

**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACION**



PROYECTO:

PAGINA WEB CON PYTHON Y DJANGO

EQUIPO DESARROLLADOR:

APELLIDOS:

NOMBRES:

CARNET

SOTO ALDANA

JUAN GABRIEL

SA01134315

NAJARRO MOREIRA

YESICA YANETH

NM01132720

PALACIOS MARMOL

WILBET ALEXANDER

PM01134162

MARTINEZ MARTINEZ

JOSE ARNULFO

MM01134085

CATEDRA: PROGRAMACION III

HORARIO: MIERCOLES 1:00 P.M. A 3:30 P.M.

FACILITADOR: LIC.RIGOBERTO ISRAEL ORELLANA

SAN SALVADOR ,10 DE NOBIEMBRE 2018

DESARROLLO

TÉCNICAS DE PROGRAMACIÓN APLICADA

Un Programación Orientada a Objetos para lo cual hemos desarrollado los distintos diagramas UML, Caso de uso, Secuencias, Componentes, etc.

HERRAMIENTAS DE DESARROLLO APLICADO

Software

Para el diseño y desarrollo de nuestra aplicación nos valdremos de las siguientes herramientas de desarrollo.

- Umbrello – UML, para el diseño y modelado de la aplicación, es una herramienta que permite modelar sistema basados en tecnología Orientada a Objetos, con ella se desarrollara cada uno de los diagramas utilizados en el desarrollo del Sistema.

- SQLite – Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB) biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

LENGUAJES DE PROGRAMACION Y SGBD

python3: es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. es gratuito y multiplataforma.

HTML: "HyperText Mark-up Lenguaje", es decir, "Lenguaje de marcado hipertextual", hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define.

una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.

CSS: Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

Django: es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt. En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Hardware

El hardware que utilizaremos para el funcionamiento de nuestra aplicación será:

- 1 Servidor (Base de Datos).
- con putadora para poder de manera pueda visualizar la pagina web.

Otros Dispositivos:

pc

movil

Como primer paso en la instalación de Django y la implementación de nuestro sitio web en linux comenzaremos por verificar si tenemos instalado pip y python aun que este ultimo ya viene instalado por defecto en linux pero a veces es necesario actualizarlo ala ultima version

```
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris: ~
izzy@Fenris:~$ python
Python 2.7.15rc1 (default, Apr 15 2018, 21:51:34)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

en este caso vemos que pip no esta instalado asi que copiamos nada mas el comando que nos muestra y procedemos ala instalación.

```
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris: ~
izzy@Fenris:~$ sudo apt install python3-venv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
python3.6-venv
Se instalarán los siguientes paquetes NUEVOS:
python3-venv python3.6-venv
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 7 392 B de archivos.
Se utilizarán 44.0 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3.6-venv amd64 3.6.6-1-18.04 [6 184 B]
Des:2 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3-venv amd64 3.6.5-3ubuntu1 [1 208 B]
Descargados 7 392 B en 1s (12.3 kB/s)
Seleccionando el paquete python3.6-venv previamente no seleccionado.
(Leyendo la base de datos ... 270427 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python3.6-venv_3.6.6-1-18.04_amd64.deb ...
Desempaquetando python3.6-venv (3.6.6-1-18.04) ...
Seleccionando el paquete python3-venv previamente no seleccionado.
Preparando para desempaquetar .../python3-venv_3.6.5-3ubuntu1_amd64.deb ...
Desempaquetando python3-venv (3.6.5-3ubuntu1) ...
Configurando python3.6-venv (3.6.6-1-18.04) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Configurando python3-venv (3.6.5-3ubuntu1) ...
izzy@Fenris:~$
```

Ahora que tenemos instalado pip y python instalamos el entorno virtual

creamos una carpeta llamada django

```
izzy@Fenris: ~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris:~$ cd Escritorio
izzy@Fenris:~/Escritorio$ mkdir django
izzy@Fenris:~/Escritorio$
```

nos vamos ala carpeta django y creamos un entorno virtual

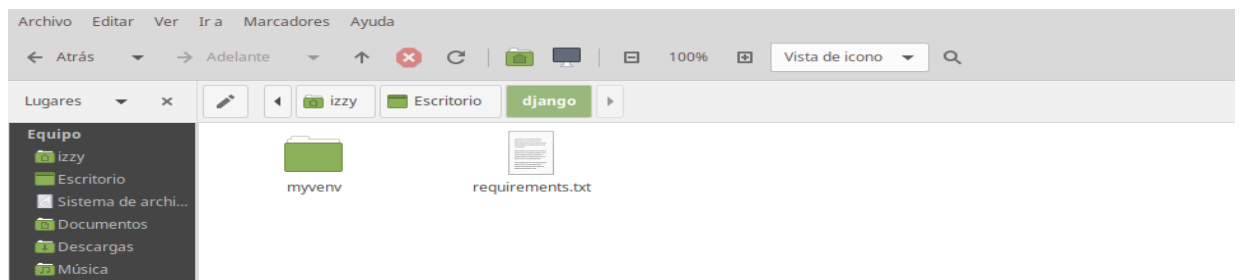
```
izzy@Fenris: ~/Escritorio/django
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris:~/Escritorio$ cd django/
izzy@Fenris:~/Escritorio/django$
```

```
izzy@Fenris: ~/Escritorio/django
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris:~/Escritorio/django$ python3 -m venv myvenv
izzy@Fenris:~/Escritorio/django$
```

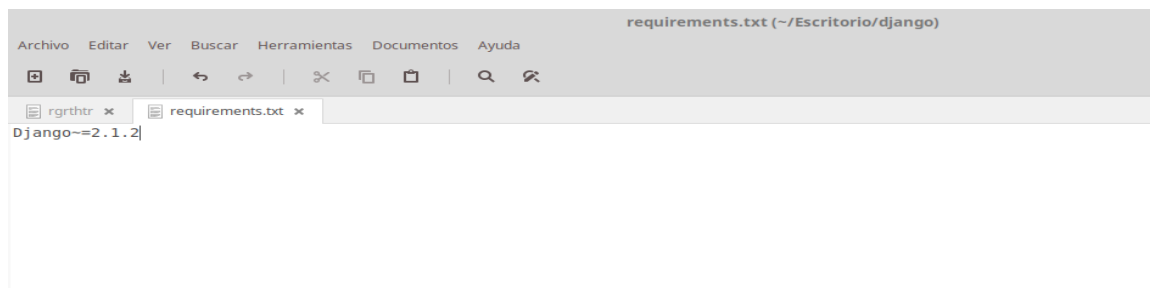
ahora encendemos el entorno virtual

```
izzy@Fenris: ~/Escritorio/django
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris:~/Escritorio/django$ source myvenv/bin/activate
(myvenv) izzy@Fenris:~/Escritorio/django$
```

ahora instalamos django en una lista de requerimientos, creamos un archivo con el nombre requirements.txt.



Ahora dentro de este documento pondremos Django~=2.1.2



ahora instamos django con el comando pip install -r requirements.txt

```
Archivo Editar Ver Buscar Terminal Ayuda
izzy@Fenris: ~/Escritorio/django
(myvenv) izzy@Fenris:~/Escritorio/django$ pip install -r requirements.txt
Collecting Django==2.1.2 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/d1/e5/2676be45ea49cf09a663f289376b3888acc57ff06c953297bfdee1fb08/Django-2.1.3-py3-none-any.whl (7.3MB)
    100% |#####| 7.3MB 123kB/s
Collecting pytz (from Django==2.1.2->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/f8/0e/2365ddc010afb3d79147f1dd544e5ee24bf4ece58ab99b16fbb465ce6dc0/pytz-2018.7-py2.py3-none-any.whl (506kB)
    100% |#####| 512kB 568kB/s
Installing collected packages: pytz, Django
Successfully installed Django-2.1.3 pytz-2018.7
(myvenv) izzy@Fenris:~/Escritorio/django$

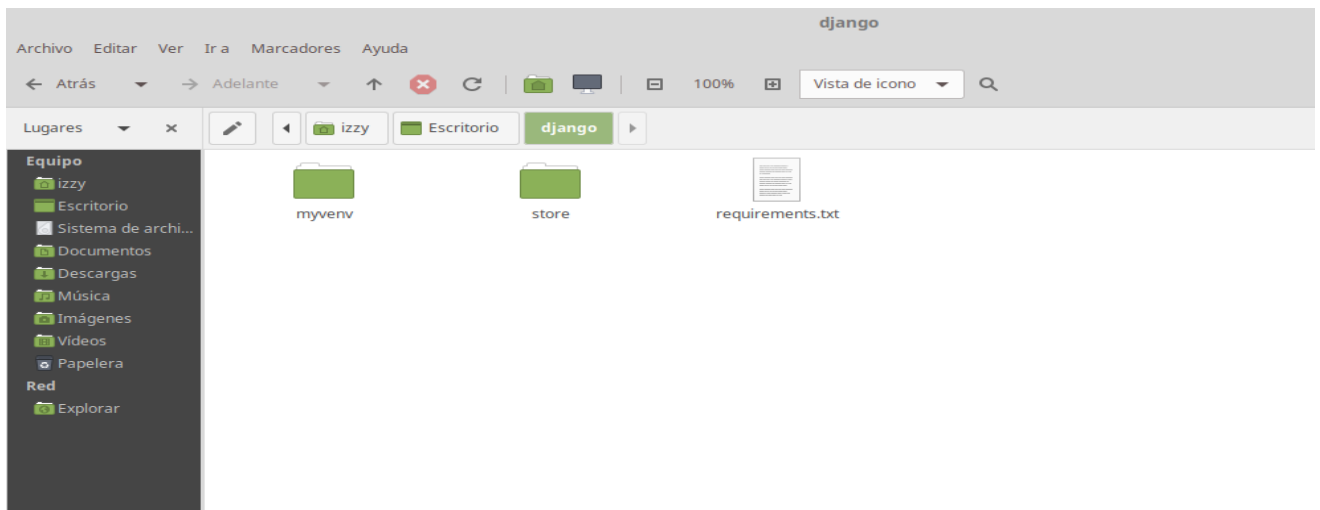
6. Lo siguiente es instalar django de una lista de requerimientos, se puede crear en la misma carpeta de django el archivo se tiene que llamar requirements.txt
dentro de ese archivo hay que poner esto
Django==2.1.2

Quedaría algo así

django
requirements.txt
myvenv

7 Ahora hay que instalar django con el comando
```

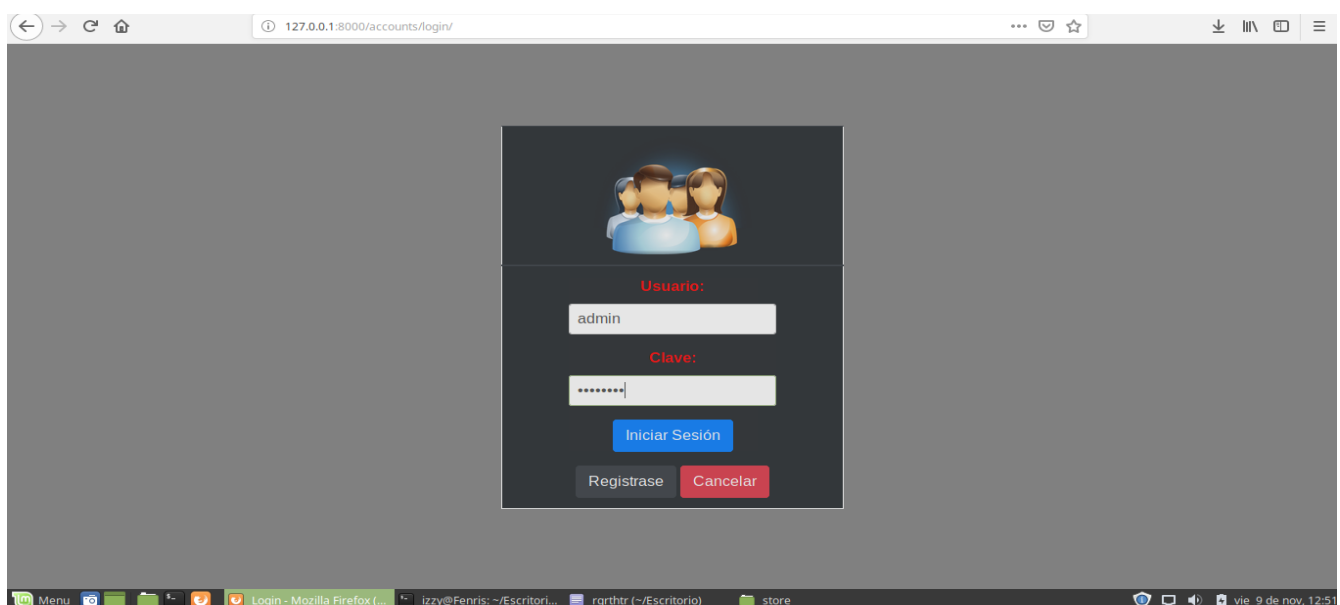
ahora nos vamos ala carpeta de django y pegamos nuestro proyecto



despues entramos ala carpera de nuestro proyecto y escribimos el comando python manage.py migrate para migrar al mostrarnos que todo es perfectamente escribimos python manage.py runserver para que nuestro sitio comienze a funcionar.

```
izzy@Fenris: ~/Escritorio/django/store
Archivo Editar Ver Buscar Terminal Ayuda
(myvenv) izzy@Fenris:~/Escritorio/django/store$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, web
Running migrations:
  No migrations to apply.
(myvenv) izzy@Fenris:~/Escritorio/django/store$ python manage.py runserver
Performing system checks...
System check identified no issues (0 silenced).
November 09, 2018 - 12:51:05
Django version 2.1.3, using settings 'store.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[09/Nov/2018 12:51:10] "GET / HTTP/1.1" 200 3247
[09/Nov/2018 12:51:10] "GET /static/js/jquery-3.3.1.slim.min.js HTTP/1.1" 200 69917
[09/Nov/2018 12:51:10] "GET /static/js/bootstrap.min.js HTTP/1.1" 200 50737
[09/Nov/2018 12:51:10] "GET /static/css/bootstrap.min.css HTTP/1.1" 200 140936
[09/Nov/2018 12:51:10] "GET /static/css/estilo.css HTTP/1.1" 200 667
[09/Nov/2018 12:51:10] "GET /static/js/estilo.min.js HTTP/1.1" 200 20340
[09/Nov/2018 12:51:10] "GET /static/js/popper.min.js HTTP/1.1" 200 20340
[09/Nov/2018 12:51:12] "GET /static/img/pc.png HTTP/1.1" 200 549519
```

Una vez hecho esto nos dirigimos a nuestro navegador y escribimos en la barra de direcciones 127.0.0.1:8000 para visualizar nuestro sitio un vez ahi veremos la parte para logearnos.



Luego de esto podemos ver que entramos como administrador donde podemos agregar o borrar todo lo que queramos.



Y
si

deseamos salir y que deje de funcionar nuestro sitio nos vamos ala terminal y damos control + c para terminar con los procesos.

```
(myvenv) izzy@Fenris:~/Escritorio/django/store$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
November 09, 2018 - 12:51:05
Django version 2.1.3, using settings 'store.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[09/Nov/2018 12:51:10] "GET / HTTP/1.1" 200 3247
[09/Nov/2018 12:51:10] "GET /static/js/jquery-3.3.1.slim.min.js HTTP/1.1" 200 69917
[09/Nov/2018 12:51:10] "GET /static/js/bootstrap.min.js HTTP/1.1" 200 50737
[09/Nov/2018 12:51:10] "GET /static/css/bootstrap.min.css HTTP/1.1" 200 140936
[09/Nov/2018 12:51:10] "GET /static/css/estilo.css HTTP/1.1" 200 667
[09/Nov/2018 12:51:10] "GET /static/js/popper.min.js HTTP/1.1" 200 20340
[09/Nov/2018 12:51:12] "GET /static/img/pc.png HTTP/1.1" 200 549519
[09/Nov/2018 12:51:29] "GET /accounts/login/ HTTP/1.1" 200 1925
[09/Nov/2018 12:51:29] "GET /static/img/user.png HTTP/1.1" 200 81472
[09/Nov/2018 12:51:51] "POST /accounts/login/ HTTP/1.1" 200 1993
[09/Nov/2018 12:52:05] "POST /accounts/login/ HTTP/1.1" 302 0
[09/Nov/2018 12:52:05] "GET / HTTP/1.1" 200 3437
[09/Nov/2018 12:52:23] "GET /Cargar_producto/ HTTP/1.1" 200 4648
[09/Nov/2018 12:52:44] "POST /Cargar_producto/ HTTP/1.1" 302 0
[09/Nov/2018 12:52:44] "GET / HTTP/1.1" 200 3437
[09/Nov/2018 12:52:51] "GET /Cargar_producto/ HTTP/1.1" 200 4648
^C(myvenv) izzy@Fenris:~/Escritorio/django/store$
Precio:
```

REQUERIMIENTOS

Navegadores compatibles

- Mozilla Firefox Extended Support Release (ESR) versión 38
- Microsoft Internet Explorer 10 y 11

- Google Chrome versiones 44, 45 y 47
- Microsoft EDGE

Requisitos de software:

- Python Versión 3.6.5
- python3 pip
- Django version 2.1.2

Requisitos de hardware:

- 1 GB de RAM