**¿Qué es Django?**

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

**Características:**

**Completo**

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y[actualizada documentación](https://docs.djangoproject.com/en/1.10/).

**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. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc). ¡El sitio que estás leyendo actualmente está basado en Django!  
  
Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos , motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

**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. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.  
   
*Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una [cryptographic hash function](https://en.wikipedia.org/wiki/Cryptographic_hash_function). Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo debido a la naturaleza "unidireccional" de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver la contraseña original.*  
  
Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (consulte [Seguridad de sitios web](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security) para obtener más detalles sobre dichos ataques).

**Escalable**

Django usa un componente basado en la arquitectura “[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)” (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de cache, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

**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. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón [Model View Controller (MVC)](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)).

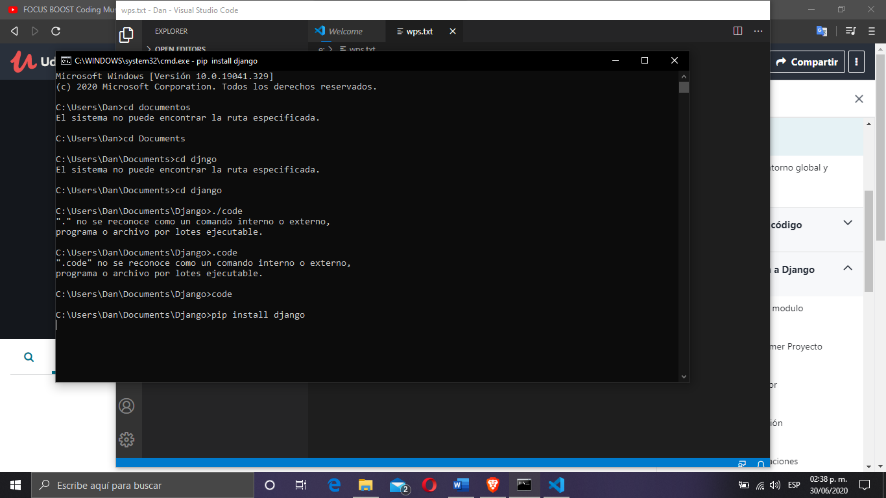
**Portable**

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

|  |  |  |  |
| --- | --- | --- | --- |
| FRAMEWORK | Django | Laravel | Rails |
| Lenguaje | Python | PHP | Ruby |
| Enfoque | Full-stack | Backend | Backend |
| Escalable | si | si | no |
| MVC | si | si | si |
| complejidad | facil | dificil | intermedio |



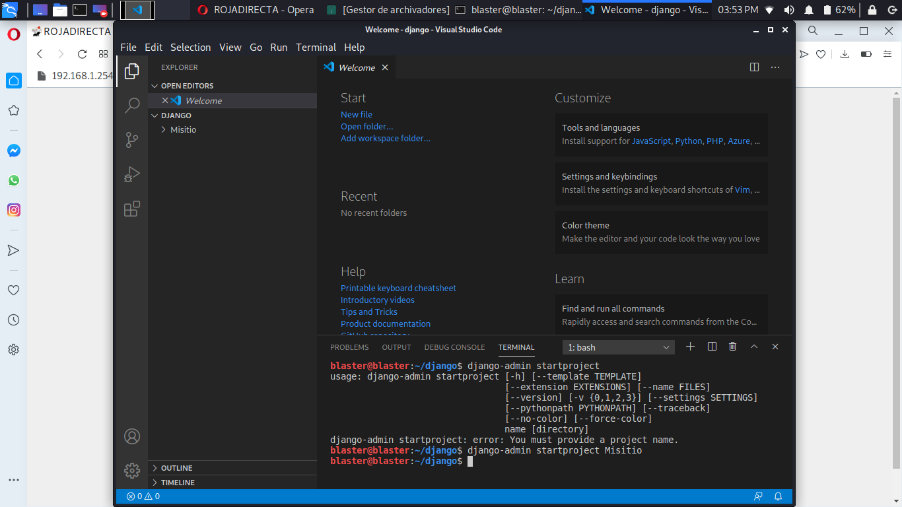
**Bitácora** Debido a que la informacion mostrada es solo para lena de comandos he optado por instalar Django desde la siguiente guía para trabajar con visual studio code: <https://www.udemy.com/course/aplicaciones-web-django-jairo-galeas/learn/lecture/16985118#overview>.

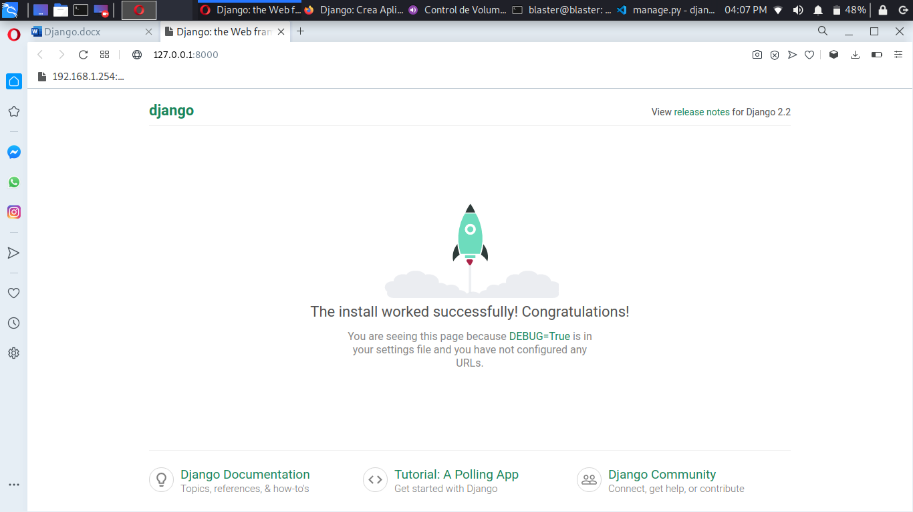




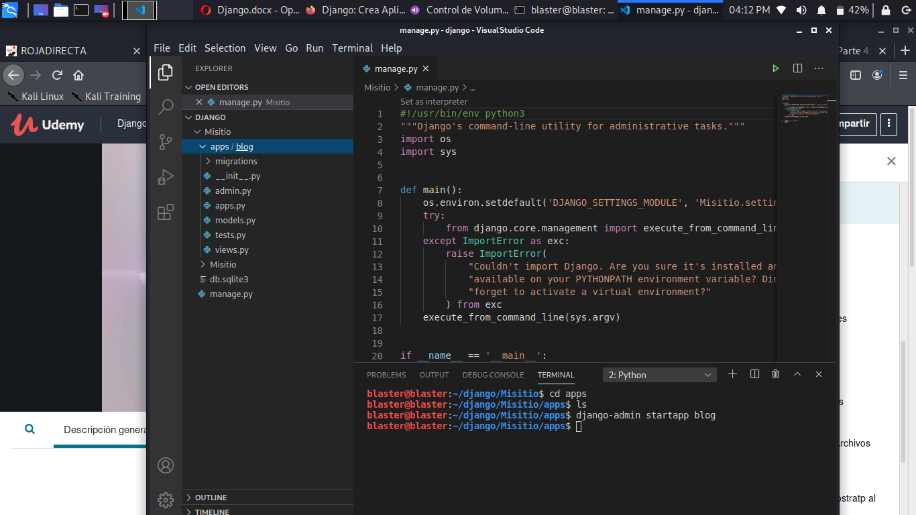
Debido a problemas con python en windows 10 2004, opte por cambiar a kali linux que ya trae por defecto instalado python.

Instalación completada de django a través del bash de visual studio code.

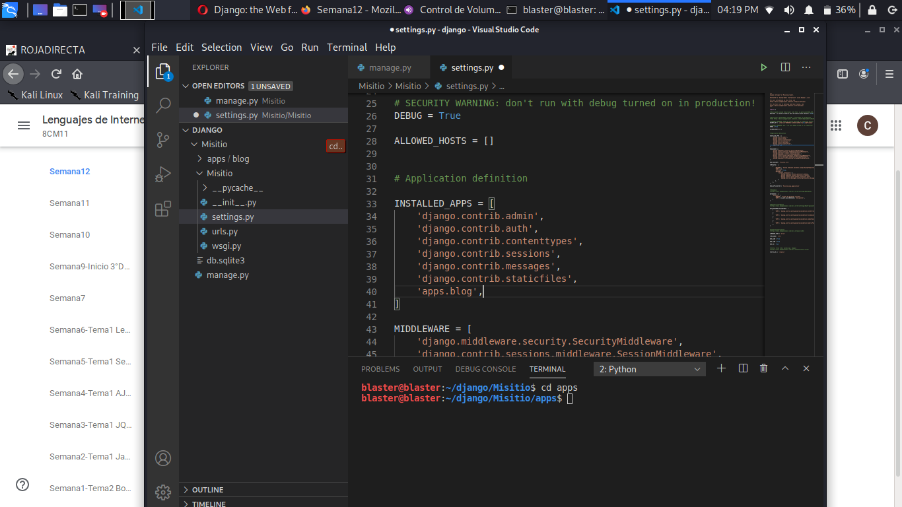




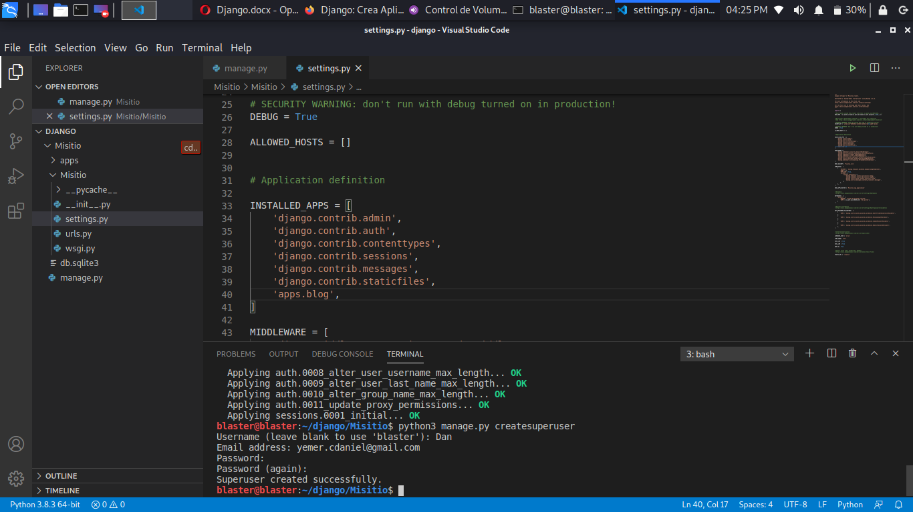
Creación del proyecto

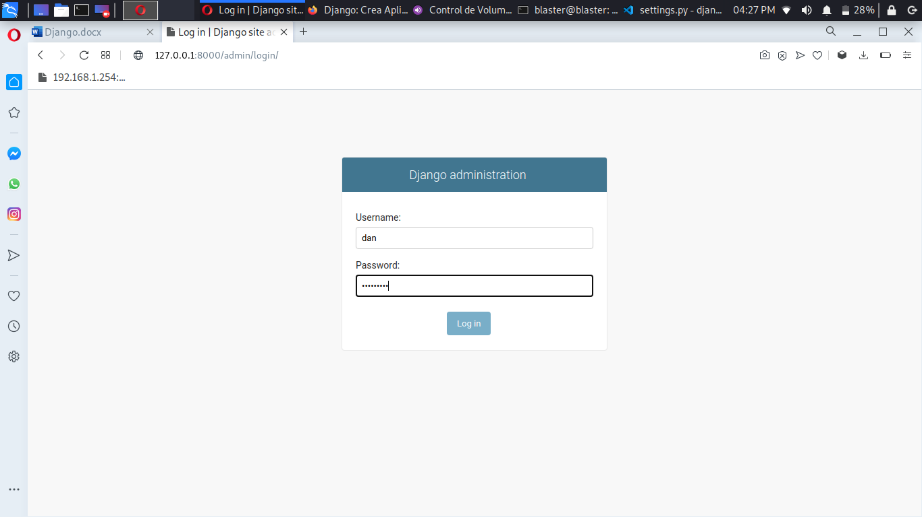


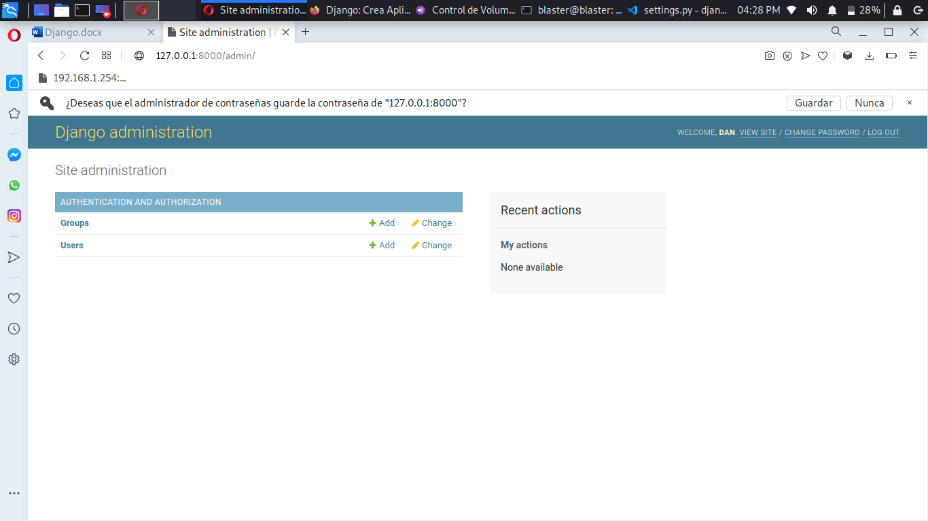
Registro de la aplicación



Creación del super usuario







Vistas y modelos

