

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO

Curso: Tecnólogo em Sistemas para Internet

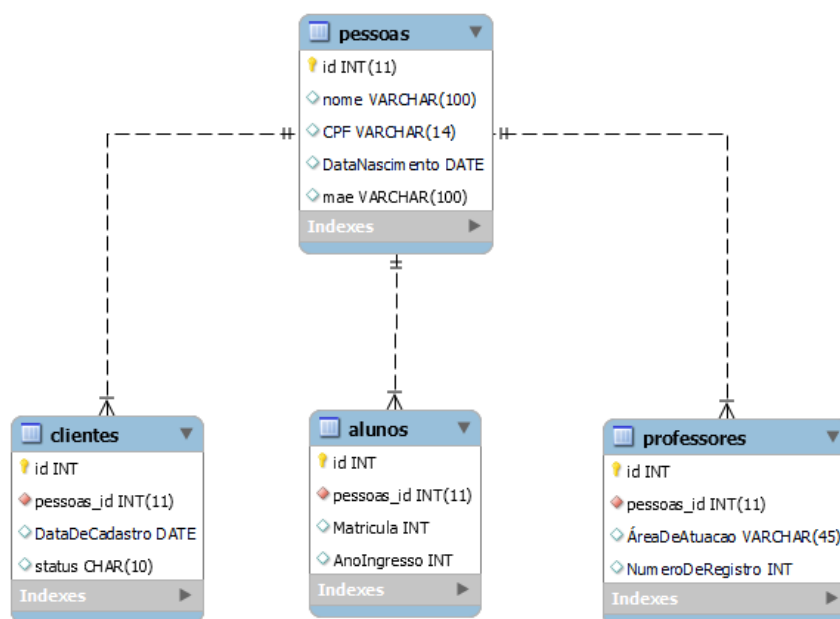
Aluno: Elizeu Cordeiro do Amaral Vailant

Disciplina: Administração de Banco de Dados

Atividade:

Construção de Modelos em Frameworks Modernos para Sistemas
para a Internet

Diagrama de Banco de Dados



Implementar o diagrama nas específicas linguagens de programação:

Laravel:

Migration: Pessoas

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
class CreatePessoasTable extends Migration
```

```
{
```

```
    public function up()
```

```
    {
```

```
        Schema::create('pessoas', function (Blueprint $table) {
```

```
            $table->id();
```

```
            $table->string('nome', 100);
```

```
            $table->string('CPF', 14)->unique();
```

```
            $table->date('DataNascimento');
```

```
            $table->string('mae', 100);
```

```
            $table->timestamps();
```

```
        });
```

```
    }
```

```
    public function down()
```

```
    {
```

```
        Schema::dropIfExists('pessoas');
```

```
    }
```

```
}
```

Migration: Clientes, Alunos, Professores

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateClientesAlunosProfessoresTables extends Migration
{
    public function up()
    {
        Schema::create('clientes', function (Blueprint $table) {
            $table->id();
            $table->foreignId('pessoas_id')->constrained('pessoas')->onDelete('cascade');
            $table->date('DataDeCadastro');
            $table->string('status', 10);
            $table->timestamps();
        });

        Schema::create('alunos', function (Blueprint $table) {
            $table->id();
            $table->foreignId('pessoas_id')->constrained('pessoas')->onDelete('cascade');
            $table->integer('Matricula');
            $table->integer('AnoIngresso');
            $table->timestamps();
        });

        Schema::create('professores', function (Blueprint $table) {
            $table->id();
            $table->foreignId('pessoas_id')->constrained('pessoas')->onDelete('cascade');
            $table->string('ÁreaDeAtuacao', 45);
```

```

        $table->integer('NumeroDeRegistro');

        $table->timestamps();

    });
}

public function down()
{
    Schema::dropIfExists('clientes');
    Schema::dropIfExists('alunos');
    Schema::dropIfExists('professores');
}
}

```

Django:

Models: Pessoas, Clientes, Alunos, Professores

```
from django.db import models
```

```
class Pessoa(models.Model):
```

```

    nome = models.CharField(max_length=100)
    cpf = models.CharField(max_length=14, unique=True)
    data_nascimento = models.DateField()
    mae = models.CharField(max_length=100)

```

```

    def __str__(self):
        return self.nome

```

```
class Cliente(models.Model):
```

```

    pessoa = models.ForeignKey(Pessoa, on_delete=models.CASCADE)
    data_de_cadastro = models.DateField()
    status = models.CharField(max_length=10)

```

```
def __str__(self):  
    return f"Cliente: {self.pessoa.nome}"
```

```
class Aluno(models.Model):  
    pessoa = models.ForeignKey(Pessoa, on_delete=models.CASCADE)  
    matricula = models.IntegerField()  
    ano_ingresso = models.IntegerField()
```

```
def __str__(self):  
    return f"Aluno: {self.pessoa.nome}"
```

```
class Professor(models.Model):  
    pessoa = models.ForeignKey(Pessoa, on_delete=models.CASCADE)  
    area_de_atuacao = models.CharField(max_length=45)  
    numero_de_registro = models.IntegerField()
```

```
def __str__(self):  
    return f"Professor: {self.pessoa.nome}"
```

Ruby on Rails:

Migration: Pessoas

```
class CreatePessoas < ActiveRecord::Migration[6.1]  
    def change  
        create_table :pessoas do |t|  
            t.string :nome, limit: 100  
            t.string :cpf, limit: 14, unique: true  
            t.date :data_nascimento  
            t.string :mae, limit: 100  
  
            t.timestamps
```

```
    end
  end
end
```

Migration: Clientes, Alunos, Professores

```
class Pessoa < ApplicationRecord
  has_many :clientes
  has_many :alunos
  has_many :professores
end
```

```
class Cliente < ApplicationRecord
  belongs_to :pessoa
end
```

```
class Aluno < ApplicationRecord
  belongs_to :pessoa
end
```

```
class Professor < ApplicationRecord
  belongs_to :pessoa
end
```

Express.js:

Modelo: Pessoa

```
const { Model, DataTypes } = require('sequelize');
const sequelize = require('../config/database');
```

```
class Pessoa extends Model {}
```

```
Pessoa.init({
  nome: {
    type: DataTypes.STRING(100),
    allowNull: false
  },
  CPF: {
    type: DataTypes.STRING(14),
    unique: true,
    allowNull: false
  },
  DataNascimento: {
    type: DataTypes.DATE,
    allowNull: false
  },
  mae: {
    type: DataTypes.STRING(100),
    allowNull: false
  }
}, {
  sequelize,
  modelName: 'Pessoa'
});
```

```
module.exports = Pessoa;
```

Modelos: Cliente, Aluno, Professor

```
const { Model, DataTypes } = require('sequelize');
const sequelize = require('../config/database');
const Pessoa = require('../Pessoa');

class Cliente extends Model {}
```

```
Cliente.init({
  DataDeCadastro: {
    type: DataTypes.DATE,
    allowNull: false
  },
  status: {
    type: DataTypes.STRING(10),
    allowNull: false
  }
}, {
  sequelize,
  modelName: 'Cliente'
});
```

```
Cliente.belongsTo(Pessoa, { foreignKey: 'pessoas_id' });
```

```
class Aluno extends Model {}
```

```
Aluno.init({
  Matricula: {
    type: DataTypes.INTEGER,
    allowNull: false
  },
  AnoIngresso: {
    type: DataTypes.INTEGER,
    allowNull: false
  }
}, {
  sequelize,
  modelName: 'Aluno'
```



```
});
```

```
Aluno.belongsTo(Pessoa, { foreignKey: 'pessoas_id' });
```

```
class Professor extends Model {}
```

```
Professor.init({  
  ÁreaDeAtuacao: {  
    type: DataTypes.STRING(45),  
    allowNull: false  
  },  
  NumeroDeRegistro: {  
    type: DataTypes.INTEGER,  
    allowNull: false  
  }  
}, {  
  sequelize,  
  modelName: 'Professor'  
});
```

```
Professor.belongsTo(Pessoa, { foreignKey: 'pessoas_id' });
```

```
module.exports = { Cliente, Aluno, Professor };


---


```