



UNIVERSIDAD ESTATAL DE MILAGRO



UNIVERSIDAD ESTATAL DE MILAGRO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

SEXTO SEMESTRE 2020

AUTOR:

YUGSAN MENDOZA EULLER EGIDIO

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:

PROGRAMACIÓN WEB

TEMA:

S6-APLICACIÓN PRACTICA “REALIZACION DE
OPERACIONES CON MODELOS USANDO EL ORM DE
DJANGO”

DOCENTE:

ING. VERA PAREDES DANIEL ALEXANDER

MILAGRO – ECUADOR

02/08/2020

VISIÓN

Ser una institución de educación superior, pública, autónoma y acreditada, de pregrado y postgrado, abierta a las corrientes del pensamiento universal, líder en la formación de profesionales emprendedores, honestos, solidarios, responsables y con un elevado compromiso social y ambiental, para contribuir al desarrollo local, nacional e internacional.

MISIÓN

Es una institución de educación superior, pública, que forma profesionales de calidad, mediante la investigación científica y la vinculación con la sociedad, a través de un modelo educativo holístico, sistémico, por procesos y competencias, con docentes altamente capacitados, infraestructura moderna y tecnología de punta, para contribuir al desarrollo de la región y el país.

Insertar registros en los modelos.

```
>>> from OrmModelo.models import Producto, Cliente, Factura, DetalleFactura
>>> P= Producto(descripcion='Aceite Girazol', precio=1.50, stock=2000)
>>> P.save()

>>> Producto.objects.create(descripcion='Coca Cola', precio=0.90, stock=10000)
<Producto: Producto object (2)>
```

Inserte dos registros de clientes, 2 registros en el modelo factura con sus dos registros en el modelo detalleFactura con el ejemplo 1 y 2 respectivamente.

Cliente:

```
>>> clien=Cliente(ruc='0604661785132', nombre='Euller Yugsan', direccion='Rcto la Victoria')
>>> clien.save()
>>> Cliente.objects.create(ruc='0603643693113', nombre='Edwin Alexander', direccion='Bucay')
<Cliente: Cliente object (2)>
```

Factura y DetalleFactura:

```
>>> Cliente.objects.create(ruc='0603643693113', nombre='Edwin Alexander', direccion='Bucay')
<Cliente: Cliente object (2)>
>>>
>>> fact=Factura(cliente=clien, fecha='2019-03-03')
>>> fact.save()
>>> detfac=DetalleFactura.objects.create(factura=fact, producto=P, cantidad=5, precio=P.precio, subtotal=P.precio*5)
>>> detfac
<DetalleFactura: DetalleFactura object (1)>
>>> P1=Producto.objects.create(descripcion='Galleta', precio=0.75, stock=45)
>>> P1
<Producto: Producto object (3)>
>>> detfac1=DetalleFactura.objects.create(factura=fact, producto=P1, cantidad=50, precio=P1.precio, subtotal=P1.precio*50)
>>> detfac1
<DetalleFactura: DetalleFactura object (2)>
>>> fact.total=detfac.subtotal+detfac1.subtotal
>>> fact.save()

>>> P2=Producto(descripcion='Leche', precio=0.75, stock=200)
>>> P2.save()
>>> P3=Producto.objects.create(descripcion='Chocolate', precio=1.00, stock=430)
>>> P3
<Producto: Producto object (5)>
>>> clien1=Cliente.objects.create(ruc='0604857489152', nombre='Andres Romero', direccion='Rcto Ines Maria')
>>> clien1
<Cliente: Cliente object (3)>
>>> fact1=Factura.objects.create(cliente=clien1, fecha='2010-01-01')
>>> fact1
<Factura: Factura object (2)>
>>> detfac2=DetalleFactura.objects.create(factura=fact1, producto=P2, cantidad=200, precio=P2.precio, subtotal=P2.precio*200)
>>> detfac2
<DetalleFactura: DetalleFactura object (3)>
>>> detfac3=DetalleFactura.objects.create(factura=fact1, producto=P3, cantidad=78, precio=P3.precio, subtotal=P3.precio*78)
>>> detfac3
<DetalleFactura: DetalleFactura object (4)>
>>> fact1.total=detfac2.subtotal+detfac3.subtotal
>>> fact1.save()
```

Actualizar registros en los modelos

```
>>> M=Producto.objects.get(id=1)
>>> M.precio
1.5
>>> M.precio=1.3
>>> M.save()
>>> M.precio
1.3
>>> Producto.objects.filter(id=1).update(precio=1.50)
1
>>> M.precio
1.3
>>> M=Producto.objects.get(id=1)
>>> M.precio
1.5
```

Modificar 2 registros de producto, cliente, factura y detalleFactura con el ejemplo 1 y 2 respectivamente:

Producto:

```
>>> MP=Producto.objects.get(id=3)
>>> MP.descripcion
'Galleta'
>>> MP.descripcion='Galleta Dulce'
>>> MP.save()
>>> MP.descripcion
'Galleta Dulce'

>>> M.descripcion
'Aceite Girazol'
>>> M.precio
1.5
>>> Producto.objects.filter(id=1).update(descripcion='Aceite
Alesol',precio=1.75)
1
>>> M=Producto.objects.get(id=1)
>>> M.descripcion
'Aceite Alesol'
>>> M.precio
1.75
_
```

Cliente:

```
>>> MC=Cliente.objects.get(id=3)
>>> MC.nombre
'Andres Romero'
>>> MC.nombre='Juan Guarnizo'
>>> MC.save()
>>> MC.nombre
'Juan Guarnizo'
```

```

>>> MC=Cliente.objects.get(id=3)
>>> MC.nombre,MC.direccion
('Andres Maza', 'El Rosario')
>>> Cliente.objects.filter(id=3).update(nombre='Ana Mendoza',
,direccion='Chillanes')
1
>>> MC=Cliente.objects.get(id=3)
>>> MC.nombre,MC.direccion
('Ana Mendoza', 'Chillanes')

```

Factura:

```

>>> fact=Factura.objects.get(id=2)
>>> fact.cliente
<Cliente: Cliente object (3)>
>>> MC=Cliente.objects.get(id=3)
>>> MC.nombre='Pedro Costa'
>>> MC.save()
>>> fact.cliente=MC
>>> fact.save()
>>> fact.cliente.nombre
'Pedro Costa'

>>> fact=Factura.objects.get(id=2)
>>> fact.fecha
datetime.date(2019, 8, 9)
>>> Factura.objects.filter(id=2).update(fecha='2018-08-01')
1
>>> fact=Factura.objects.get(id=2)
>>> fact.fecha
datetime.date(2018, 8, 1)

```

DetalleFactura:

```

>>> detf=DetalleFactura.objects.get(id=2)
>>> detf.producto
<Producto: Producto object (3)>
>>> P=Producto.objects.get(id=3)
>>> P.descripcion
'Galleta Dulce'
>>> P.descripcion='Galleta Oreo'
>>> P.save()
>>> detf.producto=P
>>> detf.save()
>>> detf.producto.descripcion
'Galleta Oreo'

```

```

>>> detf=DetalleFactura.objects.get(id=3)
>>> detf.cantidad
200.0
>>> detf.precio
0.75
>>> detf.subtotal
150.0
>>> DetalleFactura.objects.filter(id=3).update(cantidad=10)
1
>>> detf=DetalleFactura.objects.get(id=3)
>>> DetalleFactura.objects.filter(id=3).update(subtotal=detf
.cantidad*0.75)
1
>>> detf=DetalleFactura.objects.get(id=3)
>>> detf.cantidad,detf.precio,detf.subtotal
(10.0, 0.75, 7.5)

```

Eliminar registros en los modelos:

```

>>> p=Producto.objects.get(id=7)
>>> p.descripcion
'Arroz'
>>> p.delete()
(1, {'OrmModelo.Cliente_producto': 0, 'OrmModelo.DetalleFactura': 0, 'OrmModelo.Producto': 1})
>>> Producto.objects.filter(id=7).delete()
(0, {})

>>> Producto.objects.filter(id=9).delete()
(1, {'OrmModelo.Cliente_producto': 0, 'OrmModelo.DetalleFactura': 0, 'OrmModelo.Producto': 1})

```

Querys de un modelo:

p=Producto.objects.all()

```

>>> p=Producto.objects.all()
>>> p
<QuerySet [(<Producto: Producto object (1)>, <Producto: Producto object (2)>, <Producto: Producto object (3)>, <Producto: Producto object (4)>, <Producto: Producto object (5)>)>

```

p=Producto.objects.get(id=2)

```

>>> p=Producto.objects.get(id=2)
>>> p
<Producto: Producto object (2)>

```

Producto.objects.filter(id__lte=2)

```

>>> Producto.objects.filter(id__lte=2)
<QuerySet [(<Producto: Producto object (1)>, <Producto: Producto object (2)>)]>
>>>

```

Producto.objects.exclude(descripcion__icontains='Cola')

```

>>> Producto.objects.exclude(descripcion__icontains='Cola')
<QuerySet [(<Producto: Producto object (1)>, <Producto: Producto object (3)>, <Producto: Producto object (4)>)]>

```

Producto.objects.filter(id__gte=4)

```
>>> Producto.objects.filter(id__gte=4)
<QuerySet [(<Producto: Producto object (4)>), (<Producto: Producto object (5)>)]>
```

Producto.objects.filter(id__gt=4).values('id','descripcion')

```
>>> Producto.objects.filter(id__gt=4).values('id','descripcion')
<QuerySet [{ 'id': 5, 'descripcion': 'Chocolate'}]>
```

Producto.objects.filter(id__lt=4).values('id','descripcion')

```
>>> Producto.objects.filter(id__lt=4).values('id','descripcion')
<QuerySet [{ 'id': 1, 'descripcion': 'Aceite Alesol'}, { 'id': 2, 'descripcion': 'Coca Cola'}, { 'id': 3, 'descripcion': 'Galleta Oreo'}]>
```

Producto.objects.filter(descripcion='Coca Cola').values('id','descripcion')

```
>>> Producto.objects.filter(descripcion='Coca Cola').values('id','descripcion')
<QuerySet [{ 'id': 2, 'descripcion': 'Coca Cola'}]>
```

Query de varios modelos (relacionados)

Factura.objects.filter(cliente__nombre='Daniel Vera')

```
>>> Factura.objects.filter(cliente__nombre='Euller Yugsan')
<QuerySet [(<Factura: Factura object (1)>)]>
```

c= Cliente.objects.get(nombre='Daniel Vera')

c.factura_set.all()

```
>>> c= Cliente.objects.get(nombre='Euller Yugsan')
>>> c.factura_set.all()
<QuerySet [(<Factura: Factura object (1)>)]>
```

c.factura_set.filter(id=2)

```
>>> c.factura_set.filter(id=1)
<QuerySet [(<Factura: Factura object (1)>)]>
```

Factura.objects.select_related('cliente').filter(cliente__nombre='Daniel Vera')

```
>>> Factura.objects.select_related('cliente').filter(cliente__nombre='Euller Yugsan')
<QuerySet [(<Factura: Factura object (1)>)]>
```

Cliente.objects.prefetch_related('producto').filter(nombre='Daniel Vera').values('nombre','producto__descripcion')

```
>>> Cliente.objects.prefetch_related('producto').filter(nombre='Euller Yugsan').values('nombre','producto__descripcion')
<QuerySet [{ 'nombre': 'Euller Yugsan', 'producto__descripcion': None}]>
```