

DJANGO (FRAMEWORK)

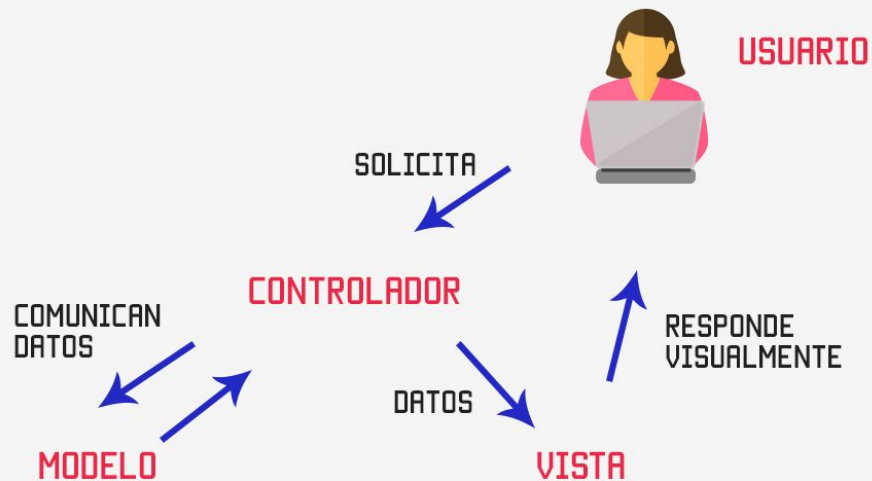
The Django logo, which features the word "django" in a white, lowercase, sans-serif font, centered within a dark green semi-circular shape.

django

Grupo 5:

- **Brenda Godoy**
- **Fernando Gardeñes**
- **Facundo Sosa**
- **Liza Strappini**

INTRODUCCIÓN



- Django es un framework de desarrollo web de código abierto, escrito en Python
- Desarrollado originalmente para gestionar páginas web orientadas a noticias de la World Company de Lawrence, Kansas
- Liberado al público bajo una licencia BSD en julio de 2005
- Nombrado en alusión al guitarrista de jazz gitano Django Reinhardt
- Meta fundamental: facilitar la creación de sitios web complejos.

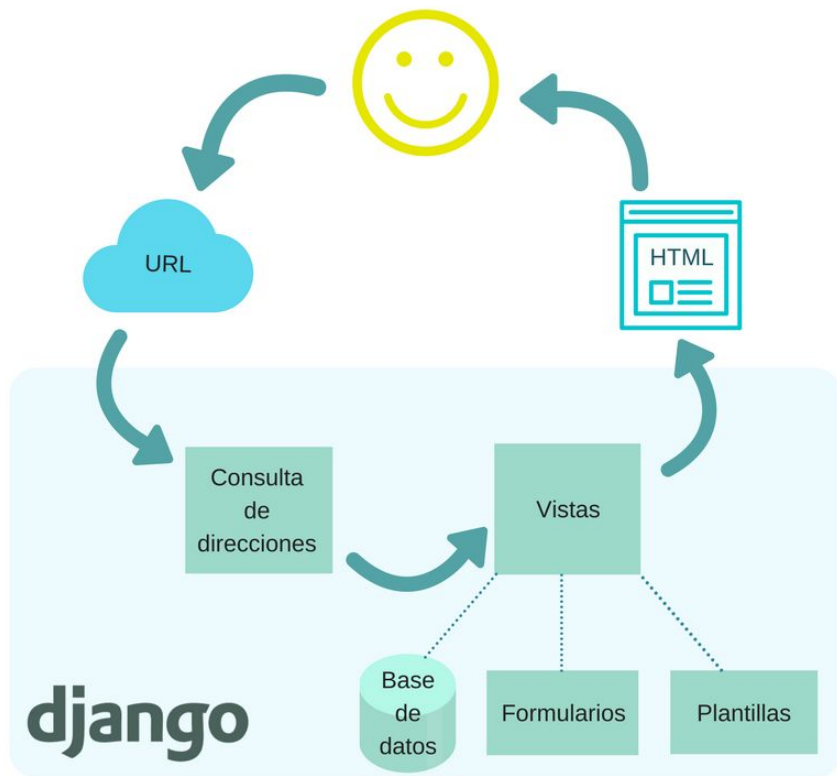
CARACTERÍSTICAS

- Aplicación administrativa incorporada

- ▶ Crear, actualizar y eliminar objetos de contenido
- ▶ Administrar varias páginas a partir de una misma instalación
- ▶ Llevar un registro de todas las acciones realizadas sobre cada objeto de contenido
- ▶ Proporcionar una interfaz para administrar los usuarios y los grupos de usuarios.

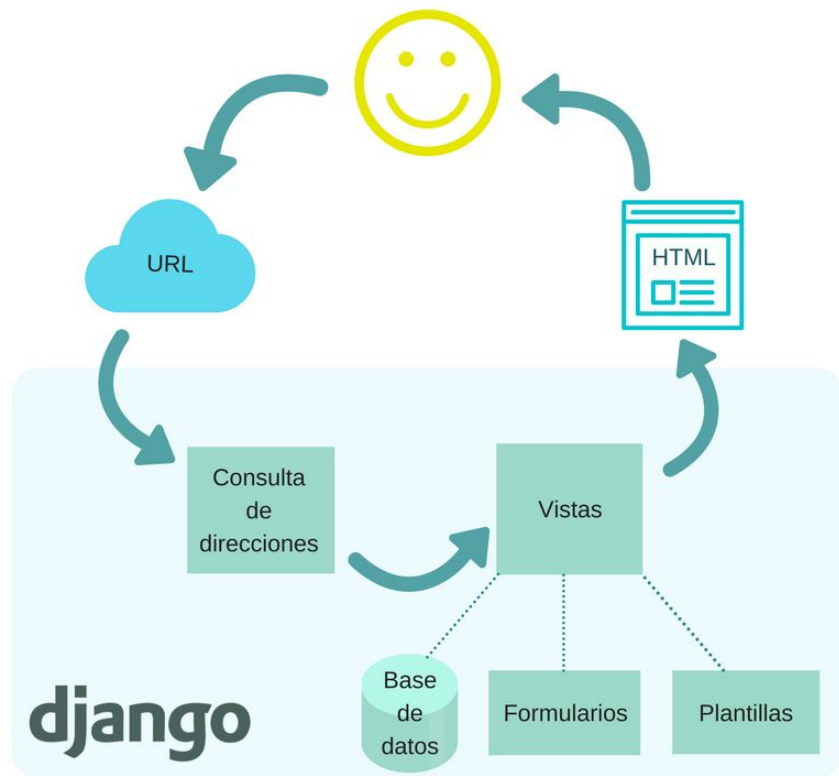
- Pone énfasis en la reusabilidad, conectividad y extensibilidad de componentes, y el principio «DRY».

- Se basa en el patrón de diseño conocido como modelo-vista-controlador (MVC).



CARACTERÍSTICAS

- Mapeador objeto-relacional. Orientado a P00
- Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django.
- API de base de datos robusta.
- Sistema de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Sistema extensible de plantillas basado en etiquetas.
- Sistema "middleware" para desarrollar características adicionales.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa



¿POR QUÉ USAR DJANGO?

- Su ORM abstrae la necesidad de escribir consultas SQL para crear tablas y consultar datos

```
class Review(models.Model):
    title = models.CharField(max_length=25)
    comment = models.TextField()
    name = models.CharField(max_length=20)
    created = models.DateTimeField(auto_now_add=True)
    modified = models.DateTimeField(auto_now=True)
    product = models.ForeignKey(
        Product, related_name="reviews", on_delete=models.CASCADE)
    user = models.ForeignKey(
        get_user_model(), related_name="reviews", null=True,
        on_delete=models.SET_NULL)
```

- Cuenta con el django admin panel que se puede usar para realizar operaciones CRUD en los modelos y posee un sólido sistema de permisos para restringir el acceso a los datos



- Ofrece seguridad ante los ataques más comunes como inyecciones SQL, Clickjacking y otros tan sólo agregando el middleware o etiqueta de plantilla correspondiente

```
<form method="post">{% csrf_token %}
```

- Incorpora más de 4000 paquetes de librerías
 - Django-haystack (Para búsquedas complejas)
 - Django-watson (Búsquedas)
 - DRF (REST)
 - Graphene (GraphQL)
 - Django-rest-auth (Autenticación)
 - Django-allauth (Autenticación)
 - Django-filter (Búsquedas)
 - Django-storage (AWS storage)
 - Django-braces (Funciones comunes)

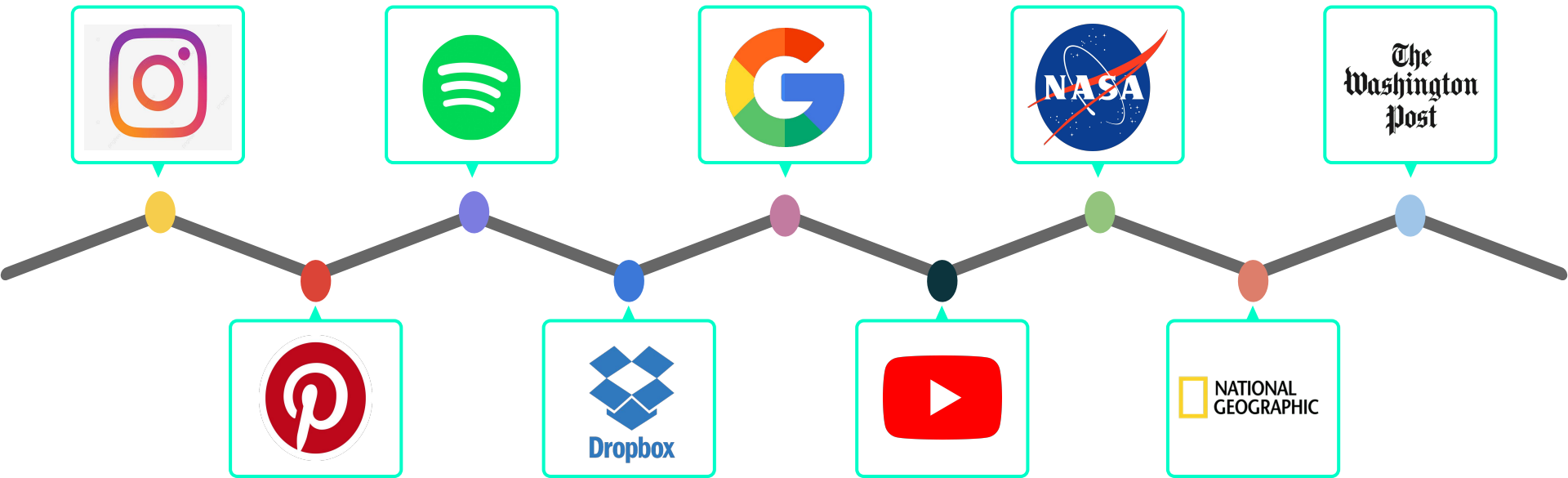
VENTAJAS

- Fácil mapeo de vistas y url: Django utiliza algo que se llama URLconf (configuración de URL). que es un conjunto de patrones que Django intentará comparar con la URL recibida para encontrar la vista correcta.
- Gratuito y de código abierto
- Simplicidad
- Increíblemente rápido
- Altamente seguro
- Escalable
- Con todas las funciones
- Fácil gestión
- Flexibilidad y versatilidad

DESVENTAJAS

- Django es un Framework antiguo que quedó algo lento en un mundo de frameworks de Javascript que evoluciona muy rápido.
- Aprender a usar el ORM con modelos y consultas, el middleware, las vistas, DRF (para las APIs) o el sistema de plantillas, el manejador de urls, traducción de cadenas de texto permite ahorrar código pero lleva mucho tiempo.
- Es más adecuado para proyectos de gama alta (full-stack) ya que sus numerosas funciones complican un proyecto simple.

¿QUÉ SITIOS WEB USAN DJANGO?



PRIMEROS PASOS CON DJANGO

INSTALACIÓN

```
PS C:\> mkdir ProyectoDjango

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          17/10/2022      12:44             ProyectoDjango

PS C:\> cd ProyectoDjango
PS C:\ProyectoDjango> python -m venv .
PS C:\ProyectoDjango> .\Scripts/Activate.ps1
(ProyectoDjango) PS C:\ProyectoDjango> python -m pip install django
Collecting django
  Using cached Django-4.1.2-py3-none-any.whl (8.1 MB)
Collecting asgiref<4,>=3.5.2
  Using cached asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting tzdata
  Downloading tzdata-2022.5-py2.py3-none-any.whl (336 kB)
----- 336.7/336.7 kB 6.9 MB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.3-py3-none-any.whl (42 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.2 sqlparse-0.4.3 tzdata-2022.5

[notice] A new release of pip available: 22.2.2 -> 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(ProyectoDjango) PS C:\ProyectoDjango>
```

Crear Entorno Virtual

python -m pip install virtualenv

Activar Entorno Virtual

activate, activate.bat, Activate.ps1

En Unix o MacOS: source .\bin\activate

Instalar Django

PRIMEROS PASOS CON DJANGO

CREACIÓN DEL PROYECTO

```
(ProyectoDjango) PS C:\ProyectoDjango> mkdir src
```

```
Directorio: C:\ProyectoDjango
```

Mode	LastWriteTime	Length	Name
d-----	17/10/2022 13:11		src

```
(ProyectoDjango) PS C:\ProyectoDjango> cd src
```

```
(ProyectoDjango) PS C:\ProyectoDjango\src> django-admin startproject primerProyectoDjango
```

```
(ProyectoDjango) PS C:\ProyectoDjango\src> cd primerProyectoDjango
```

```
(ProyectoDjango) PS C:\ProyectoDjango\src\primerProyectoDjango> dir
```

```
Directorio: C:\ProyectoDjango\src\primerProyectoDjango
```

Mode	LastWriteTime	Length	Name
d-----	17/10/2022 13:11		primerProyectoDjango
-a----	17/10/2022 13:11	698	manage.py

Crear Proyecto
usando **django-admin**

¿Qué es **django-admin**?
Es una utilidad de
línea de comandos de
Django para tareas
administrativas

Archivos creados por el
comando **startproject**

django-admin startapp

¿QUÉ ES MANAGE.PY? Y SUS COMANDOS PRINCIPALES

manage.py: Se crea automáticamente en cada proyecto de Django. Hace lo mismo que **django-admin** pero también “setea” la variable de entorno **DJANGO_SETTINGS_MODULE** para que apunte al archivo de configuración del proyecto, osea a settings.py.

- python manage.py makemigrations
- python manage.py migrate
- python manage.py createsuperuser
- python manage.py runserver

EJECUCIÓN DEL PROYECTO DJANGO

Iniciar el servidor

```
(ProyectoDjango) PS C:\ProyectoDjango\src\primerProyectoDjango> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations
for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
```

```
October 17, 2022 - 15:41:52
```

```
Django version 4.1.2, using settings 'primerProyectoDjango.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

```
[17/Oct/2022 15:41:59] "GET / HTTP/1.1" 200 10681
```

```
[17/Oct/2022 15:41:59] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
```

```
[17/Oct/2022 15:41:59] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1
```

```
[17/Oct/2022 15:41:59] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP
```

```
[17/Oct/2022 15:41:59] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.
```

Migraciones aún
no aplicadas a
la base de datos

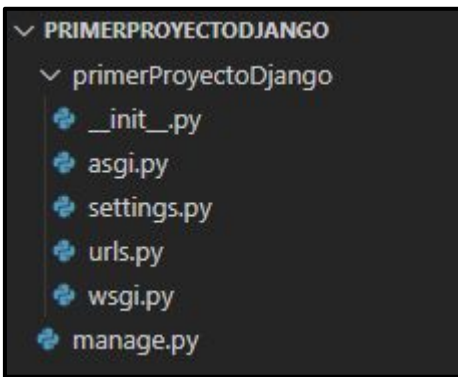
ip:puerto donde se ejecutará la aplicación



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

ESTRUCTURA DE PROYECTO DJANGO



__init__.py: Archivo necesario para indicarle a Python que el directorio se trata de un paquete.

asgi.py: Un punto de entrada al proyecto para los servidores web compatibles con ASGI.

settings.py: El archivo con la configuración del proyecto.

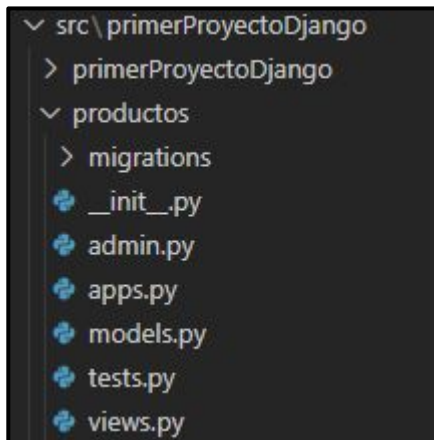
urls.py: Las declaraciones de URL para este proyecto Django; una “tabla de contenido”.

wsgi.py: Un punto de entrada al proyecto para los servidores web compatibles con WSGI.

PRIMEROS PASOS CON DJANGO

CREACIÓN DE UNA APLICACIÓN

```
python manage.py startapp productos
```



No olvidarse de registrar la aplicación en **settings.py** en la variable **INSTALLED_APPS**!

admin.py: Archivo donde se registran los modelos de la aplicación en la página de administrador.

apps.py: Archivo utilizado para configurar la aplicación, rara vez se utiliza ya que la configuración por default suele ser más que suficiente.

models.py: Representa los modelos de la aplicación, son aquellos modelos con los que va a trabajar la aplicación. Estos están representados en forma de clases y definen la estructura de la base de datos; el diseño, las relaciones, atributos y restricciones.

views.py: Representa la funcionalidad de la aplicación y cómo va a responder a las solicitudes del cliente. Puede hacerlo a través de funciones o clases. Y posee muchas clases predeterminadas de Django con fines a llevar a cabo tareas comunes (registro, logeo, operaciones CRUD, etc).

tests.py: Permite al usuario escribir pruebas para la aplicación