

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     # driscriminador
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** “Entidade”:

```
entidade.py M X
entidade.py > ...
1  """Este módulo contém a classe que representa uma entidade médica
2
3  Autor: Lucas Vargas.
4  """
5  from config import *
6  from model import Usuario
7
8  class Entidade(Usuario):
9      """Classe que representa uma entidade médica no site
10     """
11     # atributos da entidade médica
12     id = db.Column(db.Integer, db.ForeignKey('usuario.id'), primary_key=True)
13     nome_fantasia = db.Column(db.String(254))
14     razao_social = db.Column(db.String(254))
15     numero_funcionarios = db.Column(db.String(254))
16     tipo_instituicao = db.Column(db.String(254))
17     cnpj = db.Column(db.Integer())
18
19     # definindo indentidade polimórfica que ficará armazenada na classe pai
20     # no campo type
21     __mapper_args__ = {
22         'polymorphic_identity' : 'entidade',
23     }
24
25     # para printar as informações do objeto
26     def __str__(self):
27         return super().__str__() + f', {self.nome_fantasia}, '+\
28             f'{self.razao_social}, {self.numero_funcionarios}, '+\
29             f'{self.tipo_instituicao}, {self.cnpj}'
30
31     # método para dar return nas informações no padrão json
32     def json(self):
33         return super().json() | {
34             "nome_fantasia" : self.nome_fantasia,
35             "razao_social" : self.razao_social,
36             "numero_funcionarios" : self.numero_funcionarios,
37             "tipo_instituicao" : self.tipo_instituicao,
38             "cnpj" : self.cnpj,
39         }
40
```

Print do código de **teste** da classe nova "Entidade":

```
41 # comando para remover arquivo banco de dados caso já exista
42 if __name__ == "__main__":
43     if os.path.exists(arquivobd):
44         os.remove(arquivobd)
45
46     # comando para criar tabela (classe usuario)
47     db.create_all()
48
49     # criar objetos (na memória, sem persistência)
50     p1 = Entidade(estado="SC", cidade="Blumenau", endereco="Rua isadora",
51                 complemento="Fachada branca", cep="89037-255", telefone="992922070",
52                 email="lucas@email.com", senha="123", data_surgimento="10/10/2020",
53                 nome_fantasia="Santa Isabel", razao_social="NomeTeste",
54                 numero_funcionarios="40", tipo_instituicao="EIRELI", cnpj=32323)
55
56     p2 = Entidade(estado="PR", cidade="Itaporoba", endereco="Rua Barelo",
57                 complemento="Rua Roxa", cep="32076-454", telefone="85920132",
58                 email="sabrino@email.com", senha="543", data_surgimento="21/12/2020",
59                 nome_fantasia="Santo Agostinho", razao_social="NomeTeste",
60                 numero_funcionarios="63", tipo_instituicao="Sociedade Simples Pura",
61                 cnpj=344443)
62
63     # Torna os objetos persistentes
64     db.session.add(p1)
65     db.session.add(p2)
66     db.session.commit()
67
68     # exibe os objetos de Entidade
69     print("-----")
70     print(p1)
71     print(p1.json())
72     print("-----")
73     print(p2)
74     print(p2.json())
75     print("-----")
76
```

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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + - [ ] [X] ^ X
```

```
dule PSReadLine'.  
  
PS C:\xampp\htdocs\SitePlataformaMedica\Backend> & C:/Users/lucas/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/xampp/ht  
docs/SitePlataformaMedica/Backend/Entidade.py  
-----TESTE P1-----  
id = 1, SC, Blumenau, Rua isadora, Fachada branca, 89037-255, 992922070, lucas@email.com, lucas@email.com, 10/10/2020, Santa Isa  
bel, NomeTeste, 40, EIRELI, 32323  
---  
{'id': 1, 'estado': 'SC', 'cidade': 'Blumenau', 'endereco': 'Rua isadora', 'complemento': 'Fachada branca', 'cep': '89037-255',  
'telefone': '992922070', 'email': 'lucas@email.com', 'senha': '123', 'data_surgimento': '10/10/2020', 'nome_fantasia': 'Santa Is  
abel', 'razao_social': 'NomeTeste', 'numero_funcionarios': '40', 'tipo_instituicao': 'EIRELI', 'cnpj': 32323}  
-----TESTE P2-----  
id = 2, PR, Itaporoba, Rua Barelo, Rua Roxa, 32076-454, 85920132, sabrinc@email.com, sabrinc@email.com, 21/12/2020, Santo Agosti  
nho, NomeTeste, 63, Sociedade Simples Pura, 344443  
---  
{'id': 2, 'estado': 'PR', 'cidade': 'Itaporoba', 'endereco': 'Rua Barelo', 'complemento': 'Rua Roxa', 'cep': '32076-454', 'telef  
one': '85920132', 'email': 'sabrinc@email.com', 'senha': '543', 'data_surgimento': '21/12/2020', 'nome_fantasia': 'Santo Agostin  
ho', 'razao_social': 'NomeTeste', 'numero_funcionarios': '63', 'tipo_instituicao': 'Sociedade Simples Pura', 'cnpj': 344443}  
-----  
PS C:\xampp\htdocs\SitePlataformaMedica\Backend>
```

Print do **JSON** da classe nova “Entidade”:

```

C:\Users\lucas>curl localhost:5000/listar_usuarios
[
  {
    "cep": "89037-255",
    "cidade": "Blumenau",
    "cnpj": 32323,
    "complemento": "Fachada branca",
    "data_surgimento": "10/10/2020",
    "email": "lucas@email.com",
    "endereco": "Rua isadora",
    "estado": "SC",
    "id": 1,
    "nome_fantasia": "Santa Isabel",
    "numero_funcionarios": "40",
    "razao_social": "NomeTeste",
    "senha": "123",
    "telefone": "992922070",
    "tipo_instituicao": "EIRELI"
  },
  {
    "cep": "32076-454",
    "cidade": "Itaporoba",
    "cnpj": 344443,
    "complemento": "Rua Roxa",
    "data_surgimento": "21/12/2020",
    "email": "sabrino@email.com",
    "endereco": "Rua Barelo",
    "estado": "PR",
    "id": 2,
    "nome_fantasia": "Santo Agostinho",
    "numero_funcionarios": "63",
    "razao_social": "NomeTeste",
    "senha": "543",
    "telefone": "85920132",
    "tipo_instituicao": "Sociedade Simples Pura"
  }
]
C:\Users\lucas>

```

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