



UAST
Unidade Acadêmica
de Serra Talhada - PE
Desde 2006



**PROJETO DE DISCIPLINA:
PROJETO DE BANCO DE DADOS**

Vinicius Alexandre Feitoza

**SERRA TALHADA, PE
[NOVEMBRO/2019]**

Vinicius Alexandre Feitoza

Projeto desenvolvido para efetivação da [1ª ou 2ª ou 3ª] etapa apresentado para avaliação na Disciplina Projeto de Banco de Dados Relacionais ministrada pelo Prof. Hidelberg Oliveira, período letivo 2019.2.

PROJETO DE BANCO DE DADOS

01. DEFINIÇÃO DA PROBLEMÁTICA (DOMÍNIO DA APLICAÇÃO)

O Sistema acadêmico ARGUS é voltado para escolas do Ensino Fundamental e Médio. O objetivo do sistema é de prover o acompanhamento do aluno em todas as fases de seu crescimento estudantil, desde o acompanhamento de notas, pagamentos e pedagógico.

02. DEFINIÇÃO DE REQUISITOS

2.1. Requisitos Funcionais:

Identificação:	[RF01] Manter Pessoa
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca do Pessoas no banco de dados.</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF02] Manter Matricula
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca da Matricula do aluno no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF03] Manter Disciplina
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca das Disciplinas no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF04] Manter Acompanhamento Pedagógico
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca do Acompanhamento Pedagógico, no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF05] Manter Notas
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca das Notas do aluno no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF06] Cadastrar Administração
Descrição:	<i>Cadastrar perfis de usuários como Administradores do Sistema.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RF07] Realizar Login
Descrição:	<i>O Sistema solicita o Login e a Senha ao Usuário para a validação de acesso.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF08] Resetar Senha
Descrição:	<i>O Usuário poderá alterar sua senha login, sempre que esteja logado no sistema.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RF09] Quadro de Solicitações
Descrição:	<i>O quadro de solicitações apresenta as solicitações dos demais usuários ao administrador principal do sistema.</i>
Prioridade:	() Essencial () Importante (x) Desejável

Identificação:	[RF10] Gerar Log
Descrição:	<i>O sistema deverá para cada ação de inserção, atualização e remoção feita no sistema gerar um log de alteração.</i>
Prioridade:	() Essencial () Importante (x) Desejável

Identificação:	[RF11] Manter Solicitação
Descrição:	<i>O usuário poderá realizar uma solicitação de reset de senha ao ADM.</i>
Prioridade:	(x) Essencial () Importante (x) Desejável

Identificação:	[RF12] Manter Perfil Curricular
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca dos perfis curriculares de cada turma no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RF13] Manter Turma
Descrição:	<i>Manter inclui o cadastro, a remoção, a atualização e a busca da turma no banco de dados.</i>
Prioridade:	(x) Essencial () Importante () Desejável

2.2. Requisitos Não-Funcionais:

Identificação:	[RNF01] Disponibilizar Dados
Tipo:	Disponibilidade
RF Relacionado:	Nenhum

Descrição:	O sistema funcionará 24 horas por dia.
Prioridade:	(x) Essencial () Importante (x) Desejável

Identificação:	[RNF02] Utilização da Linguagem Java
Tipo:	Padrão
RF Relacionado:	Nenhum
Descrição:	O projeto foi desenvolvido com a linguagem Java.
Prioridade:	() Essencial () Importante (x) Desejável

Identificação:	[RNF03] Acessar Dados
Tipo:	Segurança
RF Relacionado:	Todos
Descrição:	O sistema é controlado por login e senha, permitindo definir diferentes níveis de acesso ao sistema através de permissões de acordo com o perfil do usuário, logo os dados estão protegidos contra modificações e exclusões não autorizadas.
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	[RNF04] Armazenamento de Dados
Tipo:	Integridade
RF Relacionado:	RF01, RF02, RF03, RF04, RF05, RF06, RF11, RF12, RF13
Descrição:	Os dados inseridos, alterados ou removidos permanecem na situação desejada pelos operadores.
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RNF05] Backup
Tipo:	Segurança
RF Relacionado:	Nenhum
Descrição:	O sistema deve ser configurado para fazer o backup do banco de dados uma vez por dia, forçando o usuário a fazer o backup que sempre que alcançar uma determinada hora do dia.
Prioridade:	() Essencial (x) Importante (x) Desejável

Identificação:	[RNF06] Padrão de Projeto
Tipo:	Implementação
RF Relacionado:	Nenhum
Descrição:	O sistema deve usar os padrões de projeto MVC e DAO.
Prioridade:	() Essencial (x) Importante (x) Desejável

2.3. Requisitos de Domínio (se existir):

Identificação:	[RNF01] Matrícula
Descrição:	<ul style="list-style-type: none"> • Um aluno não pode pular de anos. Uma vez matriculado, deve seguir o fluxo. • Considere que não há transferidos. O sistema não pode permitir matrícula em anos anteriores não cursados ou reprovados.

	<ul style="list-style-type: none"> • Um aluno pode se matricular em um currículo anual progressivo, composto por: <ul style="list-style-type: none"> • EF: Anos Iniciais (anos): 1º ao 5º • EF: Anos Finais (anos): 6º ao 9º • EM: (anos): 1º ao 3º. • Um responsável financeiro deve obrigatoriamente possuir CPF. • Se o aluno for maior de idade, ele pode ser responsável financeiro por si próprio, mas o inverso deve ser proibido. • No momento da matrícula, o sistema deve gerar as 12 parcelas referentes aos 12 meses do aluno, conforme tabela a seguir: <ul style="list-style-type: none"> • EF-Anos Iniciais: 400,00 • EF-Anos Finais: 600,00 • EM: 800,00
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RD02] Senhas
Descrição:	<ul style="list-style-type: none"> • O campo de senha dos usuários deve aceitar caracteres alfanuméricos e numéricos, com tamanho entre 6 a 20, com distinção de caracteres maiúsculos de minúsculos. A senha deverá ser criptografada no banco de dados. • Os usuários são cadastrados por grupos, dentro do sistema, definidos pelo administrador. • Cada usuário do sistema tem uma senha de acesso, com permissão de alterá-la. • No momento do cadastramento do usuário, deve ser gerada uma senha padrão. • Caso o usuário perca a senha, deverá solicitar pelo sistema ao administrador, que deverá receber um aviso no sistema da solicitação.
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RD03] Controle dos dados por usuário
Descrição:	<ul style="list-style-type: none"> • Os funcionários da secretaria deveram cadastrar as médias dos alunos e as disciplinas. • A Direção não pode fazer alterações, somente visualização. • A Coordenação Pedagógica somente deve cadastrar os dados de acompanhamento. • Somente o administrador cadastra os funcionários (em todos os níveis).
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RD04] Disciplinas
-----------------------	--------------------

Descrição:	<ul style="list-style-type: none"> As notas das disciplinas devem ser cadastradas dentro do intervalo de 0.0 à 10.0, com até um algarismo de ponto flutuante. Para ser aprovado: Por Média: precisa obter média parcial ≥ 7.0 Na Final: precisa obter média geral ≥ 5.0
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RD05] Currículo
Descrição:	<ul style="list-style-type: none"> O currículo pode ser baseado em bimestres ou trimestres, configurável pelo sistema. Se bimestres: Cada aluno em cada disciplina deve obter: <ul style="list-style-type: none"> Nota de 4 bimestres. Média Parcial. Cálculo: média aritmética dos 4 bimestres Média para aprovação: 7.0 Prova final Média Geral. Se o aluno for aprovado por média, esta média é igual à parcial. Se foi para final: (Média Parcial + Prova Final)/2 Se trimestres: <ul style="list-style-type: none"> Cada aluno em cada disciplina deve obter: Nota de 3 trimestres. Média Parcial. Cálculo: média aritmética dos 3 bimestres Média para aprovação: 7.0 Prova final Média Geral. Se o aluno for aprovado por média, esta média é igual à parcial. Se foi para final: (Média Parcial + Prova Final)/2 Para ambos (bi ou trimestres): Status dos alunos: <ul style="list-style-type: none"> Por disciplina: <ul style="list-style-type: none"> AM = aprovado por média AP = aprovado RP = reprovado. Por currículo: <ul style="list-style-type: none"> APROVADO REPROVADO

Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	[RD06] Acompanhamento pedagógico
Descrição:	<ul style="list-style-type: none"> • O sistema também deve fornecer o acompanhamento pedagógico dos alunos. • O sistema deve emitir o: <ul style="list-style-type: none"> • Relatório do acompanhamento por aluno. • Relatório de acompanhamento por profissional (relação de nome de alunos e datas de atendimento).
Prioridade:	() Essencial (x) Importante () Desejável

03. DIAGRAMA ENTIDADE-RELACIONAMENTO

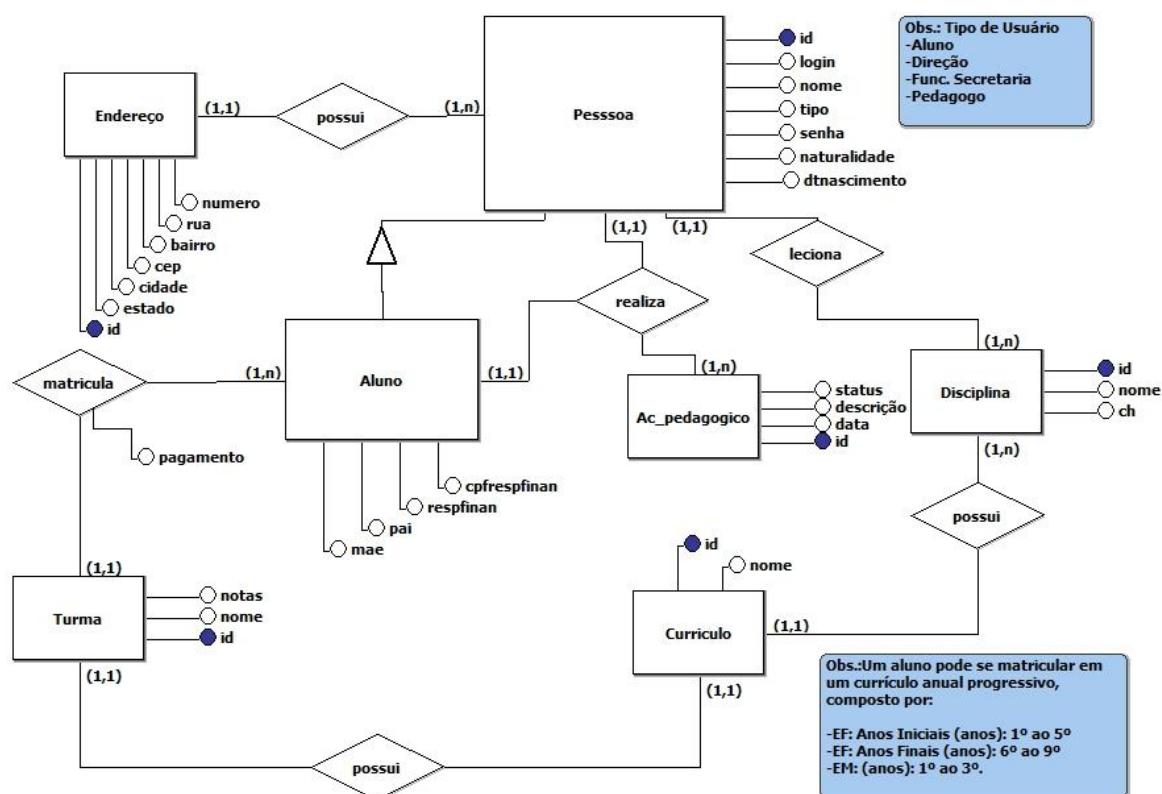


Figura 1 - Diagrama Entidade Relacionamento

04. MAPEAMENTO DO ESQUEMA CONCEITUAL PARA O ESQUEMA LÓGICO-RELACIONAL

Regra 1: Mapeamentos das entidades fortes

Endereço (**id**, estado, cidade, bairro, cep, rua)

Pessoa (**id**, nome, naturalidade, dta_nasc, tipo, login, senha)

Disciplina (**id**, nome, ch, *id_pessoa*, *id_curriculo*)

id_curriculo referência entidade Currículo.

id_pessoa referência entidade Pessoa.

Turma (**id**, nome, notas, *id_curriculo*)

id_curriculo referência entidade Currículo.

Ac_pedagogico (**id**, data, descrição, status, *id_aluno*, *id_pessoa*)

id_aluno referência entidade Aluno.

id_pessoa referência entidade Pessoa.

Aluno (**id**, pai, mae, resp_financ, cpf_resp_finance, *id_pessoa*, *id_turma*)

id_turma referência entidade Turma.

Id_pessoa referência entidade Pessoa.

id_endereco referência entidade Endereço.

Curriculo (**id**, nome)

Matricula (**id**, valortotal, valor mensalidade, *id_aluno*, *id_turma*)

id_aluno referência entidade Aluno.

id_turma referência entidade Turma.

Solicitação (**id**, nome, status, tiposolicitacao)

Regra 2: Mapeamentos das entidades fracas

Neste projeto não serão considerados entidades fracas.

Regra 3: Mapeamento de Relacionamentos Binários Um para Um (1:1)

Neste projeto não houve nenhum relacionamento binário (1:1) entre as entidades existentes no DER

Regra 4: Mapeamento de Relacionamentos Binários Um para Muitos (1:N)

Pessoa (**id**, nome, naturalidade, dta_nasc., tipo, user, senha, *id_endereco*)

id_endereco referência entidade Endereço.

Disciplina (**id**, nome, ch, *id_pessoa*)

id_pessoa faz referência a entidade Pessoa do tipo Professor

Turma (**id**, nome, notas, *id_curriculo*)

id_curriculo faz referência a entidade Curriculo

Ac_pedagogico (**id**, data, desc, status, *id_pessoa*, *id_aluno*)

id_pessoa faz referência a entidade Aluno

id_pessoa faz referência a entidade Pessoa do tipo Pedagogo

Matricula (**id**, *id_aluno*, *id_turma*, pagamento)

id_aluno faz referência a entidade Aluno

id_turma faz referência a entidade Turma

Regra 5: Mapeamento de Relacionamentos Binários Muitos para Muitos (N:M)

Obs: não houve nenhum relacionamento binário entre as entidades existentes no DER

Regra 6: Mapeamento de atributos multivalorados

Obs: não houve nenhum atributo multivalorado entre as entidades existentes no DER

Regra 7: Mapeamento de Relacionamentos N-ários

Obs: não houve nenhum relacionamento N-ário entre as entidades existentes no DER

05. DICIONÁRIO DE DADOS

Pessoa			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação da Pessoa	Long	<ul style="list-style-type: none">• Primary Key• Auto Increment• Not Null
Nome	Atributo que representa o nome da Pessoa.	String	<ul style="list-style-type: none">• Not Null
Login	Atributo que representa o login da Pessoa.	String	<ul style="list-style-type: none">• Not Null• Unique
Senha	Atributo que representa a senha da Pessoa.	String	<ul style="list-style-type: none">• Not Null
Naturalidade	Atributo que representa naturalidade da Pessoa.	String	
Dta_nasc	Atributo que representa a data de nascimento da Pessoa.	Date	
Tipo	Atributo que representar o tipo de Pessoa	String	<ul style="list-style-type: none">• Not Null• Unique
Id_endereco	Atributo que referencia o código de identificação do Endereço da Pessoa.	Long	<ul style="list-style-type: none">• Foreign Key (Tabela Endereço)

Endereço			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação do Endereço.	Long	<ul style="list-style-type: none">• Primary Key• Auto Increment• Not Null
Estado	Atributo que representa o nome do estado do Endereço.	String	<ul style="list-style-type: none">• Not Null
Cidade	Atributo que representa o nome da cidade do Endereço.	String	<ul style="list-style-type: none">• Not Null
Bairro	Atributo que representa o nome do bairro do Endereço	String	
Cep	Atributo que representa o CEP do	String	

	Endereço.		
Rua	Atributo que representa o nome da rua do Endereço	String	

Aluno			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação do Aluno.	Long	<ul style="list-style-type: none"> Primary Key Auto Increment Not Null
Mae	Atributo que representa o nome da mãe do Aluno.	String	<ul style="list-style-type: none"> Not Null
Pai	Atributo que representa o nome do pai do Aluno.	String	<ul style="list-style-type: none"> Not Null
resp_financ	Atributo que representa o responsável financeiro do Aluno.	String	<ul style="list-style-type: none"> Not Null
cpf_resp_financ e	Atributo que representa o CPF do responsável financeiro do Aluno.	String	<ul style="list-style-type: none"> Not Null
Id_pessoa	Atributo que referencia código de identificação da Pessoa do Aluno.	Long	<ul style="list-style-type: none"> Foreign Key (Tabela Pessoa)

Matricula			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação da Matricula.	Long	<ul style="list-style-type: none"> Primary Key Auto Increment Not Null
ValorTotal	Atributo que representa o valor total da Matricula.	Double	<ul style="list-style-type: none"> Not Null
ValorMensalidad e	Atributo que representa o valor mensalidade da Matricula.	Double	<ul style="list-style-type: none"> Not Null
Id_aluno	Atributo que referencia código de identificação do Aluno na Matricula.	Long	<ul style="list-style-type: none"> Foreign Key (Tabela Aluno)
Id_turma	Atributo que referencia código de identificação da Turma do Aluno na Matricula.	Long	<ul style="list-style-type: none"> Foreign Key (Tabela Turma)

Turma			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação da Turma	Long	<ul style="list-style-type: none"> Primary Key

			<ul style="list-style-type: none"> • Auto Increment • Not Null
Nome	Atributo que representa o nome da Turma.	String	<ul style="list-style-type: none"> • Not Null
Notas	Atributo que representa as notas da Turma.	String	<ul style="list-style-type: none"> • Not Null
Id_curriculo	Atributo que referência código de identificação do Currículo da Turma.	Long	<ul style="list-style-type: none"> • Foreign Key (Tabela Currículo)

Currículo			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação do Currículo.	Long	<ul style="list-style-type: none"> • Primary Key • Auto Increment • Not Null
Nome	Atributo que representa o nome do Currículo.	String	<ul style="list-style-type: none"> • Not Null

Disciplina			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação do Disciplina.	Long	<ul style="list-style-type: none"> • Primary Key • Auto Increment • Not Null
Nome	Atributo que representa o nome do Disciplina.	String	<ul style="list-style-type: none"> • Not Null
Ch	Atributo que representa a carga-horária da Disciplina.	Int	<ul style="list-style-type: none"> • Not Null
Id_curriculo	Atributo que referência código de identificação do currículo da Disciplina.	Long	<ul style="list-style-type: none"> • Foreign Key (Tabela Currículo)
Id_pessoa	Atributo que referência código de identificação da Pessoa da Disciplina.	Long	<ul style="list-style-type: none"> • Foreign Key (Tabela Pessoa)

Ac_pedagogico			
Atributo	Descrição	Tipo de Dado	Restrições
Id	Atributo que representa o código de identificação do Ac_pedagogico.	Long	<ul style="list-style-type: none"> • Primary Key • Auto Increment • Not Null
Data	Atributo que representa a data do Ac_pedagogico.	Date	<ul style="list-style-type: none"> • Not Null
Status	Atributo que representa o status do Ac_pedagogico.	String	<ul style="list-style-type: none"> • Not Null

Desc	Atributo que representa a descrição do Ac_pedagogico.	String	<ul style="list-style-type: none"> • Not Null
Id_aluno	Atributo que referência código de identificação do Aluno do Ac_pedagogico.	Long	<ul style="list-style-type: none"> • Foreign Key (Tabela Aluno)
Id_pessoa	Atributo que referência código de identificação da Pessoa do Ac_pedagogico.	Long	<ul style="list-style-type: none"> • Foreign Key (Tabela Pessoa)

05. SCRIPTS SQL

5.1 DDL

Endereco
<pre>CREATE TABLE public.endereco (id integer NOT NULL, bairro character varying(50) NOT NULL, cep character varying(15) NOT NULL, cidade character varying(80) NOT NULL, estado character varying(3) NOT NULL, numero character varying(30) NOT NULL, rua character varying(80) NOT NULL, CONSTRAINT endereco_pkey PRIMARY KEY (id)) WITH (OIDS=FALSE); ALTER TABLE public.endereco OWNER TO postgres;</pre>

Pessoa
<pre>CREATE TABLE public.pessoa (id integer NOT NULL, dtanascimento timestamp without time zone, login character varying(20), naturalidade character varying(20), nome character varying(80) NOT NULL, senha character varying(20), tipo character varying(20), endereco_id integer NOT NULL, CONSTRAINT pessoa_pkey PRIMARY KEY (id), CONSTRAINT fk150oo5tcc0u0n3i2k10pnfh8 FOREIGN KEY (endereco_id) REFERENCES public.endereco (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_fw0821si73kteng9pk5n6wwk3 UNIQUE (endereco_id))</pre>

```

WITH (
  OIDS=FALSE
);
ALTER TABLE public.pessoa
  OWNER TO postgres;

```

Aluno

```

CREATE TABLE public.aluno
(
  id integer NOT NULL,
  dtanascimento timestamp without time zone,
  login character varying(20),
  naturalidade character varying(20),
  nome character varying(80) NOT NULL,
  senha character varying(20),
  tipo character varying(20),
  endereco_id integer NOT NULL,
  cpfrespfinan character varying(20),
  mae character varying(100),
  pai character varying(100),
  respfinan character varying(100),
  turma_id integer,
  CONSTRAINT aluno_pkey PRIMARY KEY (id),
  CONSTRAINT fk_1lyv1j9qqbmi98wjamep6c2w FOREIGN KEY (endereco_id)
    REFERENCES public.endereco (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fkc2jbyom2ws00xwire0hms5bny FOREIGN KEY (turma_id)
    REFERENCES public.turma (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT uk_1lyv1j9qqbmi98wjamep6c2w UNIQUE (endereco_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.aluno
  OWNER TO postgres;

```

Matricula

```

CREATE TABLE public.matricula
(
  id integer NOT NULL,
  valormensalidade double precision,
  valortotal double precision,
  aluno_id integer,
  turma_id integer,
  CONSTRAINT matricula_pkey PRIMARY KEY (id),

```

```

CONSTRAINT fk3rpufqk0vd9j50nvu24h3vg1u FOREIGN KEY (aluno_id)
REFERENCES public.aluno (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk5cbvj745fx2o25diw6b4ccr11 FOREIGN KEY (turma_id)
REFERENCES public.turma (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.matricula
  OWNER TO postgres;

```

Turma

```

CREATE TABLE public.turma
(
  id integer NOT NULL,
  nome character varying(100),
  notas double precision,
  curriculo_id integer NOT NULL,
  CONSTRAINT turma_pkey PRIMARY KEY (id),
  CONSTRAINT fk34xmlat8wcrbyqt5q1ol68gtk FOREIGN KEY (curriculo_id)
REFERENCES public.curriculo (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT uk_jyduy8lkxaboql9jxgywr5gkk UNIQUE (curriculo_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.turma
  OWNER TO postgres;

```

Curriculo

```

CREATE TABLE public.curriculo
(
  id integer NOT NULL,
  nome character varying(100),
  CONSTRAINT curriculo_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.curriculo
  OWNER TO postgres;

```

Disciplina

```

CREATE TABLE public.disciplina
(
  id integer NOT NULL,

```



```

ch integer,
nome character varying(100),
curriculo_id integer,
pessoa_id integer,
CONSTRAINT disciplina_pkey PRIMARY KEY (id),
CONSTRAINT fkqu0gqrt0g1acye4rt8k6bi8yo FOREIGN KEY (curriculo_id)
    REFERENCES public.curriculo (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.disciplina
    OWNER TO postgres;

```

Ac_pedagogico

```

CREATE TABLE public.acpedagogico
(
    id integer NOT NULL,
    data date,
    descricao character varying(1000),
    status character varying(100),
    aluno_id integer,
    pessoa_id integer,
    CONSTRAINT acpedagogico_pkey PRIMARY KEY (id),
    CONSTRAINT fkq8m4mw8gssjtakakqg7h4c39x FOREIGN KEY (aluno_id)
        REFERENCES public.aluno (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.acpedagogico
    OWNER TO postgres;

```

LOG

```

CREATE TABLE LOG (
    id SERIAL PRIMARY KEY,
    data DATE NOT NULL,
    alteracao VARCHAR(50) NOT NULL,
    autor VARCHAR(50) NOT NULL,
    tabela VARCHAR(50) NOT NULL,
    anterior VARCHAR(255) NOT NULL
);

```

5.2. DML

Login (busca usuário por login e senha)
SELECT * FROM Pessoa where login like 'login' and senha like 'senha'

Buscar Disciplina (busca os dados cadastrados de disciplina)
select * from disciplina

Buscar Aluno a Partir da Turma (busca os alunos matriculado em uma turma)
SELECT a FROM Alunosturma a WHERE a.turma_nome like 'nome'

Buscar Acompanhamento Pedagógico (busca os dados cadastrados do acompanhamento pedagógico)
select * from acpedagogico

Buscar Matricula (busca os dados cadastrados da matricula)
select * from matricula

Buscar Pessoa (busca os dados cadastrados de pessoa)
select * from pessoa

Buscar Nome por Tabela (busca os dados cadastrados da tabela)
SELECT a FROM TABELA a WHERE nome like 'nome'

Buscar por Pedagogo na Tabela Pessoa (busca os dados cadastrados da tabela)
"SELECT nome FROM Pessoa a WHERE tipo like 'Pedagogo'"

06. ANEXOS

6.1 Diagrama de Caso de Uso

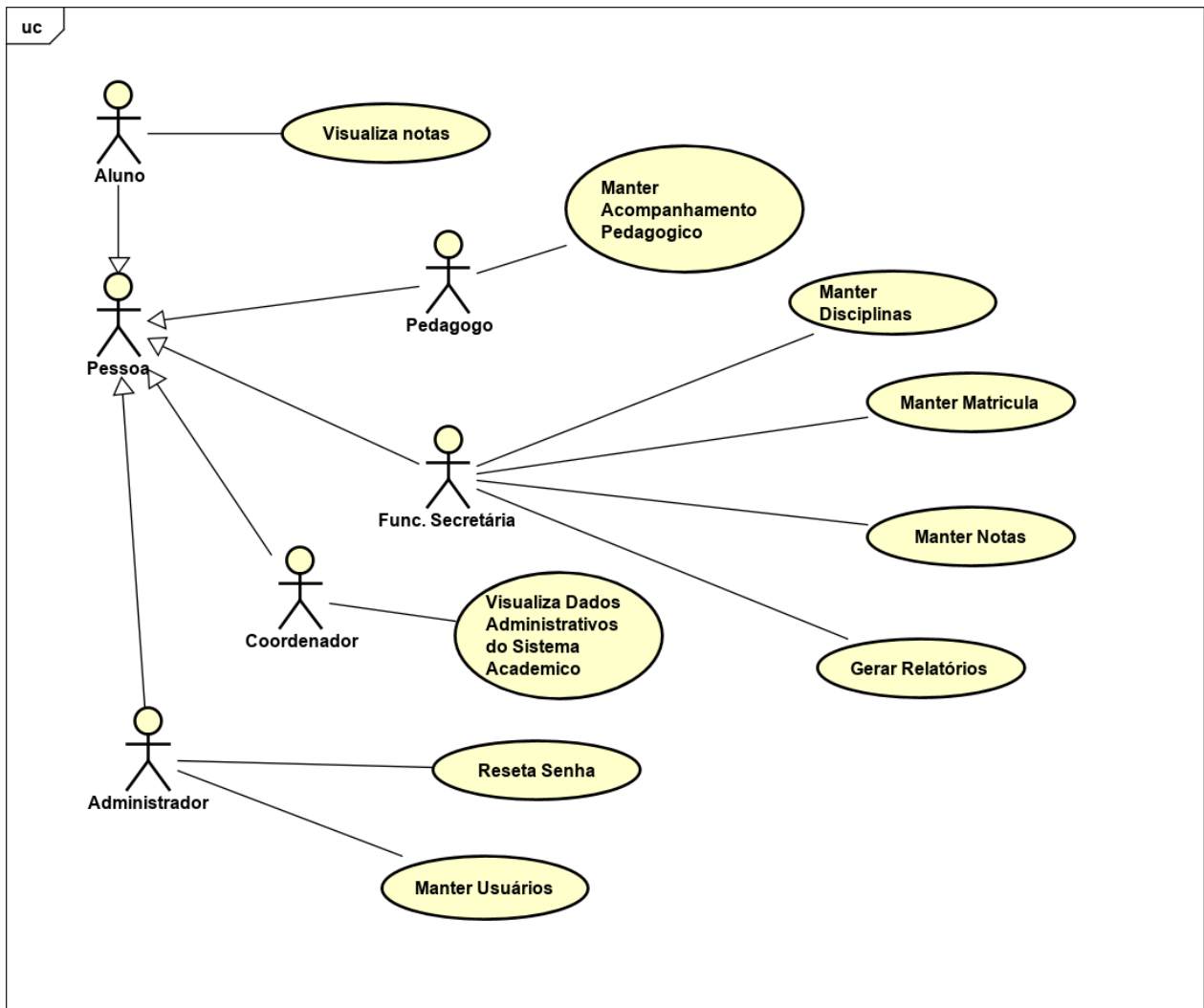


Figura 2 - Diagrama de Caso de Uso