Presentado por: grupo N°02 del GT 02

Tarea ex Aula

Manual Técnico

**Manual Técnico**

**Índice**

[Introducción 2](#_Toc452236342)

[Objetivos 3](#_Toc452236343)

[Definición de la Arquitectura del Software 4](#_Toc452236344)

[Requerimientos 5](#_Toc452236345)

[Requerimientos funcionales 5](#_Toc452236346)

[Requerimientos no funcionales 5](#_Toc452236347)

[Modelo Conceptual de Datos 6](#_Toc452236348)

[Modelo Físico de la Base de Datos 7](#_Toc452236349)

[Mapa de Navegación 8](#_Toc452236350)

[Estructura de Directorios 9](#_Toc452236351)

# Introducción

A continuación se presenta una aplicación informática para el procesamiento de la información sobre el flujo de importaciones de los 5 productos de mayor flujo en los últimos dos años. Incluyendo también el manejo de la balanza comercial de estos 5 productos y el producto interno bruto (PIB) de los dos años anteriores.

Donde en cada área esta creada para que el usuario pueda gestionar en este las diferentes importaciones consultas personalizadas con los diferentes datos de El Salvador durante los años 2014 y 2015, datos mostrados mensualmente. A su vez se proporcionará la exportación y balanza comercial de esos 5 productos que es la característica principal de este software, otra bondad es el usuario tiene la facilidad de consultar el PIB de esos años, mostrados trimestralmente. Además de consultar, podrá guardarse datos individuales o masivos sobre las importaciones de esos 5 productos, y edición de ecuaciones que el usuario considere pertinente realizar.

Siendo este un software desarrollado en un ambiente web y mostrándole al usuario la lógica con que se ha desarrollado y cada una de las fases que el proyecto debe de tener para llegar a la funcionalidad total, mostrando las diferentes limitaciones que también puede tener el proyecto.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación del aplicativo.

# Objetivos

* Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
* Definir claramente el procedimiento de instalación del aplicativo.
* Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
* Describir las herramientas utilizadas para el diseño y desarrollo del prototipo

# 

# Definición de la Arquitectura del Software

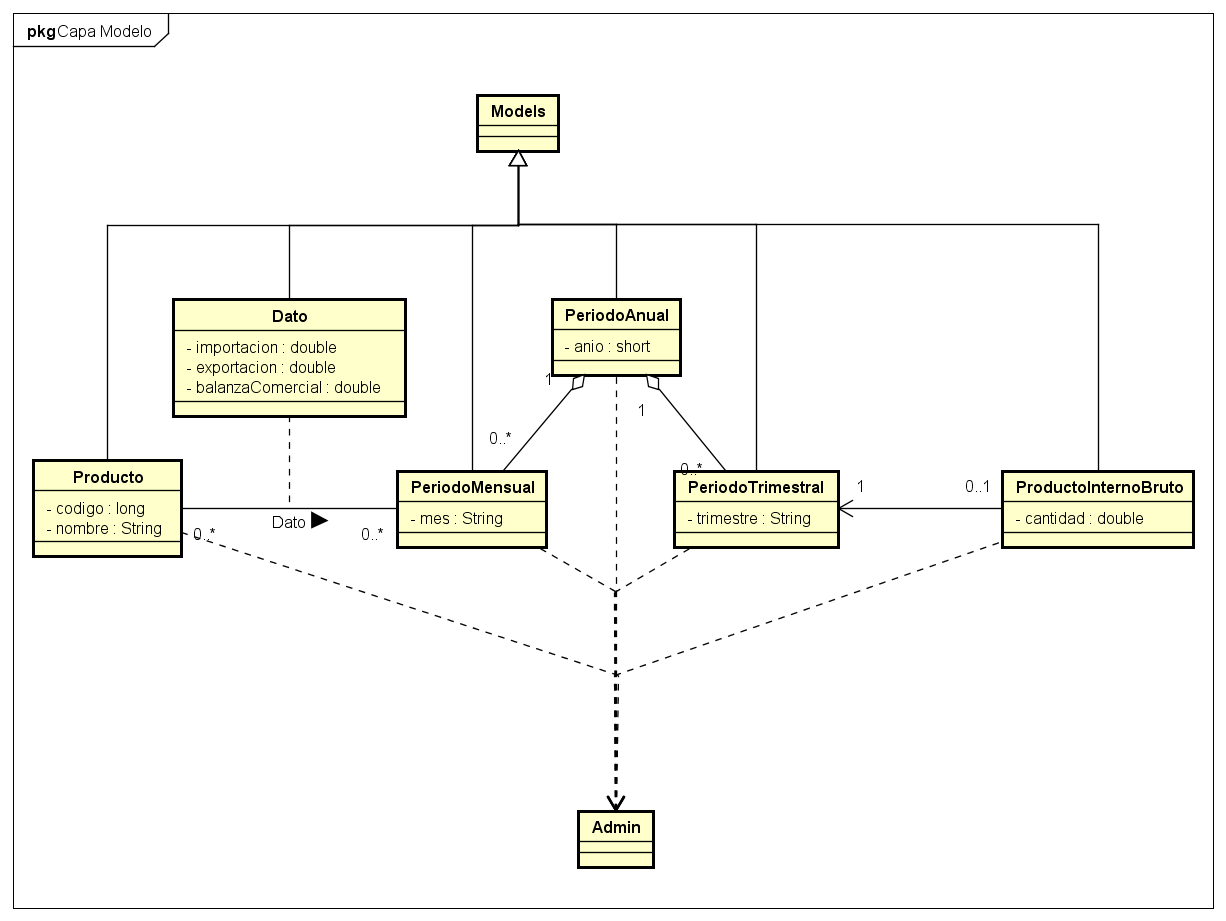
La arquitectura de software es una pieza central del desarrollo de sistemas de software modernos. El objetivo de la arquitectura consiste en desarrollar sistemas de software grandes de forma eficiente, estructurada y con capacidad de reusó. La arquitectura forma parte del proceso de diseño de software el cual también forma parte del proceso de desarrollo de software que comprende, requerimientos, diseño, implementación, prueba y mantenimiento.

Django es un framework de desarrollo Web que ahorra tiempo y hace que el desarrollo Web sea divertido. Utilizando Django puedes crear y mantener aplicaciones Web de alta calidad con un mínimo esfuerzo. En el mejor de los casos, el desarrollo web es un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y frustrante. Django nos permite enfocarnos en la parte divertida -- el quid de las aplicaciones Web -- al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. Al mismo tiempo, Django intenta no entrometerse, dejándote trabajar fuera del ámbito del framework según sea necesario.

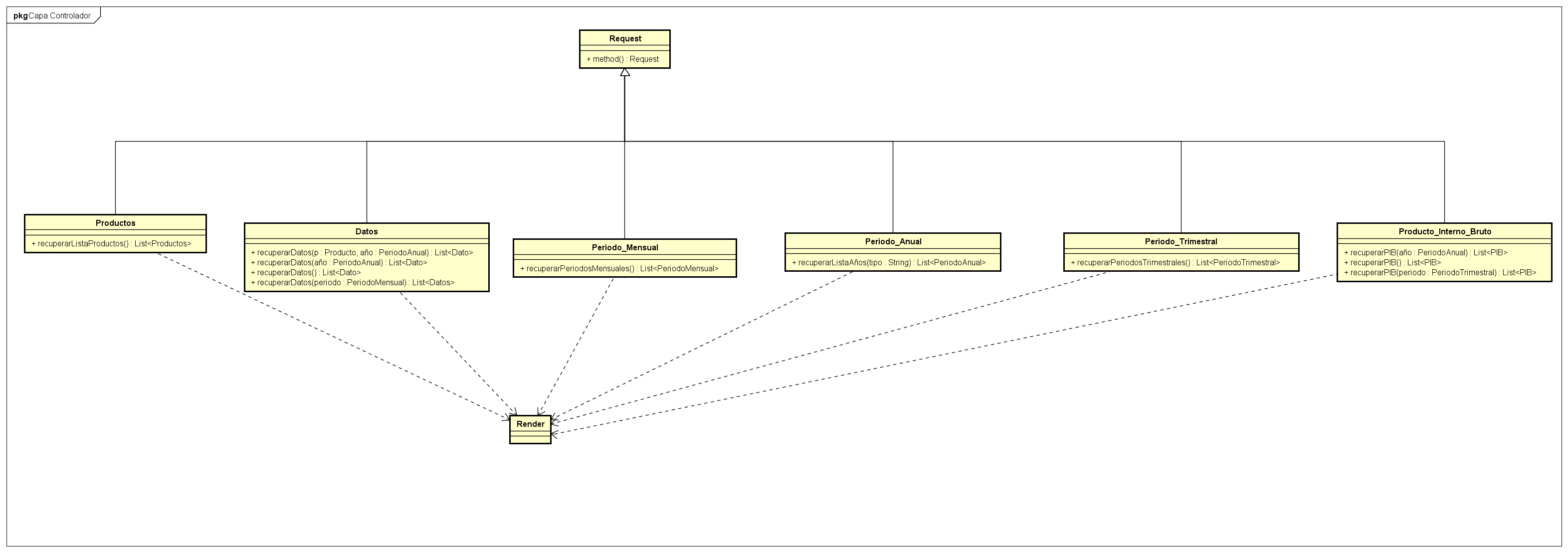
### **Modelo-Vista-Controlador**

Un punto más a favor de Django es el *patrón Modelo-Vista-Controlador* que maneja, esto quiere decir que separamos nuestra aplicación en tres componentes.

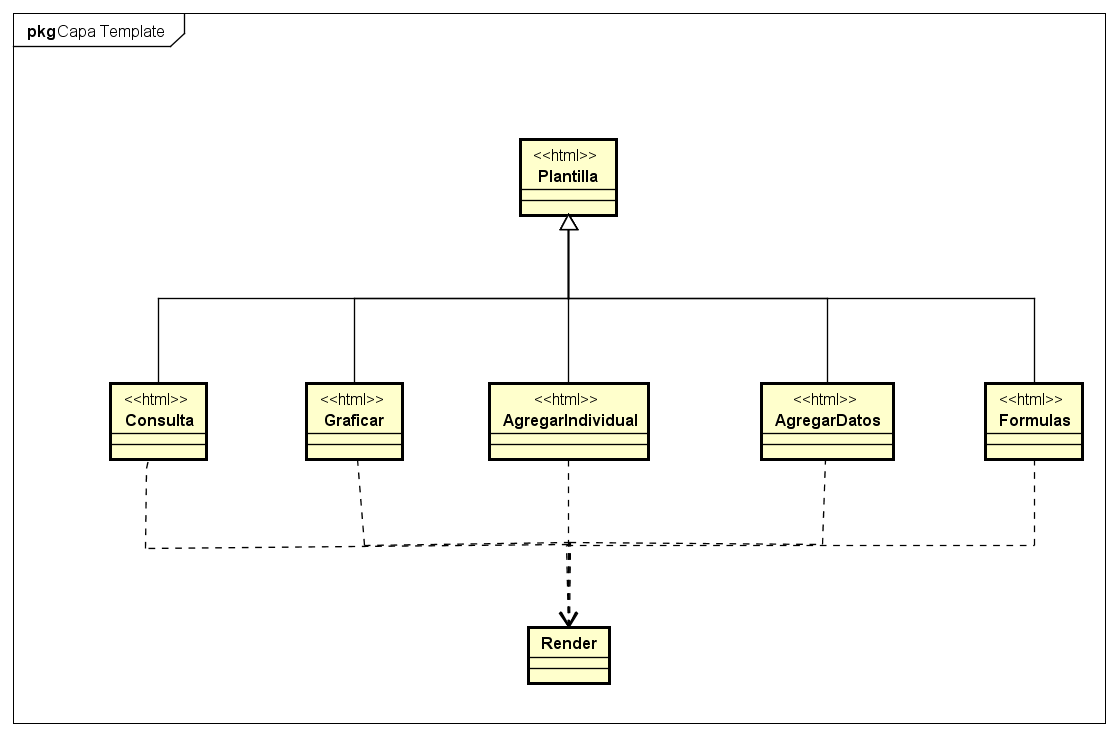
# Capa Modelo



# Capa controlador.



# Capa Template



# Configurando la base de datos

El único prerrequisito de Django es una instalación funcionando de Python. Nuestra aplicación web viene con soporte de base de datos. En nuestro caso utilizamos postgreSQL.

# Usar Django con PostgreSQL

Para utilizar PostgreSQL, necesitaremos el paquete psycopg2. En debían consiste en un simple ingreso de teclado como: #pip install postgresql psycop2

# Requerimientos

## Requerimientos funcionales

Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento). En nuestro caso los requerimientos funcionales para nuestra tarea son:

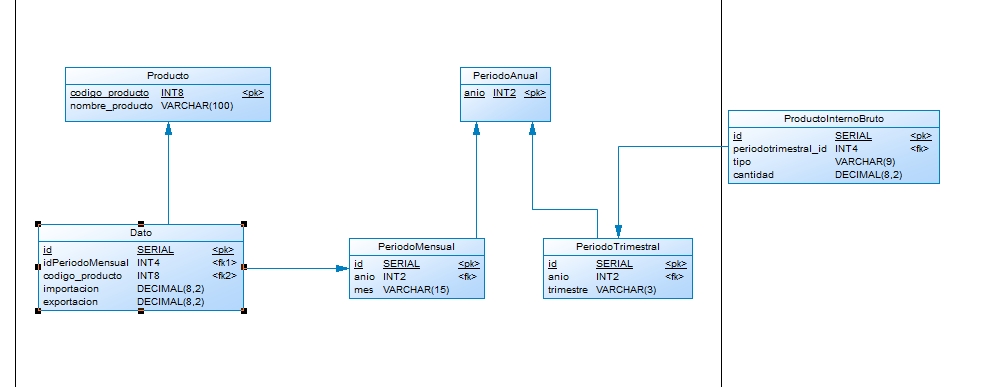
* El usuario deberá realizar un subconjunto de consultas entre categorías de importación, exportación, balanza comercial, definiendo para estas el producto y el año. Y PIB definiendo solamente el año.
* El usuario tendrá la posibilidad de graficar los datos seleccionados en la previa consulta.
* El usuario podrá agregar datos masivos de todos los datos.
* El usuario podrá agregar datos individuales de la importación de los productos.
* El usuario podrá Editar ecuaciones que representan los datos que se tiene a su disponibilidad.

## Requerimientos no funcionales

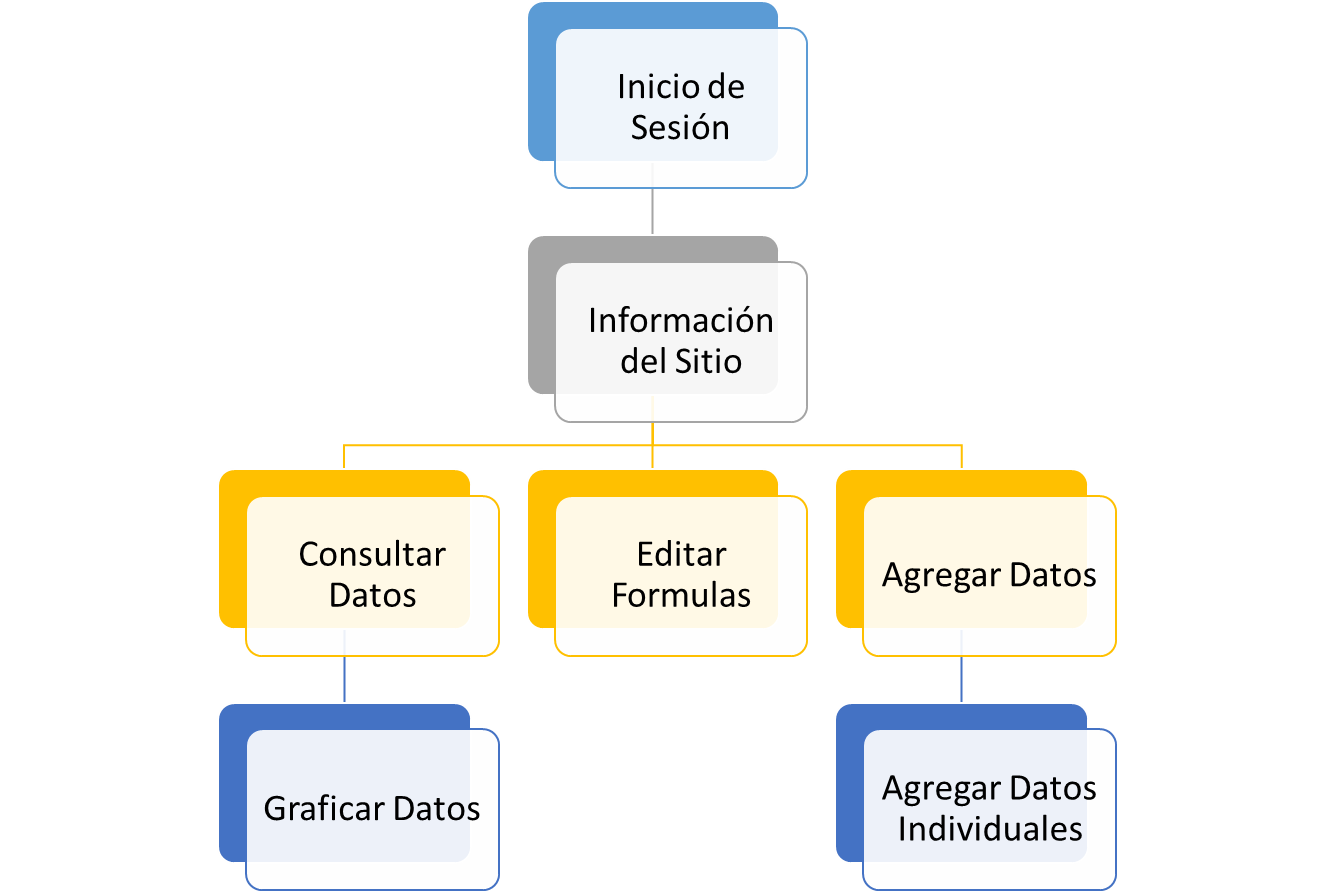
* Restricciones sobre el espacio de posibles soluciones.
* Rendimiento del sistema: Fiabilidad, tiempo de respuesta, disponibilidad.
* Interfaces: Dispositivos de E/S, usabilidad, interoperabilidad.
* Proceso de desarrollo: Estándares, herramientas, plazo de entrega.

# Modelo Conceptual de Datos

# Modelo Físico de la Base de Datos



**Mapa de Navegación**



# Estructura de Directorios

**Crear la estructura de directorios para nuestra aplicación:**

Nuestra aplicación tiene el siguiente directorio:

Tenemos un directorio destinado a montar toda la aplicación:

$ mkdir hdp

$ cd hdp

Creamos una virtualenv que cuando desarrollamos software con Django es común utilizar diferentes versiones de un mismo paquete. Un virtualenv es un espacio completamente independiente de otros entornos virtuales y de los paquetes instalados globalmente en el sistema. Una vez instalado, se ejecuta el siguiente comando dentro de nuestro directorio:

$ virtualenv ambi

Crea la siguiente estructura:

ambi/

bin/

include/

lib/

Activamos virtualenv

$ source ambi/bin/activate

(ambi)$

Una vez activada, instalamos Django:

(ambi)$ pip install django

Y podemos empezar el proyecto:

(ambi)$ django-admin.py startproject hpd115G02

Tiene la siguiente estructura:

├───manage.py

└───hdp115G02

settings.py

urls.py

wsgi.py

\_\_init\_\_.py

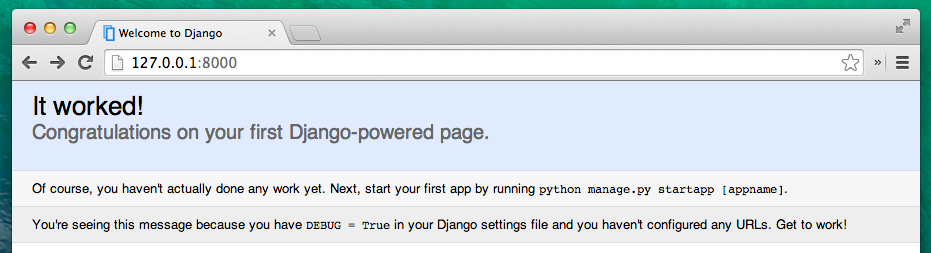
Configuramos la base de datos según el gestor que hemos de utilizar. Y una vez hecho eso, realizamos las correspondientes migraciones.

(ambi)$ python manage.py migrate

Y corremos el servidor:

(ambi) $ python manage.py runserver

Vemos lo siguiente:



Creamos la app de nuestra aplicación, que será el exoesqueleto de nuestro trabajo.

(ambi)$ python manage.py startapp importaciones

Tenemos la siguiente estructura

hdp

├── hdp115G02

| \_\_init\_\_.py

| settings.py

| urls.py

| wsgi.py

├── manage.py

└── Importaciones

├── migrations

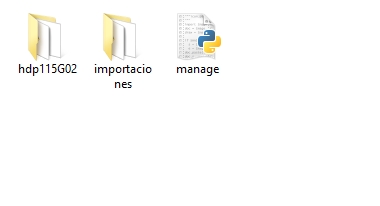
| \_\_init\_\_.py

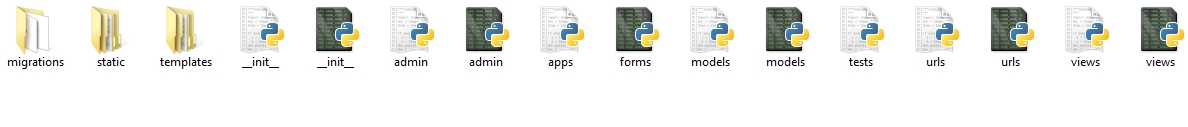
├── \_\_init\_\_.py

├── admin.py

├── models.py

├── tests.py

 └── views.py



Una vez con esto, ya podemos empezar a trabajar.