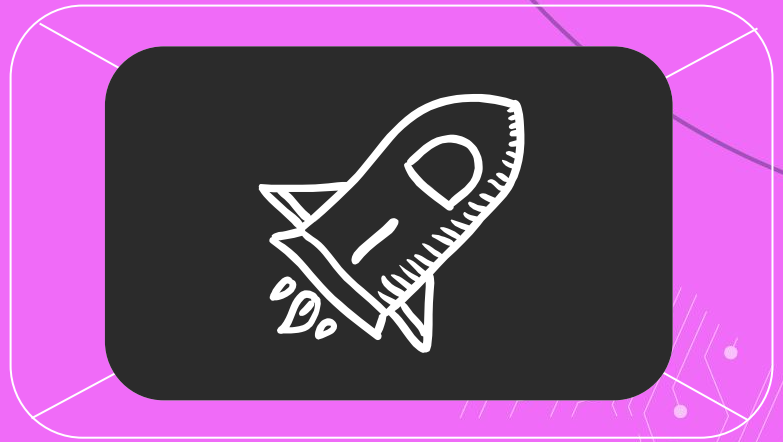
Gobierno del CHACO | Ministerio de la Producción y el Desarrollo Económico Sostenible

INFORMATORIO

Semana 10
¡Bienvenidos!



Temas de hoy

1. Introducción a Django Instalación y estructura de proyecto.
2. Introducción al Proyecto Final.

1

Introducción a Django

Cuando construimos aplicaciones web, siempre necesitamos de componentes similares: una manera de manejar la autenticación de usuarios, un panel de administración, formularios, subir archivos, etc.

Para agilizar el desarrollo de estos componentes existen los Frameworks, que nos sirven para que no tengamos que reinventar la rueda cada vez y podamos avanzar más rápido al construir una aplicación. 😉

Introducción a Django

¿Qué es Django?

Django es un **framework de aplicaciones web gratuito y de código abierto** (open source) **escrito en Python**. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.
Django te ayuda a escribir software que es:

Completo: Porque incluye y provee de todo lo necesario para que puedas desarrollar tu aplicación.

Versátil: Django puede ser (y ha sido) usado para construir casi cualquier tipo de sitio web — desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias.

Seguro: Django ayuda a los desarrolladores evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente.

Escalable: Django usa un componente basado en la arquitectura "shared-nothing" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario).

Mantenible: El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código.

Introducción a Django

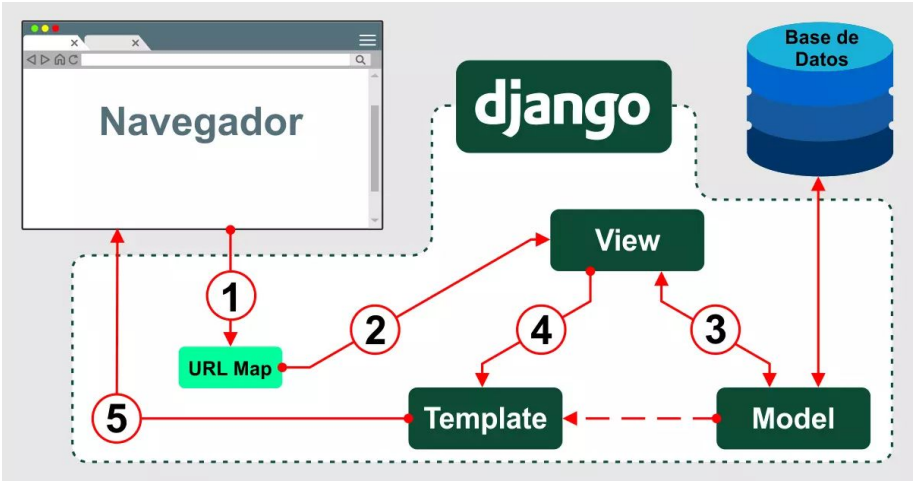
¿Por qué necesitas un framework?

Para entender para qué sirve realmente Django, necesitamos fijarnos en cómo funcionan los servidores. Lo primero es que el servidor necesita enterarse de que tú quieres que te sirva una página web.

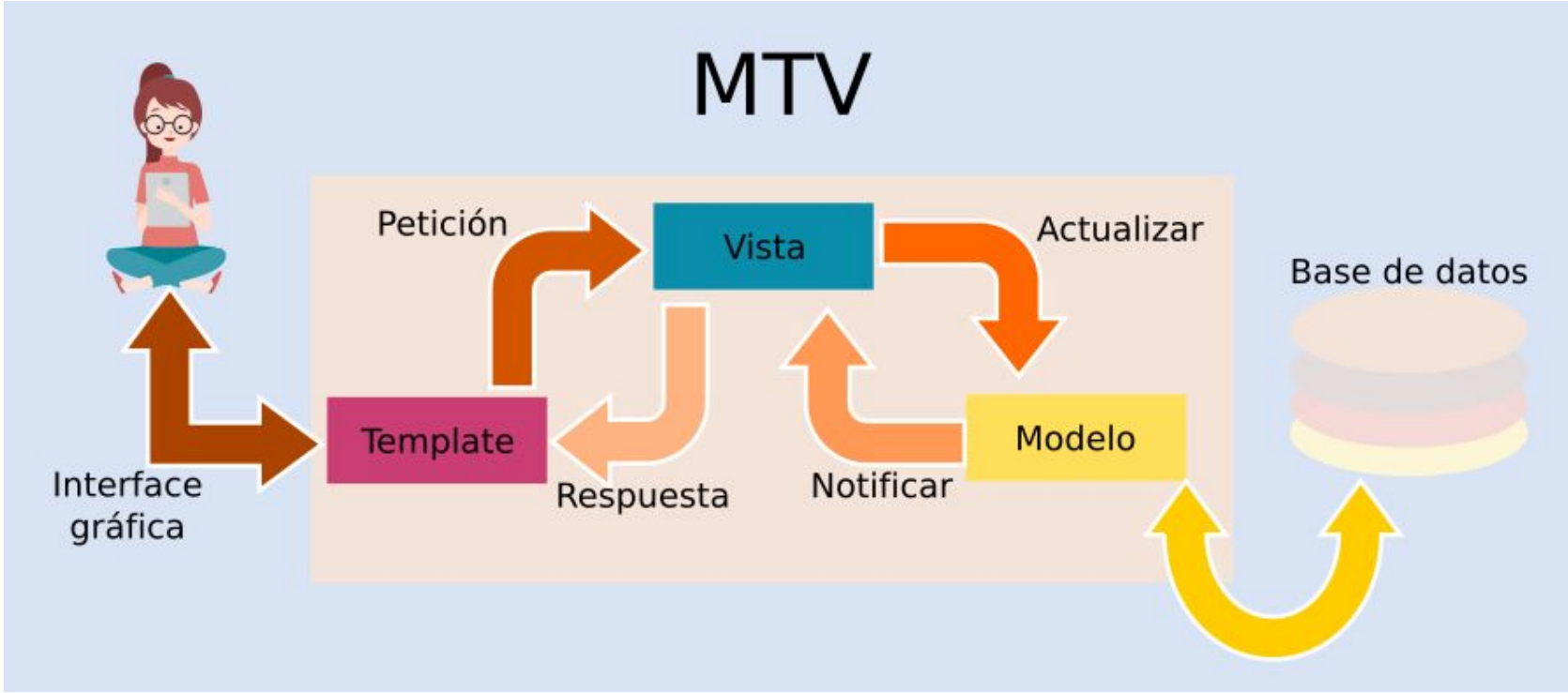
Cuando llega una petición a un servidor web, esta es pasada a Django. Toma primero la dirección de la página web e intenta averiguar qué hacer con ella. Django comprueba los patrones de arriba hacia abajo y si algo coincide entonces Django le pasa la solicitud a la función asociada (que se llama view (vista)).

En la función de view (vista) se puede mostrar información o consultar información desde la BD.

Luego la vista genera una respuesta y Django puede enviarla al navegador del usuario.



Modelo Vista Template



Modelo Vista Template

Django usa una arquitectura propia que se llama MVT, Modelo Vista Template que se basa en la MVC pero con sus propias peculiaridades.

Modelo:
Controla el comportamiento de los datos. Define los datos almacenados. Se encuentra en forma de clases de Python(el modelo es un objeto). Cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros. Posee métodos para la gestión de los datos.

Vista:
Se presenta en forma de funciones o clases en Python, su propósito es determinar que datos serán visualizados.

La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.

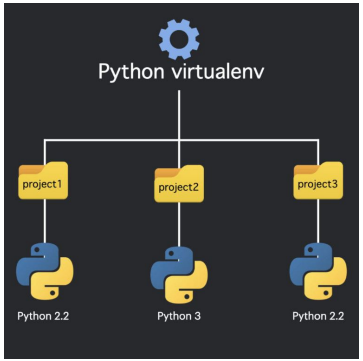
Template:
Es básicamente una página HTML con algunas etiquetas extras propias de Django, no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, etc).

La plantilla(Template) recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas permiten una presentación lógica de los datos.

Entorno virtual

¿Qué es el entorno de desarrollo de Django?

El entorno de desarrollo, es una instalación de Django en tu computadora que podés usar para desarrollar y probar apps Django antes de desplegarlas al entorno de producción.



Django es extremadamente flexible en términos de cómo y dónde puede instalarse y configurarse.
Django puede ser:

- Instalado en diferentes sistemas operativos.
- Configurado para usar una de entre varias bases de datos, que pueden también necesitar ser instaladas y configuradas por separado.
- Ejecutado en el entorno Python del sistema principal o dentro de entornos virtuales Python separados.

Si tenés proyectos con distintas versiones de Django instaladas, normalmente estas aplicaciones se ejecutan en Python dentro de entornos virtuales Python independientes. De esta forma se habilitan múltiples entornos Django diferentes en la misma computadora.

Django

Tanto para la creación del entorno virtual, como para profundizar en las características del framework Django, vas a tener a tu disposición el material complementario, donde tendras un paso a paso bastante minucioso.
Lo que si vamos a ver en esta clase (que también lo vas a tener en el material complementario) es la estructuración y comienzo del proyecto con Django.

Para crear el proyecto tenemos que tener el entorno activado y estar posicionados dentro de la carpeta donde deseamos que se genere el proyecto y, en nuestra consola escribir la siguiente línea de comando:

```
django-admin startproject mysite
```

En este caso, “mysite” vendría a ser el nombre de tu proyecto, por lo que hay que reemplazar ese nombre.

Ahora deberías tener una estructura de directorios parecida a esta:

```
mysite
├──manage.py
├──mysite
│       settings.py
│       urls.py
│       wsgi.py
│       __init__.py
└──requirements.txt
```

Ahora veamos qué función cumple cada uno de estos archivos...

Estructura de Proyecto de Django

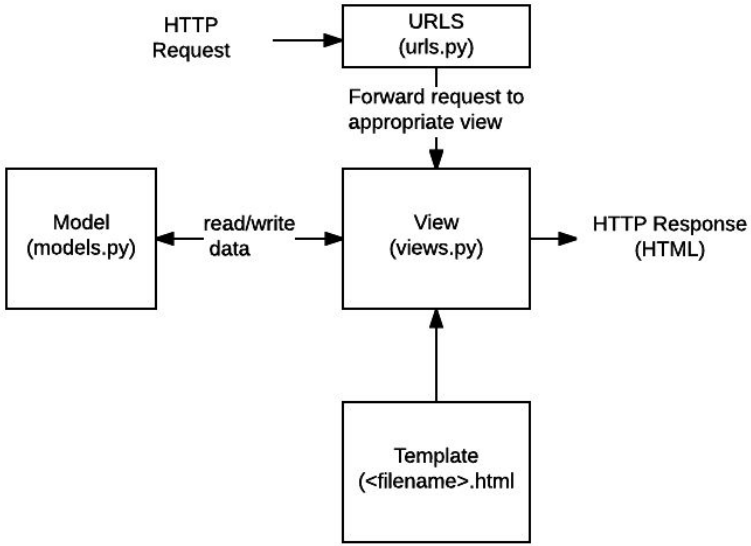
manage.py Es un script que ayuda con la administración del sitio. Con él podremos iniciar un servidor web en nuestra computadora sin necesidad de instalar nada más, entre otras cosas.

settings.py Contiene la configuración de tu sitio web. Es el primer archivo que modificaremos para poder arrancar nuestra aplicación por primera vez.

urls.py Es donde se definen las urls de nuestro proyecto. Contiene una lista de los patrones utilizados por *urlresolver*, que utiliza Django para saber que vista o template mostrar.

wsgi.py Es el mediador responsable de transmitir la comunicación entre un servidor web y una aplicación web Python. Acá configuraremos la info de conexión al servidor.

__init__.py Es el archivo que indica al intérprete de Python que el directorio *package* contiene un módulo, y que debe tratarlo como tal (es decir, hacer que sea posible importar los archivos como parte del módulo).



Configuración de Proyecto en Django

Ahora que ya tenemos creado nuestro proyecto y que conocemos un poco más acerca de su estructura y funcionamiento, vamos a hacer algunos cambios en mysite/settings.py. Abrimos el archivo con VSC y buscamos la línea que contiene BASE_DIR y modificamos para que detecte la URL base de nuestro proyecto (este punto puede realizarse de diversas formas, pero vamos a tomar como referencia una de las formas más comunes):

```
BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath(_
_file__)))
```

Luego verificamos que la línea que contiene la variable ***DEBUG*** esté configurada en ***True*** para mostrar los errores en caso que los hubiera.
Al subir nuestro sitio online tenemos que configurar esta variable en ***False***.

Buscamos el diccionario DATABASES y configuramos allí la ruta y/o credenciales de acceso a la base de datos:

Si usamos sqlite3 tiene que quedar definido así:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME':
os.path.join(os.path.dirname(BASE_DIR),'db_sqlite3'),
    }
}
```

Si usamos PostgreSQL lo definimos de otra forma:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'nombre_database',
        'USER': 'nombre_usuario_db',
        'PASSWORD': 'usuario_password',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Configuración de Proyecto en Django

Luego vamos a ir a la línea que contiene LANGUAGE_CODE y TIME_ZONE, para modificarlas y configurar idioma y zona horaria:

```
LANGUAGE_CODE = 'es-ar'

TIME_ZONE = 'America/Argentina/Buenos_Aires'
```

Además, tenemos que configurar las rutas de acceso a archivos estáticos para que los elementos como imágenes/librerías/etc. se visualicen correctamente:

```
STATIC_URL = 'static/'
STATICFILES_DIRS=(os.path.join(os.path.dirname(BASE_
DIR), 'static')),

MEDIA = '/media'
MEDIA_ROOT =
os.path.join(os.path.dirname(BASE_DIR),'media')
```

Por último, configuramos el campo DEFAULT_AUTO_FIELD solicitado por la configuración de Django para evitar errores al momento de ejecutar las migraciones para la creación de tablas en la base de datos. Agregamos esta línea al final del archivo settings.py:

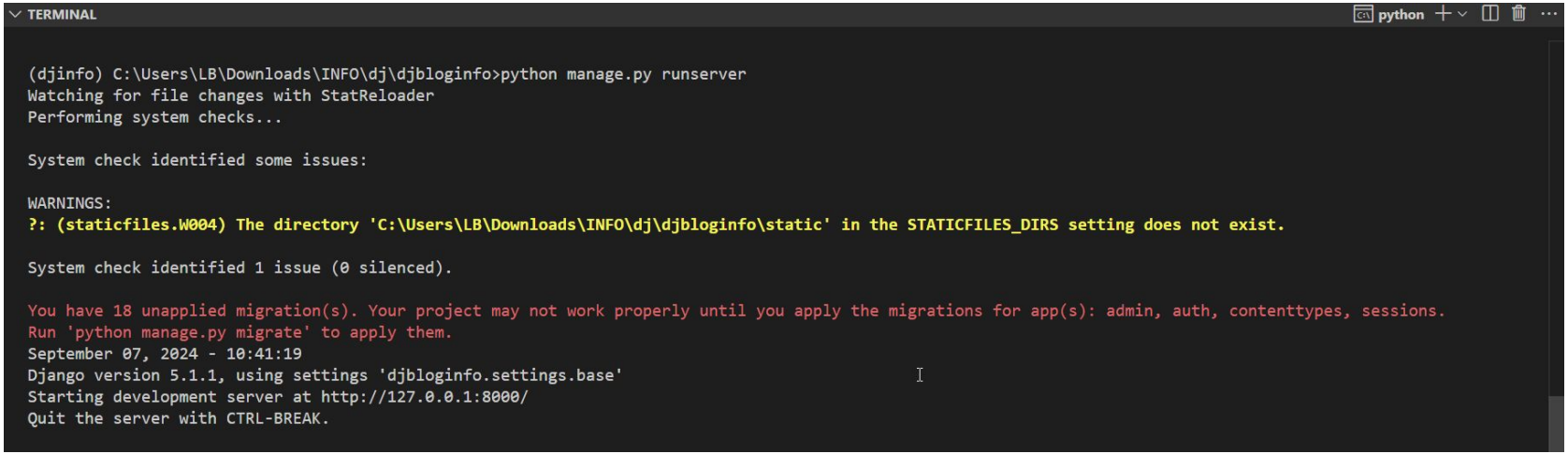
```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Con esto finalizamos la configuración inicial del proyecto en Django.
Para ejecutar nuestro proyecto y “runearlo” en el servidor, ejecutamos en la terminal el siguiente comando:

```
python manage.py runserver
```

Configuración de Proyecto en Django

Si todo se ejecuta correctamente, en la terminal vas a ver este mensaje donde te indica la URL donde se puede visualizar tu proyecto:



Por el momento ignoramos el mensaje en rojo, que nos indica que hay migraciones sin aplicar. Más adelante explicaremos y avanzaremos sobre el concepto de modelos y migraciones

Configuración de Proyecto en Django

Si ingresas a la URL que indica la terminal, en el navegador tenés que visualizar esta pantalla:

2

Introducción al Proyecto Final

Conozcamos ahora un poco más del proyecto que tendremos que completar para la parte final de este curso.

Proyecto Final

- **Objetivo**: Tener una aplicación web desarrollada utilizando DJANGO y los conceptos abordados en el curso, de acuerdo a los requerimientos indicados por profesores de cada comisión.

Modalidad: El trabajo final podrá ser desarrollado únicamente en forma grupal.
La cantidad de integrantes del equipo no debe superar 6 personas (sin embargo, cada profe podrá delimitar esta cantidad en su comisión).

Fecha de Presentación: Se estará informando la fecha de entrega, tanto en clases, como en el campus. Cada trabajo deberá poder ser accedido desde repositorios Github y el servidor Pythonanywhere, cuya URL se subirá en la tarea correspondiente dentro del aula virtual.

Proyecto Final

- Para el proyecto final nos enfocaremos en el desarrollo de una aplicación web realizada en Django con bases de datos, específicamente un Blog orientado a promocionar noticias o artículos. El tema central estará abordado en cada comisión.

Para el desarrollo del proyecto primero detallaremos cuál será el alcance, que pantallas y que tablas son las que se incluirán en esta primera versión del proyecto.

Funcionalidades Mínimas

Para la versión mínima del proyecto, se solicita que se incluyan las siguientes funcionalidades:

- El acceso de diversos perfiles (admin, registrado, etc).
- Cargar un nuevo post, eliminar post publicados (con diversas restricciones).
- Comentar posts por parte de usuarios con perfil *registrado*. Un usuario debe estar autenticado y tener perfil *registrado* para poder comentar.
- Login a usuario tipo admin y registrado.
- Filtrar post por fecha, categoría de post y comentarios recibidos.

Definición de estructura - Layouts

Para definir el flujo de pantallas que se incluirán para cumplir con las funcionalidades solicitadas pueden utilizar las siguientes herramientas :

Figma

POP

Canva

Independientemente de la herramienta, lo más importante del proyecto, es que todas las funcionalidades se ejecuten correctamente, por lo que, en principio, no es necesario detenerse mucho en el aspecto visual. Luego se va agregando progresivamente los detalles de diseño hasta llegar al boceto final.

Lo que vimos hoy 😉

Hasta acá pudimos ver la introducción y configuración inicial de un proyecto en Django.

Vimos algunas pautas a tener en cuenta y, superficialmente el comienzo del proyecto.

Te recordamos que en el material complementario, vas a encontrar a mayor profundidad y paso a paso, la base del proyecto inicial. Desde la creación del entorno virtual, hasta la finalización del proyecto, pasando por sus diversas etapas.

En este punto vas a tener que usar las metodologías ágiles para que la integridad del proyecto y los tiempos se respeten correctamente.

¡Nos vemos En la próxima clase!