



Alunos: Bruno Correia - 217120004,
Felipe Daniel França - 222215141,
João Lucas Melo - 219216216,
Rafael Correa Nagy - 216220012

Grupo: D

Data: 23/05/2023

Projeto de Banco de Dados: Sistema de vagas de estágio para alunos da UFBA

MER:

De acordo com a especificação discutida pelo grupo, submetida na etapa anterior do trabalho, entendemos que o Modelo Entidade Relacionamento seria composto por três entidades (Instituição Contratante, Vaga de Estágio e Aluno) e dois relacionamentos ('fornece' e 'seCandidataEm').

Cada atributo das entidades seguem a discussão da especificação, com exceção de:

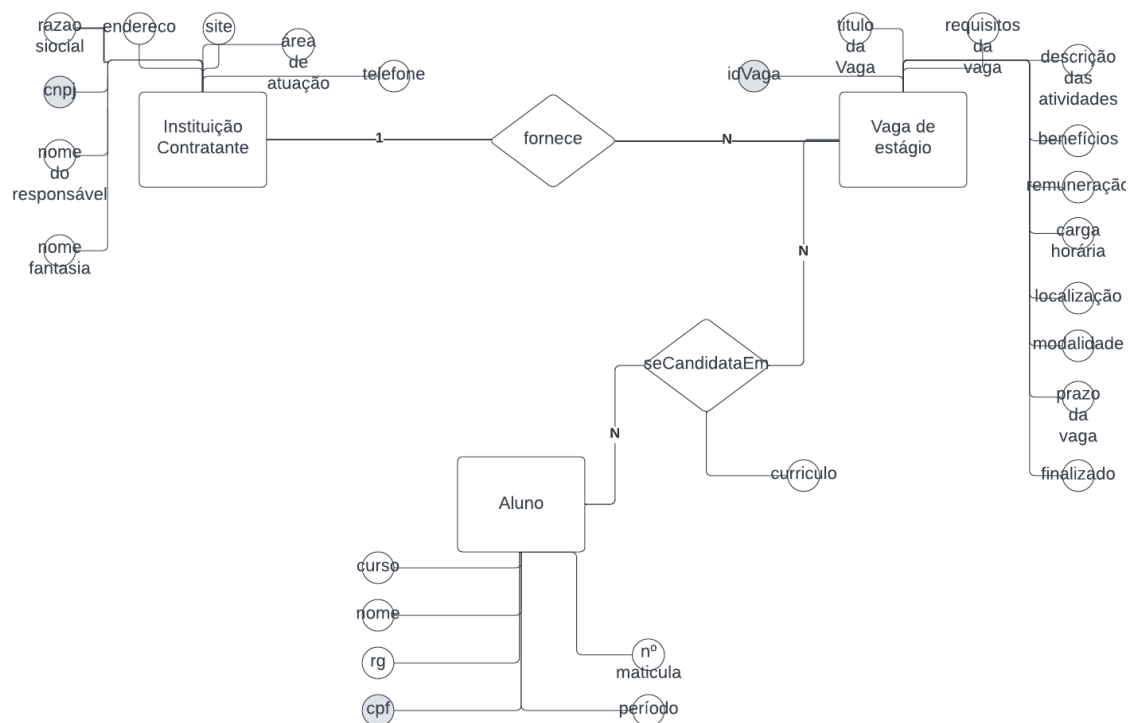
1. Instituição Contratante:
 - . Razão Social, Nome Fantasia - Achamos importante dividir o atributo 'nome' de uma instituição em dois atributos pelos quais ela pode ser referida.
 - . Telefone - Um contato direto para o representante da instituição que é responsável pela oferta da vaga.
 - . Nome do Responsável - Nome do representante da instituição que é responsável pela oferta da vaga.
2. Vaga de Estágio:
 - . Remuneração - Referente ao que é ofertado como salário na vaga.
 - . Modalidade - Referente à modalidade presencial (p), remoto (r) ou híbrido (h).
 - . Localização - Referente ao local de trabalho.
 - . Prazo da Vaga - Data de expiração da vaga.
 - . Finalizado - Sinaliza a disponibilidade da vaga (valor obtido através da consulta da data de expiração da vaga e o dia atual).

No relacionamento 'seCandidataEm', nós associamos um atributo 'currículo' referente ao link onde o currículo de um aluno para uma determinada vaga está localizado no servidor. Dessa forma, é possível que um aluno disponibilize diferentes currículos para diferentes vagas.

Uma Instituição Contratante pode fornecer diversas Vagas de Estágio, mas cada uma dessas vagas está associada a, exclusivamente, uma Instituição Contratante. Dessa forma, a cardinalidade do relacionamento 'fornece' é definida por 1:N.

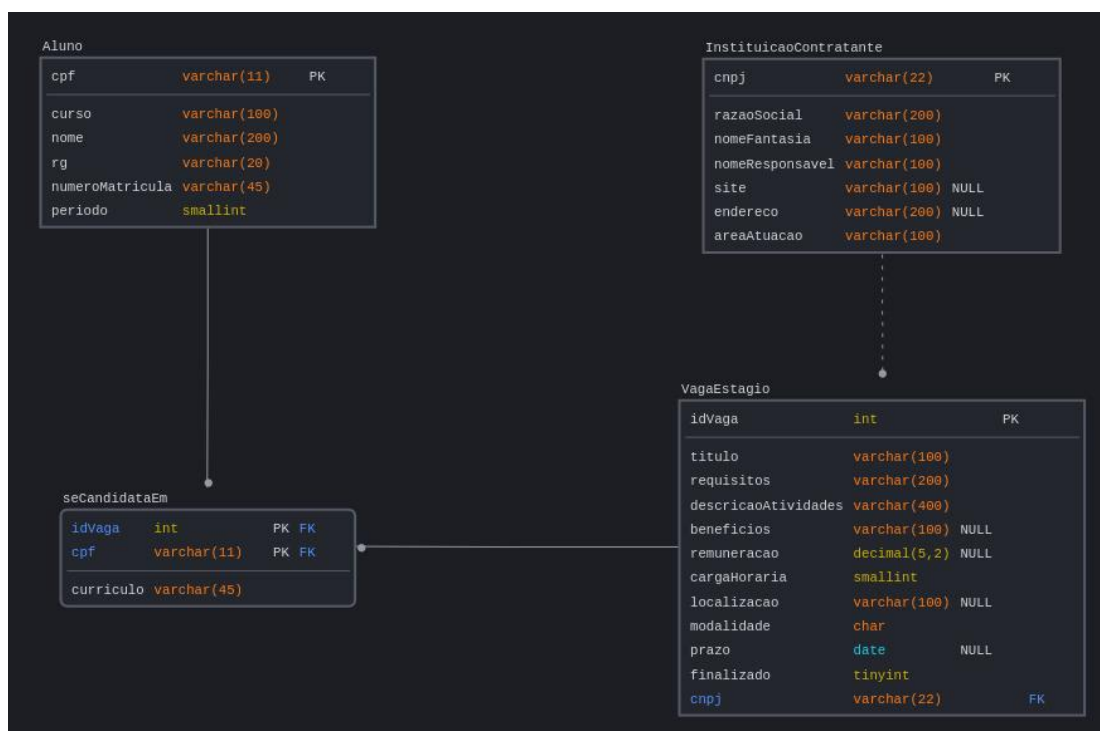
Vários Alunos podem se candidatar a várias Vagas de Estágio, onde cada vaga pode receber candidatura de diversos Alunos. Dessa forma, a cardinalidade do relacionamento

‘seCandidataEm’ é definida por M:N.



MAPEAMENTO:

De acordo com o MER anterior, discutindo sobre os tipos de dados que cada atributo exige, desenvolvemos a seguinte modelagem:



SQL:

Uma vez obtida a modelagem, desenvolvemos os seguintes códigos SQL para a criação das tabelas da nossa aplicação:

```
////DDL////
```

```
CREATE TABLE Aluno
```

```
(  
  cpf          varchar(11) NOT NULL ,  
  curso        varchar(100) NOT NULL ,  
  nome         varchar(200) NOT NULL ,  
  rg           varchar(20) NOT NULL ,  
  numeroMatricula  varchar(45) NOT NULL ,  
  periodo      smallint NOT NULL ,
```

```
PRIMARY KEY (cpf)
```

```
);
```

```
CREATE TABLE InstituicaoContratante
```

```
(  
  cnpj          varchar(22) NOT NULL ,  
  razaoSocial   varchar(200) NOT NULL ,  
  nomeFantasia  varchar(100) NOT NULL ,  
  nomeResponsavel  varchar(100) NOT NULL ,  
  site          varchar(100) ,  
  endereco     varchar(200) ,  
  areaAtuacao  varchar(100) NOT NULL ,
```

```
PRIMARY KEY (cnpj)
```

```
);
```

```
CREATE TABLE VagaEstagio
```

```
(  
  idVaga       int NOT NULL ,  
  titulo       varchar(100) NOT NULL ,  
  requisitos   varchar(200) NOT NULL ,  
  descricaoAtividades  varchar(400) NOT NULL ,  
  beneficios   varchar(100) ,  
  remuneracao  decimal(5,2) ,  
  cargaHoraria smallint NOT NULL ,  
  localizacao  varchar(100) ,  
  modalidade   char NOT NULL ,  
  prazo        date ,  
  finalizado   tinyint NOT NULL ,  
  cnpj         varchar(22) NOT NULL ,
```

```
PRIMARY KEY (idVaga)
```

```
);
```

```
CREATE TABLE seCandidataEm
```

```
(
  idVaga int NOT NULL ,
  cpf varchar(11) NOT NULL ,
  curriculo varchar(45) NOT NULL ,
```

```
PRIMARY KEY (idVaga, cpf)
);
```

```
ALTER TABLE VagaEstagio
ADD FOREIGN KEY fkInstituicao (cnpj)
REFERENCES InstituicaoContratante(cnpj);
```

```
ALTER TABLE seCandidataEm
ADD FOREIGN KEY fkVaga (idVaga)
REFERENCES VagaEstagio(idVaga);
```

```
ALTER TABLE seCandidataEm
ADD FOREIGN KEY fkAluno (cpf)
REFERENCES Aluno(cpf);
```

```
////DML////
```

```
INSERT INTO Aluno VALUES ('000000000000', 'ciencia da computacao', 'bruno',
'222222222222', '22222222', 5);
```

```
INSERT INTO Aluno VALUES ('111111111111', 'ciencia da computacao', 'renata',
'11111111111111111111', '7348717348910', 6);
```

```
INSERT INTO Aluno VALUES ('222222222222', 'ciencia da computacao', 'paulo',
'333333333333', '333333333333', 7);
```

```
INSERT INTO Aluno VALUES ('333333333333', 'ciencia da computacao', 'lucas',
'11111111110000000000', '1111111111', 6);
```

```
INSERT INTO InstituicaoContratante VALUES ('88126741000164', 'cubos', 'cubos', 'renato',
", ", 'computacao');
```

```
INSERT INTO InstituicaoContratante VALUES ('1111111111111111', 'ufba', 'ufba', 'claudia', ", ",
'computacao');
```

```
INSERT INTO VagaEstagio VALUES (222, 'dados', 'fazer coisas', 'coisas a fazer', ", null, 8, ",
'r', null, 0, '88126741000164');
```

```
INSERT INTO VagaEstagio VALUES (111, 'algebra', 'fazer coisas', 'coisas a fazer', ", null, 6,
", 'r', '2000-05-12', 0, '1111111111111111');
```

```
INSERT INTO seCandidataEm VALUES (111, '000000000000', 'curr1');
```

```
INSERT INTO seCandidataEm VALUES (111, '111111111111', 'curr2');
```

```
INSERT INTO seCandidataEm VALUES (222, '000000000000', 'curr1');
```

```
INSERT INTO seCandidataEm VALUES (222, '111111111111', 'curr2');
```

```
/* Obter título e descrição de vagas criadas por um contratante com determinado CNPJ */
```

```
SELECT titulo, descricaoAtividades
```

```
FROM VagaEstagio
```

```
WHERE cnpj = '88126741000164';
```

```
/* Obter todos os candidatos cadastrados numa vaga com determinado ID (ex. 222) */  
SELECT Aluno.*  
FROM Aluno  
JOIN seCandidataEm ON Aluno.cpf = seCandidataEm.cpf  
WHERE seCandidataEm.idVaga = 222;
```

```
/* Obter todas as vagas que têm prazo final até 31 de maio de 2023 */  
SELECT *  
FROM VagaEstagio  
WHERE prazo <= '2023-05-31';
```

```
/* Recuperar todos os alunos que se cadastraram em vagas da empresa Cubos */  
SELECT Aluno.*  
FROM Aluno  
JOIN seCandidataEm ON Aluno.cpf = seCandidataEm.cpf  
JOIN VagaEstagio ON seCandidataEm.idVaga = VagaEstagio.idVaga  
JOIN InstituicaoContratante ON VagaEstagio.cnpj = InstituicaoContratante.cnpj  
WHERE InstituicaoContratante.nomeFantasia = 'Cubos';
```

```
/* Selecionar apenas vagas que tenham "dados" no título */  
SELECT *  
FROM VagaEstagio  
WHERE titulo LIKE "%dados%";
```

```
/* Selecionar todas as vagas que tenham remuneração acima de 1500 reais e ordenar por prazo  
final */  
SELECT *  
FROM VagaEstagio  
WHERE remuneracao > 1500  
ORDER BY prazo;
```

```
/* Recuperar todas as vagas nas quais um determinado candidato se inscreveu e selecionar  
apenas as que não possuem benefícios */  
SELECT VagaEstagio.*  
FROM VagaEstagio  
JOIN seCandidataEm ON VagaEstagio.idVaga = seCandidataEm.idVaga  
LEFT JOIN Aluno ON seCandidataEm.cpf = Aluno.cpf  
WHERE Aluno.cpf = 'cpf_desejado'  
AND VagaEstagio.beneficios IS NULL OR VagaEstagio.beneficios = '';
```

SQL3:

```
CREATE TYPE instNome AS(  
`nomeFantasia`      varchar(100) NOT NULL,  
`razaoSocial`       varchar(200) NOT NULL  
);
```

```
CREATE TABLE `InstituicaoContratante`  
(
```

```
`cnpj`          varchar(22) NOT NULL ,
`nome`          instNome NOT NULL,
`nomeResponsavel` varchar(100) NOT NULL ,
`site`          varchar(100) ,
`endereco`      varchar(200) ,
`areaAtuacao`   varchar(100) NOT NULL ,
```

```
PRIMARY KEY (`cnpj`)
);
```

```
CREATE TABLE VagaEstagio
(
  idVaga        int NOT NULL ,
  titulo        varchar(100) NOT NULL ,
  requisitos    varchar(200) NOT NULL ,
  descricaoAtividades varchar(400) NOT NULL ,
  beneficios    varchar(20)[] ,
  remuneracao   decimal(5,2) ,
  cargaHoraria  smallint NOT NULL ,
  localizacao   varchar(100) ,
  modalidade    char NOT NULL ,
  prazo         date ,
  finalizado    tinyint NOT NULL ,
  cnpj          varchar(22) NOT NULL ,
```

```
PRIMARY KEY (idVaga)
);
```

```
CREATE TABLE seCandidataEm
(
  idVaga  int NOT NULL ,
  cpf     varchar(11) NOT NULL ,
  curriculo varchar(45) NOT NULL ,
```

```
PRIMARY KEY (idVaga, cpf)
);
```

```
ALTER TABLE VagaEstagio
ADD FOREIGN KEY fkInstituicao (cnpj)
REFERENCES InstituicaoContratante(cnpj);
```

```
ALTER TABLE seCandidataEm
ADD FOREIGN KEY fkVaga (idVaga)
REFERENCES VagaEstagio(idVaga);
```

```
ALTER TABLE seCandidataEm
ADD FOREIGN KEY fkAluno (cpf)
REFERENCES Aluno(cpf);
```

ÁLGEBRA RELACIONAL:

Dado que a álgebra relacional é mais abstrata do que queries em SQL de fato, optamos por incluir queries de álgebra relacional que possam representar cenários reais que eventualmente possam surgir no sistema. Esses cenários englobam, majoritariamente, operações nas quais o usuário escolhe filtrar as vagas disponíveis por diferentes campos, como por exemplo o nome da empresa, a remuneração proposta, etc.

Foram incluídas também algumas queries que o back-end possivelmente venha a fazer para fornecer ao front-end, como por exemplo, uma empresa que deseja saber quais candidatos se candidataram na sua vaga de id 123.

```
/* Obter título e descrição de vagas criadas por um contratante com determinado cnpj */  
Resultado ←  $\pi$  titulo, descricaoAtividades ( $\sigma$  cnpj = '88126741000164' (VagaEstagio  $\bowtie$  InstituicaoContratante))
```

```
/* Obter todos os candidatos cadastrados numa vaga com determinado id (ex. 222) */  
Resultado ←  $\pi$  nome, email, curso ( Aluno  $\bowtie$  seCandidataEm  $\bowtie$   $\sigma$ (idVaga = 222 (VagaEstagio)))
```

```
/* Obter todas as vagas que tem prazo final até 31 de maio de 2023 */  
Resultado ←  $\sigma$  prazo <= '2023-05-31' (VagaEstagio)
```

```
/* Recuperar todos os alunos que se cadastraram em vagas da empresa Cubos */  
Resultado ← Aluno  $\bowtie$  seCandidataEm  $\bowtie$  (VagaEstagio  $\bowtie$   $\sigma$  nomeFantasia = 'Cubos' (InstituicaoContratante))
```

```
/* Selecionar apenas vagas que tenham "dados" no título */  
Resultado ←  $\sigma$  titulo LIKE '%dados%' (VagaEstagio)
```

```
/* Selecionar todas as vagas que tenham remuneração acima de 1500 reais e ordenar por prazo final */  
Resultado ←  $\sigma$  remuneracao > 1500 (VagaEstagio)  $\bowtie$   $\pi$  * ( $\sigma$  (true) ( $\rho$  prazo (VagaEstagio)))
```

```
/* Recuperar todas as vagas na qual um determinado candidato se inscreveu e selecionar apenas as que não possuem benefícios */  
CandidatoEspecifico ←  $\rho$  cpf ← '999999999999' (Aluno)  
CandidatoVagas ← CandidatoEspecifico  $\bowtie$  seCandidataEm  
VagasSemBeneficios ←  $\sigma$  beneficios = "  $\vee$  beneficios IS NULL (VagaEstagio)  
Resultado ← CandidatoVagas  $\bowtie$  VagasSemBeneficios
```