



pythonTM

django

Contenido

- Modelos
- Seguridad
- Ejemplos extras

Modelos

Modelo Basico

```
from django.db import models

class Musico(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    instrumento = models.CharField(max_length=100)

class Disco(models.Model):
    artista = models.ForeignKey(Musico, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)
    fecha_lanzamiento = models.DateField()
    numero_estrellas = models.IntegerField()
```

Modelos

Modelo Basico

```
from django.db import models

class Musico(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    instrumento = models.CharField(max_length=100)

class Disco(models.Model):
    artista = models.ForeignKey(Musico, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)
    fecha_lanzamiento = models.DateField()
    numero_estrellas = models.IntegerField()
```

Modelos

Modelo choices

```
from django.db import models

class Person(models.Model):
    SHIRT_SIZES = (
        ('S', 'Small'),
        ('M', 'Medium'),
        ('L', 'Large'),
    )
    name = models.CharField(max_length=60)
    shirt_size = models.CharField(max_length=1,
    choices=SHIRT_SIZES)
```

Modelos

Modelo choices

```
from django.db import models

class Person(models.Model):
    SHIRT_SIZES = (
        ('S', 'Small'),
        ('M', 'Medium'),
        ('L', 'Large'),
    )
    name = models.CharField(max_length=60)
    shirt_size = models.CharField(max_length=1,
    choices=SHIRT_SIZES)
```

Modelos - Instrucciones de manejo de registros

Creación

```
fruit = Fruit.objects.create(name='Apple')
```

Obtener todos los registros

```
Membership.objects.all()
```

Filtrar Registros

```
Group.objects.filter(members__name__startswith='Paul')
```

```
Group.objects.filter(create_date__gte='2021-12-12')
```

```
Group.objects.filter(name__icontains='art')
```

Recuperar un registro

```
Group.objects.get(id=123)
```

Borrar

```
Group.objects.get(id=123).delete()
```

Seguridad

Django viene con un sistema de autenticación de usuarios. Maneja cuentas de usuario, grupos, permisos y sesiones de usuario basadas en cookies.

Link: [Documentación](#)

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, set against a dark green rounded rectangular background.

Seguridad - Login

```
from django.contrib.auth import authenticate, login

def my_view(request):
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(request, username=username,
password=password)
    if user is not None:
        login(request, user)
        # Redirect to a success page.
        ...
    else:
        # Return an 'invalid login' error message.
        ...
```

Seguridad - Logout

```
from django.contrib.auth import logout

def logout_view(request):
    logout(request)

    # Redirect to a success page.
```

Seguridad - Register

```
from django.contrib.auth.models import User

def my_view(request):
    username = request.POST['username']
    email = request.POST['email']
    password = request.POST['password']
    user = User.objects.create_user(username, email, password)
    ...
```

Seguridad - Login views

Agregar URL

```
urlpatterns = [  
    path('accounts/', include('django.contrib.auth.urls')),  
]
```

URL disponibles

```
accounts/login/ [name='login']  
accounts/logout/ [name='logout']  
accounts/password_change/ [name='password_change']  
accounts/password_change/done/ [name='password_change_done']  
accounts/password_reset/ [name='password_reset']  
accounts/password_reset/done/ [name='password_reset_done']  
accounts/reset/<uidb64>/<token>/ [name='password_reset_confirm']  
accounts/reset/done/ [name='password_reset_complete']
```

Seguridad - Login views

Agregar config

```
LOGIN_REDIRECT_URL = '/alumnos'
```

Agregar template registration/login.html.

```
<form method="post" action="{% url 'login' %}">
{% csrf_token %}
<div>
  <td>{{ form.username.label_tag }}</td>
  <td>{{ form.username }}</td>
</div>
<div>
  <td>{{ form.password.label_tag }}</td>
  <td>{{ form.password }}</td>
</div>
<div>
  <input type="submit" value="login" />
  <input type="hidden" name="next" value="{% next %}" />
</div>
</form>
```

Ejemplos extras

Esto es contenido para ver en el código....

; -)

django

django