
PUCP

Desarrollo Web con Python

CETAM - PUCP

Sesión 6:

- Django - Modelos y Migraciones
  - Modelos y migraciones
  - Relaciones entre modelos
  - Modelo User
- Django - Responses
  - HttpResponse, HttpResponseRedirect
  - FileResponse
  - JsonResponse

Django - Modelos y migraciones

Desarrollo Web con Python - Sesión 6

Modelos y relaciones

En Django se cuenta con el archivo models.py que permiten definir la estructura de las tablas que se crearán en la base de datos, cada clase es una tabla mientras que cada objeto de la clase es una fila de la tabla. Al SQL ser un lenguaje relacional se pueden crear relaciones entre los campos y objetos de la tabla:

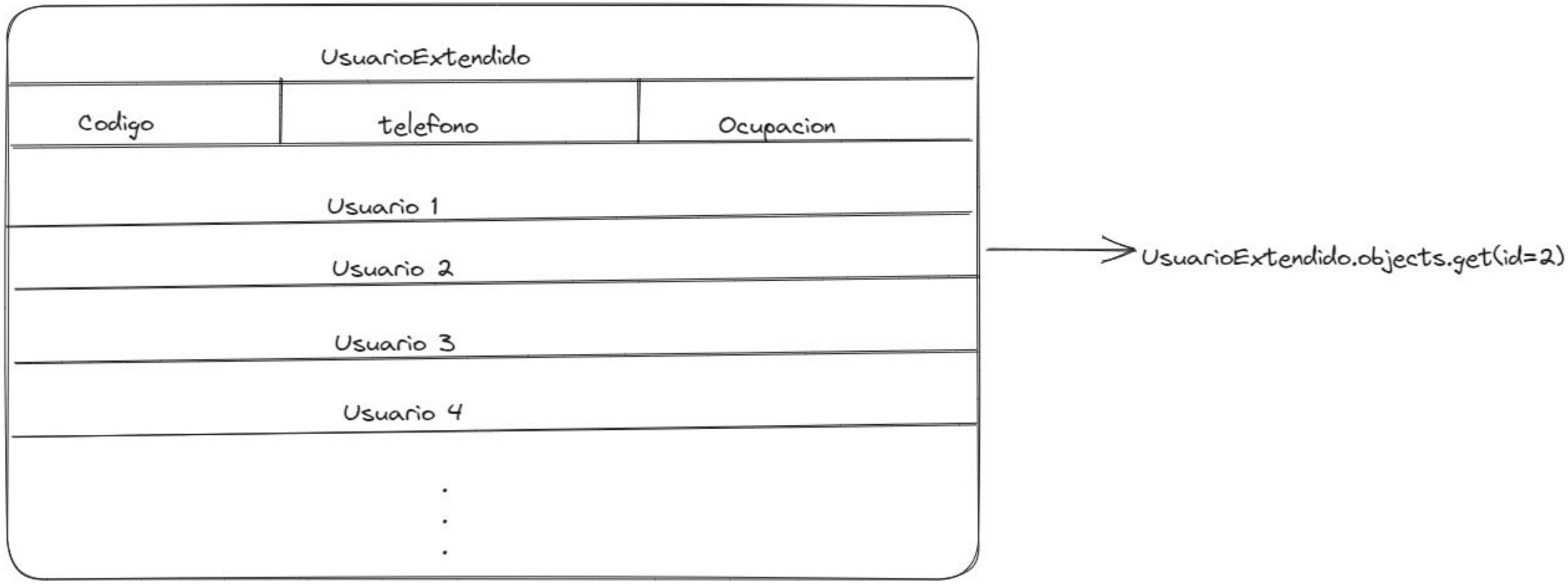
- OneToOne
- OneToMany
- ManyToMany

Modelos en Django

En el archivo models.py

```
class usuarioExtendido(models.Model):
    codigo = models.CharField(max_lenght=128,default='')
    telefono = models.CharField(max_lenght=128,default='')
    ocupacion = models.CharField(max_lenght=128,default='')
```

Clases y objetos en Django



Creando migraciones

Cuando se realizan modificaciones en los modelos o bases de datos es necesario registrarlas y modificar la base de datos. Django posee un sistema automatizado para este proceso y que se puede acceder desde la terminal de comandos a través de:

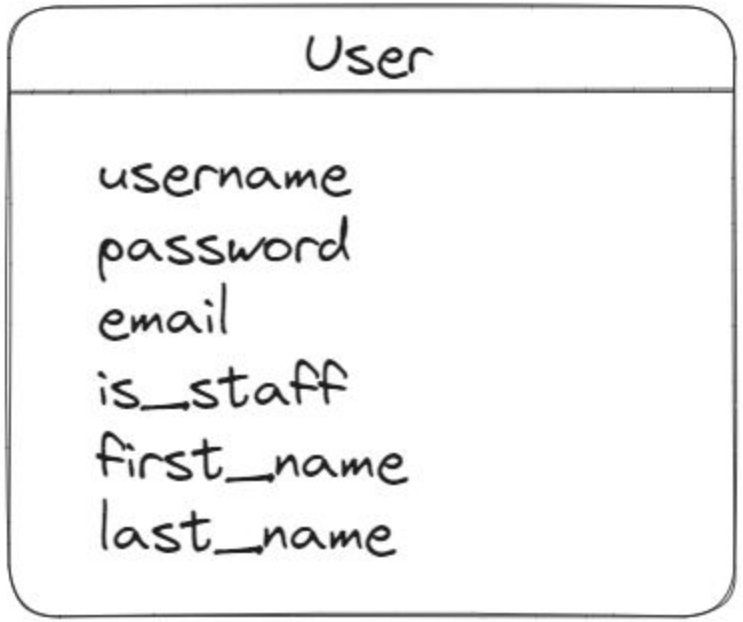
- python3 manage.py makemigrations
- python3 manage.py migrate

Querys desde python

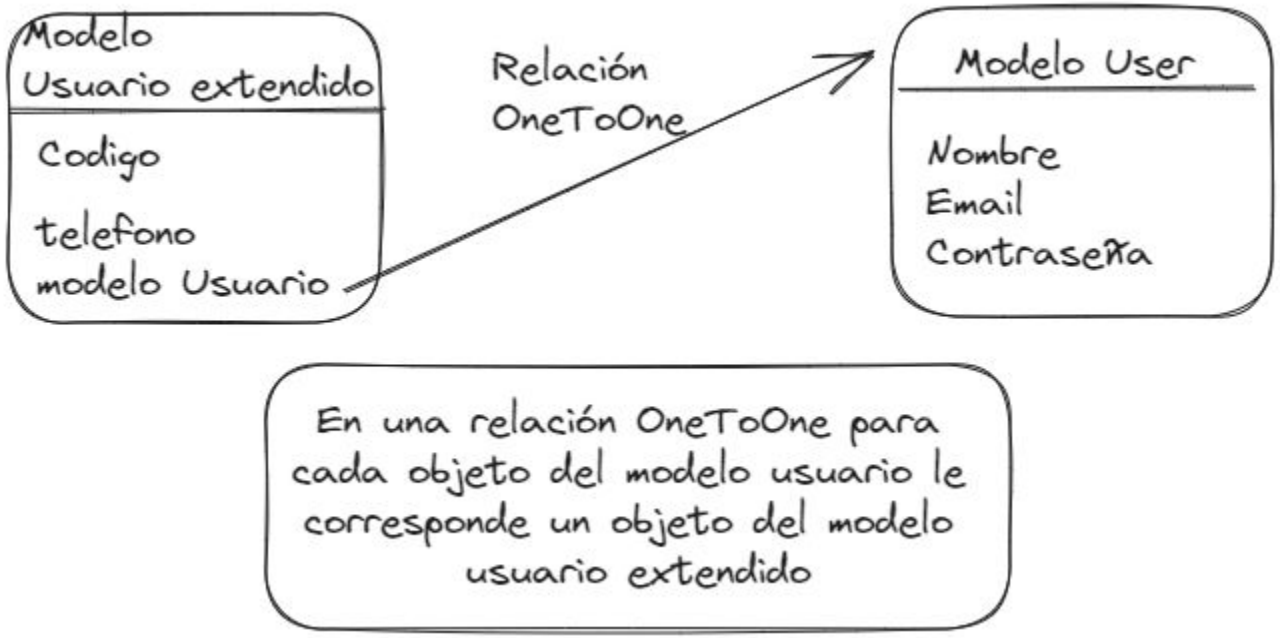
Para poder obtener información de la base de datos se trabaja con los siguientes querys:

- <class-name>.objects.create()
- <class-name>.objects.all()
- <class-name>.objects.get(id=4)
- <class-name>.objects.filter(name='alexander').first()
- <class-name>.objects.filter(fecha__range = ['f_i','f_f'])
- <class-name>.save()

Modelo User: django.contrib.auth.models.User

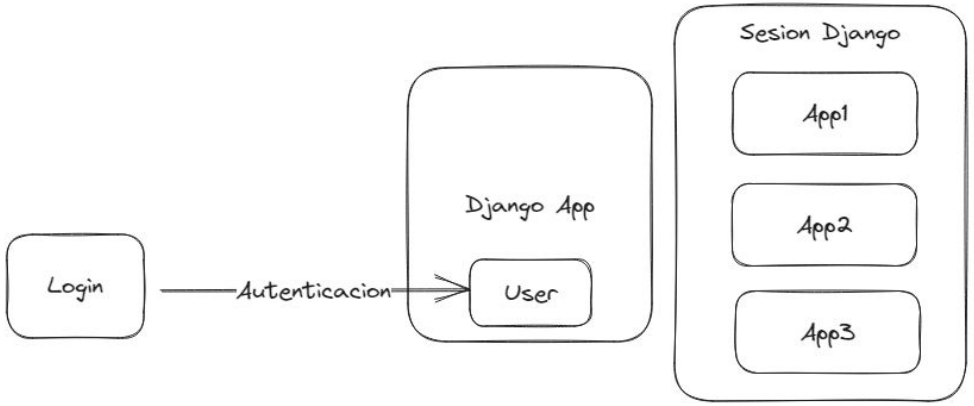


Relaciones en Django



Modelo Usuario en Django

Django cuenta con una tabla integrada para el modelo de usuarios que permite la autenticación y acceso a las aplicaciones de Django. El enlace a este modelo se encuentra en la librería:
from django.contrib.auth.models import User

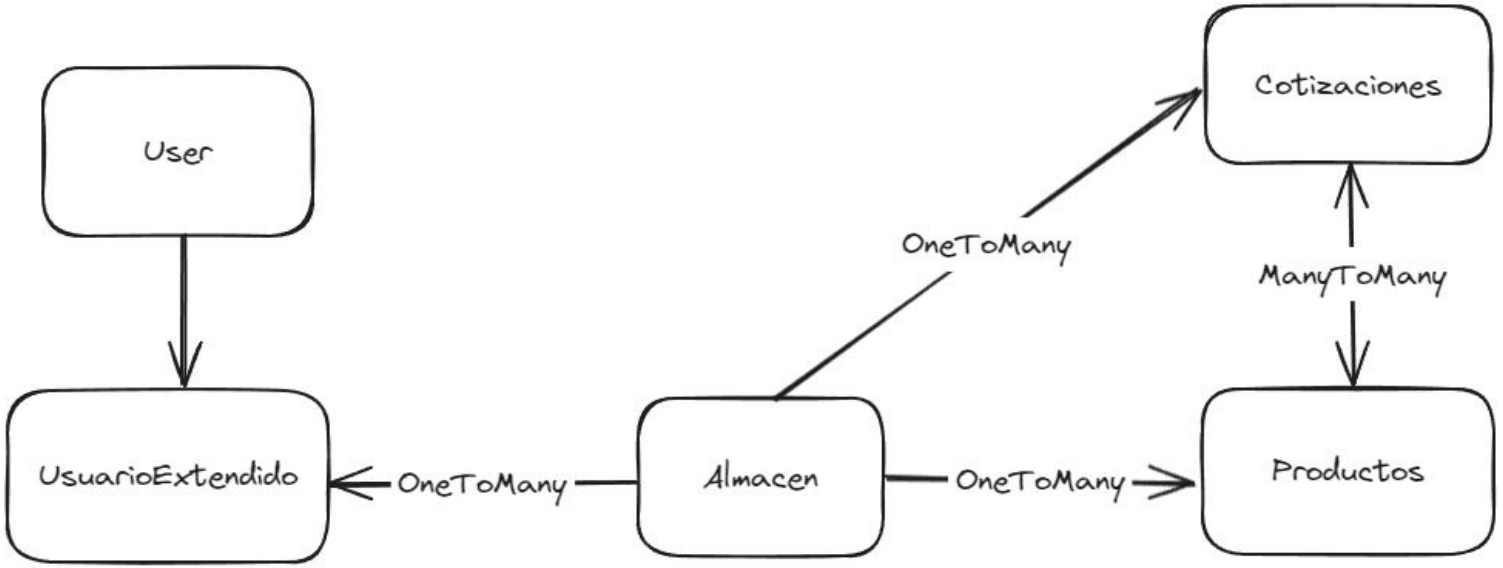


Ejemplo de aplicación

En la presente aplicación se desarrollarán las funciones de una web de gestión de inventarios en donde se registrarán usuarios, productos, almacenes y cotizaciones. El sistema a desarrollar debe cumplir con gestionar los accesos de los usuarios a través de un formulario de ingreso, además se debe de permitir la creación de nuevos usuarios, asignar roles y relacionarlos con almacenes y productos dentro del proyecto.

Modelos de la aplicación

En esta clase nos centraremos en el desarrollo de los modelos usuariosExtendido, el modelo de almacenes y productos.



Django - Responses

Desarrollo Web con Python - Sesión 7

Responses en Django

Las peticiones en Django reciben una respuesta del servidor, la cual puede ser de diferentes tipos.

Responder con HTML	: return render()
Respuesta genérica	: return HttpResponse()
Redirección en la URL	: return HttpResponseRedirect(reverse('app:ruta'))
Mensaje Json	: return JsonResponse()
Respuesta de archivo	: return FileResponse()

File Response

En las aplicaciones web es muy común el manejo de archivos o la generación de estos a partir de las bases de datos. Hasta este punto hemos devuelto los siguientes objetos:

- render
- JsonResponse
- HttpResponse
- HttpResponseRedirect

Sin embargo también existe el método **FileResponse** que permite enviar archivos desde el servidor para que el cliente los pueda visualizar

Función de retorno de archivos

Para utilizar FileResponse se utiliza la siguiente funcion:

```
from django.http import FileResponse
response = FileResponse(open('archivo.pdf','rb'),as_attachment=True)
return response
```

Gracias