

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA BANCO DE DADOS I

MARIA LETHÍCIA CARTAXO LIMA ABRANTES
ISLEIMAR DE SOUZA OLIVEIRA

GESTÃO ESCOLAR

CAJAZEIRAS
2018

MARIA LETHÍCIA CARTAXO LIMA ABRANTES
ISLEIMAR DE SOUZA OLIVEIRA

GESTÃO ESCOLAR

Trabalho desenvolvido como requisito parcial para obtenção da aprovação na disciplina Banco de Dados I, no curso de Análise e Desenvolvimento de Sistemas, no Instituto Federal de Educação, Ciência e Tecnologia, *Campus* Cajazeiras.

Prof. Dr. Fábio Gomes de Andrade

Cajazeiras

2018

Sumário

1	Introdução.....	4
2	Modelo Conceitual.....	5
2.1	Levantamento dos requisitos.....	5
2.2	Diagrama de Entidade-Relacionamento.....	8
2.3	Dicionário conceitual de dados.....	9
3	Modelo lógico.....	16
3.1	Mapeamento Entidade Relacionamento.....	16
3.2	Dicionário Lógico de Dados.....	16
4	Modelo físico.....	28
4.1	Script SQL.....	28
4.1.1	Criando as relações.....	28
4.1.2	Criando Índices.....	39
4.1.3	Visões.....	40
4.1.4	Funções agregadas.....	41
4.1.5	Gatilhos.....	43
4.1.6	Povoando as relações.....	49
4.1.7	Consultas.....	61

1 Introdução

Considerando que o período útil da semana pode ser bastante exaustivo graças ao trabalho e demais afazeres, nem todos os responsáveis têm a disponibilidade de ir à instituição escolar para verificar como os seus filhos estão a comportar-se, verificar suas notas ou também saber quando acontecem as avaliações e eventos. Dessa maneira, visando achar formas de melhorar a comunicação e promover flexibilidade aos pais, foi consultada a instituição Colégio Nossa Senhora de Lourdes de modo a colher requisitos que auxiliem na chegada ao objetivo desejado – o diálogo é escasso, o que eventualmente rende problemas de perda ou inconsistência de informações, além de causar problemas caso haja algo que a instituição não conseguiu fazer alcançar os responsáveis.

2 Modelo Conceitual

2.1 Levantamento dos requisitos

Uma instituição escolar possui várias pessoas vinculadas a ela, gerando a necessidade de armazenar as seguintes informações: Nome, Data de Nascimento, Endereço, Telefones, E-mails, Sexo, Identidade e CPF. De modo mais específico, devemos armazenar para os responsáveis, que respondem pelos alunos, o endereço e os telefones do trabalho, além do que já foi supracitado, e o parentesco que possui com o aluno. Este, por sua vez, deve possuir o nome do pai, da mãe, o rendimento escolar e a matrícula, além de estar matriculado em apenas uma turma. Para o professor, funcionário da empresa, deve-se constar uma matrícula que o distinga e sua especialização; ele deve ministrar pelo menos uma disciplina.

As formas de organização interestelar se dão por meio de turmas, que devem armazenar o número da sala, série e um código de identificação. De modo a organizar o andamento letivo da turma, ela organiza-se em trimestres, que devem constar a data final previsto, a data inicial e o qual número que lhe singulariza (primeiro, segundo ou terceiro). Ademais, deve-se conter também informações sobre as disciplinas, estas armazenando o nome, a carga horária e um código único; cada disciplina deve possuir um diário, de modo a permitir o professor armazenar a frequência. O diário também possuirá um código e um “atualizado”, que dirá se o professor está com o diário em dia ou não.

Havendo disciplinas, elas deverão ser ministradas por um professor. Ele está encarregado, igualmente, de agendar as avaliações de sua respectiva disciplina. Para a avaliação agendada, o tipo, o conteúdo e um número (primeira, segunda ou terceira, se for o caso) tem que estar armazenados, permitindo uma futura checagem. Esta avaliação será feita por um aluno.

A avaliação faz parte de um maior escopo, que são os acontecimentos escolares. Eles são pautados por um calendário anual e devem, de modo geral, armazenar a data, contendo dia e mês, em que eles acontecerão, o ano do acontecimento, esta informação tendo que ser única para evitar que dois acontecimentos semelhantes não se confundam, a hora de início, a de término e o local onde o acontecimento será realizado. Desses acontecimentos, alguns cairão na pauta de evento, pois a escola proporciona várias celebrações, como o dia dos pais,

das mães, etc. Para estes deverá constar uma descrição e um código de identificação. Quem registra eventos é o administrador, funcionário da empresa, que, além das informações comuns a todos, também precisa ter uma chave singular de acesso e informar qual o cargo que ocupa dentro da instituição.

Para efetuar a comunicação entre as pessoas, é preciso fazer uso de uma mensagem, que deve armazenar uma hora e data de envio, hora e data de visualização, e a natureza se é privada ou pública, permitindo que os responsáveis comuniquem-se com a instituição de forma mais reservada sobre os assuntos que conferem seus filhos ou para que todos possam ver uma mensagem de âmbito geral, e também um código de envio que torne a mensagem ímpar.

Ainda que o armazenamento destas informações seja essencial, para que o sistema funcione adequadamente, algumas funcionalidades devem estar inclusas. Para disponibilizar corretamente o conteúdo das mensagens, a visibilidade das mesmas devem ser consultadas, assim, uma lista de mensagens consideradas públicas ficarão disponíveis para o acesso geral, enquanto as mensagens consideradas privadas só poderão ser acessadas por quem as enviou e o administrador, que possui a chave de acesso.

Caso a necessidade de uma constatação mais séria sobre um determinado aluno, em que esta deva ser feita pessoalmente, deve-se haver uma consulta para reaver os dados pessoais do responsável, como telefone ou e-mail. Igualmente necessária é uma consulta sobre quais as avaliações que já foram feitas, para que desse modo possa-se ter acesso às notas já computadas. Ainda nesse espectro, é importante recuperar, também, as datas em que os acontecimentos do tipo avaliação estão agendadas para que os responsáveis tenham conhecimento da relação de quais dias elas estão previstas para suceder.

Verificar, a partir do diário, a quantidade de faltas referentes a um aluno, para que os pais possam ser alertados sendo o número muito elevado, tendo o cuidado de verificar se ele está atualizado, para que não exista a chance de propagação de informações erradas quanto a porcentagem de presença.

É interessante uma consulta nos eventos propiciados pela instituição, provendo uma lista dos eventos que acontecerão mensalmente, informando a descrição associada a cada um deles, bem como a descrição sobre o que será o evento. Dessa forma, a disseminação das informações será bem mais ampla do que através de notificações físicas.

De forma a disponibilizar informações sobre a duração do trimestre, é importante

consultar o dia de início e término; à vista disso, tornar-se-á possível verificar quantos dias já foram computados e quantos ainda faltam até o término do período escolar em questão.

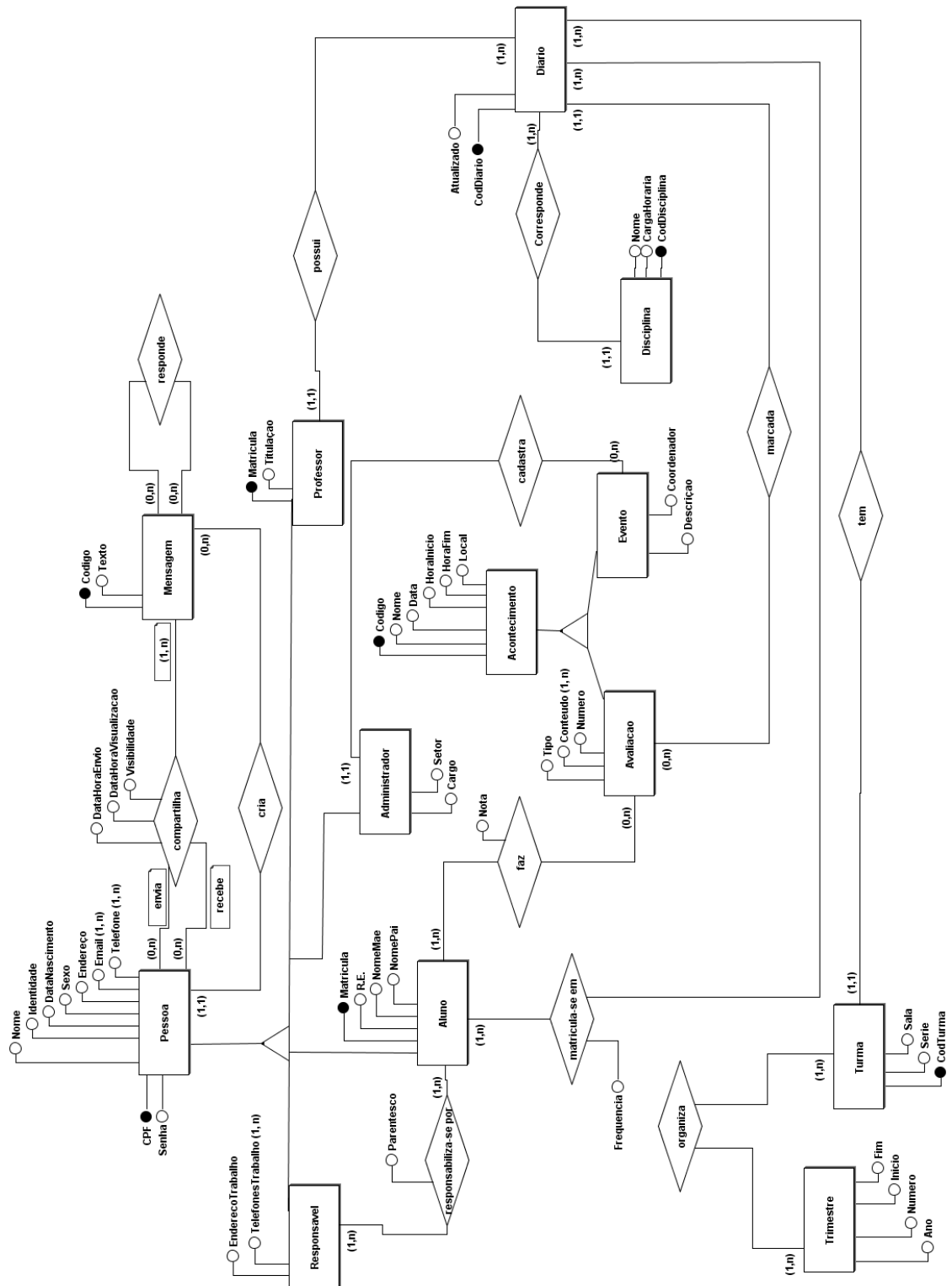
Para fins gerenciais, consultar quantas e quais disciplinas cada professor ministra acaba por ser significativo, pois a direção poderá controlar qual a carga horária que o professor está desempenhando na escola e prevenir que ela ultrapasse o valor máximo requisitado pelo docente.

Outra consulta significativa para o bom funcionamento dentro da empresa é a verificação do rendimento escolar dos alunos vinculados a uma turma, selecionando-os de acordo com a série. Promovendo uma melhor análise de como cada série está se saindo academicamente.

A escola deseja, também, monitorar qual o período de maior troca de mensagens: manhã, tarde ou noite; para este fim, deve ser consultado o horário de envio, organizando-o entre essas categorias. O retorno deve ser aquele que possuir a maior quantidade de mensagens selecionadas. Com essa informação a escola poderá organizar o melhor tempo para compartilhar dados referentes à escola.

Ela também confere aos alunos uma premiação por desempenho nas avaliações do tipo simulado, os três alunos com a maior nota recebem uma bonificação. De modo a filtrar mais facilmente quem são esses alunos, é preciso selecionar, dentre as avaliações, aquelas do tipo simulado e, a partir dessa informação, filtrar os alunos com melhor nota.

2.2 Diagrama de Entidade-Relacionamento



2.3 Dicionário conceitual de dados

- **Entidade Pessoa:**

- É uma generalização entre Responsável, Aluno, Administrador e Professor, que foi criada para armazenar todas as informações que são comuns a todas as pessoas.

Atributo(s):

- Nome: é um atributo que armazena o nome completo da pessoa.
- CPF: é um atributo chave que armazena o CPF da pessoa.
- Senha: é um atributo que armazena o código de acesso ao sistema.
- Identidade: é um atributo que armazena a identidade, registro geral, da pessoa.
- DataNascimento: é um atributo que armazena a data de nascimento da pessoa.
- Sexo: é um atributo que armazena o sexo da pessoa.
- Endereço: é um atributo que armazena o endereço completo da pessoa.
- Email: é um atributo multivalorado que armazena os e-mails de uma pessoa.
- Telefone: é um atributo multivalorado que armazena os números de telefone de uma pessoa.

- **Entidade Responsável:**

- É uma especificação da entidade Pessoa e foi criada com o propósito de armazenar as informações restritas ao responsável.

Atributo(s):

- EnderecoTrabalho: é um atributo que armazena o endereço de trabalho do responsável.
- TelefonesTrabalho: é um atributo multivalorado que armazena os telefones de trabalho do responsável.

- **Entidade Aluno:**

- É uma especialização da entidade Pessoa e foi criada com o propósito de armazenar as informações restritas ao aluno.

Atributo(s):

- Matricula: é um atributo chave que armazena um valor de matrícula.
- R.E.: é um atributo que armazena um valor de rendimento escolar.
- NomePai: é um atributo que armazena o nome do pai.
- NomeMae: é um atributo que armazena o nome da mãe.

- **Entidade Administrador:**

- É uma especialização da entidade Pessoa que foi criada para armazenar informações que somente um administrador possui.

Atributo(s):

- Cargo: é um atributo que armazena a função que o administrador exerce na instituição.
- Setor: é um atributo que armazena o setor que o administrador trabalha na instituição.

- **Entidade Professor:**

- É uma especialização da entidade Pessoa que tem por principal função armazenar informações específicas referentes aos professores.

Atributo(s):

- Matricula: é um atributo chave que armazena a matrícula do professor.
- Titulacao: é um atributo que armazena a maior titulação do professor.

- **Entidade Turma:**

- É uma entidade criada para armazenar informações referentes a estrutura de uma turma escolar.

Atributo(s):

- Sala: é um atributo que armazena o número da turma.
- Série: é um atributo que armazena o número da série.
- CodTurma: é um atributo chave que armazena um código único da turma.

- **Entidade Acontecimento:**

- É uma generalização de Avaliacao e Evento que foi criada para armazenar todas as informações que esses dois acontecimentos têm em comum.

Atributo(s):

- Codigo: é um atributo chave que armazena o código de identificação do acontecimento.
- Nome: é um atributo que armazena a descrição do acontecimento.
- Data: é um atributo composto que armazena data.
- HoraInicio: é um atributo que armazena o horário de inicio de um acontecimento.
- HoraFim: é um atributo que armazena o horário do fim de um acontecimento.
- Local: é um atributo que armazena o local onde o acontecimento acontecerá.

- **Entidade Diario:**

- É uma entidade que foi criada com o propósito de armazenar informações sobre os diários dos professores.

Atributo(s):

- CodDiario: é um atributo chave que armazena um código identificador do diário.
- Atualizado: é um atributo que armazena o *status* do diário, ou seja, se ele está em dia ou não.

- **Entidade Avaliacao:**

- É uma especificação da entidade Acontecimento que armazena informações referentes apenas a entidade Avaliacao.

Atributo(s):

- Tipo: é um atributo que armazena qual o tipo de avaliação aplicada.
- Conteudo: é um atributo multivalorado que armazena os conteúdos referentes à avaliação.
- Numero: é um atributo que armazena qual o número da avaliação que está sendo aplicada.

- **Entidade Evento:**

- É uma especificação da entidade Acontecimento que armazena informações que referem-se apenas à entidade Evento.

Atributo(s):

- Descricao: é um atributo que armazena a descrição do evento.
- Coordenador: é um atributo que armazena o nome do coordenador do evento.

- **Entidade Disciplina:**

- É uma entidade que foi criada com o propósito de armazenar informações sobre as disciplinas.

Atributo(s):

- Nome: é um atributo que armazena o nome da disciplina.
- CargaHoraria: é um atributo que armazena a quantidade de horas-aula da disciplina.
- CodDisciplina: é um atributo chave que armazena um código identificador da disciplina.

- **Entidade Mensagem:**

- É uma entidade criada por pessoa com o propósito de armazenar informações sobre as mensagens compartilhadas.

Atributo(s):

- Codigo: é um atributo chave que armazena um código identificador da mensagem.
- Texto: é um atributo que armazena o corpo textual da mensagem.
- Privada: é um atributo que armazena o tipo, se a mensagem é privada ou pública.

- **Entidade Trimestre:**

- É uma entidade que foi criada com o propósito de armazenar informações sobre os trimestres que dividem o ano escolar.

Atributo(s):

- Ano: é um atributo que armazena o ano letivo.
- Numero: é um atributo que armazena o número do trimestre letivo, caracterizando-o em primeiro, segundo ou terceiro.
- Inicio: é um atributo que armazena a data em que o trimestre começa.
- Fim: é um atributo que armazena a data em que o trimestre termina.

- **Relacionamentos**

- responde: é um relacionamento recursivo entre duas instâncias da entidade **Mensagem**. Uma Mensagem responde, no mínimo, uma outra Mensagem e, no máximo, várias outras Mensagens. Ela também é respondida por ,no mínimo, nenhuma Mensagem e, no máximo, várias Mensagens.
- compartilha: é um relacionamento recursivo ternário das entidades **Pessoa** e **Mensagem**. Duas instâncias de Pessoa, devem estar relacionadas a, no mínimo, uma Mensagem e, no máximo, várias Mensagens. Uma instância de Pessoa (envia) e outra de Mensagem devem estar relacionadas a, no mínimo, nenhuma outra pessoa e, no máximo, várias outras pessoas. Uma instância de Pessoa (recebe) e outra de Mensagem devem estar relacionadas a, no mínimo, nenhuma outra Pessoa e, no máximo, várias outras Pessoas, possui os atributos **DataHoraEnvio** que armazena a data e hora em que a mensagem foi enviada, **DataHoraVisualizacao** que armazena a data e hora em que a mensagem foi visualizada e **Visibilidade** que armazena se a mensagem é pública ou privada.
- responsabiliza-se por: é um relacionamento da entidade **Responsável** e a entidade **Aluno**, onde armazena o parentesco. Um Responsável deve estar relacionado a, no mínimo, um Aluno e, no máximo, vários Alunos. Um Aluno

deve estar relacionado a, no mínimo, um Responsável e, no máximo, vários Responsáveis.

- organiza: é um relacionamento das entidades **Trimestre e Turma**. Uma Turma organiza-se em, no mínimo, um trimestre, e, no máximo, vários trimestres. Um Trimestre é organizado para, no mínimo, uma Turma, e, no máximo, várias Turmas.
- matricula-se em: é um relacionamento que envolve as entidades **Aluno e Diário**, armazenando a frequência do aluno no Diário. Um Aluno deve estar matriculado em, no mínimo, em um Diário e, no máximo, muitos Diários. Um Diário pode ter matriculado, no mínimo, um Aluno, e, no máximo, vários Alunos.
- faz: é um relacionamento das entidades **Aluno e Avaliação**, armazenando a nota que o Aluno recebeu. Um Aluno pode fazer, no mínimo, nenhuma Avaliação e no máximo várias Avaliações. Uma avaliação pode ser feita por, no mínimo, um aluno, e no máximo vários Alunos.
- cadastra: é um relacionamento entre as entidades **Administrador e Evento**. Um Administrador pode cadastrar, no mínimo, nenhum Evento e, no máximo, vários Eventos. Um Evento pode ser cadastrado por, no mínimo e no máximo, apenas um Administrador.
- marca: é um relacionamento das entidades **Diário e Avaliação**. Uma Avaliação pode ser marcada por, no mínimo e no máximo, apenas um Diário. Um Diário pode marcar, no mínimo, nenhuma e, no máximo, muitas avaliações.
- corresponde: é um relacionamento das entidades **Disciplina e Diário**. Uma Disciplina pode corresponder, no mínimo e no máximo, apenas um diário. Um Diário é correspondido por sua vez, no mínimo, uma Disciplina e, no máximo, várias Disciplinas.
- está contida: é um relacionamento entre as entidades **Turma e Diário**. Um Diário pode ter, no mínimo e no máximo, uma Turma vinculada a ele. Uma Turma pode ter, no mínimo, um Diário e, no máximo, vários Diários vinculados a ela.
- possui: é um relacionamento entre as entidades Professor e Diário. Um professor pode possuir, no mínimo, um Diário, e, no máximo, vários Diários. Um Diário, por sua vez, pode ter, no máximo e no mínimo, apenas um

Professor vinculado a ele.

- cria: é um relacionamento entre as entidade **Pessoa** e **Mensagem**. Uma Pessoa pode criar, no mínimo, nenhuma e no máximo muitas Mensagens. Uma Mensagem pode ser criada por, no máximo e no mínimo, apenas uma Pessoa.

3 Modelo lógico

3.1 Mapeamento Entidade Relacionamento

PESSOA(nome, cpf, senha, identidade, dataNascimento, sexo, endereco);

RESPONSAVEL(cpf, enderecoTrabalho);

ALUNO(cpf, matricula, re, nomeMae, nomePai);

ADMINISTRADOR(cpf, cargo, setor);

PROFESSOR(cpf, matricula, titulacao);

ACONTECIMENTO(codAcontecimento, nome, data, horarioInicio, horarioFim, local);

DIARIO(codDiario, atualizado, matProf, codTurma, codDisciplina);

DISCIPLINA(nome, cargaHoraria, codDisciplina);

EVENTO(codAcontecimento, coordenador, descricao, cpfAdm);

TURMA(codTurma, serie, sala);

TRIMESTRE(ano, numero, dataInicio, dataFim);

MENSAGEM(codMensagem, texto, cpf);

AVALIACAO(codAcontecimento, codDiario, numero, tipo,);

RESPONSABILIZA(codResponsavel, cpfAluno, parentesco);

MATRICULA(cpfAluno, codDiario, frequencia);

RESULTADOAVALIACAO(cpfAluno, codAcontecimento, nota);

RESPONDEMENSAGEM(codMensagem, codResposta);

ORGANIZATRIMESTRE(ano, numero, codTurma);

COMPARTILHAMENSAGEM(cpfEnvia, cpfRecebe, codMensagem, visibilidade, dataHoraEnvio, dataHoraVisualizacao);

CONTEUDOAVALIACAO(codAcontecimento, conteudo);

EMAILPESSOA(cpf, email);

TELEFONEPESSOA(cpf, telefone);

TELEFONERESPONSAVEL(cpf, telefone);

3.2 Dicionário Lógico de Dados

PESSOA: Relação que armazena os dados gerais de responsável, aluno, administrador e professor.				
Atributo	Descrição	Tipo	Domínio	Restrição
nome	Representa o nome de um responsável, aluno, administrador ou professor.	VARCHAR (100)	VARCHAR (100)	• Não nulo
cpf	Representa o número do CPF de um responsável, aluno, administrador ou professor.	VARCHAR (20)	VARCHAR (20)	• Chave Primária
senha	Representa a palavra-chave de acesso ao sistema de um responsável, aluno administrador ou professor.	VARCHAR (32)	VARCHAR (32)	• Não nulo
identidade	Representa o número e o órgão emissor do documento de identificação de um responsável, aluno administrador ou professor	VARCHAR (32)	VARCHAR (32)	• Não nulo
dataNascimento	Representa a data de nascimento de um responsável, aluno, administrador ou professor.	DATE	DATE	• Não nulo
sexo	Representa o sexo de um responsável, aluno, administrador ou professor.	VARCHAR (10)	VARCHAR (10)	• Não nulo
endereço	Representa o endereço de um responsável, aluno, administrador ou professor.	VARCHAR (100)	VARCHAR (100)	• Não nulo

Tabela 1: Relação PESSOA.

RESPONSÁVEL: Relação que armazena os dados específicos do responsável por aluno.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o número do CPF de um responsável por aluno.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave estrangeira referenciando o CPF na tabela Pessoa • Chave Primária

RESPONSAVEL: Relação que armazena os dados específicos do responsável por aluno.				
Atributo	Descrição	Tipo	Domínio	Restrição
endereçoTrabalho	Representa o endereço de trabalho do responsável pelo aluno.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> Sem restrição

Tabela 2: Relação RESPONSAVEL.

ALUNO: Relação que armazena os dados específicos do aluno.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o número do CPF do aluno.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> Chave estrangeira referencian do o CPF na tabela Pessoa. Chave Primária.
matricula	Representa a matrícula do aluno.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave.
RE	Representa o rendimento acadêmico do aluno.	REAL	Números reais não negativos.	<ul style="list-style-type: none"> Sem restrição.
nomeMae	Representa o nome da mãe do aluno.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> Sem restrição.
nomePai	Representa o nome do pai do aluno.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> Sem restrição.

Tabela 3: Relação ALUNO.

ADMINISTRADOR: Relação que armazena os dados específicos do administrador.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o número do CPF do administrador.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> Chave estrangeira referenciando CPF na tabela Pessoa Chave Primária
cargo	Representa o cargo do administrador.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> Não nulo.
setor	Representa o setor onde o administrador trabalha.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> Não nulo.

Tabela 4: Relação ADMINISTRADOR.

PROFESSOR: Relação que armazena os dados específicos do professor.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o número do CPF do professor.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave estrangeira referenciando o CPF na tabela Pessoa. • Chave Primária.
matricula	Representa a matrícula do professor.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Atributo Chave.
titulacao	Representa a titulação do professor.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Não nulo.

Tabela 5: Relação PROFESSOR.

ACONTECIMENTO: Relação que armazena os dados gerais de Avaliacao e Evento.				
Atributo	Descrição	Tipo	Domínio	Restrição
codAcontecimento	Representa o código que identifica o acontecimento	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Primária
nome	Representa o nome do acontecimento.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Não nulo.
data	Representa o ano do acontecimento.	INTEGER	Número inteiros positivos.	<ul style="list-style-type: none"> • Não nulo.
horarioInicio	Representa a hora prevista para o início do evento.	TIME	TIME	Nulo.
horarioFim	Representa a hora prevista para o fim do evento.	TIME	TIME	<ul style="list-style-type: none"> • Não nulo.
local	Representa o local do evento.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Sem restrição.

Tabela 6: Relação ACONTECIMENTO.

DIARIO: Relação que armazena os dados do diário.				
Atributo	Descrição	Tipo	Domínio	Restrição
codDiario	Representa o código de identificação do diário.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Primária.
atualizado	Representa estado do diário, se está atualizado ou não.	BOOLEAN	BOOLEAN	<ul style="list-style-type: none"> • Não nulo. • Default True.

DIARIO: Relação que armazena os dados do diário.				
Atributo	Descrição	Tipo	Domínio	Restrição
matProf	Representa a matrícula do professor proprietário do diário.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Professor. • Não Nulo.
codTurma	Representa o código da turma que contém o diário.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Turma. • Não Nulo.
codDisciplina	Representa o código da disciplina a que o diário corresponde.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Disciplina.

Tabela 7: Relação DIARIO.

DISCIPLINA: Relação que armazena os dados da disciplina.				
Atributo	Descrição	Tipo	Domínio	Restrição
nome	Representa o nome da disciplina.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Não Nula.
cargaHoraria	Representa a total de horas aulas que compõem a carga horária da disciplina.	INTEGER	Inteiro não negativo	<ul style="list-style-type: none"> • Não nulo. • Default 0.
codDisciplina	Representa o código de identificação da disciplina.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Primária.

Tabela 8: Relação DISCIPLINA.

EVENTO: Relação que armazena os dados específicos do evento.				
Atributo	Descrição	Tipo	Domínio	Restrição
codAcontecimento	Representa o código de identificação do Acontecimento.	VARC HAR (32)	VARC HAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Acontecimento. • Chave Primária.
coordenador	Representa o nome do responsável por coordenar o evento.	VARC HAR (100)	VARC HAR (100)	<ul style="list-style-type: none"> • Não nulo.
descricao	Representa a descrição do evento.	VARC HAR (100)	VARC HAR (100)	<ul style="list-style-type: none"> • Não nulo.
cpfAdm	Representa o CPF do administrador responsável por cadastrar o evento.	VARC HAR (20)	VARC HAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Administrador. • Não nulo.

Tabela 9: Relação EVENTO.

TURMA: Relação que armazena os dados da turma.				
Atributo	Descrição	Tipo	Domínio	Restrição
codTurma	Representa o código de identificação da turma.	VARCHAR (32)	VARCHAR (32)	• Chave Primária.
serie	Representa a série da turma.	VARCHAR (32)	VARCHAR (32)	• Não nulo.
sala	Representa a identificação da sala.	VARCHAR (32)	VARCHAR (32)	• Não nulo.

Tabela 10: Relação TURMA.

TRIMESTRE: Relação que armazena os dados do trimestre.				
Atributo	Descrição	Tipo	Domínio	Restrição
ano	Representa o ano do trimestre.	INTEGER	Número inteiro positivo	• Compõe a Chave Primária junto com o número.
numero	Representa o número do trimestre no ano.	INTEGER	Número entre os valores entre 1 e 3.	• Compõe a Chave Primária junto com ano.
dataInicio	Representa a data prevista para o início do trimestre no ano.	DATE	DATE	• Não nulo.
dataFim	Representa a data prevista para o fim do trimestre no ano.	DATE	DATE	• Não nulo.

Tabela 11: Relação TRIMESTRE.

MENSAGEM: Relação que armazena os dados da mensagem.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o CPF da pessoa que criou a mensagem.	VARC HAR (20)	VARC HAR (20)	• Chave Estrangeira da tabela Pessoa.
codMessage m	Representa o código de identificação da mensagem.	VARC HAR (32)	VARC HAR (32)	• Chave Primária.
texto	Representa o conteúdo da mensagem.	VARC HAR (100)	VARC HAR (100)	• Não nulo.

Tabela 12: Relação MENSAGEM.

AVALIACAO: Relação que armazena os dados da avaliação.				
Atributo	Descrição	Tipo	Domínio	Restrição
codDiario	Representa o código de identificação do diário.	VARC HAR (32)	VARC HAR (32)	• Chave Estrangeira da tabela Diário. • Não nulo.
numero	Representa o número da avaliação (1 – Primeira, 2 – Segunda e 3 – terceira).	INTEGE R	Número inteiro positivo	• Não nulo.
tipo	Representa o tipo da avaliação ministrada (Avaliação, Trabalho ou Simulado).	VARC HAR (32)	VARC HAR (32)	• Não nulo.
codAconteciment o	Representa o código da tabela Acontecimento.	VARC HAR (32)	VARC HAR (32)	• Chave Estrangeira da tabela Acontecimento. • Não nulo.

Tabela 13: Relação AVALIACAO.

RESPONSABILIZA: Relação que armazena os dados da relacionamento do responsável pelo aluno.				
Atributo	Descrição	Tipo	Domínio	Restrição
codResponsavel	Representa o CPF da especialização responsável.	VARC HAR (32)	VARC HAR (32)	• Chave Estrangeira da tabela Responsável • Compõem a Chave Primária junto com matrícula.

RESPONSABILIZA: Relação que armazena os dados da relacionamento do responsável pelo aluno.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpfAluno	Representa o CPF da especialização Aluno.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Aluno Compõe a Chave Primária.
parentesco	Representa o tipo relação entre o responsável e o aluno.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Não nulo.

Tabela 14: Relação RESPONSABILIZA.

MATRICULA: Relação que armazena os dados da matrícula do aluno no diário.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpfAluno	Representa o CPF do aluno.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Aluno. Compõe a Chave Primária junto com codDiário.
codDiario	Representa o código do diário.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Diário. Compõe junto com matrícula do aluno a Chave Primária.
frequencia	Representa a frequência do aluno ao diário.	REAL	Número real não negativo.	<ul style="list-style-type: none"> Não nulo.

Tabela 15: Relação MATRICULA.

RESULTADOAVALIACAO: Relação que armazena os resultados da avaliação.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpfAluno	Representa o CPF do aluno.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Aluno. Compõe a Chave Primária junto com codAcontecimento.

RESULTADOAVALIACAO: Relação que armazena os resultados da avaliação.				
Atributo	Descrição	Tipo	Domínio	Restrição
codAcontecimento	Representa o código do Acontecimento.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Acontecimento. Compõe junto com cpfAluno a Chave Primária.
nota	Representa o nota do aluno na avaliação.	REAL	Número real não negativo.	<ul style="list-style-type: none"> Não nulo.

Tabela 16: Relação RESULTADOAVALIACAO.

RESPONDEMENSAGEM: Relação que armazena os resultados da avaliação.				
Atributo	Descrição	Tipo	Domínio	Restrição
codMensagem	Representa o código da mensagem original.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Mensagem. Compõe a Chave Primária junto com codResposta.
codResposta	Representa o código da mensagem reposta.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Mensagem. Compõe junto com codMensagem a Chave Primária.

Tabela 17: Relação RESPONDEMENSAGEM.

ORGANIZATRIMESTRE: Relação que armazena a organização das turmas no trimestre.				
Atributo	Descrição	Tipo	Domínio	Restrição
ano	Representa o ano do trimestre.	Integer	Número inteiro positivo.	<ul style="list-style-type: none"> Chave Estrangeira que referencia a tabela Trimestre. Compõe a Chave Primária.
numero	Representa o número do trimestre.	Integer	Número inteiro positivo	<ul style="list-style-type: none"> Chave Estrangeira que referencia a tabela Trimestre. Compõe a Chave Primária.
codTurma	Representa o código da turma.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> Chave Estrangeira da tabela Turma. Compõe a Chave Primária.

Tabela 18: Relação ORGANIZATRIMESTRE.

COMPARTILHAMENSAGEM: Relação que armazena a o compartilhamento das mensagens.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpfEnvia	Representa o CPF da pessoa que compartilha a mensagem.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira que referencia a tabela Pessoa. • Compõe a Chave Primária.
cpfRecebe	Representa o CPF da pessoa que recebe a mensagem.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira que referencia a tabela Pessoa. • Compõe a Chave Primária.
codMensagem	Representa o código da mensagem que está sendo compartilhada.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira que referencia a tabela Mensagem. • Compõe a Chave Primária. •
visibilidade	Representa a visibilidade da mensagem que está sendo compartilhada se ela é pública ou privada.	BOOLEAN	BOOLEAN	<ul style="list-style-type: none"> • Assume os valores TRUE para visibilidade Pública e FALSE para visibilidade Privada. • Não nulo
dataHoraEnvio	Representa a data e hora em que a mensagem foi compartilhada.	TIMESTAMP	TIMESTAMP	<ul style="list-style-type: none"> • Não nulo
dataHoraVisualizacao	Representa a data e hora em que a mensagem foi visualizada, caso seja nulo indica que a mensagem ainda não foi visualizada.	TIMESTAMP	TIMESTAMP	<ul style="list-style-type: none"> • Sem restrição • Não houve visualização quando o valor for NULL.

Tabela 19: Relação COMPARTILHAMENSAGEM.

CONTEUDO AVALIACAO: Relação que armazena a os conteúdo da avaliação.				
Atributo	Descrição	Tipo	Domínio	Restrição
codAcontecimento	Representa o código da Avaliação.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Avaliação. • Compõe a Chave Primária.
conteudo	Representa o conteúdo cobrado na avaliação.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Compõe a Chave Primária.

Tabela 20: Relação CONTEUDO AVALIACAO.

EMAILPESSOA: Relação que armazena a os e-mails da generalização pessoa.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o CPF da relação pessoa.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Pessoa. • Compõem a Chave Primária.
email	Representa o e-mail da generalização pessoa.	VARCHAR (100)	VARCHAR (100)	<ul style="list-style-type: none"> • Compõe a Chave Primária.

Tabela 21: Relação EMAILPESSOA.

TELEFONEPESSOA: Relação que armazena a os telefones da generalização pessoa.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o CPF da relação pessoa.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Pessoa. • Compõe a Chave Primária junto com email.
telefone	Representa o telefone da generalização pessoa.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Compõe junto com cpf a Chave Primária.

Tabela 22: Relação TELEFONEPESSOA.

TELEFONERESPONSAVEL: Relação que armazena a os telefones do responsável.				
Atributo	Descrição	Tipo	Domínio	Restrição
cpf	Representa o CPF da relação responsável.	VARCHAR (20)	VARCHAR (20)	<ul style="list-style-type: none"> • Chave Estrangeira da tabela Responsável. • Compõe a Chave Primária junto com email.
telefone	Representa o telefone da especialização responsável.	VARCHAR (32)	VARCHAR (32)	<ul style="list-style-type: none"> • Compõe junto com cpf a Chave Primária.

Tabela 23: Relação TELEFONERESPONSAVEL.

4 Modelo físico

4.1 Script SQL

4.1.1 Criando as relações

- **Criando a relação PESSOA**

```
CREATE TABLE pessoa (
    nome VARCHAR (100) NOT NULL,
    cpf VARCHAR (20),
    senha VARCHAR (32) NOT NULL,
    identidade VARCHAR (32) NOT NULL,
    dataNascimento DATE NOT NULL,
    sexo VARCHAR (10) NOT NULL,
    endereco VARCHAR (100) NOT NULL,
    CONSTRAINT pessoa_pk PRIMARY KEY (cpf)
);
```

- **Criando a relação RESPONSVEL**

```
CREATE TABLE responsavel (
    cpf VARCHAR (20),
    enderecoTrabalho VARCHAR (100),
    CONSTRAINT responsavel_pk PRIMARY KEY (cpf),
    CONSTRAINT responsavel_fk FOREIGN KEY (cpf)
REFERENCES pessoa (cpf)
);
```

- **Criando a relação ALUNO**

```
CREATE TABLE aluno(  
    cpf VARCHAR (20),  
    matricula VARCHAR (32) UNIQUE NOT NULL,  
    re REAL,  
    nomeMae VARCHAR (100),  
    nomePai VARCHAR (100),  
    CONSTRAINT re_positivo CHECK (re>=0),  
    CONSTRAINT aluno_pk PRIMARY KEY (cpf),  
    CONSTRAINT aluno_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação ADMINISTRADOR**

```
CREATE TABLE administrador(  
    cpf VARCHAR (20),  
    cargo CHAR(100) NOT NULL,  
    setor VARCHAR(100) NOT NULL,  
    CONSTRAINT adm_pk PRIMARY KEY (cpf),  
    CONSTRAINT amd_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação PROFESSOR**

```
CREATE TABLE professor(  
    cpf VARCHAR (20),  
    matricula VARCHAR (32) UNIQUE NOT NULL,  
    titulacao VARCHAR (100) NOT NULL,  
    CONSTRAINT prof_pk PRIMARY KEY (cpf),  
    CONSTRAINT prof_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação TRIMESTRE**

```
CREATE TABLE trimestre(  
    ano INT,  
    numero INT,  
    dataInicio DATE NOT NULL,  
    dataFim DATE NOT NULL,  
    CONSTRAINT trimestre_pk  
PRIMARY KEY (ano, numero),  
    CONSTRAINT ano_positivo CHECK (ano>0),  
    CONSTRAINT trimestre_num  
CHECK (numero>=1 AND numero<=3)  
);
```

- **Criando a relação TURMA**

```
CREATE TABLE turma(  
    codTurma VARCHAR(32),  
    serie VARCHAR(32) NOT NULL,  
    sala VARCHAR(32) NOT NULL,  
    CONSTRAINT turma_pk PRIMARY KEY(codTurma)  
);
```

- **Criando a relação ACONTECIMENTO**

```
CREATE TABLE acontecimento(  
    codAcontecimento VARCHAR(32),  
    nome VARCHAR(100) NOT NULL,  
    data DATE NOT NULL,  
    horarioInicio TIME NOT NULL,  
    horarioFim TIME NOT NULL,  
    local VARCHAR(100),  
    CONSTRAINT acont_pk  
PRIMARY KEY (codAcontecimento)  
);
```

- **Criando a relação DISCIPLINA**

```
CREATE TABLE disciplina(  
    nome VARCHAR(100) NOT NULL,  
    cargaHoraria INT NOT NULL DEFAULT 0,  
    codDisciplina VARCHAR(32),  
    CONSTRAINT codDisciplina_pK  
PRIMARY KEY(codDisciplina)  
);
```

- **Criando a relação DIARIO**

```
CREATE TABLE diario(  
    codDiario VARCHAR(32),  
    atualizado BOOLEAN NOT NULL DEFAULT TRUE,  
    matProf VARCHAR(32) NOT NULL,  
    codTurma VARCHAR(32) NOT NULL,  
    codDisciplina VARCHAR(32) NOT NULL,  
    CONSTRAINT diario_pk PRIMARY KEY(codDiario),  
    CONSTRAINT matProf_fk  
FOREIGN KEY(matProf)  
REFERENCES professor(matricula),  
    CONSTRAINT codTurma_fk FOREIGN KEY(codTurma)  
REFERENCES turma(codTurma),  
    CONSTRAINT codDisciplina_fk
```

```
FOREIGN KEY (codDisciplina)
REFERENCES disciplina(codDisciplina)
);
```

- **Criando a relação EVENTO**

```
CREATE TABLE evento(
    codAcontecimento VARCHAR(32),
    coordenador VARCHAR(100) NOT NULL,
    descrição VARCHAR(100) NOT NULL,
    cpfAdm VARCHAR(20) NOT NULL,
    CONSTRAINT evento_pk
PRIMARY KEY (codAcontecimento),
    CONSTRAINT cpfAdm_fk FOREIGN KEY (cpfAdm)
REFERENCES administrador(cpf)
);
```

- **Criando a relação MENSAGEM**

```
CREATE TABLE mensagem(
    cpf VARCHAR(20),
    codMensagem VARCHAR(32),
    texto VARCHAR(100) NOT NULL,
    CONSTRAINT mensagem_pk
PRIMARY KEY (codMensagem),
    CONSTRAINT cpf_fk
FOREIGN KEY (cpf) REFERENCES pessoa(cpf)
);
```

- **Criando a relação AVALIACAO**

```
CREATE TABLE avaliacao(
    codDiario VARCHAR(32) NOT NULL,
    numero INT NOT NULL,
    tipo VARCHAR(32) NOT NULL,
    codAcontecimento VARCHAR(32) NOT NULL,
    CONSTRAINTS codDiario_pk
PRIMARY KEY (codAcontecimento,
    CONSTRAINT codDiario_fk
FOREIGN KEY (codDiario)
REFERENCES diario(codDiario)
);
```

- **Criando a relação RESPONSABILIZA**

```
CREATE TABLE responsabiliza(
    codResponsavel VARCHAR(32),
    cpfAluno VARCHAR(20),
```

```

parentesco VARCHAR(32) NOT NULL,
CONSTRAINT responsabiliza_pk
PRIMARY KEY(codResponsavel, cpfAluno),
CONSTRAINT codResponsavel_fk
FOREIGN KEY(codResponsavel)
REFERENCES responsavel(cpf),
CONSTRAINT cpfAluno_fk
FOREIGN KEY(cpfAluno)
REFERENCES aluno(cpf)
);

```

- **Criando a relação MATRICULA**

```

CREATE TABLE matricula(
    cpfAluno VARCHAR(20),
    codDiario VARCHAR(32),
    frequencia REAL NOT NULL,
    CONSTRAINT matricula_pk
    PRIMARY KEY(cpfAluno, codDiario),
    CONSTRAINT cpfAluno_fk
    FOREIGN KEY(cpfAluno)
    REFERENCES Aluno(cpf),
    CONSTRAINT codDiario_pk
    FOREIGN KEY(codDiario)
    REFERENCES diario(codDiario),
    CONSTRAINT freq_positiva
    CHECK(frequencia>=0)
);

```

- **Criando a relação RESULTADOAVALIACAO**

```

CREATE TABLE resultadoavaliacao(
    cpfAluno VARCHAR(20),
    codAcontecimento VARCHAR(32),
    nota REAL NOT NULL,
    CONSTRAINT resultadoAvaliacao_pk
    PRIMARY KEY(cpfAluno, codAcontecimento),
    CONSTRAINT cpfAluno_fk
    FOREIGN KEY(cpfAluno)REFERENCES aluno(cpf),
    CONSTRAINT codAcontecimento_pk
    FOREIGN KEY (codAcontecimento)
    REFERENCES acontecimento(codAcontecimento),
    CONSTRAINT nota_positiva CHECK(nota>=0)
);

```

- **Criando a relação RESPONDEMENSAGEM**

```

CREATE TABLE respondemensagem(
    codMensagem VARCHAR(32),

```

```

codResposta VARCHAR(32),
CONSTRAINT respondemensagem_pk
PRIMARY KEY(codMensagem, codResposta),
CONSTRAINT codMensagem_pk
FOREIGN KEY (codMensagem)
REFERENCES mensagem(codMensagem),
CONSTRAINT codResposta_pk
FOREIGN KEY (codMensagem)
REFERENCES mensagem(codMensagem)
);

```

- **Criando a relação ORGANIZATRIMESTRE**

```

CREATE TABLE organizatrimestre(
    ano INT,
    numero INT,
    codTurma VARCHAR(32),
    CONSTRAINT organizatrimestre_pk
    PRIMARY KEY(ano, numero, codTurma),
    CONSTRAINT trimestre_fk
    FOREIGN KEY (ano, numero)
    REFERENCES trimestre(ano, numero),
    CONSTRAINT codTurma_pk FOREIGN KEY (codTurma)
    REFERENCES turma(codTurma)
);

```

- **Criando a relação COMPARTILHAMENSAGEM**

```

CREATE TABLE compartilhamensagem(
    cpfEnvia VARCHAR(20),
    cpfRecebe VARCHAR(20),
    codMensagem VARCHAR(32) NOT NULL,
    visibilidade BOOLEAN NOT NULL,
    dataHoraEnvio TIMESTAMP NOT NULL,
    dataHoraVisualizacao TIMESTAMP,
    CONSTRAINT compartilharmsg_pk
    PRIMARY KEY (cpfEnvia, cpfRecebe, codMensagem),
    CONSTRAINT cpfEnvia_fk FOREIGN KEY(cpfEnvia)
    REFERENCES pessoa(cpf),
    CONSTRAINT cpfRecebe_fk FOREIGN KEY(cpfRecebe)
    REFERENCES pessoa(cpf),
    CONSTRAINT codMensagem_fk
    FOREIGN KEY(codMensagem)
    REFERENCES mensagem(codMensagem)
);

```


- **Criando a relação PESSOA**

```
CREATE TABLE pessoa(  
    nome VARCHAR (100) NOT NULL,  
    cpf VARCHAR (20),  
    senha VARCHAR (32) NOT NULL,  
    identidade VARCHAR (32) NOT NULL,  
    dataNascimento DATE NOT NULL,  
    sexo VARCHAR (10) NOT NULL,  
    endereco VARCHAR (100) NOT NULL,  
    CONSTRAINT pessoa_pk PRIMARY KEY (cpf)  
);
```

- **Criando a relação RESPONSÁVEL**

```
CREATE TABLE responsavel(  
    cpf VARCHAR (20),  
    enderecoTrabalho VARCHAR (100),  
    CONSTRAINT responsavel_pk PRIMARY KEY(cpf),  
    CONSTRAINT responsavel_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação ALUNO**

```
CREATE TABLE aluno(  
    cpf VARCHAR (20),  
    matricula VARCHAR (32) UNIQUE NOT NULL,  
    RE REAL,  
    nomeMae VARCHAR (100),  
    nomePai VARCHAR (100),  
    CONSTRAINT re_positivo CHECK(RE>=0),  
    CONSTRAINT aluno_pk PRIMARY KEY (cpf),  
    CONSTRAINT aluno_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação ADMINISTRADOR**

```
CREATE TABLE administrador(  
    cpf VARCHAR (20),  
    cargo CHAR(100) NOT NULL,  
    setor VARCHAR(100) NOT NULL,  
    CONSTRAINT adm_pk PRIMARY KEY(cpf),  
    CONSTRAINT amd_fk FOREIGN KEY (cpf)  
REFERENCES pessoa(cpf)  
);
```

- **Criando a relação PROFESSOR**

```
CREATE TABLE professor(  
    cpf VARCHAR (20),  
    matricula VARCHAR (32) UNIQUE NOT NULL,  
    titulacao VARCHAR (100) NOT NULL,  
    CONSTRAINT prof_pk PRIMARY KEY (cpf),  
    CONSTRAINT prof_fk FOREIGN KEY (cpf)  
    REFERENCES pessoa(cpf)  
);
```

- **Criando a relação TRIMESTRE**

```
CREATE TABLE trimestre(  
    ano INT,  
    numero INT,  
    dataInicio DATE NOT NULL,  
    dataFim DATE NOT NULL,  
    CONSTRAINT trimestre_pk  
    PRIMARY KEY(ano, numero),  
    CONSTRAINT ano_positivo CHECK (ano>0),  
    CONSTRAINT trimestre_num  
    CHECK(numero>=1 AND numero<=3)  
);
```

- **Criando a relação TURMA**

```
CREATE TABLE turma(  
    codTurma VARCHAR(32),  
    serie VARCHAR(32) NOT NULL,  
    sala VARCHAR(32) NOT NULL,  
    CONSTRAINT turma_pk PRIMARY KEY(codTurma)  
);
```

- **Criando a relação ACONTECIMENTO**

```
CREATE TABLE acontecimento(  
    codAcontecimento VARCHAR(32),  
    nome VARCHAR(100) NOT NULL,  
    data DATE NOT NULL,  
    horarioInicio TIME NOT NULL,  
    horarioFim TIME NOT NULL,  
    local VARCHAR(100),  
    CONSTRAINT acont_pk  
    PRIMARY KEY (codAcontecimento)  
);
```

- **Criando a relação DISCIPLINA**

```
CREATE TABLE disciplina(  
    nome VARCHAR(100) NOT NULL,  
    cargaHoraria INT NOT NULL DEFAULT 0,  
    codDisciplina VARCHAR(32),  
    CONSTRAINT codDisciplina_pK  
PRIMARY KEY(codDisciplina)  
);
```

- **Criando a relação DIARIO**

```
CREATE TABLE diario(  
    codDiario VARCHAR(32),  
    atualizado BOOLEAN NOT NULL DEFAULT TRUE,  
    matProf VARCHAR(32) NOT NULL,  
    codTurma VARCHAR(32) NOT NULL,  
    codDisciplina VARCHAR(32) NOT NULL,  
    CONSTRAINT diario_pk PRIMARY KEY(codDiario),  
    CONSTRAINT matProf_fk  
FOREIGN KEY(matProf)  
REFERENCES professor(matricula),  
    CONSTRAINT codTurma_fk FOREIGN KEY(codTurma)  
REFERENCES turma(codTurma),  
    CONSTRAINT codDisciplina_fk  
FOREIGN KEY (codDisciplina)  
REFERENCES disciplina(codDisciplina)  
);
```

- **Criando a relação EVENTO**

```
CREATE TABLE evento(  
    codAcontecimento VARCHAR(32),  
    coordenador VARCHAR(100) NOT NULL,  
    descrição VARCHAR(100) NOT NULL,  
    cpfAdm VARCHAR(20) NOT NULL,  
    CONSTRAINT evento_pk  
PRIMARY KEY (codAcontecimento),  
    CONSTRAINT cpfAdm_fk FOREIGN KEY (cpfAdm)  
REFERENCES administrador(cpf)  
);
```

- **Criando a relação MENSAGEM**

```
CREATE TABLE mensagem(  
    cpf VARCHAR(20),  
    codMensagem VARCHAR(32),  
    texto VARCHAR(100) NOT NULL,  
    CONSTRAINT mensagem_pk  
PRIMARY KEY(codMensagem),
```

```

CONSTRAINT cpf_fk
FOREIGN KEY (cpf) REFERENCES pessoa (cpf)
);

```

- **Criando a relação AVALIACAO**

```

CREATE TABLE avaliacao(
    codDiario VARCHAR(32) NOT NULL,
    numero INT NOT NULL,
    tipo VARCHAR(32) NOT NULL,
    codAcontecimento VARCHAR(32) NOT NULL,
    CONSTRAINTS codDiario_pk
    PRIMARY KEY (codAcontecimento,
    CONSTRAINT codDiario_fk
    FOREIGN KEY (codDiario)
    REFERENCES diario (codDiario)
);

```

- **Criando a relação RESPONSABILIZA**

```

CREATE TABLE responsabiliza(
    codResponsavel VARCHAR(32),
    cpfAluno VARCHAR(20),
    parentesco VARCHAR(32) NOT NULL,
    CONSTRAINT responsabiliza_pk
    PRIMARY KEY (codResponsavel, cpfAluno),
    CONSTRAINT codResponsavel_fk
    FOREIGN KEY (codResponsavel)
    REFERENCES responsavel (cpf),
    CONSTRAINT cpfAluno_fk
    FOREIGN KEY (cpfAluno)
    REFERENCES aluno (cpf)
);

```

- **Criando a relação MATRICULA**

```

CREATE TABLE matricula(
    cpfAluno VARCHAR(20),
    codDiario VARCHAR(32),
    frequencia REAL NOT NULL,
    CONSTRAINT matricula_pk
    PRIMARY KEY (cpfAluno, codDiario),
    CONSTRAINT cpfAluno_fk
    FOREIGN KEY (cpfAluno)
    REFERENCES Aluno (cpf),
    CONSTRAINT codDiario_pk
    FOREIGN KEY (codDiario)
    REFERENCES diario (codDiario),
    CONSTRAINT freq_positiva

```

```
CHECK(frequencia>=0)
);
```

- **Criando a relação RESULTADOAVALIACAO**

```
CREATE TABLE resultadoavaliacao(
    cpfAluno VARCHAR(20),
    codAcontecimento VARCHAR(32),
    nota REAL NOT NULL,
    CONSTRAINT resultadoAvaliacao_pk
```

- **Criando a relação CONTEUDOAVALIACAO**

```
CREATE TABLE conteudoavaliacao(
    codAcontecimento VARCHAR(32),
    conteudo VARCHAR(100),
    CONSTRAINT conteudoavaliacao_pk
    PRIMARY KEY(codAcontecimento, conteudo),
    CONSTRAINT codAcontecimento_fk
    FOREIGN KEY(codAcontecimento)
    REFERENCES acontecimento(codAcontecimento),
);
```

- **Criando a relação EMAILPESSOA**

```
CREATE TABLE emailpessoa(
    cpf VARCHAR(20),
    email VARCHAR(100),
    CONSTRAINT emailpessoa_pk
    PRIMARY KEY(cpf, email),
    CONSTRAINT cpf_fk FOREIGN KEY(cpf)
    REFERENCES pessoa(cpf)
);
```

- **Criando a relação TELEFONEPESSOA**

```
CREATE TABLE telefonepessoa(
    cpf VARCHAR(20),
    telefone VARCHAR(32),
    CONSTRAINT telefonepessoa_pk
    PRIMARY KEY(cpf, telefone),
    CONSTRAINT cpf_fk FOREIGN KEY(cpf)
    REFERENCES pessoa(cpf)
);
```

- **Criando a relação TELEFONERESPONSAVEL**

```
CREATE TABLE telefoneresponsavel(
    cpf VARCHAR(20),
    telefone VARCHAR(32),
```

```

CONSTRAINT telefoneresponsavel_pk
PRIMARY KEY(cpf, telefone),
CONSTRAINT cpf_fk FOREIGN KEY(cpf)
REFERENCES responsavel(cpf)
);

```

4.1.2 Criando Índices

```

CREATE INDEX codacontindice ON
acontecimento(codacontecimento);
CREATE INDEX cpfindice ON administrador(cpf);
CREATE INDEX cpfalunoindice ON aluno(cpf);
CREATE INDEX coddinarioindice ON avaliacao(coddinario);
CREATE INDEX numeroindice ON avaliacao(numero);
CREATE INDEX codmensagemindice ON
compartilhamensagem(codmensagem);
CREATE INDEX codacontindice ON
conteudoavaliacao(codacontecimento);
CREATE INDEX coddinarioindice ON diario(coddinario);
CREATE INDEX coddiscindice ON disciplina(coddisciplina);
CREATE INDEX emailpessindice ON emailpessoa(cpf);
CREATE INDEX eventoindice ON evento(codacontecimento);
CREATE INDEX matriculaindice ON matricula(cpfaluno);
CREATE INDEX mensagemindice ON mensagem(cpf);
CREATE INDEX organtrimanoindice ON
organizatrimestre(ano);
CREATE INDEX codacontindice ON
acontecimento(codacontecimento);
CREATE INDEX cpfindice ON administrador(cpf);
CREATE INDEX cpfalunoindice ON aluno(cpf);
CREATE INDEX coddinarioindice ON avalia INDEX
organtrimnumeroindice ON organizatrimestre(numero);
CREATE INDEX pessoaindice ON pessoa(cpf);
CREATE INDEX profindice ON professor(cpf);
CREATE INDEX respmensindice ON
respondemensagem(codmensagem);
CREATE INDEX responsabilizaindice ON
responsabiliza(codresponsavel);
CREATE INDEX responsavelindice ON responsavel(cpf);
CREATE INDEX resultavalindice ON
resultadoavaliacao(codacontecimento);
CREATE INDEX telpesscpfindice ON telefonepessoa(cpf);
CREATE INDEX telpesstelindice ON
telefonepessoa(telefone);
CREATE INDEX trimestreanoindice ON trimestre(ano);
CREATE INDEX trimestrenumindice ON trimestre(numero);
CREATE INDEX turmaindice ON turma(codturma);
CREATE INDEX telrespccpfindice ON

```

```
telefoneresponsavel(cpf);  
CREATE INDEX telresptelindice ON  
telefoneresponsavel(telefone);
```

4.1.3 Visões

- Essa visão cria uma junção natural entre a relação PROFESSOR e PESSOA onde auxilia na recuperação dos dados do professor.

```
CREATE VIEW VisaoProfessorPessoa  
AS SELECT *  
FROM professor NATURAL LEFT JOIN pessoa;
```

- Essa visão cria uma junção natural entre a relação ALUNO e PESSOA onde auxilia na recuperação dos dados do aluno.

```
CREATE VIEW VisaoAlunoPessoa  
AS SELECT *  
FROM aluno NATURAL LEFT JOIN pessoa;
```

- Essa visão cria uma junção natural entre a relação ADMINISTRADOR e PESSOA onde auxilia na recuperação dos dados do aluno.

```
CREATE VIEW VisaoAdministradoPessoa  
AS SELECT *  
FROM administrador NATURAL LEFT JOIN pessoa;
```

- Essa visão cria uma junção natural entre a relação RESPONSÁVEL e PESSOA onde auxilia na recuperação dos dados do aluno.

```
CREATE VIEW VisaoResponsavelPessoa  
AS SELECT *  
FROM responsavel NATURAL LEFT JOIN pessoa;
```

- Essa visão cria uma junção natural entre a relação EVENTO e ACONTECIMENTO onde auxilia na recuperação dos dados do aluno.

```
CREATE VIEW VisaoEvento  
AS SELECT *  
FROM evento NATURAL LEFT JOIN acontecimento;
```

- Essa visão cria uma junção natural entre a relação AVALIACAO e ACONTECIMENTO onde auxilia na recuperação dos dados do aluno.

```
CREATE VIEW VisaoAvaliacao
AS SELECT
    av.codAcontecimento    AS codAcontecimento,
    av.tipo                AS tipoAvaliacao,
    av.numero              AS numAvaliacao,
    ac.nome                AS nomeAvaliacao,
    ac.data                AS dataAvaliacao,
    ac.horarioInicio      AS horaIniAvaliacao,
    ac.horarioFim          AS horaFimAvaliacao,
    ac.local               AS localAvaliacao,
    dc.nome                AS disciplina,
    di.codDiario           AS codDiario,
    ps.cpf                 AS cpfProfessor,
    ps.nome                AS professor
FROM
    avaliacao av,
    acontecimento ac,
    diario di,
    disciplina dc,
    professor pr,
    pessoa ps
WHERE
    av.codAcontecimento = ac.codAcontecimento AND
    av.codDiario = di.codDiario AND
    di.codDisciplina = dc.codDisciplina AND
    di.matProf = pr.matricula AND
    ps.cpf = pr.cpf;
```

- Essa visão cria uma junção entre DIARIO, TURMA, DISCIPLINA e PROFESSOR onde auxilia na recuperação dos dados do diário.

```
CREATE VIEW VisaoDiario
AS SELECT d.codDiario, d.atualizado, p.matricula, p.nome
AS Professor, t.serie, t.sala, dc.codDisciplina, dc.nome
AS Disciplina, dc.cargaHoraria
FROM diario d, disciplina dc, turma t,
visaoProfessorPessoa p
WHERE d.codDisciplina = dc.codDisciplina AND t.codTurma =
d.codTurma AND p.matricula=d.matProf
```


4.1.4 Funções agregadas

- Procedimento armazenado que retorna a quantidade de dias entre duas datas.

```
CREATE OR REPLACE FUNCTION diasEntreDatas (DATE, DATE)
RETURNS INTEGER
AS $$
    DECLARE
        dataMaior DATE;
        dataMenor DATE;
        dias INTEGER;
    BEGIN
        IF ($1<$2) THEN
            dataMenor := $1;
            dataMaior := $2;
        ELSE
            dataMenor := $2;
            dataMaior := $1;
        END IF;
        dias := dataMaior - dataMenor;
        RETURN dias;
    END;
$$ LANGUAGE PLPGSQL;
```

- Procedimento armazenado que retorna a duração de um trimestre em número de dias.

```
CREATE OR REPLACE FUNCTION duracaoTrimestre (INTEGER,
INTEGER) RETURNS INTEGER
AS $$
    DECLARE
        an ALIAS FOR $1;
        nu ALIAS FOR $2;
        dias INTEGER;
        tr trimestre%ROWTYPE;
    BEGIN
        SELECT * INTO tr FROM trimestre WHERE ano=an
AND numero=nu;
        dias :=
diasEntreDatas(tr.dataInicio, tr.dataFim);
        RETURN dias;
    END
$$ LANGUAGE PLPGSQL;
```

- Procedimento armazenado que retorna a quantidade de dias até o final de um trimestre.

```
CREATE OR REPLACE FUNCTION
diasParaFinalTrimestre(INTEGER, INTEGER) RETURNS INTEGER
AS $$
    DECLARE
        an ALIAS FOR $1;
        nu ALIAS FOR $2;
        dias INTEGER;
        hoje DATE;
        tr trimestre%ROWTYPE;
    BEGIN
        hoje := CAST(CURRENT_TIMESTAMP AS DATE);
        SELECT * INTO tr FROM trimestre WHERE ano=an
AND numero=nu;
        IF (tr.dataInicio > hoje ) THEN
            RETURN -1;
        ELSE IF (tr.dataFim < hoje) THEN
            RETURN 0;
        END IF;
        END IF;
        dias := diasEntreDatas(hoje,tr.dataFim);
        RETURN dias;
    END
$$ LANGUAGE PLPGSQL;
```

- Procedimento armazenado que retorna o próximo código da mensagem.

```
CREATE OR REPLACE FUNCTION proximoCodMensagem() RETURNS
mensagem.codMensagem%TYPE
AS $$
    DECLARE
        prx INTEGER;
        cod mensagem.codMensagem%TYPE;
    BEGIN
        SELECT MAX(to_number(SUBSTRING(codMensagem FROM
3)::VARCHAR,'000'))+1
        INTO prx
        FROM MENSAGEM;
        SELECT LPAD(CAST(prx AS VARCHAR),3,'0')
        INTO cod;
        cod := '#D'||cod;
        RETURN cod;
    END
$$ LANGUAGE PLPGSQL;
```

4.1.5 Gatilhos

- Gatilho responsável por enviar mensagem, para o aluno e responsável, sempre que uma avaliação for marcada.

```
CREATE OR REPLACE FUNCTION CriaMensagensAvaliacao()
  RETURNS TRIGGER AS $$
  DECLARE
    nomeAcont acontecimento.nome%TYPE;
    dataAcont acontecimento.data%TYPE;
    horaIni acontecimento.horarioInicio%TYPE;
    localAcont acontecimento.local%TYPE;

    cpfProfessor professor.cpf%TYPE;

    nomeDisc disciplina.nome%TYPE;

    mensagem mensagem.texto%TYPE;
    codMsg mensagem.codMensagem%TYPE;
    cpfAln aluno.cpf%TYPE;
    cpfRsp responsavel.cpf%TYPE;
  BEGIN

    SELECT INTO
      nomeAcont, dataAcont, horaIni, localAcont
      nome, data, horarioInicio, local
    FROM acontecimento WHERE codAcontecimento =
NEW.codAcontecimento;

    SELECT INTO
      cpfProfessor, nomeDisc
      cpf, nome
    FROM diario, professor, disciplina
    WHERE diario.codDiario = NEW.codDiario AND
diario.matProf = professor.matricula
    AND diario.codDisciplina = disciplina.codDisciplina;
    /*Criando mensagem*/
    mensagem := nomeAcont || ' - ' ||
to_char(dataAcont,'DD/MM/YYYY') || ' ' || horaIni || ' - '
|| localAcont ;
    SELECT proximoCodMensagem() INTO codMsg;
    INSERT INTO mensagem VALUES(
      cpfProfessor,
      codMsg,
      mensagem);
    /*Compartilhando com o aluno*/
    FOR cpfAln IN
      SELECT cpfAluno FROM matricula WHERE
```

```

codDiario=NEW.codDiario
    LOOP
        INSERT INTO CompartilhaMensagem
            VALUES (
                cpfProfessor,
                cpfAln,
                codMsg,
                TRUE,
                CURRENT_TIMESTAMP(0),
                NULL);

    END LOOP;
/*Compartilhando com o responsavel*/
FOR cpfRsp IN
    SELECT codResponsavel
    FROM matricula NATURAL JOIN responsabiliza
    WHERE codDiario=NEW.codDiario
    LOOP
        INSERT INTO CompartilhaMensagem
            VALUES (
                cpfProfessor,
                cpfRsp,
                codMsg,
                TRUE,
                CURRENT_TIMESTAMP(0),
                NULL);

    END LOOP;
RETURN NULL;
END
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER InformaAvaliacao
AFTER INSERT ON Avaliacao
FOR EACH ROW
EXECUTE PROCEDURE CriaMensagensAvaliacao();

```

- Gatilho responsável por enviar mensagem, para todas as pessoas sempre que um evento for marcado.

```

CREATE OR REPLACE FUNCTION CriaMensagensEvento()
RETURNS TRIGGER AS $$
DECLARE
    nomeAcont acontecimento.nome%TYPE;
    dataAcont acontecimento.data%TYPE;
    horaIni acontecimento.horarioInicio%TYPE;
    localAcont acontecimento.local%TYPE;

```

```

mensagem mensagem.texto%TYPE;
codMsg mensagem.codMensagem%TYPE;

cpfPess pessoa.cpf%TYPE;

BEGIN
    SELECT INTO
        nomeAcont, dataAcont, horaIni, localAcont
        nome, data, horarioInicio, local
    FROM acontecimento WHERE codAcontecimento =
NEW.codAcontecimento;

    /*Criando mensagem*/
    mensagem := nomeAcont || ' - ' ||
to_char(dataAcont,'DD/MM/YYYY') || ' ' || horaIni || ' - '
|| localAcont ;
    SELECT proximoCodMensagem() INTO codMsg;
    INSERT INTO mensagem VALUES (
        NEW.cpfAdm,
        codMsg,
        mensagem);

    /*Compartilhando com o pessoa*/
    FOR cpfPess IN
        SELECT cpf FROM pessoa
    LOOP
        INSERT INTO CompartilhaMensagem
        VALUES (
            NEW.cpfAdm,
            cpfPess,
            codMsg,
            TRUE,
            CURRENT_TIMESTAMP(0),
            NULL);
    END LOOP;
    RETURN NULL;
END
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER InformaEvento
AFTER INSERT ON Evento
FOR EACH ROW
EXECUTE PROCEDURE CriaMensagensEvento();

```

- Gatilho responsável por modificar o RE do alunos sempre que for cadastrado o resultado de uma avaliação.

```
CREATE OR REPLACE FUNCTION ModificaREALuno()
RETURNS TRIGGER AS $$
DECLARE
    media ResultadoAvaliacao.nota%TYPE;
BEGIN
    SELECT INTO media AVG(nota) FROM ResultadoAvaliacao
    WHERE cpfAluno = NEW.cpfAluno
    GROUP BY cpfAluno;

    UPDATE aluno SET re=media WHERE cpf = NEW.cpfAluno;

    RETURN NULL;
END
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER ResultadoAvaliacao
AFTER INSERT ON ResultadoAvaliacao
FOR EACH ROW
EXECUTE PROCEDURE ModificaREALuno();
```

- Gatilho para informar quando o aluno apresentar frequência inferior a 75%.

```
/*Mensagem quando frequencia ficar abaixo de 75 */
CREATE OR REPLACE FUNCTION InformaFrequencia()
RETURNS TRIGGER AS $$
DECLARE
    cpfProfessor professor.cpf%TYPE;
    nomeDisc disciplina.nome%TYPE;
    mensagem mensagem.texto%TYPE;
    codMsg mensagem.codMensagem%TYPE;
    nomeAluno pessoa.nome%TYPE;
    cpfRsp responsavel.cpf%TYPE;
BEGIN

    SELECT INTO
        nomeAluno
        nome
    FROM pessoa
    WHERE cpf = NEW.cpfAluno;
    SELECT INTO
        cpfProfessor, nomeDisc
        cpf, nome
    FROM diario, professor, disciplina
    WHERE diario.codDiario = NEW.codDiario AND
    diario.matProf = professor.matricula
```

```

        AND diario.codDisciplina = disciplina.codDisciplina;
        /*Criando mensagem*/
        mensagem := 'A frequencia do aluno '||nomeAluno|| '
na disciplina '||nomeDisc||' é de '||
NEW.frequencia||'.';
        SELECT proximoCodMensagem() INTO codMsg;
        INSERT INTO mensagem VALUES(
            cpfProfessor,
            codMsg,
            mensagem);
        /*Compartilhando com o aluno*/

        INSERT INTO CompartilhaMensagem
        VALUES(
            cpfProfessor,
            NEW.cpfAluno,
            codMsg,
            TRUE,
            CURRENT_TIMESTAMP(0),
            NULL);

        /*Compartilhando com o responsavel*/
        FOR cpfRsp IN
            SELECT codResponsavel
            FROM matricula NATURAL JOIN responsabiliza
            WHERE codDiario=NEW.codDiario AND
matricula.cpfAluno=NEW.cpfAluno
        LOOP
            INSERT INTO CompartilhaMensagem
            VALUES(
                cpfProfessor,
                cpfRsp,
                codMsg,
                TRUE,
                CURRENT_TIMESTAMP(0),
                NULL);

        END LOOP;
        RETURN NULL;
    END
    $$ LANGUAGE PLPGSQL;

/*Criando o gatilho quando a frequencia ficar abaixo de
75%*/
CREATE TRIGGER FrequenciaInferior
AFTER INSERT OR UPDATE ON matricula
FOR EACH ROW
WHEN (NEW.frequencia < 0.75)
EXECUTE PROCEDURE InformaFrequencia();

```

4.1.6 Povoando as relações

- **Povoando a relação PESSOA - RESPONSÁVEL**

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES ('Ana Rodrigues', '947.564.740-
63', 'gDNyZQXL', '34.642.816-
6', '10/02/1988', 'feminino', 'Cajazeiras - Paraíba,
Brasil');
```

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES ('Beatriz Pereira', '816.582.500-
36', 'FkT8D2Je', '38.045.510-
9', '04/07/1985', 'feminino', 'Cajazeiras - Paraíba,
Brasil');
```

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES ('Antonella Fernandes', '455.270.350-
11', 'UkRp9t2G', '39.900.321-
6', '23/12/1979', 'feminino', 'Cajazeiras - Paraíba,
Brasil');
```

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES ('Matheus Ricardo Almada', '492.332.324-
84', 'DLqdK7xeNG', '48.684.360-
9', '21/02/1996', 'masculino', 'Cajazeiras - Paraíba,
Brasil');
```

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES ('Theo Henry Campos', '829.340.194-03',
'RoMI1lhQPn', '35.940.274-
4', '06/11/1992', 'masculino', 'Cajazeiras - Paraíba,
Brasil');
```


- **Povoando a relação PESSOA - PROFESSOR**

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Jennifer Amanda Porto','974.336.374-27',
'Ajh5W9Rrpm', '19.067.861-6', '12/12/1981', 'feminino',
'Sousa - Paraíba, Brasil');

INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Henry Murilo Raul de Paula', '490.437.174-79',
'CM9gJC6huG', '13.096.888-2', '09/11/1985', 'masculino',
'Sousa - Paraíba, Brasil');

INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Vitor Guilherme Moura', '979.536.644-02',
'QhgLRfwnHa', '38.528.504-8', '03/10/1985', 'masculino',
'Sousa - Paraíba, Brasil');

INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Mariana Fernanda Souza', '725.430.454-
73', 'Ed01twUIxx', '24.860.270-
6', '20/02/1988', 'feminino', 'Sousa - Paraíba, Brasil');

INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Augusto Mário Almada', '421.019.024-
19', 'wG2PWATc6Y', '10.088.219-
5', '20/12/1988', 'masculino', 'Sousa - Paraíba, Brasil');
```

- **Povoando a relação PESSOA - ADMINISTRADOR**

```
INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
VALUES('Maitê Duarte', '071.053.094-
31', 'hexiL6ltSq', '40.368.856-
5', '19/01/1968', 'feminino', 'Pombal - Paraíba, Brasil');

INSERT INTO
pessoa (nome, cpf, senha, identidade, dataNascimento, sexo, ende
reco)
```

```
VALUES('Raimunda Nogueira','140.261.534-52','taZ40aHcln','28.267.872-4','24/01/1960','feminino','Pombal - Paraíba, Brasil');

INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Alessandra Nascimento','866.110.874-86','wWjo9OyKRq','16.326.035-7','26/06/1972','feminino','Pombal - Paraíba, Brasil');

INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Joaquim Assis','591.310.934-18','mmaKhkhslc','35.784.148-7','02/03/1979','masculino','Pombal - Paraíba, Brasil');

INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Oliver Santos','561.077.524-80','0SJW6RarF6','44.583.476-6','27/12/1981','masculino','Pombal - Paraíba, Brasil');
```

• Povoando a relação PESSOA - ALUNO

```
INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Estella Costa','537.759.154-27','LcfswMbUMi','31.584.762-1','25/09/2003','feminino','Cajazeiras - Paraíba, Brasil');

INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Luana Souza','003.962.124-33','pB6tERtui5','24.046.447-3','01/11/2005','feminino','Cajazeiras - Paraíba, Brasil');

INSERT INTO
pessoa(nome,cpf,senha,identidade,dataNascimento,sexo,ende reco)
VALUES('Carlos','963.234.750-14','5cxuHnEM','39.871.131-8','14/01/2005','masculino','Cajazeiras - Paraíba, Brasil');
```

```

INSERT INTO
pessoa (nome,cpf,senha,identidade,dataNascimento,sexo,ende
reco)
VALUES ('Daniel','358.595.280-19','txzeMrNt','47.919.186-
4','10/03/2005','masculino','Cajazeiras - Paraíba,
Brasil');

INSERT INTO
pessoa (nome,cpf,senha,identidade,dataNascimento,sexo,ende
reco)
VALUES ('Ellen','236.717.500-49','ubhKs5Wy','26.190.992-
7','19/08/2003','feminino','Cajazeiras - Paraíba,
Brasil');

```

- **Povoando a relação RESPONSÁVEL**

```

INSERT INTO responsavel (cpf,enderecoTrabalho)
VALUES ('947.564.740-63','Cajazeiras - Paraíba, Brasil');
INSERT INTO responsavel (cpf,enderecoTrabalho)
VALUES ('816.582.500-36','Cajazeiras - Paraíba, Brasil');
INSERT INTO responsavel (cpf,enderecoTrabalho)
VALUES ('455.270.350-11','Cajazeiras - Paraíba, Brasil');
INSERT INTO responsavel (cpf,enderecoTrabalho)
VALUES ('492.332.324-84','Cajazeiras - Paraíba, Brasil');
INSERT INTO responsavel (cpf,enderecoTrabalho)
VALUES ('829.340.194-03','Cajazeiras - Paraíba, Brasil');

```

- **Povoando a relação ALUNO**

```

INSERT INTO aluno (cpf,matricula,RE,nomeMae,nomePai)
VALUES ('537.759.154-27','2018140001',9.3,'Ana','João');
INSERT INTO aluno (cpf,matricula,RE,nomeMae,nomePai)
VALUES ('003.962.124-
33','2018140002',8.3,'Beatriz','Márcio');
INSERT INTO aluno (cpf,matricula,RE,nomeMae,nomePai)
VALUES ('963.234.750-
14','2018140003',9.1,'Antonella','André');
INSERT INTO aluno (cpf,matricula,RE,nomeMae,nomePai)
VALUES ('358.595.280-
19','2018140004',7.6,'Marcella','Matheus');
INSERT INTO aluno (cpf,matricula,RE,nomeMae,nomePai)
VALUES ('236.717.500-
49','2018140005',8.4,'Edilsa','Theo');

```

- **Povoando a relação ADMINISTRADOR**

```

INSERT INTO administrador (cpf,cargo,setor)
VALUES ('071.053.094-31','Diretora','Administrativo');

```

```

INSERT INTO administrador(cpf,cargo,setor)
VALUES('140.261.534-52','Cordenadora','Administrativo');
INSERT INTO administrador(cpf,cargo,setor)
VALUES('561.077.524-80','Técnico de T.I.','T.I.');
```

```

INSERT INTO administrador(cpf,cargo,setor)
VALUES('866.110.874-86','Vice-
Diretora','Administrativo');
INSERT INTO administrador(cpf,cargo,setor)
VALUES('591.310.934-18','Coordenador','Administrativo');
```

• Povoando a relação PROFESSOR

```

INSERT INTO professor(cpf,matricula,titulacao)
VALUES('974.336.374-27','2018120001','mestrado');
```

```

INSERT INTO professor(cpf,matricula,titulacao)
VALUES('490.437.174-79','2018120002','especialização');
```

```

INSERT INTO professor(cpf,matricula,titulacao)
VALUES('979.536.644-02','2018120003','mestrado');
```

```

INSERT INTO professor(cpf,matricula,titulacao)
VALUES('725.430.454-73','2018120004','doutorado');
```

```

INSERT INTO professor(cpf,matricula,titulacao)
VALUES('421.019.024-19','2018120005','doutorado');
```

• Povoando a relação ACONTECIMENTO

```

INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#001','Dia das
Mães','13/05/2018','17:30','21:00','Ginásio');
```

```

INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#002','Dia dos
Pais','08/08/2018','18:00','22:00','Ginásio');
```

```

INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#003','Dia do
Estudante','11/08/2018','10:45','11:45','Ginásio');
```

```

INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#004','Natal','22/12/2018','18:00','20:00','Praça
Padre Rolim');
```

```

INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#005','Dia dos
```

```

Professores','14/10/2018','19:00','23:00','Ginásio');
INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#006','Avaliação de
Matemática','13/09/2018','11:00','12:05','Sala-01');
INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#007','Avaliação de
Português','18/09/2018','07:00','09:15','Sala-02');
INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#008','Avaliação de Geografia',
'20/09/2018','08:30','09:15','Sala-01');
INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#009','Avaliação de
Inglês','12/09/2018','09:30','11:00','Sala-02');
INSERT INTO
acontecimento(codAcontecimento,nome,data,horarioInicio,ho
rarioFim,local)
VALUES('#010','Avaliação de
Espanhol','14/09/2018','10:45','12:05','Sala 01');

```

• Povoando a relação TURMA

```

INSERT INTO turma(codTurma, serie, sala)
VALUES('#B001','1A-EM','Sala 01');
INSERT INTO turma(codTurma, serie, sala)
VALUES('#B002','8A-EFII','Sala 02');
INSERT INTO turma(codTurma, serie, sala)
VALUES('#B003','8B-EFII','Sala 03');
INSERT INTO turma(codTurma, serie, sala)
VALUES('#B004','8C-EFII','Sala 04');
INSERT INTO turma(codTurma, serie, sala)
VALUES('#B005','1B-EM','Sala 05');

```

• Povoando a relação TRIMESTRE

```

INSERT INTO trimestre(ano, numero, dataInicio, dataFim)
VALUES(2018,2,'05/09/2018','11/09/2018');
INSERT INTO trimestre(ano, numero, dataInicio, dataFim)
VALUES(2018,1,'22/01/2018','08/05/2018');
INSERT INTO trimestre(ano, numero, dataInicio, dataFim)
VALUES(2018,3,'09/12/2018','07/12/2018');
INSERT INTO trimestre(ano, numero, dataInicio, dataFim)

```

```
VALUES (2019,1,'21/01/2019','07/05/2018');
INSERT INTO trimestre(ano, numero, dataInicio, dataFim)
VALUES (2019,2,'05/08/2019','10/09/2019');
```

• Povoando a relação ORGANIZATRIMESTRE

```
INSERT INTO organizatrimestre(ano,numero,codTurma)
VALUES (2018,2,'#B001');
INSERT INTO organizatrimestre(ano,numero,codTurma)
VALUES (2018,2,'#B002');
INSERT INTO organizatrimestre(ano,numero,codTurma)
VALUES (2018,1,'#B003');
INSERT INTO organizatrimestre(ano,numero,codTurma)
VALUES (2018,1,'#B004');
INSERT INTO organizatrimestre(ano,numero,codTurma)
VALUES (2018,3,'#B005');
```

• Povoando a relação MENSAGEM

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('947.564.740-63','#D001','Olá, mundo!');
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('974.336.374-27','#D002','Tudo tem que acabar
eventualmente. Senão nada poderia começar.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('071.053.094-31','#D003','Não faz sentido ser
adulto se por vezes não se pode ser criança.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('140.261.534-52','#D004','Inteligência superior e
crueldade insensível simplesmente não vão bem juntas');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('358.595.280-19','#D005','Existe sempre uma
solução.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('236.717.500-49','#D006','Nunca esteja certo de
tudo. É um sinal de fraqueza.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('490.437.174-79','#D007','Violência não acaba com
a violência, expande-a.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('591.310.934-18','#D008','A dor e a perda nos
define tanto quanto a felicidade e o amor.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('963.234.750-14','#D009','Algumas coisas valem a
pena, mesmo que partam seu coração.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
VALUES ('829.340.194-03','#D010','Amor não é sentimento...
Amor é uma promessa.');
```

```
INSERT INTO mensagem(cpf,codMensagem,texto)
```

```
VALUES('071.053.094-31','#D011','A aluna Maité Duarte  
apresentou compormatamento inadequado durante a aula.');
```

- **Povoando a relação COMPARTILHAMENSAGEM**

```
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('947.564.740-63','236.717.500-  
49','#D001',FALSE,'11/10/2018 11:29:29','11/10/2018  
13:47:34');  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('974.336.374-27','490.437.174-  
79','#D002',FALSE,'12/10/2018 15:11:23','12/10/2018  
17:20:00');  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('071.053.094-31','591.310.934-  
18','#D003',FALSE,'13/10/2018 16:00:02','13/10/2018  
18:59:37');  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('140.261.534-52','963.234.750-  
14','#D004',FALSE,'14/10/2018 09:45:36','14/10/2018  
09:47:11');  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('358.595.280-19','829.340.194-  
03','#D005',FALSE,'15/10/2018 07:43:34','15/10/2018  
08:53:11');  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('071.053.094-31','947.564.740-  
63','#D011',FALSE,'10/09/2018 07:43:34',null);  
INSERT INTO  
compartilhamensagem(cpfEnvia,cpfRecebe,codMensagem,visibi  
lidade,dataHoraEnvio,dataHoraVisualizacao)  
VALUES('829.340.194-03','816.582.500-  
36','#D010',FALSE,'20/09/2018 07:43:34','20/09/2018  
08:43:34');
```

- **Povoando a relação RESPONDEMENSAGEM**

```
INSERT INTO respondemensagem(codMensagem,codResposta)
VALUES ('#D001', '#D002');
INSERT INTO respondemensagem(codMensagem,codResposta)
VALUES ('#D003', '#D004');
INSERT INTO respondemensagem(codMensagem,codResposta)
VALUES ('#D005', '#D006');
INSERT INTO respondemensagem(codMensagem,codResposta)
VALUES ('#D007', '#D008');
INSERT INTO respondemensagem(codMensagem,codResposta)
VALUES ('#D009', 'D009');
```

- **Povoando a relação RESULTADOAVALIACAO**

```
INSERT INTO
resultadoavaliacao(cpfAluno,codAcontecimento,nota)
VALUES ('537.759.154-27', '#006', 8.2);
INSERT INTO
resultadoavaliacao(cpfAluno,codAcontecimento,nota)
VALUES ('003.962.124-33', '#007', 9.3);
INSERT INTO
resultadoavaliacao(cpfAluno,codAcontecimento,nota)
VALUES ('963.234.750-14', '#008', 9.9);
INSERT INTO
resultadoavaliacao(cpfAluno,codAcontecimento,nota)
VALUES ('358.595.280-19', '#009', 9.1);
INSERT INTO
resultadoavaliacao(cpfAluno,codAcontecimento,nota)
VALUES ('236.717.500-49', '#010', 10.0);
```

- **Povoando a relação DISCIPLINA**

```
INSERT INTO disciplina(nome, cargaHoraria, codDisciplina)
VALUES ('Matemática', 40, '#C001');
INSERT INTO disciplina(nome, cargaHoraria, codDisciplina)
VALUES ('Português', 40, '#C002');
INSERT INTO disciplina(nome, cargaHoraria, codDisciplina)
VALUES ('Geografia', 40, '#C003');
INSERT INTO disciplina(nome, cargaHoraria, codDisciplina)
VALUES ('Inglês', 20, '#C004');
INSERT INTO disciplina(nome, cargaHoraria, codDisciplina)
VALUES ('Espanhol', 20, '#C005');
```

- **Povoando a relação DIARIO**

```
INSERT INTO diario(codDiario, atualizado, matProf,
codTurma, codDisciplina)
VALUES ('#AA01', TRUE, '2018120001', '#B001', '#C001');
```



```

INSERT INTO diario(codDiario, atualizado, matProf,
codTurma, codDisciplina)
VALUES('#AA02',TRUE, '2018120002','#B002','#C002');
INSERT INTO diario(codDiario, atualizado, matProf,
codTurma, codDisciplina)
VALUES('#AA03',TRUE,'2018120003','#B003','#C003');
INSERT INTO diario(codDiario, atualizado, matProf,
codTurma, codDisciplina)
VALUES('#AA04',TRUE,'2018120004','#B004','#C004');
INSERT INTO diario(codDiario, atualizado, matProf,
codTurma, codDisciplina)
VALUES('#AA05',TRUE,'2018120005','#B005','#C005');

```

• Povoando a relação EVENTO

```

INSERT INTO
evento(codAcontecimento,coordenador,descrição,cpfAdm)
VALUES('#001','Mário','Dia dos mães proposto para
enaltecer a figura materna.','561.077.524-80');
INSERT INTO
evento(codAcontecimento,coordenador,descrição,cpfAdm)
VALUES('#002','Andressa','Dia dos pais proposto para
enaltecer a figura paterna.','561.077.524-80');
INSERT INTO
evento(codAcontecimento,coordenador,descrição,cpfAdm)
VALUES('#003','Anália','Dia dos estudantes proposto para
enaltecer os alunos.','561.077.524-80');
INSERT INTO
evento(codAcontecimento,coordenador,descrição,cpfAdm)
VALUES('#004','Natal','Auto de natal para comemorar o
evento cristão.','071.053.094-31');
INSERT INTO
evento(codAcontecimento,coordenador,descrição,cpfAdm)
VALUES('#005','Dia dos Professores','Jantar de
comemoração para o dia dos professores.','591.310.934-
18');

```

• Povoando a relação AVALIACAO

```

INSERT INTO
avaliacao(codDiario,numero,tipo,codAcontecimento)
VALUES('#AA01',1,'Prova','#006');
INSERT INTO
avaliacao(codDiario,numero,tipo,codAcontecimento)
VALUES('#AA02',1,'Prova','#007');
INSERT INTO
avaliacao(codDiario,numero,tipo,codAcontecimento)
VALUES('#AA03',1,'Prova','#008');
INSERT INTO

```

```

avaliacao(codDiario,numero,tipo,codAcontecimento)
VALUES('#AA04',1,'Prova','#009');
INSERT INTO
avaliacao(codDiario,numero,tipo,codAcontecimento)
VALUES('#AA05',1,'Prova','#010');

```

- **Povoando a relação RESPONSABILIZA**

```

INSERT INTO responsabiliza(codResponsavel, cpfAluno,
parentesco)
VALUES('947.564.740-63','537.759.154-27','Mãe');
INSERT INTO responsabiliza(codResponsavel, cpfAluno,
parentesco)
VALUES('816.582.500-36','003.962.124-33','Mãe');
INSERT INTO responsabiliza(codResponsavel, cpfAluno,
parentesco)
VALUES('455.270.350-11','963.234.750-14','Mãe');
INSERT INTO responsabiliza(codResponsavel, cpfAluno,
parentesco)
VALUES('492.332.324-84','358.595.280-19','Pai');
INSERT INTO responsabiliza(codResponsavel, cpfAluno,
parentesco)
VALUES('829.340.194-03','236.717.500-49','Pai');

```

- **Povoando a relação MATRICULA**

```

INSERT INTO matricula(cpfAluno,codDiario,frequencia)
VALUES('537.759.154-27','#AA01', 91);
INSERT INTO matricula(cpfAluno,codDiario,frequencia)
VALUES('003.962.124-33','#AA02', 93);
INSERT INTO matricula(cpfAluno,codDiario,frequencia)
VALUES('963.234.750-14','#AA02', 94);
INSERT INTO matricula(cpfAluno,codDiario,frequencia)
VALUES('358.595.280-19','#AA02', 79);
INSERT INTO matricula(cpfAluno,codDiario,frequencia)
VALUES('236.717.500-49','#AA01', 88);

```

- **Povoando a relação CONTEUDO-AVALIACAO**

```

INSERT INTO conteudoavaliacao(codAcontecimento,conteudo)
VALUES('#006','Álgebra');
INSERT INTO conteudoavaliacao(codAcontecimento,conteudo)
VALUES('#007','Figuras de Linguagem');
INSERT INTO conteudoavaliacao(codAcontecimento,conteudo)
VALUES('#008','Relevo');
INSERT INTO conteudoavaliacao(codAcontecimento,conteudo)
VALUES('#009','Present Continuous');
INSERT INTO conteudoavaliacao(codAcontecimento,conteudo)
VALUES('#010','Préterito pluscuamperfecto');

```

- **Povoando a relação EMAILPESSOA**

```
INSERT INTO emailpessoa(cpf,email)
VALUES('947.564.740-63','ana@gmail.com');
INSERT INTO emailpessoa(cpf,email)
VALUES('816.582.500-36','beatriz@hotmail.com');
INSERT INTO emailpessoa(cpf,email)
VALUES('974.336.374-27','jennifer@hotmail.com');
INSERT INTO emailpessoa(cpf,email)
VALUES('561.077.524-80','oliver@hotmail.com');
INSERT INTO emailpessoa(cpf,email)
VALUES('003.962.124-33','luana@gmail.com');
```

- **Povoando a relação TELEFONEPESSOA**

```
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('947.564.740-63','(83) 9954-1234');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('816.582.500-36','(83) 9876-5678');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('974.336.374-27','(83) 4209-1105');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('561.077.524-80','(83) 5797-2171');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('003.962.124-33','(83) 8739-5343');
```

- **Povoando a relação TELEFONERESPONSAVEL**

```
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('947.564.740-63','(83) 9954-3695');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('816.582.500-36','(83) 9876-6013');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('455.270.350-11','(83) 4209-2929');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('492.332.324-84','(83) 5797-2984');
INSERT INTO telefonepessoa(cpf,telefone)
VALUES('829.340.194-03','(83) 8739-5493');
```

4.1.7 Consultas

- Listar os professores com as suas respectivas disciplinas e carga horária.

```
SELECT p.nome as Professor, dp.nome as Disciplina,
dp.cargaHoraria
FROM pessoa p ,professor pf, diario d, disciplina dp
WHERE (p.cpf=pf.cpf) and (d.matProf = pf.matricula) and
(dp.codDisciplina = d.codDisciplina);
```

- Listar todos os telefones do responsável pela aluno Estella Costa.

```
SELECT telefone
FROM visaoalunopessoa a, responsabiliza r,
telefoneresponsavel tr
WHERE nome='Estella Costa' and r.cpfaluno = a.cpf and
tr.cpf=r.codResponsavel;
```

- Listar mensagens que não foram visualizadas e seus destinatários.

```
SELECT p.nome AS Destinatario, m.texto as Mensagem
FROM mensagem m, compartilhamensagem c, pessoa p
WHERE c.codMensagem = m.codMensagem AND c.cpfRecebe =
p.cpf and dataHoraVisualizacao IS NULL;
```

- Listar mensagens destinadas a Carlos que já foram visualizadas.

```
SELECT m.texto AS Mensagem
FROM pessoa p, mensagem m, compartilhamensagem c
WHERE p.nome='Carlos' AND p.cpf=c.cpfRecebe AND
c.codMensagem=m.codMensagem AND c.dataHoraVisualizacao IS
NOT NULL;
```

- Listar mensagens com seus respectivos remetente e destinatário que contenha a palavra “Amor”.

```
SELECT r.nome AS Remetente, d.nome AS Destinatario,
m.texto AS Mensagem
FROM mensagem m, compartilhamensagem c, pessoa d, pessoa
r
WHERE m.texto LIKE '%Amor%' AND m.codMensagem =
c.codMensagem AND d.cpf=c.cpfRecebe AND r.cpf=c.cpfEnvia;
```

- Listar professores que residem em Sousa.

```
SELECT nome AS Professor
FROM visaoProfessorpessoa
```

```
WHERE endereco LIKE '%Sousa%';
```

- Listar alunos ordenando de acordo com seu respectivo rendimento escolar.

```
SELECT nome AS Aluno, re AS RendimentoEscolar
FROM visaoalunopessoa
ORDER BY re DESC;
```

- Listagem de alunos em ordem alfabética.

```
SELECT nome AS Alunos
FROM VisaoAlunoPessoa
ORDER BY nome;
```

- Listar alunos matriculados na disciplina “Português”.

```
SELECT a.matricula, a.nome AS Aluno
FROM visaoAlunoPessoa a JOIN matricula m ON
(a.cpf=m.cpfAluno)
WHERE m.codDiario IN (
SELECT codDiario
FROM visaoDiario
WHERE disciplina ='Português');
```

- Listar responsáveis que possuam mais de um telefone do trabalho.

```
SELECT nome AS Responsavel
FROM visaoResponsavelPessoa*
WHERE cpf IN
(SELECT cpf
FROM telefoneResponsavel
GROUP BY cpf
HAVING COUNT(*) > 1);
```

- Listar professores que possuam avaliações do tipo prova.

```
SELECT professor
FROM visaoDiario vd
WHERE EXISTS
(SELECT *
FROM Avaliacao a
WHERE a.codDiario = vd.codDiario AND a.tipo='Prova');
```

- Listar professores que não possuam email.

```
SELECT p.nome AS Professor
FROM visaoProfessorPessoa p
WHERE NOT EXISTS (
```

```
SELECT *
FROM emailPessoa e
WHERE e.cpf=p.cpf);
```

- Listar alunos que assistem aula nas salas “Sala 02” ou “Sala 03”.

```
(SELECT a.matricula, a.nome As Aluno
FROM (visaoAlunoPessoa a JOIN matricula m ON
(a.cpf=m.cpfAluno)) JOIN visaoDiario d ON
(m.codDiario=d.codDiario)
WHERE d.sala ='Sala 02') UNION (SELECT a.matricula,
a.nome As Aluno
FROM (visaoAlunoPessoa a JOIN matricula m ON
(a.cpf=m.cpfAluno)) JOIN visaoDiario d ON
(m.codDiario=d.codDiario)
WHERE d.sala ='Sala 03');
```

- Listar a quantidade de mensagem que cada pessoa compartilhou ordenando-as por número de mensagem.

```
SELECT p.nome , COUNT(c.codMensagem) AS QtdMensagem
FROM pessoa p LEFT JOIN compartilhamensagem c
ON(p.cpf=c.cpfEnvia)
GROUP BY p.cpf, p.nome
ORDER BY COUNT(c.codMensagem) DESC;
```

- Listar diários que não marcaram avaliação.

```
(SELECT CodDiario
FROM diario)
EXCEPT
(SELECT CodDiario
FROM avaliacao);
```

- Verificar quantos dias foram computados e quantos faltam para o final do trimestre 2018.3.

```
SELECT
duracaoTrimestre(2018,3) - diasParaFinalTrimestre(2018,3)
AS DiasComputados,
diasParaFinalTrimestre(2018,3) AS DiaParaFinal;
```

- Listar as mensagens visualizadas e a quantidade de dias entre o envio e sua visualização.

```
SELECT r.Nome AS Remetente,d.Nome As Destinatario,
m.texto AS Mensagem,
```

```

diasEntreDatas (CAST(c.dataHoraEnvio AS
DATE),CAST(c.dataHoraVisualizacao AS DATE)) AS DemoraDias
FROM pessoa r, pessoa d, compartilhaMensagem c, mensagem
m
WHERE r.cpf = c.cpfEnvia AND d.cpf=c.cpfRecebe AND
c.codMensagem=m.codMensagem AND c.dataHoraVisualizacao IS
NOT NULL
ORDER BY diasEntreDatas (CAST(c.dataHoraEnvio AS
DATE),CAST(c.dataHoraVisualizacao AS DATE)) DESC

```

- Listar o período do dia em que existe o maior volume de mensagens.

```

SELECT COUNT(*) AS QuantMensagens,
CASE
WHEN (CAST(datahoraenvio AS TIME)<CAST('06:00:00' AS
TIME))
THEN 'Noite'
WHEN (CAST(datahoraenvio AS TIME)<CAST('12:00:00' AS
TIME))
THEN 'Manhã'
WHEN (CAST(datahoraenvio AS TIME)<CAST('18:00:00' AS
TIME))
THEN 'Tarde'
ELSE 'Noite'
END AS HORARIO
FROM compartilhamensagem
GROUP BY HORARIO
ORDER BY COUNT(*) DESC;

```

- Relacionar os 3 primeiro colocados no simulado de geografia.

```

SELECT acontecimento.nome, visaoAlunoPessoa.nome AS
Aluno, nota, *
FROM (avaliacao NATURAL JOIN resultadoavaliacao) NATURAL
JOIN acontecimento, VisaoAlunoPessoa

WHERE tipo='Simulado' AND acontecimento.nome = 'Simulado
de Geografia' AND cpfaluno = cpf
ORDER BY codacontecimento, nota DESC
LIMIT(3);

```