

PRACTICAS DE DJANGO

1) Instalar Django, revisar variables de entorno, comprobar en una consola que esta correctamente instalado. Consultar la versión. Crear un proyecto, lanzarlo y probar con el Servidor. Ejecutar el comando migrate y consultar en la BD las tablas creadas,

2) Modelos: Crear proyecto, aplicación y definir modelos dentro del contexto de una Biblioteca: con las siguientes entidades: Auto (nombre, apellidos, correo ...), Editor (nombre, domicilio, ciudad, provincia, país, web). Y los libros: donde un libro tiene uno o más autores, relación de muchos a muchos) y un único editor. La relación es: 1 editor - muchos libros (editor sería una clave foránea en los libros) y fecha de publicación. Definir modelos, migraciones y migrate.

Hacer modificaciones en el modelo, añadiendo algún campo: precio, ... etc,

Hacer pruebas a crear, modificar y borrar registros. Desde la consola: **python manage.py shell**

3) Diseño del modelo: Una **empresa** de venta telefónica dispone de un catálogo de productos clasificados por Categoría (nombre y descripción). Por ejemplo:

- | | |
|---------------|-----------------|
| • Categoría | Producto |
| • Informática | Portátil |
| • Informática | Ratón |
| • Papelería | Paquete folios. |

Los productos (nombre, precio, existencias, existencias mínimas), se surten con proveedores (nombre, empresa) que tienen que quedar registrados en el sistema. El funcionamiento de la tienda es el siguiente: Un cliente (DNI, nombre, dirección, país, fecha de alta, teléfono, email) llama por teléfono a la empresa y le recibe un Empleado (nombre, cargo, fecha de alta) que genera un pedido con todos los productos que pide. El pedido posteriormente se gestiona a través de una empresa de transportes (nombre). Pensar en las relaciones de las entidades, que campos serían necesarios y las relaciones: 1 a muchos y muchos a muchos. Del pedido tener en cuenta que registra el cliente, el empleado, la compañía de transporte, fechas: recepción, pedido, envío, lista de productos con su precio, cantidad y descuento, mantener también un campo de observaciones. Registrar la app en settings.py

4) Modelos. Funciones de agregado, consultas, etc. Utilizar el proyecto: **TUTORIAL_BIBLIO** ya tiene una BD cargada con datos.

Las consultas se pueden realizar desde la consola. Lo primero importar los modelos para poder trabajar con ellos. Probarlo en la consola: **python manage.py shell**

- Número de editores, autores, libros y almacenes

- Ver los nombres de los autores con una sentencia:
- Contar los libros de un determinado editor, ver primero los editores:
- Libros que contenga una C:
- El precio medio de los libros, suma total, mínimo, máximo:
- Calcular lo mismo del anterior, pero para los libros del 'editor 1'.
- Baremos de precios por cada almacén.

5) Vistas: Definir vistas (views.py) y los mapeos (en urls.py) :

- Definir una URL que permita enviar una operación sum / sub / mul / div y que reciba 2 números. Devolverá la operación realizada. La operación no válida se controla en la vista y mostrará un mensaje para ello.
- Idem de la anterior, pero con re_path recibe sum, sub, mul o div y 2 números de 1 o 2 cifras.
- Definir un conversor que compruebe el número de una matrícula y un DNI. Luego implementar la vista.
- Con alguno de los modelos anteriores enviar un id y devolver el objeto.
- Generar HTML (*ojo, como no hemos visto plantillas se genera con HTML en la vista*). Tabla, combo, lista. Con contenidos del modelo.

6.1) Formularios1: Implementar un proyecto con una vista y una plantilla para recuperar información de request y de las cabeceras META y mostrarlo con HTML. Mostrar los métodos / att de la request.

- Crear el proyecto y la app
- En settings.py añadir la app: **INSTALLED_APPS ...**
- La ruta de las plantillas:
- 'DIRS':[os.path.join(BASE_DIR,'templates')],
- Crear una carpeta **templates** dentro de la app.
- La información se tiene que mostrar con la petición: http://localhost:8000
- Mostrará una salida similar a esta:

Información de la Request	
CLAVE	VALOR
path	/
host	localhost:8000
full_path	/?info=000
seguro	NO seguro
Información de las cabeceras: META	
CLAVE	VALOR
ALLUSERSPROFILE	C:\ProgramData
ANDROID_HOME	C:\android\sdk
ANT_HOME	C:\apache-ant-1.9.4
APPDATA	C:\Users\INVES_Propietario\AppData\Roaming
AXIS2_HOME	C:\axis2
CATALINA_HOME	C:\apache-tomcat-8.5.6
CLASSPATH	C:\Program Files\Java\jdk1.8.0_65\lib\tools.jar;C:\mysql_conector\mysql-connector-java-5.1.13-bin.jar;
COMMONPROGRAMFILES	C:\Program Files (x86)\Common Files
COMMONPROGRAMFILES(X86)	C:\Program Files (x86)\Common Files
COMMONPROGRAMW6432	C:\Program Files\Common Files
COMPUTERNAME	WORKER-PC

6.2) Formularios2: Definir una calculadora con gestión de errores.

- Falta el número 2
- Falta la operación

Introduce los números y selecciona la operación:

Numero 1:

Numero 2:

Operación

Cuando estén bien todos los datos, se mostrará la operación así: $12 + 100 = 120$

6.3) Formularios basados en clases:

- Hay que mostrar la fecha por defecto: `datetime.date.today()`
- El texto: "Al menos 8 chars." es un texto de ayuda en el usuario
- Meter validaciones: en el campo de observaciones tiene que haber 4 palabras al menos.
- Al mostrar los errores en la plantilla, aplicar estilos.
- Mostrar en el desplegable unos cuantos países.

Información del usuario

¡Se requieren mínimo 4 palabras!

Nombre:

Usuario:
Al menos 8 chars.

Password:

Email:

Edad:

Fecha:

Recibir publicidad: ☒

Pais:

- ¡Se requieren mínimo 4 palabras!

Observaciones:

7) Panel de Administración: Probar la interfaz de administración con el modelo de Empresa. Crear el super usuario. Personalizar los formularios: filtros de búsqueda, jerarquías de fechas, campos de muchos a muchos, foreignkey. Ordenar los clientes por país y nombre. Ojo, registrar la app en settings.py, rellenar las clases Admin en admin.py

8) Archivos estáticos: utilizando el contenido de la carpeta practicas/estático crear una CSS y utilizar imágenes para crear esta página:



Implementar una nueva versión en otra plantilla, las fotos quedarán en vertical, pero los nombres de los ficheros los pasamos a través de la vista por el contexto y desde la plantilla definir variables con {% with 'valor' as var %} y concatenar la parte fija de la ruta.

10.1) Generar formatos: Definir un modelo para los datos de una persona: nombre y dos apellidos, dni, teléfono, email.

a. El sitio web va a tener las siguientes URLs:

```
generar/  
ver/  
exportar/pdf/  
exportar/csv/  
exportar/xml/
```

b. Instalar la librería django-import-export y probar a crear vistas para csv, json y excel

generar/ Genera una serie de registros de una forma más o menos aleatoria y los graba en la base de datos. ***Emite un mensaje con el número de registros generados.***

ver/ Visualiza los datos de la tabla de persona en una tabla de HTML.

LISTADO DE PERSONAS					
NOMBRE	APELLIDO1	APELLIDO2	DNI	EMAIL	TELEFONO
VVUHE	TOWHZ	RAUZO	51000000	webmaster@hotmail0.com	12345678
YEHD	ESUUD	ETFZC	51000001	webmaster@hotmail1.com	12345678
BNGVB	ASQMB	OSGFS	51000002	webmaster@hotmail2.com	12345678
OPSNB	GPBBE	QPJNR	51000003	webmaster@hotmail3.com	12345678
WELIR	XFCVD	HLYRG	51000004	webmaster@hotmail4.com	12345678
IBYAP	MICCY	AMDIK	51000005	webmaster@hotmail5.com	12345678

exportar/pdf Genera un pdf con el contenido de la tabla

LISTADO DE PERSONAS

NOMBRE	APELLIDO1	APELLIDO2	DNI	TELEFONO	EMAIL
VVUHE	TOWHZ	RAUZO	51000000	12345678	webmaster@hotmail0.com
YEHD	ESUUD	ETFZC	51000001	12345678	webmaster@hotmail1.com
BNGVB	ASQMB	OSGFS	51000002	12345678	webmaster@hotmail2.com
OPSNB	GPBBE	QPJNR	51000003	12345678	webmaster@hotmail3.com
WELIR	XFCVD	HLYRG	51000004	12345678	webmaster@hotmail4.com
IBYAP	MICCY	AMDIK	51000005	12345678	webmaster@hotmail5.com

exportar/csv Genera un csv con el contenido de la tabla

```
NOMBRE,APELLIDO1,APELLIDO2,DNI,TELEFONO,EMAIL
VVUHE,TOWHZ,RAUZO,51000000,12345678,webmaster@hotmail0.com
YEHD,ESUUD,ETFZC,51000001,12345678,webmaster@hotmail1.com
BNGVB,ASQMB,OSGFS,51000002,12345678,webmaster@hotmail2.com
OPSNB,GPBBE,QPJNR,51000003,12345678,webmaster@hotmail3.com
WELIR,XFCVD,HLYRG,51000004,12345678,webmaster@hotmail4.com
IBYAP,MICCY,AMDIK,51000005,12345678,webmaster@hotmail5.com
HYKES,BJXJK,ZOROI,51000006,12345678,webmaster@hotmail6.com
PVBSU,PHMFB,IOOWY,51000007,12345678,webmaster@hotmail7.com
GDRWK,PZPHR,HUVSG,51000008,12345678,webmaster@hotmail8.com
OABVW,WMHIW,RSBYL,51000009,12345678,webmaster@hotmail9.com
```

exportar/xml Genera un xml con el contenido de la tabla

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<personas>
  <persona>
    <nombre>VVUHE</nombre>
    <apellido1>TOWHZ</apellido1>
    <apellido2>RAUZO</apellido2>
    <dni>51000000</dni>
    <email>webmaster@hotmail0.com</email>
    <telefono>12345678</telefono>
  </persona>
  <persona>
    <nombre>YEHD</nombre>
    <apellido1>ESUUD</apellido1>
    <apellido2>ETFZC</apellido2>
    <dni>51000001</dni>
    <email>webmaster@hotmail1.com</email>
    <telefono>12345678</telefono>
  </persona>
```

10.2) Se puede hacer un formulario que rellene el usuario y generar un contenido en un PDF

11) VISTAS BASADAS EN CLASES: Implementar la gestión de libros utilizando vistas basadas en clases: Listado de libros, ver detalle, crear, modificar y borrar libros. Según los pantallazos adjuntos:

Primera página: Si se pulsa en "nuevo libro" aparece un formulario vacío para dar de alta un nuevo libro. Si pincha en un ISBN se muestran los detalles del libro y se muestran enlaces para borrar y editar un libro.

LISTADO DE LIBROS			
isbn1	libro django	34.00	Editor 1
isbn2	libro cobol	54.00	Editor 1
isbn3	libro python 3	24.00	Editor 1
isbn4	libro cobol avanzado	123.00	Editor 2
isbn5	libro C++	67.00	Editor 2
I.S.B.N.6	Programación C	44.67	Editor 2
isbn7	Tutorial Android	25.00	Editor 3
isbn8	Tutorial PHP	65.00	Editor 4
isbn9	Tutorial C Linux	35.00	Editor 5
isbn10	Manual de C	55.00	Editor 5
6868585	El Gran Libro de Android	45.00	Editor 1
87887	Java J2EE	89.00	Editor 1
332322	Telefonia iphone	45.00	Editor 3
677756454	Maven	78.00	Editor 1
88999888	PHP Avanzado 2	90.00	Editor 2
NUEVO LIBRO			

Pantalla de nuevo libro:

Ver libro:

NUEVOS LIBROS	TITULO: libro C++
Isbn: <input type="text"/>	ISBN: isbn5
Name: <input type="text"/>	Precio: 67.00
Price: <input type="text"/>	Editar Borrar Volver
Publisher: <input type="text"/>	
Authors: <div>Autor 2 edad:64 Autor 3 edad:33 Autor 4 edad:28 Autor 5 edad:49</div>	
<input type="button" value="GRABAR"/>	

Edición de Libros

Eliminación de Libros

NUEVOS LIBROS	BORRAR LIBROS
<p>Isbn: <input type="text" value="isbn5"/></p> <p>Name: <input type="text" value="libro C++"/></p> <p>Price: <input type="text" value="67,00"/></p> <p>Publisher: <input type="text" value="Publisher object (2) ▼"/></p> <p>Authors: <div><div>Autor 2 edad:64</div><div>Autor 3 edad:33</div><div>Autor 4 edad:28</div><div>Autor 5 edad:49</div></div></p> <p><input type="button" value="GRABAR"/></p>	<p>Estás seguro que quieres borrar el libro libro C++ precio: 67.00?</p> <p><input type="button" value="CONFIRMAR"/></p>

12) **ModelUser**: Tenemos dos usuarios.

- Admin: Nos manda al panel de administración
- Login: Hace login y habilita la session
- Logout: Cierra session

Al iniciar no hay un usuario identificado, por tanto muestra **AnonymousUser**. Las dos primeras operaciones son vistas en funciones, una libre y la otra el usuario se tiene que identificar.

superusuario: antonio / antonio

kraken / asd123_*

[Admin](#) | [Login](#) | [Logout](#)

[Operacion1 \(libre\)](#) | [Operacion2 \(protegida\)](#)

Paginas con vistas genéricas

[Pagina1 \(protegida\)](#) | [Escuchar canción \(protegida\)](#) | [Bajar canción \(protegida\)](#)

Información sobre el usuario

Usuario actual: AnonymousUser

Las otras 3, necesitan plantilla:

- Pagina1: Sólo requiere identificación
- Escuchar Canción: Se necesita el permiso: escuchar_cancion
- Bajar Canción: Se necesita el permiso: Bajar_cancion
- El usuario Kraken sólo tendrá permiso para escuchar canción. Y el super usuario podrá hacer todo.
- Las plantillas o vistas sólo muestran un mensaje sencillo.
- Crear un Model Canción, con los dos permisos en la clase interna Meta.
- El permiso de escuchar canción se otorga a Kraken desde el panel de administración.

13) Galería de imágenes: subir imágenes al servidor.

[Inicio](#) | [Galería](#) | [Administración](#)

GALERIA

Estas son las imágenes que tenemos en la BD.

En la Cumbre



En Andorra



En la parte de administración, posibilidad de crear fotografías con su título, fichero de imagen, etc.

En la parte pública se mostrarán formando una galería de imágenes.