

Print da classe **já existente** “Usuário”:

```
model.py > ...
1  """Este módulo contém a classe que representa o Usuario
2
3  Autor: Gustavo Guerreiro, Johannes Wachholz José e Lucas Vargas
4  """
5
6  from config import *
7
8  class Usuario(db.Model):
9      # atributos do usuário
10     id = db.Column(db.Integer, primary_key=True)
11     estado = db.Column(db.String(254))
12     cidade = db.Column(db.String(254))
13     endereco = db.Column(db.String(254))
14     complemento = db.Column(db.String(254))
15     cep = db.Column(db.String(254))
16     telefone = db.Column(db.String(254))
17     email = db.Column(db.String(254))
18     senha = db.Column(db.String(254))
19     data_surgimento = db.Column(db.String(254))
20
21     # discriminador
22     type = db.Column(db.String(50))
23
24     # definições de mapeamento da classe mãe
25     __mapper_args__ = {
26         'polymorphic_identity': 'usuario',
27         'polymorphic_on': type
28     }
29
30     def __str__(self):
31         return f'id = {self.id}, {self.estado}, {self.cidade}, '+\
32             f'{self.endereco}, {self.complemento}, {self.cep}, '+\
33             f'{self.telefone}, {self.email}, {self.email}, '+\
34             f'{self.data_surgimento}'
35
36     def json(self):
37         return {
38             "id" : self.id,
39             "estado" : self.estado,
40             "cidade" : self.cidade,
41             "endereco" : self.endereco,
42             "complemento" : self.complemento,
43             "cep" : self.cep,
44             "telefone" : self.telefone,
45             "email" : self.email,
46             "senha" : self.senha,
47             "data_surgimento" : self.data_surgimento
48         }
49
```

Print da classe nova “Paciente”:

```
"""Este módulo contém a classe que representa um paciente
No arquivo encontra-se a implementação de uma classe: Paciente.
Autor: Johannes Wachholz José.
"""

from config import *
from model import Usuario

class Paciente(Usuario):
    """Classe que representa um Paciente no site
    """
    # atributos exclusivos do Paciente
    id = db.Column(db.Integer, db.ForeignKey('usuario.id'), primary_key=True)
    nome_completo = db.Column(db.String(254))
    sexo = db.Column(db.String(254))
    cpf = db.Column(db.Integer())
    alergias = db.Column(db.String(254))

    __mapper_args__={
        'polymorphic_identity' : 'paciente',
    }

    def __str__(self):
        return super().__str__() + f', {self.nome_completo}, '+\
            f'{self.sexo}, {self.cpf}, '+\
            f'{self.alergias}'

    def json(self):
        return super().json() | {
            "nome_completo" : self.nome_completo,
            "sexo" : self.sexo,
            "cpf" : self.cpf,
            "alergias" : self.alergias
        }
```

Print do código de **teste** da classe nova “Paciente”:

```
# comando para remover arquivo banco de dados caso já exista
if __name__ == "__main__":
    if os.path.exists(arquivobd):
        os.remove(arquivobd)

    # comando para criar tabela (classe usuario)
    db.create_all()

    # criar objetos (na memória, sem persistência)
    p1 = Paciente(estado="SC", cidade="Pomerode", endereco="Rua Hermann Weege",
complemento="casa", cep="89107-000", telefone="992922070", email="joao8856@email.com",
senha="senha123", data_surgimento="11/06/2002", nome_completo="João Machado", sexo="Masculino",
cpf=16465232154, alergias="poeira")

    p2 = Paciente(estado="SC", cidade="Blumenau", endereco="Rua Itajaí",
complemento="Apartamento", cep="12512000", telefone="994210500", email="amanda5521@gmail.com",
senha="amanda2251", data_surgimento="21/04/1997", nome_completo="Amanda da Silva", sexo="Feminino",
cpf=52145561244, alergias="polém")

    # para tornar os objetos persistentes
    db.session.add(p1)
    db.session.add(p2)
    db.session.commit()

    # exibir paciente
    print(p1.json())
    print(p2)
```

Print do **teste** da classe nova “Paciente”:

```
{'id': 1, 'estado': 'SC', 'cidade': 'Pomerode', 'endereco': 'Rua Hermann Weege', 'complement
o': 'casa', 'cep': '89107-000', 'telefone': '992922070', 'email': 'joao8856@email.com', 'sen
ha': 'senha123', 'data_surgimento': '11/06/2002', 'nome_completo': 'João Machado', 'sexo': '
Masculino', 'cpf': 16465232154, 'alergias': 'poeira'}
id = 2, SC, Blumenau, Rua Itajaí, Apartamento, 12512000, 994210500, amanda5521@gmail.com, am
anda5521@gmail.com, 21/04/1997, Amanda da Silva, Feminino, 52145561244, polém
```

Print do **JSON** da classe nova "Paciente":

```
C:\Users\Johannes>curl localhost:5000/listar_usuarios
[
  {
    "alergias": "poeira",
    "cep": "89107-000",
    "cidade": "Pomerode",
    "complemento": "casa",
    "cpf": 16465232154,
    "data_surgimento": "11/06/2002",
    "email": "joao8856@email.com",
    "endereco": "Rua Hermann Weege",
    "estado": "SC",
    "id": 1,
    "nome_completo": "Jo\u00e3o Machado",
    "senha": "senha123",
    "sexo": "Masculino",
    "telefone": "992922070"
  },
  {
    "alergias": "pol\u00e9m",
    "cep": "12512000",
    "cidade": "Blumenau",
    "complemento": "Apartamento",
    "cpf": 52145561244,
    "data_surgimento": "21/04/1997",
    "email": "amanda5521@gmail.com",
    "endereco": "Rua Itaja\u00ed",
    "estado": "SC",
    "id": 2,
    "nome_completo": "Amanda da Silva",
    "senha": "amanda2251",
    "sexo": "Feminino",
    "telefone": "994210500"
  }
]
```

LINK GITHUB: <https://github.com/GuerreiroG/SitePlataformaMedica>