

# Ejercicios POLIMENTES

## Capacitación

### Práctica 10082021

Cada vez que creamos una nueva app en nuestro proyecto debemos registrarla en nuestras configuraciones, en el archivo settings.py en la sección de “Installed Apps”.

Dentro de la carpeta de nuestra app también se genera la carpeta “migrations” donde se guardarán los archivos que creamos al generar una migración. El término migración hace referencia a cuando con sintaxis de Python con ayuda del ORM de Django hacemos alguna modificación o actualización a nuestra base de datos.

Los documentos generados en una app son los siguientes

admin.py → donde se registran los modelos/tablas para que sean incluidos al administrador incluido en Django.

apps.py → donde se hace referencia a la aplicación con su nombre nativo y sus configuraciones

models.py → en él es donde se genera un modelo, un modelo es una clase de Python que representa a una tabla en la base de datos. Dentro de este archivo se deben encontrar todas las tablas que vayamos a tener y deseamos que existan en nuestra DB.

tests.py → Django incluye código para poder hacer pruebas y testeos de nuestra app, de nuestros modelos creados y que lo hayan sido correctamente, que la lógica sea la adecuada, etc.

Views.py → recordemos que en el modelo vista controlador tenemos la capa de modelo y la capa de vista que es la parte visual, sin embargo aquí ya no es así. Básicamente este documento contendrá la lógica de nuestra app, el código que reflejará la lógica de la creación de un usuario por ejemplo o de uno de los recursos de nuestra app (por ejemplo, si nuestro proyecto es de una biblioteca y nuestra app se llama “libro” pues la creación de un libro).

#### MODELOS

Los modelos son la representación de las tablas de la DB pero representados a través de sintaxis propia de Python y del framework de Django.

Se basan en clases que son tal cual la representación de las entidades (tablas) de la DB

#### Modelo vista-Controlador

Tenemos un usuario que interactúa con la parte visible, de ahí el nombre “vista” la cual puede responder porque esta conectada a un controlador que toma esa acción y que ya tiene indicado qué es lo que va a hacer (modificaciones en las bases de datos) a través de su conexión con el modelo que es quien realmente accede a la base de datos y le retorna la respuesta entonces al controlador. Este envía la respuesta que obtuvo del modelo a la vista y la vista la manda de una manera entendible al usuario.

#### Modelo vista- Template

En este caso el usuario hace una interacción a través de un sitio web que es capturada por una URL la cual a su vez está conectada a una vista. Esta vista contiene la lógica DE PROGRAMACIÓN, EL CÓDIGO que va a servir para recibir la interacción que el usuario ha hecho y tiene asociado un modelo ya que es a través de la interacción y en el código de la vista donde está indicada la acción que se quiere llevar a cabo con la base de datos y es por eso que a través del modelo lo hace. El modelo retorna su respuesta a la vista y la vista tendrá configurada también en DONDE se verá la respuesta y lo manda entonces al template donde es que está definido cómo es que se va a apreciar todo y finalmente es el template quien se conecta al usuario y le muestra lo que desea.

Entonces, en este modelo la vista únicamente contiene la lógica de programación y dónde es que se va a mostrar. El cómo se va a mostrar al usuario lo contiene el template que es quien interactúa con el usuario.

