

Init_py

#Run app by 'run.py' file

*from flask import Flask from flask_sqlalchemy import
SQLAlchemy from flask_script import Manager from
flask_migrate import Migrate, MigrateCommand*

*app = Flask(__name__)
app.config.from_object('config')*

*db = SQLAlchemy(app)
migrate = Migrate(app, db)*

*manager = Manager(app)
manager.add_command('db', MigrateCommand)*

Config_py

*import os.path basedir =
os.path.abspath(os.path.dirname(__file__))*

DEBUG = True

*SQLALCHEMY_DATABASE_URI = 'sqlite:/// ' + os.path.join(basedir, 'storage.db')
SQLALCHEMY_TRACK_MODIFICATIONS = True*

Cliente_py

"""

Módulo cliente -

Classe Cliente -

Atributos:

_id - chave - informado
_nome - cliente - informado
_codigo - codigo - informado
_cnpjcpf - cnpj ou cpf - informado
_tipo - tipo do cliente - informado
(Pessoa Fisica ou Juridica)

"""

from tipocliente import TipoCliente

from app import db

```
class Cliente(db.Model):  __tablename__ =  
"TB_CLIENTE"  id = db.Column(db.Integer,  
primary_key=True)  nome =  
db.Column(db.String)  codigo =  
db.Column(db.String, unique=True)  cnpjcpf =  
db.Column(db.String, unique=True)
```

```
def __init__(self, id, nome, codigo, cnpjcpf, tipo):  
    self._id = id  
    self._nome = nome
```

```
self._codigo = codigo
self._cnpjcpf = cnpjcpf
self._tipo = tipo
```

```
def str(self):
    string="\nId={4} Codigo={2} Nome={3} CNPJ/CPF={1} Tipo={0}".format(self._tipo,
self._cnpjcpf, self._codigo, self._nome, self._id)
    return string
```

```
if __name__ == '__main__':
    cliente=Cliente(1, "Francisco", 100, '200.100.345-34', TipoCliente.PESSOA_FISICA)
    print(cliente.str())
```

"""

Módulo itemnotafiscal

Classe ItemNotaFiscal

Atributos :

id - informado

sequencial - informado

quantidade - informado

produto - informado

valor - calculado.

"""

from produto import Produto

from app import db

Notafiscal_py

```

class ItemNotaFiscal(db.Model): __tablename__ = "TB_ITEM_NF" id =
db.Column(db.Integer, primary_key=True) id_notafiscal =
db.Column(db.Integer, db.ForeignKey("TB_NOTA_FISCAL.id")) sequencial =
db.Column(db.String)

quantidade = db.Column(db.Integer) produto = db.Column(db.String,
db.ForeignKey("TB_PRODUTO.codigo")) descricao = db.Column(db.String,
db.ForeignKey("TB_PRODUTO.descricao")) valorUnitario = db.Column(db.Float,
db.ForeignKey("TB_PRODUTO.valorUnitario")) valorItem = db.Column(db.Float)

nota_fiscal = db.relationship('NotaFiscal', foreign_keys=id) produto
= db.relationship('Produto', foreign_keys=codigo) descricao =
db.relationship('Produto', foreign_keys=descricao) valor_unitario =
db.relationship('Produto', foreign_keys=valorUnitario)

def __init__(self, id, sequencial, quantidade, produto):
    self._id=id    self._sequencial=sequencial
self._quantidade=quantidade    self._produto=produto
self._descricao=self._produto.getDescricao()
self._valorUnitario=self._produto.getValorUnitario()
self._valorItem=float(self._quantidade * self._valorUnitario)

def str(self):
    string="\nId={5} Sequencial={4} Quantidade={3} Produto={2} Valor Unitario={1} Valor
Item={0}".format(self._valorItem,
                                self._valorUnitario,
self._descricao,
self._quantidade,
self._sequencial,
self._id)
    return string

```

```
if __name__ == '__main__':  
    produto = Produto(1,100,'Arroz', 5.5)  
    item=ItemNotaFiscal(1, 1, 12, produto)  
    print(item.str())
```

Main_py

"""

*Módulo main - instancia objetos de classes definidas em
módulos do pacote projeto01.*

```
""" from produto    import Produto from  
cliente    import Cliente from notafiscal  
import NotaFiscal from itemnotafiscal  
import ItemNotaFiscal from tipocliente  
import TipoCliente
```

```
def main():
```

```
    cli=Cliente(1, "Francisco", 100, "200.100.345-34", 1)
```

```
    p1=Produto(1,100,"Carne de Lata", 5.5)  
    it1=ItemNotaFiscal(1, 1, 10, p1)
```

```
    p2=Produto(2,200,"Feijao", 8.5)  
    it2=ItemNotaFiscal(2, 2, 10, p2)
```

```
p3=Produto(3,300,"Arroz", 4.5)
it3=ItemNotaFiscal(3, 3, 10, p3)
nf = NotaFiscal(1,100,cli)
```

```
nf.adicionarItem(it1)
```

```
nf.adicionarItem(it2)
```

```
nf.adicionarItem(it3)
```

```
nf.calcularNotaFiscal()
```

```
print("Valor Nota Fiscal= " + str(nf.valorNota))
```

```
nf.imprimirNotaFiscal()
```

```
if __name__ == '__main__':
    main()
```

Notafiscal_py

"""

Módulo notafiscal -

Classe NotaFiscal -

Atributos :

id - informado.

codigo - informado.

data - informado.

cliente - informado.

items - informado

valornota - calculado.

"""

import datetime from cliente import

Cliente from produto import Produto

from itemnotafiscal import

ItemNotaFiscal

from app import db

class NotaFiscal(db.Model): __tablename__ = "TB_NOTA_FISCAL"

id = db.Column(db.Integer, primary_key=True) codigo =

db.Column(db.String) cliente = db.Column(db.String,

db.ForeignKey("TB_CLIENTE.codigo")) data =

db.Column(db.DateTime) valorNota = db.Column(db.Float)

cliente = db.relationship('Cliente', foreign_keys=codigo)

def __init__(self, Id, codigo, cliente):

self._Id = Id

self._codigo=codigo

self._cliente=cliente

self._data=datetime.datetime.now()

self._itens=[]

self._valorNota=0.0

def setCliente(self, cliente):

if isinstance(cliente, Cliente):

self._cliente=cliente

```
def adicionarItem(self, item):  
    if isinstance(item, ItemNotaFiscal):  
        self._itens.append(item)
```

```
def calcularNotaFiscal(self):  
    valor=0.0    for item in  
self._itens:    valor = valor +  
item._valorItem  
self.valorNota=valor
```

```
def imprimirNotaFiscal(self):  
    pass
```

Notafiscal_produto_py

```
"""  
"""
```

```
from notafiscal import NotaFiscal  
from produto    import Produto
```

```
class NotaFiscal_Produto():
```

```
    def __init__(self):  
self._notasFiscais=[]  
self._produtos=[]
```

```
    def adicionarNotaProduto(self, nota, produto):    if isinstance  
(nota, NotaFiscal) and isinstance(produto, Produto):
```



```
        self._notasFiscais.append(nota)
self._produtos.append(produto)
```

Projeto_py

```
"""
```

Módulo produto

Classe Produto

Atributos :

id - informado
codigo - informado
descricao - informado
valorUnitario - informado.

```
""" from app
```

```
import db
```

```
class Produto(db.Model): __tablename__ =
"TB_PRODUTO" id = db.Column(db.Integer,
primary_key=True) codigo =
db.Column(db.String, unique=True) descricao
= db.Column(db.String) valorUnitario =
db.Column(db.Float)
```

```
def __init__(self, id, codigo, descricao, valorUnitario):
    self._id = id
self._codigo=codigo
self._descricao=descricao
self._valorUnitario=valorUnitario
```

```
def getDescricao(self):  
    return self._descricao
```

```
def getValorUnitario(self):  
    return self._valorUnitario
```

```
def str(self):  
    string="\nId={3} Codigo={2} Descricao={1} Valor  
Unitario={0}".format(self._valorUnitario, self._descricao, self._codigo, self._id)    return  
string
```

```
if __name__ == '__main__':  
    produto=Produto(1,100,'Carne de Lata', 5.5)  
    print(produto.str())
```