

<div> <div>Proyecto</div> <div> <div>setting.py</div> <div>Añadir las app nuevas al setting.py</div> </div> <div> <div>SESSION_SERIALIZER =</div> <div>'django.contrib.sessions.serializers.PickleSe</div> <div>rializer'</div> </div> <div> <div>TEMPLATE_DIRS = {</div> <div>os.path.join(BASE_DIR,</div> <div>'TIENDA/templates'),</div> <div>}</div> </div> <div> <div># Redirect when login is correct.</div> <div>LOGIN_REDIRECT_URL = "/myusers/perfil"</div> <div># Redirect when login is not correct.</div> <div>LOGIN_URL = '/myusers/login'</div> </div> <div> <div>urls.py</div> <div>from django.conf.urls import include, url</div> <div>from django.contrib import admin</div> </div> <div> <div>urlpatterns = [</div> <div># Examples:</div> <div># url(r'^\$', 'TIENDA.views.home',</div> <div>name='home'),</div> <div># url(r'^blog/', include('blog.urls')),</div> <div></div> <div>url(r'^\$', include('productos.urls')),</div> <div>url(r'^admin/',</div> <div>include(admin.site.urls)),</div> <div>url(r'^productos/',</div> <div>include('productos.urls')),</div> <div>url(r'^myusers/',</div> <div>include('myusers.urls')),</div> <div>]</div> </div> <div> <div>Templates</div> <div>//Comprobar si el usuario está autenticado</div> <div>{% if user.is_authenticated %}</div> <div>&lt;td&gt;&lt;a href="/myusers/perfil/"&gt;perfil&lt;/a&gt;&lt;/td&gt;</div> <div>&lt;td&gt;&lt;a href="/myusers/logout/"&gt;logout&lt;/a&gt;&lt;/td&gt;</div> <div>{% else %}</div> <div>&lt;td&gt;&lt;a href="/myusers/login/"&gt;login&lt;/a&gt;&lt;/td&gt;</div> <div>&lt;td&gt;&lt;a href="/myusers/registrarse/"&gt;Registrarse&lt;/a&gt;&lt;/td&gt;</div> <div>{% endif %}</div> <div></div> <div>{% if user.is_authenticated %}</div> <div>&lt;h2&gt;Welcome, {{ user.username }}. Thanks for logging</div> <div>in.&lt;/h2&gt;</div> <div>{% else %}</div> <div>&lt;h2&gt;Welcome, new user. Please log in.&lt;/h2&gt;</div> <div>{% endif %}</div> </div> </div>	<div> <div>APP Productos</div> <div> <div>admin.py</div> <div>from django.contrib import admin</div> <div>from productos.models import Ordenador, tabletYSmartphone,</div> <div>Customsobremesa, Sobremesa, Portatil, Tablet, Smartphone</div> <div>admin.site.register(Ordenador)</div> <div>admin.site.register(tabletYSmartphone)</div> <div>admin.site.register(Customsobremesa)</div> <div>admin.site.register(Sobremesa)</div> <div>admin.site.register(Portatil)</div> <div>admin.site.register(Tablet)</div> <div>admin.site.register(Smartphone)</div> </div> <div> <div>models.py</div> <div>from django.db import models</div> <div>from django import forms</div> </div> <div> <div>class Productos(models.Model):</div> <div>modelo=models.CharField(max_length =</div> <div>100)</div> </div> <div> <div>class Ordenador(Productos):</div> <div>procesador=models.CharField(max_length</div> <div>= 100)</div> <div>sistemaOperativo=models.CharField(max_length</div> <div>= 100)</div> <div>precio=models.IntegerField()</div> </div> <div> <div>class Sobremesa(Ordenador):</div> <div>marca=models.CharField(max_length = 30)</div> <div>modeloPlacaBase=models.CharField(max_length =</div> <div>100)</div> <div></div> <div>def __unicode__(self):</div> <div>return self.modelo</div> </div> <div> <div>PROCESADORORDENADORES_CHOICES = (</div> <div>('None', 'No aplicar'),</div> <div>('Intel i7', 'Intel i7'),</div> <div>('Intel i5', 'Intel i5'),</div> <div>('Intel i3', 'Intel i3'),</div> <div>)</div> </div> <div> <div>class listarSobremesasForm(forms.Form):</div> <div>procesador =</div> <div>forms.ChoiceField(choices=PROCESADORORDENADOR</div> <div>ES_CHOICES)</div> </div> <div> <div>class anadirACestaForm(forms.Form):</div> <div>unidades = forms.CharField(initial="1",</div> <div>label='Und.', max_length=3)</div> </div> </div>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Views.py

```
from django.shortcuts import render,
render_to_response
from django.conf.urls.static import static

# Create your views here.

from django.template import RequestContext,
loader
from django.http import HttpResponse,
HttpResponseRedirect
from productos.models import Sobremesa

def index(request):
    template =
loader.get_template('productos/base.html')
    context = RequestContext(request)
    return
HttpResponse(template.render(context))

def listarSobremesas(request):
    if request.method == 'POST':
        form = listarSobremesasForm(request.POST)
        if form.is_valid():
            procesador =
form.cleaned_data["procesador"]
            filtro = None
            sobremesas=[]
            latest_question_list =
Sobremesa.objects.all()
            for sobremesa in latest_question_list:
                apto = False
                noApto = False
                if procesador != 'None':
                    filtro = 'true'
                    if sobremesa.marcaModeloProcesador ==
procesador:
                        apto = True
                    else:
                        noApto = True

                if noApto == False:
                    if apto == True:
                        sobremesas.append(sobremesa)

            if filtro != None:
                latest_question_list = sobremesas

    template =
loader.get_template('productos/listarSobremesas.html')
    context = RequestContext(request,
{'latest_question_list':
latest_question_list, 'form':form})
    return
HttpResponse(template.render(context))
    else:
        form = listarSobremesasForm()
        latest_question_list =
Sobremesa.objects.all()
        template =
loader.get_template('productos/listarSobremesas.html')
        context = RequestContext(request,
{'latest_question_list':
latest_question_list, 'form':form})
        return
HttpResponse(template.render(context))
```

```
def detallesSobremesas(request, question_id):
    if request.method == 'POST':
        form = anadirACestaForm(request.POST)
        if form.is_valid():
            sobremesa = Sobremesa.objects.get(id =
question_id)
            unidades = form.cleaned_data["unidades"]
            if "cesta_compra" in request.session:
                productos =
request.session["cesta_compra"]
                encontrado = None
                if productos:
                    index = 0
                    for producto in productos:
                        producto_aux = producto[0]
                        id_sobremesa = int(sobremesa.id)
                        id_producto = int(producto_aux.id)
                        if id_sobremesa == id_producto:
                            encontrado = 'True'
                            unidades_aux = producto[1]
                            unidades_aux = int(unidades_aux)
                            unidades_aux = unidades_aux +
int(unidades)
                            producto[1]=unidades_aux
                            precio = producto_aux.precio *
unidades_aux
                            producto[2] = precio
                            productos[index]=producto
                            request.session["cesta_compra"] =
productos

                    index = index + 1

                if encontrado == None:
                    producto = []
                    producto.append(sobremesa)
                    unidades = int(unidades)
                    producto.append(unidades)
                    precio = sobremesa.precio * unidades
                    producto.append(precio)
                    producto.append("sobremesas")
                    productos.append(producto)
                    request.session["cesta_compra"] =
productos
                else:
                    request.session["cesta_compra"] = []
                    producto = []
                    producto.append(sobremesa)
                    unidades = int(unidades)
                    producto.append(unidades)
                    precio = sobremesa.precio * unidades
                    producto.append(precio)
                    producto.append("sobremesas")
                    productos.append(producto)
                    request.session["cesta_compra"] =
productos

            return
HttpResponseRedirect('/productos/cesta')
    else:
        form = anadirACestaForm()
        sobremesa = Sobremesa.objects.get(id =
question_id)
        if sobremesa.memoriaDisco >= 1000:
            sobremesa.memoriaDisco =
sobremesa.memoriaDisco/1000
            sobremesa.memoriaDisco =
str(sobremesa.memoriaDisco) + ' TB'
        else:
```

```

sobremesa.memoriaDisco =
str(sobremesa.memoriaDisco) + ' GB'

sobremesa.memoriaRAM =
str(sobremesa.memoriaRAM) + ' GB'

template =
loader.get_template('productos/detallesSobrem
esas.html')
context = RequestContext(request,
{'sobremesa': sobremesa, 'form':form})
return
HttpResponse(template.render(context))

def cesta(request, actualizar_id=None):
if request.method == 'POST':
form = anadirACestaForm(request.POST)
if form.is_valid():
unidades = form.cleaned_data["unidades"]
if "cesta_compra" in request.session:
productos =
request.session["cesta_compra"]
else:
request.session["cesta_compra"] = []
productos =
request.session["cesta_compra"]

index = 0
for producto in productos:
producto_aux = producto[0]
actualizar_id = int(actualizar_id)
id_producto = int(producto_aux.id)
if actualizar_id == id_producto:
unidades_aux = int(unidades)
producto[1] = unidades_aux
precio = producto_aux.precio *
unidades_aux
producto[2] = precio
productos[index]=producto
request.session["cesta_compra"] =
productos
index = index + 1
return
HttpResponseRedirect('/productos/cesta')

else:
if "cesta_compra" in request.session:
productos =
request.session["cesta_compra"]
else:
request.session["cesta_compra"] = []
productos =
request.session["cesta_compra"]

form = anadirACestaForm()
precioTotal = 0
unidadesTotal = 0
for producto in productos:
precioTotal = precioTotal + producto[2]
unidadesTotal = unidadesTotal +
producto[1]

template =
loader.get_template('productos/cesta.html')
context = RequestContext(request,
{'productos':productos, 'form':form,
'precioTotal':precioTotal,
'unidadesTotal':unidadesTotal})
return
HttpResponse(template.render(context))

```

```

def eliminarProductoCesta(request,
eliminar_id):
cesta_compra =
request.session["cesta_compra"]

nuevo_productos = []

for producto in cesta_compra:
aux_producto = producto[0]
id_producto =
int(aux_producto.id)
eliminar_id = int(eliminar_id)

if eliminar_id != id_producto:

nuevo_productos.append(producto)

request.session["cesta_compra"] =
nuevo_productos

return
HttpResponseRedirect('/productos/cesta')

urls.py

from django.conf.urls import patterns, url
from productos import views

urlpatterns = patterns('',
url(r'^$', views.index, name='index'),
url(r'^sobremesas/$',
views.listarSobremesas,
name="listarSobremesas"),
url(r'^sobremesas/(?P<question_id>\d+)/$',
views.detallesSobremesas,
name='detallesSobremesas'),
url(r'^cesta/$', views.cesta,
name='cesta'),
url(r'^cesta/eliminarproducto/(?
P<eliminar_id>\d+)/$',
views.eliminarProductoCesta,
name='eliminarProductoCesta'),
url(r'^cesta/actualizarproducto/(?
P<actualizar_id>\d+)/$', views.cesta,
name='actualizarProductoCesta'),
)

```

## //ListarSobremesas.html

```
{% extends "productos/base.html" %}

{% block NavigationLeft %}
    <form action="." method="POST">
        {% csrf_token %}
        <table>
            <tr>

                <td><h3>Filtrar</h3></td>
                <td><input
type="submit" value="Aplicar filtro"/></td>
            </tr>
            <tr>
                <td>Procesador: </td>

                <td>{{ form.procesador }}</td>
            </tr>
        </table>
    </form>
{% endblock %}

{% block Contents %}
    {% if latest_question_list %}
        <ul>
            {% for question in
latest_question_list %}
                <li><a
href="/productos/sobremesas/
{{ question.id }}/"/>{{ question.modelo }}</a>
                </li>
            {% endfor %}
        </ul>
    {% else %}
        <p>No hay sobremesas
disponibles.</p>
    {% endif %}
{% endblock %}
```

## //detallesSobremesas.html

```
{% extends "productos/base.html" %}

{% block Contents %}
    <h2>{{ sobremesa.modelo}}</h2>
    <table>
        <tr>
            <td>Imagen</td>
            <td>
                <table>
                    <tr>
                        {% if sobremesa.existencias > 0 %}
                            <td>Disponibilidad: En stock. unidades
disponibles: {{ sobremesa.existencias }}
Und.</td>
                        {% else %}
                            <td>Disponibilidad: Agotado</td>
                        {% endif %}
                    </tr>
                    <tr>
                        <form method="post" action="">
                            {% csrf_token %}
                            <td>{{ sobremesa.precio }}€
x{{ form.unidades }}Und. <input type="submit"
value="Comprar"/></td>
                        </form>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
```

```
<ul>
    <li>Marca: {{ sobremesa.marca }}</li>
    <li>Procesador:
{{ sobremesa.procesador }}</li>
    <li>Memoria RAM:
{{ sobremesa.memoriaRAM }}</li>
    <li>Disco duro:
{{ sobremesa.memoriaDisco }}</li>
    <li>Gráfica:
{{ sobremesa.modeloGrafica }}</li>
    <li>Placa base:
{{ sobremesa.modeloPlacaBase }}</li>
    <li>Lector/Grabador discos:
{{ sobremesa.lectorGrabador }}</li>
    <li>Sistema Operativo:
{{ sobremesa.sistemaOperativo }}</li>
</ul>
{% endblock %}
```

### Para cesta es interesante:

```
{% for producto, unidades, precio, clase in
productos %}
{% endfor %}
```

### APP myUsers: models.py

```
from django.db import models
from django.forms import ModelForm
from django import forms
from django.contrib.auth.models import User
from django.forms.widgets import
PasswordInput

class registrarUsuarioForm(ModelForm):
    class Meta:
        model = User
        fields = ['username', 'password',
'email', 'first_name', 'last_name']
        widgets = {
            'password':
forms.PasswordInput(),
        }

class perfilUsuarioForm(ModelForm):
    class Meta:
        model = User
        fields = ['username', 'password',
'email', 'first_name', 'last_name']
        widgets = {
            'password':
forms.PasswordInput(),
        }

class cambiarContrasenaForm(forms.Form):
    passwordAntigua =
forms.CharField(label='Antigua contrasena:',
widget = PasswordInput(), max_length=30)
    passwordNueva1 =
forms.CharField(label='Nueva
contrasena:', widget = PasswordInput(),
max_length=30)
    passwordNueva2 =
forms.CharField(label='Repita nueva
contrasena:', widget = PasswordInput(),
max_length=30)

class cambiarEmailForm(ModelForm):
    class Meta:
        model = User
        fields = ['email']
```

```

class cambiarDatosPersonalesForm(ModelForm):
    class Meta:
        model = User
        fields = ['first_name',
'last_name']

views.py:
from django.contrib.auth.forms import
AuthenticationForm
from django.contrib.auth.decorators import
login_required
from django.contrib.auth.hashers import
check_password

# Create your views here.

def index(request):
    return
render_to_response("productos/base.html",
request)

def registrarUsuario(request):
    if request.method == 'POST':
        form =
registrarUsuarioForm(request.POST)
        if form.is_valid():
            username =
form.cleaned_data["username"]
            password =
form.cleaned_data["password"]
            email =
form.cleaned_data["email"]
            first_name =
form.cleaned_data["first_name"]
            last_name =
form.cleaned_data["last_name"]
            user =
User.objects.create_user(username, email,
password)
            user.first_name =
first_name
            user.last_name = last_name

            # Save new user attributes
            user.save()

            return
HttpResponseRedirect('/') # Redirect after
POST
        else:

            form = registrarUsuarioForm()

            data = {
                'form': form,
            }
            return
render_to_response('myusers/registrarUsuario.
html', data,
context_instance=RequestContext(request))

```

```

@login_required()
def perfilUsuario(request):
    user = request.user
    user = User.objects.get(id = user.id)
    template =
loader.get_template('myusers/perfilUsuario.ht
ml')

    context = RequestContext(request,
{'user': user,})
    return
HttpResponse(template.render(context))

@login_required()
def cambiarContrasena(request):
    if request.method == 'POST':
        form =
cambiarContrasenaForm(request.POST)
        if form.is_valid():
            user = request.user
            passwordOriginal =
user.password
            passwordAntigua =
form.cleaned_data["passwordAntigua"]
            passwordNueva1 =
form.cleaned_data["passwordNueva1"]
            passwordNueva2 =
form.cleaned_data["passwordNueva2"]

            if
check_password(passwordAntigua,
passwordOriginal):

                if passwordNueva1 ==
passwordNueva2:

                    user.set_password(passwordNueva1)
                    user.save()

                    update_session_auth_hash(request, user)
                    return
HttpResponseRedirect('/')
                else:
                    mensaje = "La
nueva contrasena y su confirmacion no
coinciden."

                    return
render(request,
'myusers/cambiarContrasena.html', {'form':
form, 'mensaje':mensaje})
            else:
                mensaje = "La
contrasena original introducida no es
correcta."

                return render(request,
'myusers/cambiarContrasena.html', {'form':
form, 'mensaje':mensaje})
            else:

                form = cambiarContrasenaForm()

                data = {
                    'form': form,}
                return render(request,
'myusers/cambiarContrasena.html', {'form':
form})

```

```

@login_required()
def cambiarEmail(request):
    if request.method == 'POST':
        form =
cambiarEmailForm(request.POST)
        if form.is_valid():
            user = request.user

            # Process the data in
form.cleaned_data
            email =
form.cleaned_data["email"]

            user.email = email
            user.save()

            return
HttpResponseRedirect('/myusers/perfil') #
Redirect after POST
        else:

            form = cambiarEmailForm()

            data = {
                'form': form,
            }
            return render(request,
'myusers/cambiarEmail.html', {'form': form})

@login_required()
def cambiarDatosPersonales(request):
    if request.method == 'POST': # If the
form has been submitted...
        form =
cambiarDatosPersonalesForm(request.POST) # A
form bound to the POST data
        if form.is_valid(): # All
validation rules pass

            user = request.user

            # Process the data in
form.cleaned_data
            first_name =
form.cleaned_data["first_name"]
            last_name =
form.cleaned_data["last_name"]

            user.first_name =
first_name
            user.last_name = last_name

            # Save new user attributes
            user.save()

            return
HttpResponseRedirect('/myusers/perfil') #
Redirect after POST
        else:

            form =
cambiarDatosPersonalesForm()

            data = {
                'form': form,
            }
            return render(request,
'myusers/cambiarDatosPersonales.html',
{'form': form})

```

```

Urls.py:
from django.conf.urls import patterns, url
from myusers import views
from django.contrib.auth.views import login,
logout

urlpatterns = patterns('',
    url(r'^$', views.index, name='index'),
    url(r'^perfil/$', views.perfilUsuario,
name='perfilUsuario'),
    url(r'^registrarse/$',
views.registrarUsuario,
name='registrarUsuario'),
    url(r'^login/$', login,
{'template_name':
'myusers/loginUsuario.html', },
name='loginUsuario'),
    url(r'^logout/$', logout,
{'template_name': 'productos/base.html', },
name='logoutUsuario'),
    url(r'^perfil/cambiarcontrasena/$',
views.cambiarContrasena,
name='cambiarContrasena'),
    url(r'^perfil/cambiaremail/$',
views.cambiarEmail, name='cambiarEmail'),
    url(r'^perfil/cambiardatospersonales/
$', views.cambiarDatosPersonales,
name='cambiarDatosPersonales'),
)

//perfilUsuario.html
{% extends "myusers/base.html" %}

{% block Contents %}
    <table>
        <tr>
            <td>Nombre de
usuario: {{ user.username }}</td>
        </tr>
        <tr>
            <td>Contraseña:
*****</td>
            <td><a
href="/myusers/perfil/cambiarcontrasena/">Cam
biar</a></td>
        </tr>
        <tr>
            <td>Email:
{{ user.email }}</td>
            <td><a
href="/myusers/perfil/cambiaremail/">Cambiar<
/a></td>
        </tr>
        <tr>
            <td>Datos Personales:
{{ user.first_name }}
{{ user.last_name }}</td>
            <td><a
href="/myusers/perfil/cambiardatospersonales/
">Cambiar</a></td>
        </tr>
    </table>
{% endblock %}

```

#### RegistrarUsuario.html:

```
{% extends "myusers/base.html" %}

{% block Contents %}
    <form method="post" action="">

        {% csrf_token %}

        <table>
            <tr>

                <td>{{ form.username.label_tag }}</td>

                <td>{{ form.username }}</td>
            </tr>
            <tr>

                <td>{{ form.password.label_tag }}</td>

                <td>{{ form.password }}</td>
            </tr>
            <tr>

                <td>{{ form.email.label_tag }}</td>

                <td>{{ form.email }}</td>
            </tr>
            <tr>

                <td>{{ form.first_name.label_tag }}</td>

                <td>{{ form.first_name }}</td>
            </tr>
            <tr>

                <td>{{ form.last_name.label_tag }}</td>
                <td>{{ form.last_name
            }}</td>
            </tr>
        </table>

        <input type="submit"
value="Aceptar"/>
    </form>
{% endblock %}
```

#### Comandos interesantes:

```
python manage.py makemigrations productos
python manage.py migrate
python manage.py runserver
python manage.py createsuperuser
```

#### loginUsuario:

```
{% extends "myusers/base.html" %}

{% block Contents %}
    {% if form.errors %}
        <p>Your username and password
didn't match. Please try again.</p>
    {% endif %}

    <form method="post" action="{% url
'django.contrib.auth.views.login' %}">
        {% csrf_token %}
        <table>
            <tr>

                <td>{{ form.username.label_tag }}</td>

                <td>{{ form.username }}</td>
            </tr>
            <tr>

                <td>{{ form.password.label_tag }}</td>

                <td>{{ form.password }}</td>
            </tr>
        </table>

        <input type="submit"
value="login"/> <input type="hidden"
name="next" value="{{ next }}" />
    </form>
{% endblock %}
```

#### cambiarContraseña.html:

```
{% extends "myusers/base.html" %}

{% block Contents %}
    <form method="post" action="">

        {% csrf_token %}
        {% if mensaje %}
            <p>{{mensaje}}</p>
        {% endif %}
        <table>
            <tr>

                <td>{{ form.passwordAntigua.label_tag }}</td>

                <td>{{ form.passwordAntigua }}</td>
            </tr>
            <tr>

                <td>{{ form.passwordNueva1.label_tag }}</td>

                <td>{{ form.passwordNueva1 }}</td>
            </tr>
            <tr>

                <td>{{ form.passwordNueva2.label_tag }}</td>

                <td>{{ form.passwordNueva2 }}</td>
            </tr>
        </table>

        <input type="submit"
value="Aceptar"/>
    </form>
{% endblock %}
```



# Referencia del Modelo

Tipos de campos	Opciones de campo
<b>AutoField</b>	<b>null</b> = False
<b>BigIntegerField</b>	<b>blank</b> = False
<b>BooleanField</b>	<b>choices</b> = tupla_de_opciones
<b>NullBooleanField</b>	<b>db_column</b> = 'nombre de columna'
<b>CharField</b>	<b>db_index</b> = False
<b>max_length</b>	<b>db_tablespace</b> = 'nombre_tablespace'
<b>CommaSeparatedIntegerField</b>	<b>default</b> = 'valor'
<b>DateField</b>	<b>editable</b> = True
<b>auto_now</b> = False	<b>error_messages</b> = diccionario_de_mensajes
<b>auto_now_add</b> = False	<b>help_text</b> = 'text'
<b>DateTimeField</b>	<b>primary_key</b> = True
<b>auto_now</b> = False	<b>unique</b> = True
<b>auto_now_add</b> = False	<b>unique_for_date</b> = 'campo_fecha'
<b>DecimalField</b>	<b>unique_for_month</b> = 'campo_fecha'
<b>max_digits</b> = 10	<b>unique_for_year</b> = 'campo_fecha'
<b>decimal_places</b> = 2	<b>verbose_name</b> = 'nombre'
<b>IntegerField</b>	<b>validators</b> = lista_de_validadores
<b>SlugField</b>	
<b>max_length</b> = 50	
<b>SmallIntegerField</b>	
<b>TextField</b>	
<b>auto_now</b> = False	
<b>auto_now_add</b> = False	
<b>URLField</b>	
<b>verify_exists</b> = True	
<b>max_length</b> = 200	
<b>FloatField</b>	
<b>IPAddressField</b>	
<b>GenericIPAddressField</b>	
<b>EmailField</b>	
<b>max_length</b> = 75	
<b>FileField</b>	
<b>upload_to</b> = 'cargas/'	
<b>storage</b> = FileSystemStorage	
<b>FilePathField</b>	
<b>path</b> = '/home/archivos'	
<b>match</b> = r'\.png\$'	
<b>recursive</b> = False	
<b>ImageField</b>	
<b>upload_to</b> = 'cargas/'	
<b>height_field</b> = 'nombre_campo'	
<b>width_field</b> = 'nombre_campo'	
<b>PositiveIntegerField</b>	
<b>PositiveSmallIntegerField</b>	
<b>ForeignKey</b>	
<b>related_name</b> = 'modelo'	
<b>limit_choices_to</b> = query_kwargs	
<b>to_field</b> = 'llave_campo'	
<b>ManyToManyField</b>	
<b>related_name</b> = 'modelo'	
<b>limit_choices_to</b> = query_kwargs	
<b>symmetrical</b> = True	
<b>through</b> = 'ModeloIntermedio'	
<b>OneToOneField</b>	
<b>parent_link</b> = 'campo'	