Introducción a Django

Django

Elizabeth León Guzman, Ph.D.
eleong@unal.edu.co

Carlos Andrés Sierra, M.Sc.
casierrav@unal.edu.co

Joseph Gallego, M.Sc.
jagallegom@unal.edu.co

Jhon López, M.Sc.(C)
jalopezfa@unal.edu.co

Grupo de Investigación en Minería de Datos – (MIDAS)
Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia

Agenda

Frameworks (Marcos de Trabajo)

Un marco de trabajo es un conjunto de funcionalidades genéricas predeterminadas basadas en:

- Herramientas
- Librerías
- Buenas prácticas

También se conoce como un eco-sistema que permite desarrollar aplicaciones de una manera rápida y sencilla, especialmente por la reutilización de código.

Uso Recomendado

- Cuando se está iniciando un nuevo proyecto web con funcionalidades comunes: no reinventar la rueda.
- Cuando no se tiene demasiada experiencia en desarrollo y se quiere hacer un desarrollo ágil con baja cantidad de código.
- Cuando se quiere mantener un estándar de desarrollo en el equipo de trabajo siguiendo prácticas recomendadas.

Recuerde

El uso de un *framework* depende del proyecto, encontrará algunos casos donde quizás usarlo no es la mejor opción.

Agenda

Modelo-Vista-Controlador

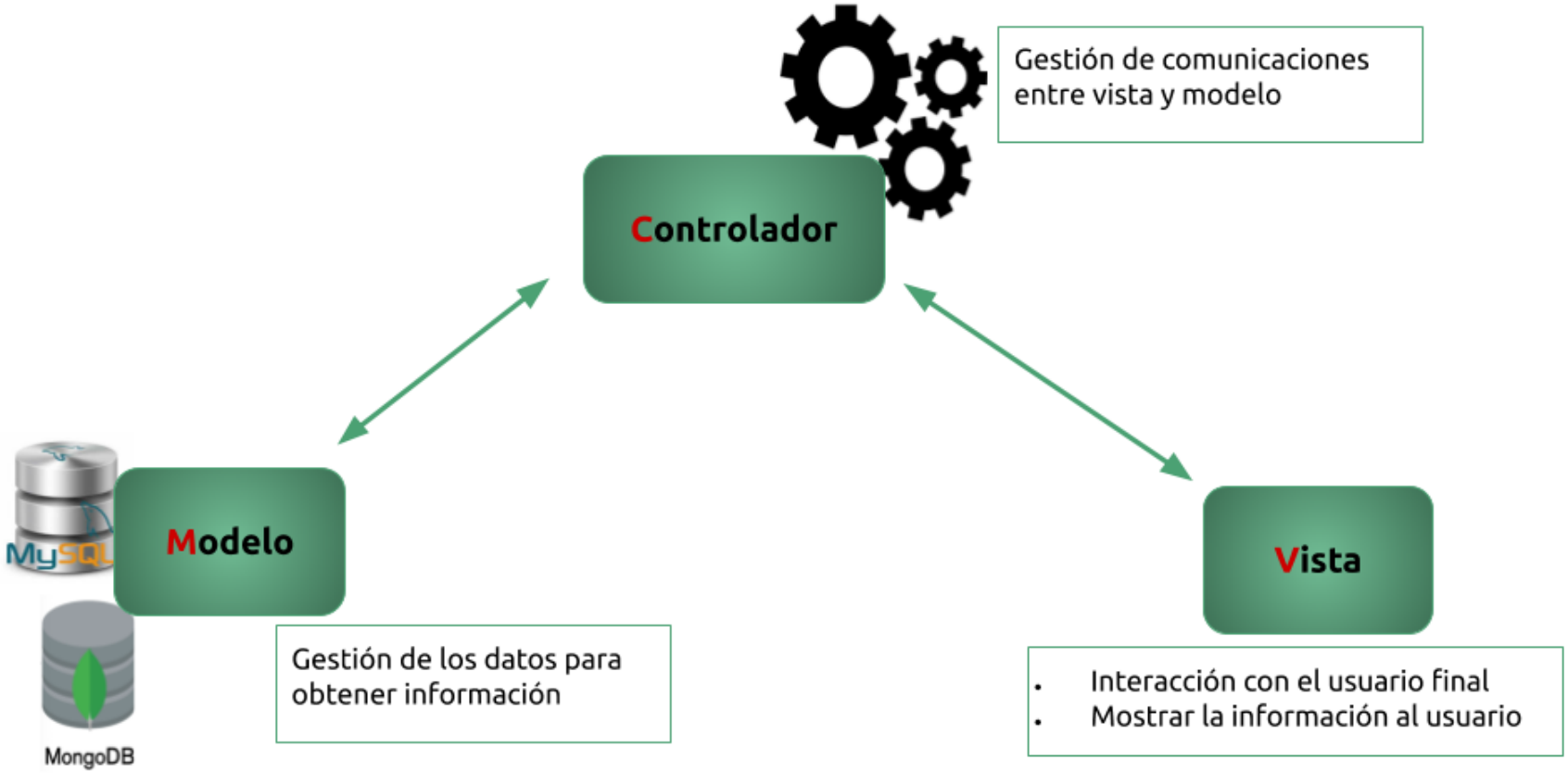


Figure: Esquema Básico del MVC

Flujo de Información

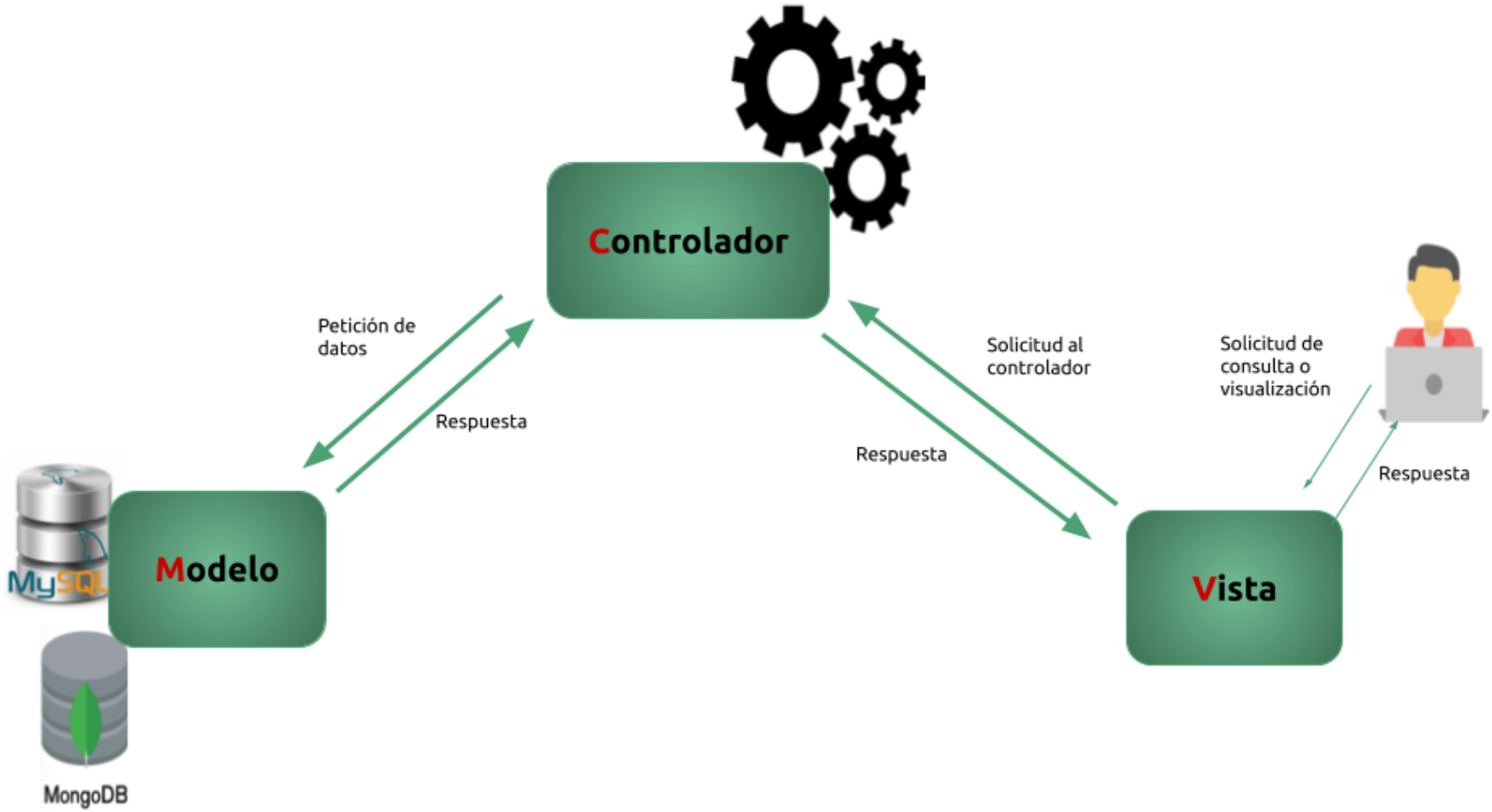


Figure: Peticiones en el MVC

Ventajas

- Muy utilizado en web porque flexibiliza el desarrollo por componentes.
- Permite desarrollo desacoplado, ágil, y organizado...además de fácil de mantener.
- Ayuda al mapeo de las URLs, brindando mayor seguridad y mejor indexación para los buscadores.
- Se pueden reutilizar componentes de manera simple, contribuyendo a la escalabilidad.

Agenda

Django

Django es un framework basado en Python, lanzado en el 2005, aunque sus inicios se remontan al año 2003.

Basado en un patrón similar al MVC, utiliza una estructura denominada MTV (Model-Templates-View), en español: *Modelo-Plantilla-Vista*. Aunque la filosofía es similar a MVC, en **Django** se sustituye la *Vista* por *Plantillas*, y el *Controlador* es la *Vista*.

Sitio Web Oficial

DjangoProject.com

Casos de Éxito

Figure: Sitios Web más famosos hechos con Django

Instalación de Django

Django se instala como cualquier otro paquete de Python, haciendo uso de *pip*. Se aclara que previamente se debe tener instalado Python $\geq$ 3.6, y ajustado para que el comando *pip* sea reconocido por el sistema operativo.

```
(python -m) pip install Django==3.1.2
```

PyPI

En el sitio web de PyPI encontrará la información de las últimas versiones de los paquetes de Python.

Si falla pip, acá tiene una opción...

curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py

Agenda

Creación de Proyectos de Django

En Django un proyecto se define como un conjunto de aplicaciones (o componentes) que hacen parte de un desarrollo web completo.
Para crear un proyecto se utiliza la siguiente instrucción:

```
django-admin startproject nombre_proyecto
```

Django fue originalmente diseñado para trabajar con bases de datos relacionales. Es por ello que mantiene un proceso de migraciones para mantener un control sencillo sobre los modelos. Para hacer las migraciones existen las siguientes instrucciones:

```
python manage.py migrate
python manage.py makemigrations
```

Estructura de un Proyecto de Django

Cuando se crea un proyecto de Django, se generan de manera automática los siguientes archivos:

- `manage.py`:
- `nombre_proyecto`: Carpeta con los siguientes archivos.
  - `__init__.py`: define el contenido de la carpeta como un módulo de Python.
  - `asgi.py`: permite asociar el proyecto para ser lanzado en un canal asincrónico.
  - `settings.py`: configuraciones generales del proyecto en Django.
  - `urls.py`: Listado de urls definidas para ser accedidas por el usuario en el MTV.
  - `wsgi.py`: permite asociar la aplicación para ser lanzada automáticamente con el servidor Apache.

Configuración de un Proyecto de Django

En el archivo llamado `settings.py` se pueden configurar los siguientes elementos:

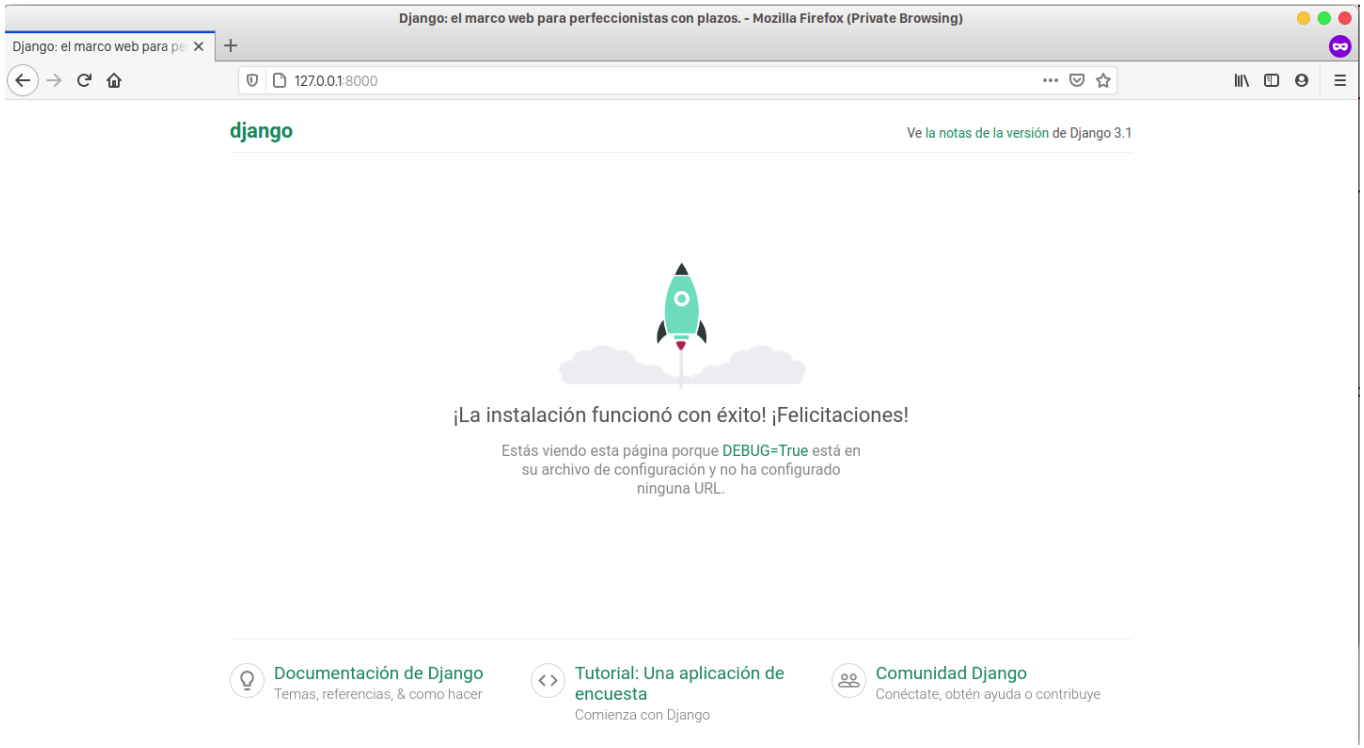
- Aspectos de seguridad y acceso a las aplicaciones.
- Aplicaciones (componentes) del proyecto.
- Credenciales de la base de datos SQL.
- Ajuste de valores para la internacionalización.
- Definición de la ubicación para archivos estáticos.

Puesta en Marcha del Proyecto en Django

Para colocar el proyecto de Django en el servidor local (*localhost*, *127.0.0.1*), solo se requiere ejecutar la siguiente instrucción:

```
python manage.py runserver
```

Si va a la dirección *http://127.0.0.1:8000* verá lo siguiente:



Agenda

Aplicación Vs. Proyecto

En Django se considera una aplicación como un componente, el cual bien puede ser un portal web, o un módulo de una gran aplicación web.

La importancia de trabajar con la noción de múltiples aplicaciones es poder trabajar de manera desacoplada, distribuida, y con flexibilidad.

Cada aplicación maneja de manera independiente el MTV, es decir, tiene su propio mapeo, sus propias plantillas, e incluso su propio modelo de datos.

Creación de Aplicaciones

En Django una aplicación se puede crear usando la siguiente instrucción que usa el archivo principal del proyecto `manage.py`:

```
python manage.py startapp nombre_aplicacion
```

De manera similar al proyecto, en Django se pueden hacer migraciones independientes para cada aplicación de acuerdo al modelo de datos que se le defina. Para hacer las migraciones de aplicaciones existen las siguientes instrucciones:

```
python manage.py migrate nombre_aplicacion
python manage.py makemigrations nombre_aplicacion
```

Estructura de la Aplicación

Cuando se crea la aplicación se genera una carpeta homónima que contiene los siguientes archivos:

- `__init__.py`: define el contenido de la carpeta como un módulo de Python.
- `admin.py`: en donde se gestiona el administrador por defecto de los modelos.
- `apps.py`: en donde se ajustan parámetros de configuración de la aplicación.
- `models.py`: en donde se define el modelo de datos de la aplicación.
- `tests.py`: en donde se colocan las pruebas unitarias de la aplicación.
- `views.py`: en donde se relacionan las vistas de la aplicación.

Gracias!

¿Preguntas?