo: dd/mm/aaaa;
- i. Idioma(0 , N): Representa os idiomas que o aluno domina para ser capaz de se comunicar, é uma cadeia de caracteres(String) com combinação de no máximo 20 caracteres. Ex:Inglês, francês, espanhol e etc..

j. Endereço é um atributo composto:

- i. 'Rua (1 , 1)': Nome da rua. Representado por uma cadeia de caracteres (String) de no máximo 60 caracteres;
- ii. 'Complemento (1 , 1)': Complemento do endereço, por exemplo: bloco 2 apartamento 111, cobertura, loja A. É representado por uma cadeia de caracteres (String) de no máximo 15 caracteres;
- iii. 'Bairro (1 , 1)': Nome do bairro. Representado por uma cadeia de caracteres (String) de no máximo 20 caracteres;
- iv. 'Cidade (1 , 1)': Nome da cidade. Representado por uma cadeia de caracteres (String) de no máximo 20 caracteres;
- v. 'UF (1 , 1)': Sigla da unidade federativa. Representado por uma cadeia de caracteres (String) de no máximo 2 caracteres;
- vi. 'CEP (1 , 1)': Código de Endereçamento Postal. Cadeia de caracteres (String) de no máximo 9 caracteres, seguindo o seguinte formato: xxxxx-xxx;

k. Telefone é um atributo composto:

- i. 'Telefone fixo' (1 , N): Combinação de até no máximo 14 caracteres seguindo os seguintes modelos: (**)*****-**** ou (**)****-****. Sendo obrigatório informar pelo menos um telefone fixo, mesmo sendo aluno, curso ou instituição;
 - ii. 'Celular' (0 , N): Combinação de até no máximo 14 caracteres seguindo os seguintes modelos: (**)*****-**** ou (**)****-****. Sendo não obrigatório informar pelo menos um celular, mesmo sendo aluno, curso ou instituição;
- l. 'Email (1 , 1)': Endereço de e-mail do coordenador do aluno representado por uma cadeia de caracteres (String) de no máximo 60 caracteres. Respeitando a seguinte formatação, é obrigatório que o símbolo '@' esteja presente uma única vez na formatação e após esse, tenha pelo menos um símbolo '.', por exemplo, nomedousuario@dominio.com.

2. Entidade 'Curso': representa os cursos disponibilizados pela PUC-Rio, podendo ser estes serem de graduação, pós-graduação, extensão e etc. Além disso, possui seus próprios atributos, como:

Atributos:

- a. 'Nome (1 , 1)' : Nome do curso oferecido pela PUC-Rio, representado por uma cadeia de caracteres (String) de no máximo 20 caracteres;
 - b. 'Coord (1 , 1)' : Nome completo do coordenador do curso representado por uma cadeia de caracteres (String) de no máximo 50 caracteres;
 - c. 'Depto (1 , 1)': Nome do departamento responsável pelo curso representado por uma cadeia de caracteres (String) de no máximo 30 caracteres;
 - d. 'QtdAluno (1 , 1)': Número total de alunos inscritos no curso, representado por um número(pertencente ao conjunto dos números naturais);
3. Entidade 'Instituição': Representa todas as instituições que tem parcerias com a Rede de Empreendimentos Sociais da PUC-Rio (RESPUC). Além disso, possui seus próprios atributos, como:

Atributos:

- a. 'Nome Fantasia (1 , 1)': Nome Fantasia da Instituição representada por uma cadeia de caracteres (String) de no máximo 100 caracteres;
- b. 'Razão Social (1 , 1)': Nome comercial da Instituição representada por uma cadeia de caracteres (String) de no máximo 150 caracteres;
- c. 'Ano de fundação (1 , 1)': Ano em que foi fundada a instituição, representado por uma cadeia de caracteres(String) de no máximo 4 caracteres;
- d. 'Site (0 , 1)': Endereço da instituição dentro da web, ou seja, endereço do seu website. É uma combinação de caracteres de no máximo 60 caracteres, seguindo os seguintes formatos: www.nomedoweb site .xxxx ou www.nomedoweb site .xxx.xx;
- e. 'Vínculo (1 , 1)' : com a PUC-Rio: relação que a Instituição possui com a PUC-Rio representado por uma cadeia de caracteres (String) de no máximo 150 caracteres;
- f. 'Qtd_Membros (1 , 1)' de membros na Instituição: É número de pessoas que fazem parte da instituição, representado por um número(pertencente ao conjunto dos naturais);
- g. Responsável: é um atributo composto que representa o Relações Públicas da Instituição e o email dele:

- i. 'Responsável Relações Públicas (0 , 1)': Nome completo do responsável por contatos com a sociedade representado por uma cadeia de caracteres (String) de no máximo xxx caracteres;
 - ii. 'E-mail do Relações Públicas (0 , 1)': Endereço de e-mail do responsável pelo contato com a sociedade representado por uma cadeia de caracteres (String) de no máximo xxx caracteres. Respeitando a seguinte formatação, é obrigatório que o símbolo '@' esteja presente uma única vez na formatação e após esse, tenha pelo menos um símbolo '.', por exemplo, nomedousuario@dominio.com.
- h. Endereço é um atributo composto:
- i. 'Rua (1 , 1)': Nome da rua. Representado por uma cadeia de caracteres (String) de no máximo 60 caracteres;
 - ii. 'Complemento (1 , 1)': Complemento do endereço, por exemplo: bloco 2 apartamento 111, cobertura, loja A. É representado por uma cadeia de caracteres (String) de no máximo 15 caracteres;
 - iii. 'Bairro (1 , 1)': Nome do bairro. Representado por uma cadeia de caracteres (String) de no máximo 20 caracteres;
 - iv. 'Cidade (1 , 1)': Nome da cidade. Representado por uma cadeia de caracteres (String) de no máximo 20 caracteres;
 - v. 'UF (1 , 1)': Sigla da unidade federativa. Representado por uma cadeia de caracteres (String) de no máximo 2 caracteres;
 - vi. 'CEP (1 , 1)': Código de Endereçamento Postal. Cadeia de caracteres(String) de no máximo 9 caracteres, seguindo o seguinte formato: xxxxx-xxx;
- i. Telefone é um atributo composto:
- i. 'Telefone fixo' (1 , N): Combinação de até no máximo 14 caracteres seguindo os seguintes modelos: (**)*****-**** ou (**)****-****. Sendo obrigatório informar pelo menos um telefone fixo, mesmo sendo aluno, curso ou instituição;
 - ii. 'Celular' (0 , N): Combinação de até no máximo 14 caracteres seguindo os seguintes modelos: (**)*****-**** ou (**)****-****. Sendo não obrigatório informar pelo menos um celular, mesmo sendo aluno, curso ou instituição;

- j. 'Email (1 , 1)': Endereço de e-mail da instituição representado por uma cadeia de caracteres (String) de no máximo 60 caracteres. Respeitando a seguinte formatação, é obrigatório que o símbolo '@' esteja presente uma única vez na formatação e após esse, tenha pelo menos um símbolo '.', por exemplo, nomedousuario@dominio.com.
4. Entidade 'Projeto': Representa os projetos em que os alunos podem se inscrever, oferecidos pelas instituições através da sua parceria com a Rede de Empreendimentos Sociais da PUC-Rio (RESPUC), também são os projetos em que os gerentes atuam. Além de estabelecer um relacionamento com a entidade 'Aluno','Gerente' e 'Instituição', 'Projeto'.

Atributos:

- a. 'Vagas Disponíveis (1 , 1)': Número de vagas disponíveis para voluntários naquele projeto, sendo representado por um numeric(número tem que pertencer ao conjunto dos naturais);
- b. 'NomeProj (1 , 1)': Nome do projeto representado por uma cadeia de caracteres (String) de no máximo 50 caracteres;
- c. 'Info (0 , 1)': É um texto com no máximo 450 caracteres, contendo informações sobre em que consiste o projeto;
- d. 'Área (1 , 1)': É uma cadeia de caracteres(String) de no máximo 15 caracteres que representa a área em que o projeto aceita voluntários, por exemplo: saúde, esporte, educação e etc.;
- e. 'PúblicoAlvo (1 , 1)': É uma cadeia de caracteres(String) de no máximo 10 caracteres que representa público alvo do projeto, por exemplo: criança, idosos, adultos e etc.;
- f. 'DataIni (1 , 1)': É uma cadeia de caracteres(String) que representa a data de início de um projeto, com combinação de no máximo 10 caracteres, seguindo o seguinte modelo: dd/mm/aaaa;
- g. 'DataFim (0 , 1)': É uma cadeia de caracteres(String) que representa a data de fim de um projeto, com combinação de no máximo 10 caracteres, seguindo o seguinte modelo: dd/mm/aaaa. Lembrando que a data de fim tem que ser posterior a data de início;
- h. 'Quantidade Equipe (1 , 1)': É número de pessoas que fazem parte do projeto, representado por um número(pertencente ao conjunto dos naturais);
- i. 'Status (1 , 1)': Representa o estado atual do projeto, por exemplo: iniciado, em andamento, finalizado e etc. Representado por uma cadeia de

caracteres(String) de no máximo 20 caracteres.

5. Entidade 'Gerente': Representa o responsável pelo projeto que as instituições oferecerem

Atributos:

- a. 'Nome (1 , 1)': Nome completo do gerente representado por uma cadeia de caracteres (String) de no máximo 50 caracteres;
- b. 'CPF (1 , 1)': Cadastro de pessoas físicas válido e representado por uma combinação de 14 caracteres seguindo o seguinte modelo: xxx.xxx.xxx-xx, que também o identifica na inscrição;
- c. 'Telefone(1 , N)': Combinação de até no máximo 14 caracteres seguindo os seguintes modelos: (**)*****-**** ou (**)****-****. Sendo obrigatório informar pelo menos um telefone fixo;

6. Relacionamento 'Inscrição': Representa o relacionamento entre aluno e projeto, todo aluno que quiser se inscrever em um projeto precisa de uma inscrição.

Atributos:

- a. 'DataInscrição (1 , 1)': (Autoexplicativo) É uma cadeia de caracteres(String) que representa a data de inscrição de um aluno, com combinação de no máximo 10 caracteres, seguindo o seguinte modelo: dd/mm/aaaa;
- b. 'DiasDisponíveis (1 , 1)' disponível: É uma cadeia de caracteres(String) que representa os dias da semana que um aluno pode atuar em projeto, com combinação de no máximo 50 caracteres. Deve estar entre as seguintes opções: 'Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta', 'Sábado' e 'Domingo';
- c. 'HorárioDisponível (1 , 1)' durante o dia disponível: uma cadeia de caracteres(String) que representa os dias da semana que um aluno pode atuar em projeto, com combinação de no máximo 10 caracteres. Deve estar entre as seguintes opções: 'Manhã', 'Tarde' e 'Noite';
- d. 'ComoConheceu (0 , 1)' o programa: É um texto com no máximo 450 caracteres, contendo informações sobre como conheceu o projeto;
- e. 'StatusInscrição (1 , 1)': Representa o estado atual da inscrição do aluno, por exemplo: Aceita, Em Análise, Recusada. Representado por uma cadeia de caracteres(String) de no máximo 20 caracteres.

Nesta parte, discutiremos sobre os relacionamentos entre as entidades retratadas no modelo conceitual. Detalharemos os relacionamentos e suas cardinalidades.

Relacionamento: **Aluno se inscreve em um Projeto**

Um aluno pode se inscrever em 1 ou mais projetos, ou seja, pode não estar vinculado em nenhum ou quantos projetos quiser. Cada projeto pode se relacionar com 0 ou vários alunos.

Relacionamento: **Aluno está matriculado em um Curso**

Um aluno tem que estar matriculado somente em um curso, ou seja, não pode estar vinculado em mais de um curso ou não estar matriculado em nenhum curso. Cada Curso pode se relacionar com 0 ou vários alunos.

Relacionamento: **Projeto pertence a uma Instituição**

Um projeto tem que pertencer a somente uma instituição, ou seja, não pode estar vinculado em nenhum ou quantas instituições quiser. Cada instituição pode se relacionar com 0 ou vários projetos.

Relacionamento: **Gerente atua em um Projeto**

Um gerente atua em 1 ou mais projetos, ou seja, todo gerente tem que estar atuando em pelo menos um projeto ou vários projetos. Cada projeto tem que se relacionar com pelo menos um ou vários gerentes.

Relacionamento: **Gerente participa de uma Instituição**

Um gerente participa de somente uma instituição, ou seja, não pode estar vinculado em nenhum ou quantas instituições quiser. Cada instituição pode se relacionar com 0 ou vários gerentes.

Restrições de Integridade Semântica

Foram definidas as seguintes Regras de Negócio:

1. Toda instância de 'DataFim' de um projeto tem que ter um valor posterior ao de 'DataInicio' ou nula caso não tenha uma data determinada.
2. Consideramos o par identidade, número de identificação junto com órgão de emissão, dessa forma impossibilitando que haja mais de um número por órgão de emissão para cada pessoa;

3. Toda instância de endereço de e-mail deve respeitar a seguinte formatação, é obrigatório que o símbolo '@' esteja presente uma única vez na formatação, por exemplo, nomedousuario@dominio.com
4. Todo aluno tem um limite por lei específico para de carga horária semanal de horas trabalhadas, limitando a quantidade de projetos que ele pode participar.
5. Toda instância de projeto haverá uma quantidade de membros em uma equipe, essa quantidade não pode ser nula e se tiver apenas um membro, esse membro deverá ser o gerente.
6. Toda instância de inscrição haverá um status da inscrição e este deverá ter somente três opções: aceita, em análise, reprovada.
7. Toda instância de projeto haverá um status da inscrição e este deverá ter somente três opções: iniciado, em análise, terminado.

Esquema Lógico Relacional

Objetivo deste esquema lógico é retratar como a estrutura de dados de um banco de dados é visto pelo usuário de SGBD. A seguir são apresentadas as tabelas geradas a partir do modelo conceitual proposto anteriormente. As chaves primárias estão sublinhadas e as chaves estrangeiras estão em itálico. Para cada chave estrangeira foi explicitada a relação à qual ela faz referência.

1. Aluno(CPE, *Curso*, Data de Nascimento, Gênero, Identidade, Idioma, Nome, Matrícula, *OrgaoDeEmissao*, Período, Rua, Complemento, Bairro, Cidade, UF, CEP, Telefonefixo, Celular, Email)
2. Curso(Nome, Coord, Depto, QdAluno)
3. Inscrição(DiasDisponiveis, HorarioDisponivel, ComoConheceu, StatusInscricao, DataInscricao, *NomeDoProjeto*, *CPFDoAluno*)
4. Projeto(NomeProj, AreaVoluntario, PublicoAlvo, DataIni, DataFim, QuantidadeEquipe, Status, VagasDisponiveis, Info, RazaoSocialInsituicao)
5. Instituição(NomeFantasia, RazaoSocial, AnoDeFundacao, Site, Vínculo, Qtd_Membros, RelaçõesPúblicas, EmailRelaçõesPúblicas, Rua, Complemento, Bairro, Cidade, UF, CEP, Telefonefixo, Celular, Email)

6. Gerente(Nome,CPE,Telefone)

7. Participação(NomeProj, CPFDoGerente)

DDL Esquema Lógico

A seguir estão os códigos em SQL referentes à criação das tabelas especificadas na seção anterior.

```
CREATE TABLE aluno(  
  nome VARCHAR(50) Not Null,  
  cpf CHAR(14) Not Null,  
  identidade CHAR(14),  
  orgaodeemissao CHAR(5) ,  
  matricula integer() Not Null,  
  genero CHAR(20) Not Null,  
  periodo integer() Not Null,  
  data_de_nascimento DATE Not Null,  
  idioma CHAR(20) Not Null,  
  nomecurso CHAR(20) Not Null,  
  rua VARCHAR(60) Not Null,  
  complemento CHAR(15) Not Null,  
  bairro VARCHAR(20) Not Null,  
  cidade VARCHAR(20) Not Null,  
  uf VARCHAR(2) Not Null,  
  cep CHAR(9) Not Null,  
  telefonefixo CHAR(14) Not Null,  
  celular CHAR(14),  
  email CHAR(60) Not Null,
```

```
  CONSTRAINT chk_email CHECK( like (email = '%@%')),  
  CONSTRAINT pk_aluno PRIMARY KEY (cpf),  
  CONSTRAINT matricula_uni UNIQUE (matricula)  
);
```

```
CREATE TABLE curso(  
  nome VARCHAR(20) Not Null,  
  coord VARCHAR(50) Not Null,  
  depto VARCHAR(30) Not Null,  
  qtdaluno NUMERIC(2) Not Null,
```

```
  CONSTRAINT pk_nome PRIMARY KEY (nome),  
  CONSTRAINT coordenador_uni UNIQUE (coordenador)  
);
```

```

CREATE TABLE inscricao(
diasdisponiveis VARCHAR(50) Not Null,
horariodisponivel CHAR(5) Not Null,
comoconheceu CHAR(450) Not Null,
statusinscricao CHAR(20) Not Null,
datainscricao date Not Null,
nomeproj VARCHAR(50) Not Null,
cpfaluno VARCHAR(14) Not Null,

CONSTRAINT chk_statusinscricao CHECK(statusinscricao in ('aceita', 'em analise',
'reprovada')),
CONSTRAINT pk_inscricao PRIMARY KEY (nomeproj,cpfaluno),
CONSTRAINT fk_nomeproj FOREIGN KEY nomeproj references projeto(nomeproj),
CONSTRAINT fk_cpfaluno FOREIGN KEY cpfaluno references aluno(cpfl)
);

```

```

CREATE TABLE projeto(
nomeproj VARCHAR(50) Not Null,
areavoluntario VARCHAR(15) Not Null,
publicoalvo VARCHAR(10) Not Null,
dataini date Not Null,
datafim date,
quantidadeequipe integer Not Null,
status VARCHAR(20) Not Null,
vagasdisponiveis integer Not Null,
info VARCHAR(450) Not Null,
razaosocialinstituicao VARCHAR(150) Not Null,

```

```

CONSTRAINT chk_status CHECK (status in ('iniciado', 'em andamento', 'terminado')),
CONSTRAINT chk_quantidadeequipe CHECK ((quantidadeequipe > 0)),
CONSTRAINT pk_nomeproj PRIMARY KEY (nomeproj),
CONSTRAINT fk_razaosocialinstituicao FOREIGN KEY razaosocialinstituicao references
instituicao(razaosocial)
);

```

```

CREATE TABLE instituicao(
nomefantasia VARCHAR(100) Not Null,
razaosocial VARCHAR(150) Not Null,
anodefundacao CHAR(4) Not Null,
site VARCHAR(60),
vinculo VARCHAR(150) Not Null,
qtd_membros integer Not Null,
relacoespublicas VARCHAR(50),
emailrelacoespublicas VARCHAR(60),
rua VARCHAR(60) Not Null,

```

complemento CHAR(15) Not Null,
bairro VARCHAR(20) Not Null,
cidade VARCHAR(20) Not Null,
uf VARCHAR(2) Not Null,
cep CHAR(9) Not Null,
telefonefixo CHAR(14) Not Null,
celular CHAR(14),
email CHAR(60) Not Null,

CONSTRAINT chk_email CHECK(like (email = '%@%')),

CONSTRAINT chk_quantidadeequipe CHECK((qtd_membros > 0)),
CONSTRAINT pk_razaosocial PRIMARY KEY (razaosocial)
);

CREATE TABLE gerente(
nome VARCHAR(50) Not Null,
cpf CHAR(14) Not Null,
telefone CHAR(14) Not Null,

CONSTRAINT pk_gerente PRIMARY KEY (cpf)
CONSTRAINT telefone _uni UNIQUE (telefone)
);

CREATE TABLE participacao(
nomeproj VARCHAR(50) Not Null,
cpfgerente CHAR(14) Not Null,

CONSTRAINT pk_participacao PRIMARY KEY (nomeproj , cpfgerente),
CONSTRAINT fk_nomeproj FOREIGN KEY nomeproj references projeto(nomeproj),
CONSTRAINT fk_cpfgerentel FOREIGN KEY cpfgerente references gerente(cpf));

Consultas em Álgebra Relacional e em SQL

1. Quais os nomes e cursos dos alunos que se inscreveram em 2016?

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}, \text{Curso}} (\sigma_{\text{DataInscrição} = 2016} (\sigma_{\text{CPF} = \text{CPFDoAluno}} (\text{Aluno} \times \text{Inscrição})))$

SQL: SELECT a.nome, a.curso
FROM inscrição as i, aluno as a
WHERE i.cpfdoaluno = a.cpf
AND year(i.dataInscricao) = 2016

2. Quais os nomes e cursos dos alunos que se inscreveram para projeto 'X' ou projeto 'Y' em 2016?

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}, \text{Curso}} (\sigma_{\text{NomeProj} = X \wedge \text{NomeProj} = Y \wedge \text{DataInscrição} = 2016} (\sigma_{\text{CPF} = \text{CPFAluno}}$

(Aluno x Inscrição)))

SQL: SELECT a.nome, a.curso
FROM aluno a, inscrição i, projeto px, projeto py
WHERE (px.nomeproj = 'X'
OR py.nomeproj = 'Y')
AND i.cpf = a.cpf
AND year(i.dataInscricao) = 2016

3. Quais os nomes e cursos dos alunos que tiveram suas inscrições aprovadas em 2016?

AR: Resp $\leftarrow \pi_{\text{Nome, Curso}} (\sigma_{\text{Status de Inscrição} = \text{Aceita}} (\sigma_{\text{CPF} = \text{CPFDoAluno}} (\text{Aluno} \times \text{Inscrição})))$

SQL: SELECT nome, curso
FROM Aluno a, Inscrição i
WHERE i.cpf = a.cpf
AND i.StatusInscricao = 'Aceita'
AND year(i.DataInscricao) = 2016

4. Listar os nomes dos projetos que começaram em 2013 e tem mais de 5 membros em sua equipe.

AR: Resp $\leftarrow \pi_{\text{NomeProj}} (\sigma_{\text{DataIni} = 2013 \wedge \text{QuantidadeEquipe} > 5} (\text{Projeto}))$

SQL: SELECT NomeDoProjeto
FROM Projeto
WHERE QuantidadeEquipe > 5
AND year(DataIni) = 2013

5. Listar o nome fantasia das instituições que foram fundadas em 1990.

AR: Resp $\leftarrow \pi_{\text{NomeFantasia}} (\sigma_{\text{AnoDeFundação} = 1990} (\text{Instituição}))$

SQL: SELECT NomeFantasia
FROM instituição
WHERE AnoDeFundacao = '1990'

6. Quais os nomes e cursos dos alunos que estão aguardando a aprovação para entrar em determinado projeto?

AR: Resp $\leftarrow \pi_{\text{Nome, Curso}} (\sigma_{\text{StatusInscrição} = \text{'Em Analise'}} (\sigma_{\text{CPF} = \text{CPFDoAluno}} (\text{Aluno} \times \text{Inscrição})))$

SQL: SELECT nome, curso
FROM Aluno, Inscrição
WHERE StatusInscricao = 'Em Analise'

7. Listar nome e matricula dos alunos que falam os idiomas inglês e espanhol, e tenham disponibilidade na parte da tarde.

AR: $\text{Resp} \leftarrow \pi_{\text{Nome, matricula}}(\sigma_{\text{cpfdoaluno} = \text{cpf} \wedge \text{horariodisponivel} = \text{'tarde'} \wedge \text{idioma} = \text{'ingles'} \vee \text{idioma} = \text{'espanhol'}}(\text{Aluno} \times \text{Inscrição}))$

SQL: SELECT a.Nome, a.matricula
FROM aluno a, inscrição i
WHERE a.idioma IN ('inglês', 'espanhol')
AND i.horariodisponivel = tarde
AND i.cpfdoaluno = a.cpf

8. Listar CPF e nome de todos os alunos que estejam entre o segundo e o quarto período.

AR: $\text{Resp} \leftarrow \pi_{\text{CPF, Nome}}(\sigma_{\text{Periodo} > 2 \wedge \text{Periodo} < 4}(\text{Alunos}))$

SQL: SELECT CPF, Nome
FROM alunos
WHERE periodo BETWEEN 2 and 4

9. Exibir o número de alunos que tiveram a inscrição aceita.

AR: $\text{Resp} \leftarrow \mathbf{Gcount}(\text{StatusInscrição})(\sigma_{\text{StatusInscrição} = \text{'Aceita'}}(\text{Inscrição}))$

SQL: SELECT count(StatusInscricao)
FROM Inscrição
WHERE StatusInscricao = 'Aceita'

10. Listar os nomes dos projetos e o nome fantasia a qual instituição ele pertence.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj, NomeFantasia}}(\sigma_{\text{RazaoSocialInstituicao} = \text{RazaoSocial}}(\text{Projeto} \times \text{Instituição}))$

SQL: SELECT p.NomeProj, i.NomeFantasia
FROM Projeto p, Instituição i
WHERE p.RazaoSocialDaInstituicao = i.RazaoSocial

11. Listar todos os projetos em que o público alvo seja 'X'.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}}(\sigma_{\text{PublicoAlvo} = \text{'X'}}(\text{Projeto}))$

SQL: SELECT NomeProj
FROM Projeto
WHERE PublicoAlvo= 'X'

12. Listar o nome dos alunos que foram aprovados em projetos e tem domínio no idioma francês.

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}} (\sigma_{\text{StatusInscrição} = \text{'Aceita'} \wedge \text{CPF} = \text{CPFDoAluno} \wedge \text{Idioma} = \text{'Francês'}} (\text{Aluno} \times \text{Inscrição}))$

SQL: SELECT nome
FROM Aluno, Inscrição
WHERE StatusInscricao = 'Aceita'
AND Aluno.CPF = Inscrição.CPFDoAluno
AND Idioma = 'Francês'

13. Listar os nomes dos projetos e os nomes de seus respectivos gerentes/responsáveis.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}, \text{Nome}} (\sigma_{\text{NomeProj} = \text{NomeProj}} (\text{Projeto} \times \sigma_{\text{CPF} = \text{CPFDoGerente}} (\text{Gerente} \times \text{Participação})))$

SQL: SELECT p.NomeProj, g.Nome
FROM Projeto p, Gerente g, Participação pa
WHERE g.cpf = pa.CPFDoGerente
AND pa.NomeProj = p.NomeProj

14. Listar quais os projetos que ainda não tem aluno inscrito.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}} (\sigma_{\text{QuantidadeEquipe} = 1} (\text{Projeto}))$

SQL: SELECT p.NomeProj
FROM Projeto p
WHERE p.NomeProj NOT IN
(SELECT i.NomeProj
FROM Inscrição i)

15. Listar os projetos que não possuem mais vagas para alunos.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}} (\sigma_{\text{VagasDisponiveis} = 0} (\text{Projeto}))$

SQL: SELECT NomeProj
FROM Projeto
WHERE VagasDisponiveis = 0

16. Em quais projetos (nome) os alunos do curso de informática estão atuando?

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}} (\sigma_{\text{StatusInscrição} = \text{'Aceita'}} (\text{Projeto} \bowtie (\sigma_{\text{CPF} = \text{CPFDoAluno}} (\text{Aluno} \times \text{Inscrição}))))$

SQL: SELECT p.NomeProj
FROM Projeto p, Aluno a, Inscrição i

```
WHERE a.Departamento = 'Informática'
AND i.StatusInscricao = 'Aceita'
AND p.DataFim IS NULL
AND a.CPF = i.CPFDoAluno
AND i.NomeProj = p.NomeProj
```

17. Listar os nomes dos projetos de todas as instituições em que o gerente de nome João participa.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}, \text{Nome}, \text{RazaoSocial}} ((\text{Projeto} \bowtie \text{Instituicao})_{(\text{razaoSocialInstituicao} = \text{razaoSocial})} \bowtie (\text{Participação} \bowtie \text{Gerente})_{(\text{Nome} = \text{Joao})})$

```
SQL: SELECT p.NomeProj
      FROM Participação p, Gerente g
      WHERE g.nome like '%João%'
      AND p.CPFDogerente = g.CPF
```

18. Listar todos os gerentes que tenham Maria no nome.

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}}(\text{Gerente}) - \pi_{\text{Nome}}(\sigma_{\text{Nome} \neq \text{'Maria'}}(\text{Gerente}))$

```
SQL: SELECT *
      FROM gerente
      WHERE nome LIKE '%Maria%'
```

19. Exibir o nome do projeto que tenha a menor quantidade de participantes.

AR: $\text{Resp} \leftarrow \text{TTNomeProj}(\text{ProjetoGmin}(\text{QuantidadeEquipe}))$

```
SQL: SELECT NomeProj
      FROM Projeto
      WHERE NomeProj IN
            (SELECT min(p.QuantidadeEquipe),
              NomeProj
              FROM Projeto p)
```

20. Exibir o nome do projeto que tenha a maior quantidade de vagas disponíveis.

AR: $\text{Resp} \leftarrow \Pi_{\text{NomeProj}} (\text{ProjetoG}_{\max}(\text{VagasDisponiveis}))$

```
SQL: SELECT NomeProj
      FROM Projeto
      WHERE NomeProj IN
                                (SELECT max(p.VagasDisponiveis),
                                NomeProj
                                FROM Projeto p)
```

21. Exibir a média da quantidade de pessoas de cada projeto.

AR: $\text{Resp} \leftarrow \pi_{\text{QuantidadeEquipe}} (\text{ProjetoGavq}(\text{QuantidadeEquipe}))$

SQL: SELECT avg(QuantidadeEquipe)

FROM Projeto

22. Exibir a soma das vagas disponíveis dos projetos.

AR: $\text{Resp} \leftarrow \text{Projeto} \mathbf{G} \text{sum}(\text{VagasDisponiveis})$

SQL: SELECT sum(VagasDisponiveis)
FROM Projeto

23. Listar todos os nomes dos alunos exceto os de Design.

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}}(\text{Aluno}) - \pi_{\text{Nome}}(\sigma_{\text{Curso} = \text{'Design'}}(\text{Aluno}))$

SQL: (SELECT Nome
FROM Aluno)
EXCEPT
(SELECT Nome
FROM Aluno
WHERE Curso = 'Design')

24. Exibir o nome e data de inscrição de todos os alunos que realizaram inscrição para algum projeto, ordenado pela data de inscrição.

AR: $\text{Resp} \leftarrow \tau_{\text{datainscrição asc}}(\pi_{\text{Nome, datainscrição}}(\sigma_{\text{cpf} = \text{cpfdosaluno}}(\text{Aluno} \times \text{Inscrição})))$

SQL: SELECT a.Nome, i.DataInscricao
FROM Aluno a, Inscrição i
WHERE a.CPF = i.CPFDaAluno
ORDER BY DataInscricao

25. Exibir todos os alunos (nome) e instituições (nome fantasia) que tem seu endereço no bairro do centro.

AR: $\text{Resp} \leftarrow \pi_{\text{Nome}}(\sigma_{\text{Bairro} = \text{'Centro'}}(\text{Aluno})) \cup \pi_{\text{NomeFantasia}}(\sigma_{\text{Bairro} = \text{'Centro'}}(\text{Instituição}))$

SQL: (SELECT Nome
FROM Aluno
WHERE Bairro = 'Centro')
UNION
(SELECT NomeFantasia
FROM Instituição
WHERE Bairro = 'Centro')

26. Listar Nome e site de alguma instituição localizada na Gávea.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeFantasia, Site}}(\sigma_{\text{Bairro} = \text{'Gávea'}}(\text{Instituição}))$

SQL: SELECT NomeFantasia, Site
FROM Instituição
WHERE Bairro = 'Gávea'

27. Exibir os nomes dos projetos que tenham o maior número de vagas disponíveis.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}} (\text{Projeto} \bowtie_{\text{Gmax}(\text{VagasDisponiveis})})$

SQL: SELECT NomeDoProjeto
FROM Projeto
WHERE VagasDisponiveis >= ALL
(SELECT p.VagasDisponiveis
FROM Projeto as p)

28. Quais os CPF's de todos os alunos que se inscreveram para participar de algum projeto?

AR: $\text{Resp} \leftarrow \pi_{\text{CPF}} (\text{Aluno} \bowtie \text{Inscrição})(\text{CPF} = \text{CPFDoAluno})$

SQL: SELECT aluno.cpf,
FROM aluno
INNER JOIN inscricao
ON aluno.cpf = inscricao.cpfaluno

29. Listar todos os nomes fantasia das instituições e os seus respectivos projetos (nome) caso o tenham.

AR: $\text{Resp} \leftarrow \pi_{\text{NomeProj}, \text{NomeFantasia}} (\text{Projeto} \bowtie \text{Instituição})(\text{RazaoSocialInstituicao} = \text{RazaoSocial})$

SQL: SELECT projeto.nomeproj, instituicao.nomefantasia
FROM projeto
RIGHT OUTER JOIN instituicao
ON projeto.razaosocialinstituicao = instituicao.razaocial