

# Manual técnico de aplicación



## PRESENTADO POR:

GERMAN JASSET CUESTA BECERRA  
CINDI LORENA DIAZ CARDENAS  
SEBASTIAN RODRÍGUEZ DÍAZ  
MAIKOLL YESID TRUJILLO TALERO

SENA CENTRO MINERO  
SOGAMOSO BOYACÁ

SEPTIEMBRE DE 2022



## MANUAL TECNICO DE APLICACIÓN

German Jaset Cuesta Becerra

Cindi Lorena Diaz Cardozo

Sebastián Rodríguez Díaz

Maikoll Yesid Trujillo Talero

SENA CENTRO MINERO

SOGAMOSO BOYACÀ

2022

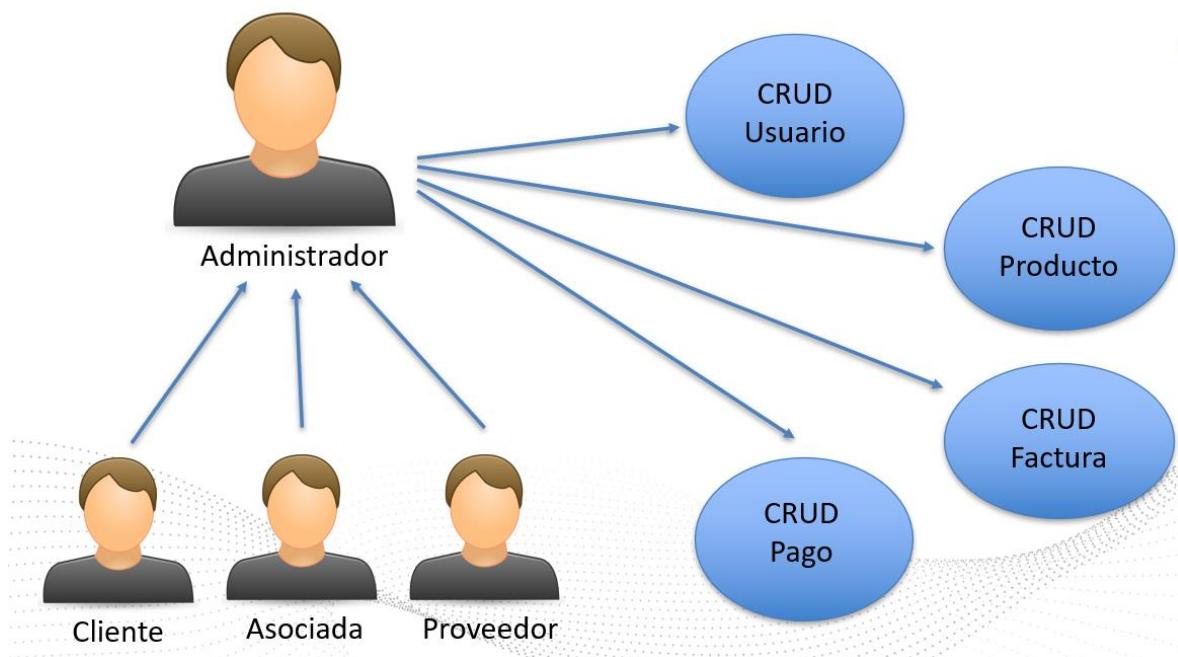
## TABLA DE CONTENIDO

1. ALCANCE.....	4
2. DESCRIPCIÓN DE PROCESOS .....	5
3. DIAGRAMAS UML.....	6
3.1 Diagrama de clases.....	6
3.1.1 Factura.....	6
3.1.2 Pago.....	6
3.1.4 Usuario .....	7
3.2.1 Cliente .....	8
3.2.2 Compra.....	8
3.2.4 Proveedor.....	9
3.2.5 Venta .....	10
4. MODELO ENTIDAD RELACIÓN .....	11
5. ADMINISTRACIÓN DE USUARIOS .....	12
5.1 Crear usuario .....	12
5.2 Ver usuario .....	15
5.3 Editar usuario .....	16
5.4 Desactivar usuario.....	16
5.5 Crear super usuario .....	17
Un super usuario es el encargado de administrar todo el sistema con todos los permisos necesarios para modificar el sistema. Se crea mediante comandos: .....	17
Python manage.py createsuperuser .....	17
6. DESCRIPCIÓN DE LA PLATAFORMA.....	18
6.1 Instalación del servidor .....	18
6.2 Mantenimiento del software .....	35
Es recomendable actualizar las librerías en un periodo de 6 meses a 1 año.....	35
7. DOCUMENTACIÓN DEL CODIGO FUENTE.....	36
8. ESTRUCTURA DE DIRECTORIOS .....	37
9. RELACIÓN DE PROGRAMAS.....	38
10. DESCRIPCIÓN DE LOS ACUERDOS DE NIVELES DE SERVICIOS O ANS .....	39
11. GLOSARIO .....	40
12. REFERENCIAS.....	43

## 1. ALCANCE

Este manual explica los detalles técnicos de la aplicación Unidad Productiva Tejidos Y Variedades Tota, permitirá que un nuevo equipo técnico conozca la estructura del proyecto y el funcionamiento de este, el lenguaje de programación con sus respectivas librerías y herramientas que componen el proyecto.

## 2. DESCRIPCIÓN DE PROCESOS



### 3. DIAGRAMAS UML

#### 3.1 Diagrama de clases

##### 3.1.1 Factura

Campos requeridos para crear una factura con su respectiva especificación de tipo de datos, utilizando métodos setters y getters.

FACTURA
fecha: date tipo factura: ENUM total pagar: int estado: ENUM  «create» LISTAR/BUSCAR () «create» (fecha : date, tipo factura : Enum, total : int, estado : Enum) # buscar (consulta : String): USUARIO () # insertar (): Boolean () # actualizar (): Boolean () # borrar (): Boolean () # colocar () # guardar (query : String): Boolean ()

##### 3.1.2 Pago

Campos requeridos para crear un pago con la especificación de los tipos de datos que se deben usar, utilizando métodos setters y getters.

PagoTejidoMensual
- fecha: date - total: Real - estado: ENUM  «create» LISTAR/BUSCAR () «create» (fecha : date, total : real, estado : Enum) # buscar (consulta : String): USUARIO () # insertar (): Boolean () # actualizar (): Boolean () # borrar (): Boolean () # colocar () # guardar (query : String): Boolean ()

### 3.1.3 Producto

Campos requeridos para crear un producto con la especificación de los tipos de datos que se deben utilizar, además se usan métodos setters y getters.

PRODUCTO
categoria: varchar - nombre: varchar - stock: int - precio: Real precio_venta: Real porcentaje: varchar valor_porcentaje: double estado: varchar  «create» LISTAR/BUSCAR () «create» (nombre : varchar, stock : int, precio : Real, precio _venta : Real, porcentaje : varchar, valor_porcentaje : double, estado : varchar) # buscar (consulta : String): USUARIO () # insertar (): Boolean () # actualizar (): Boolean () # borrar (): Boolean () # colocar () # guardar (query : String): Boolean ()

### 3.1.4 Usuario

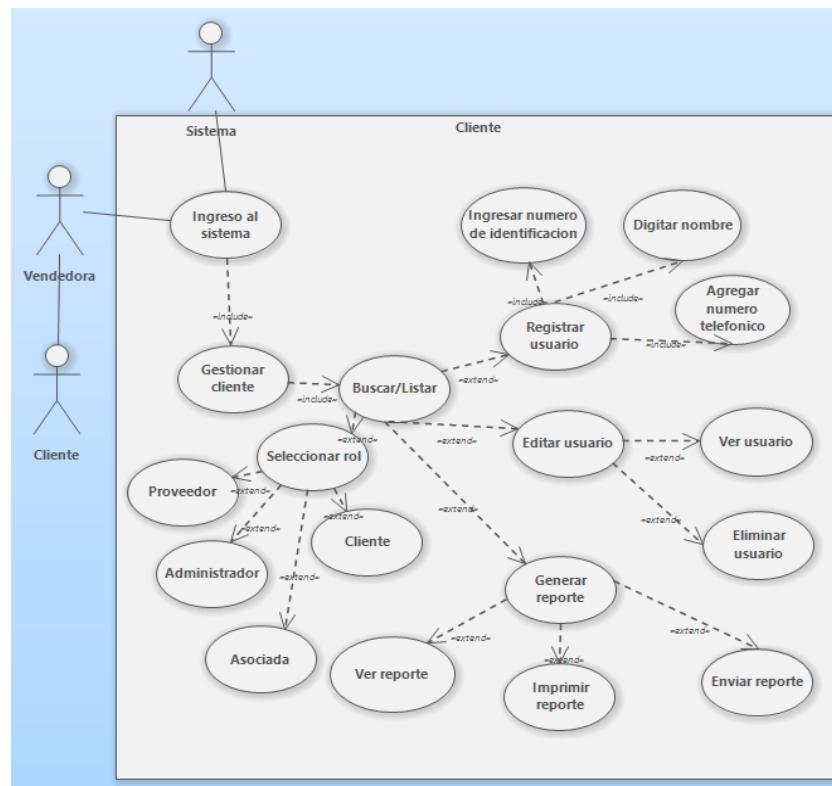
Campos requeridos para la creación de un usuario, utilizando constructores, métodos getters y setters.

USUARIO
- nombre: String - apellido: String - numeroDocumento: Int - telefono: Int - rol: ROL - tipoDocumento: TIPODOCUMENTO  # «create» USUARIOS () # «create» USUARIOS (nombre : String, apellido : String, numeroDocumento : Int, telefono : Smallint, rol : ROL, tipoDocumento : TIPODOCUMENTO) # colocar () # buscar (consulta : String): USUARIO # buscarPorId (id : integer): USUARIO # obtenerTodos (): USUARIO # guardar (query : String): Boolean # insertar (): Boolean # actualizar (): Boolean # borrar (): Boolean # adquirir ()

### 3.2 Diagrama de casos de uso

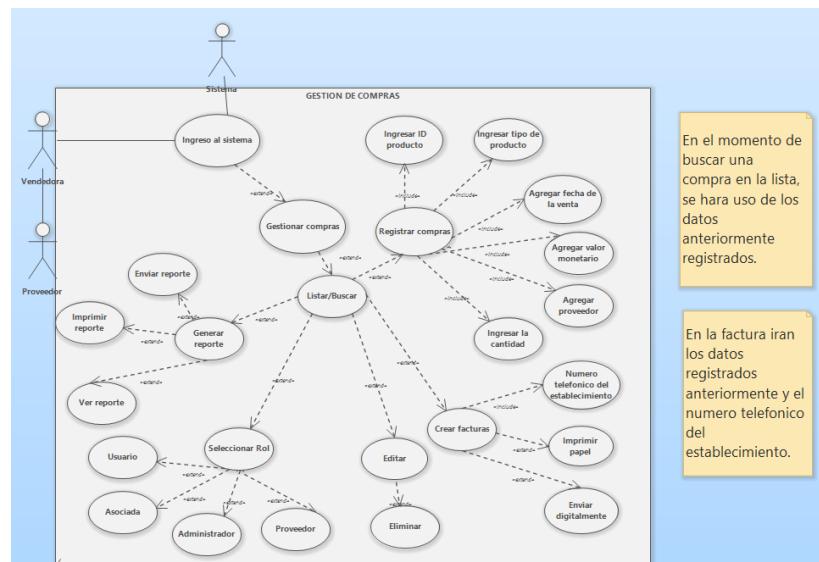
#### 3.2.1 Cliente

Cuando un cliente compra un producto tiene relación directa con el sistema, el vendedor y el producto de igual forma trabaja con el CRUD.



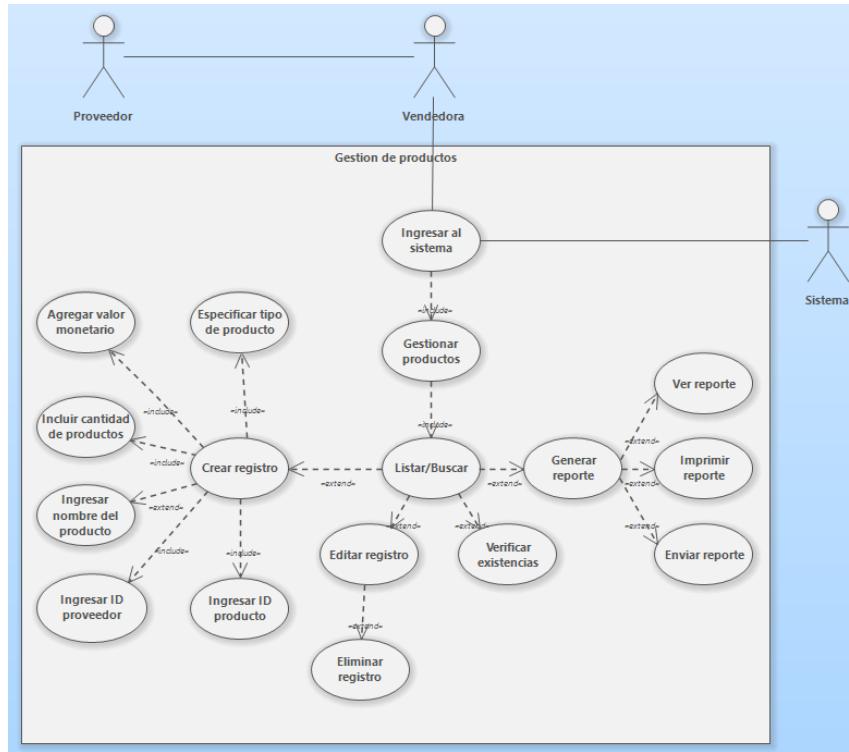
#### 3.2.2 Compra

Al hacer una compra se tiene relación con el proveedor y la factura.



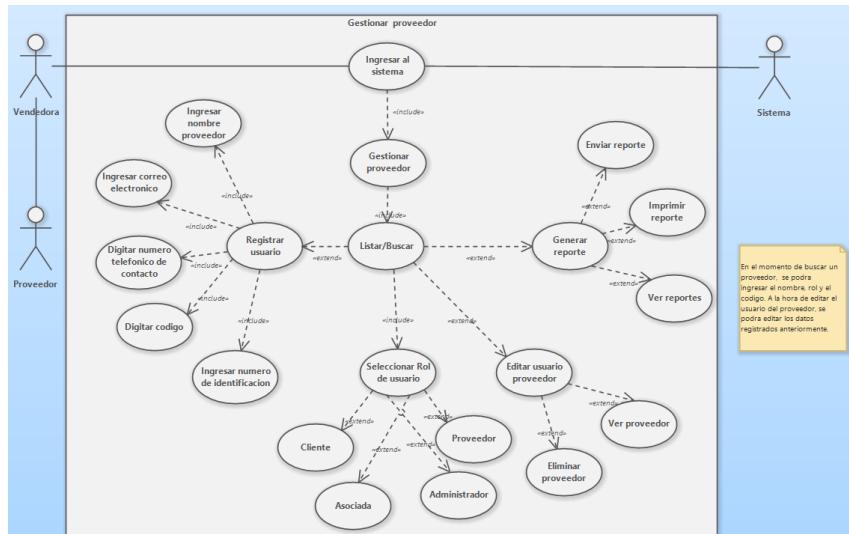
### 3.2.3 Producto

El proveedor trae productos los cuales serán vendidos a los clientes y se hará uso del CRUD.



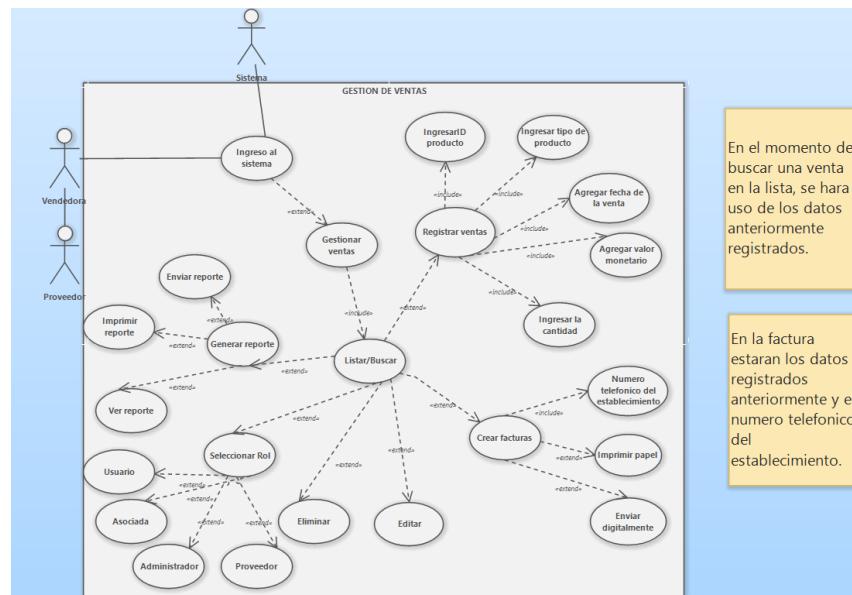
### 3.2.4 Proveedor

El proveedor tiene relación directa con el sistema y con el administrador encargándose de suministrar productos al establecimiento, haciendo uso del CRUD.

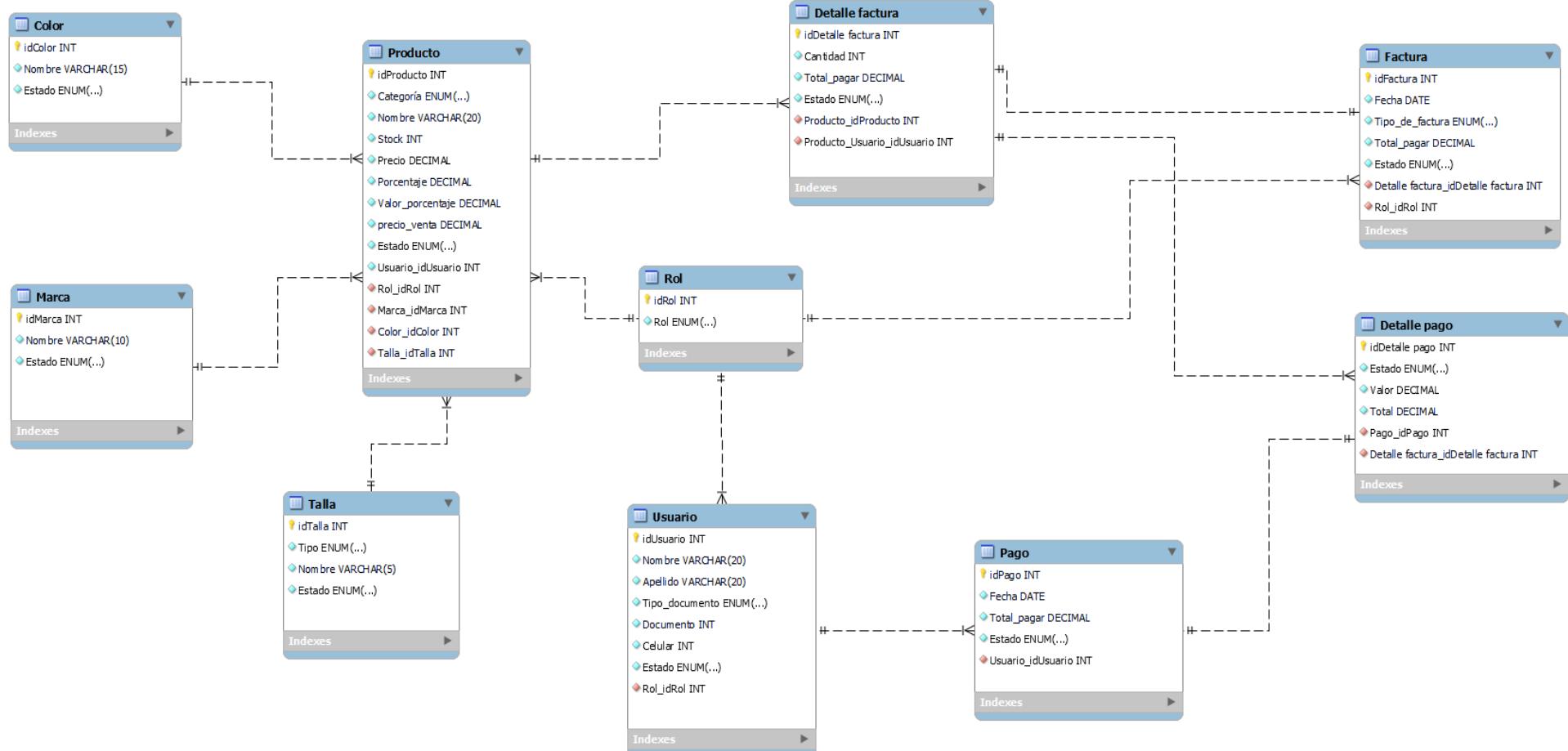


### 3.2.5 Venta

Se generan a partir de los productos que se venden al cliente, haciendo uso del CRUD.



## 4. MODELO ENTIDAD RELACIÓN



## 5. ADMINISTRACIÓN DE USUARIOS

### 5.1 Crear usuario

Para crear un usuario en el sistema, selecciona en la barra de navegación el botón **Usuario**, el sistema lo redirigera a la tabla usuarios.

En la parte inferior derecha de la tabla, se encuentra botón **Agregar usuario**, da clic en el botón y lo redirigira a la siguiente página.

En el campo de seleccionar (Rol) elegir el rol respectivo del usuario que se está creando:

**Unidad Productiva  
Tejidos y Variedades Tota**

**Inicio Usuario Productos Facturas Pagos**

**Agregar usuario**

**Rol\***

- Cliente
- Accesaria
- Proveedor
- C.E.

Apellido\*

Tipo documento\*

Documento\*

Celular\*

**Guardar**

**Ayuda**

En el campo de **Tipo de documento** elegir la opción correspondiente al documento que será ingresado (C.C: Cedula Ciudadanía, T.I: Tarjeta de identidad, C.E: Cedula de extranjería).

**Unidad Productiva  
Tejidos y Variedades Tota**

**Inicio Usuario Productos Facturas Pagos**

**Agregar usuario**

**Rol\***

- Cliente
- Nombre\*
- Apellido\*
- Tipo documento\*
- C.C
- T.I.
- C.E

Apellido\*

Tipo documento\*

Documento\*

Celular\*

**Guardar**

**Ayuda**

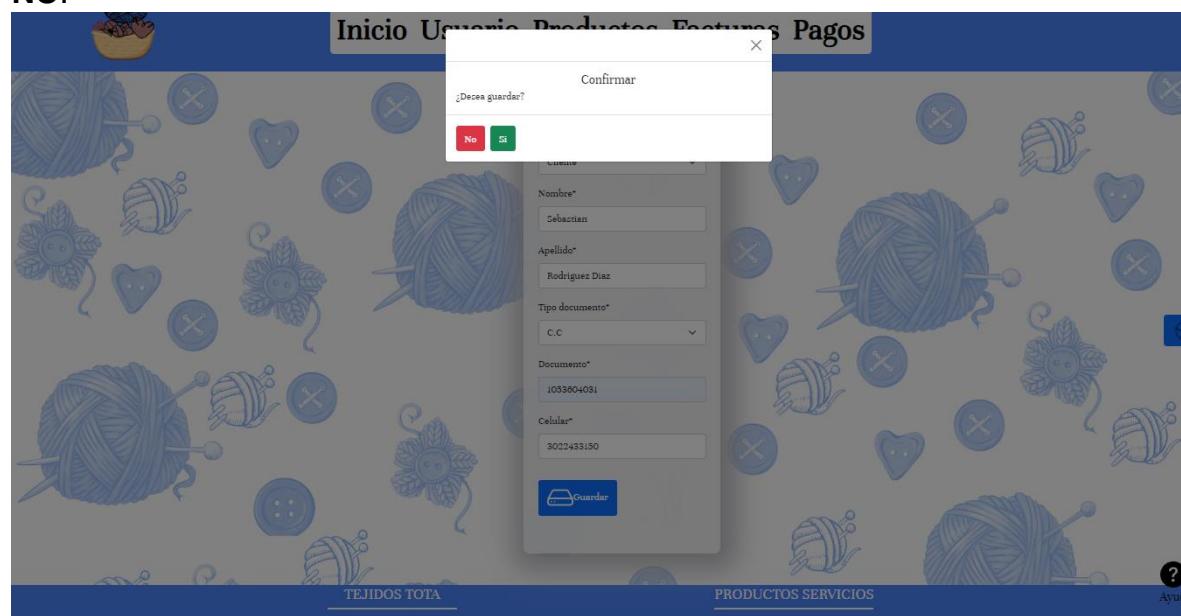
Ingresar los siguientes datos:

1. Nombre
2. Apellido
3. Tipo de documento
4. Numero de documento
5. Celular

Por último, dar en el botón azul **Guardar**.



Aparecerá una ventana emergente, donde debe confirmar guardar los datos, en el botón verde **SI**, de lo contrario si desea cambiar los datos dará en el botón rojo **NO**.



Nuevamente nos redirigirá a la tabla de usuarios, donde se puede ver el usuario anteriormente creado. Cada usuario tiene una serie de opciones, ver, editar y desactivar.

## 5.2 Ver usuario

Dando clic en el ícono de ojo al final de la tabla, este corresponde a visualizar los detalles del usuario registrado, nos redirigirá a la siguiente interfaz.

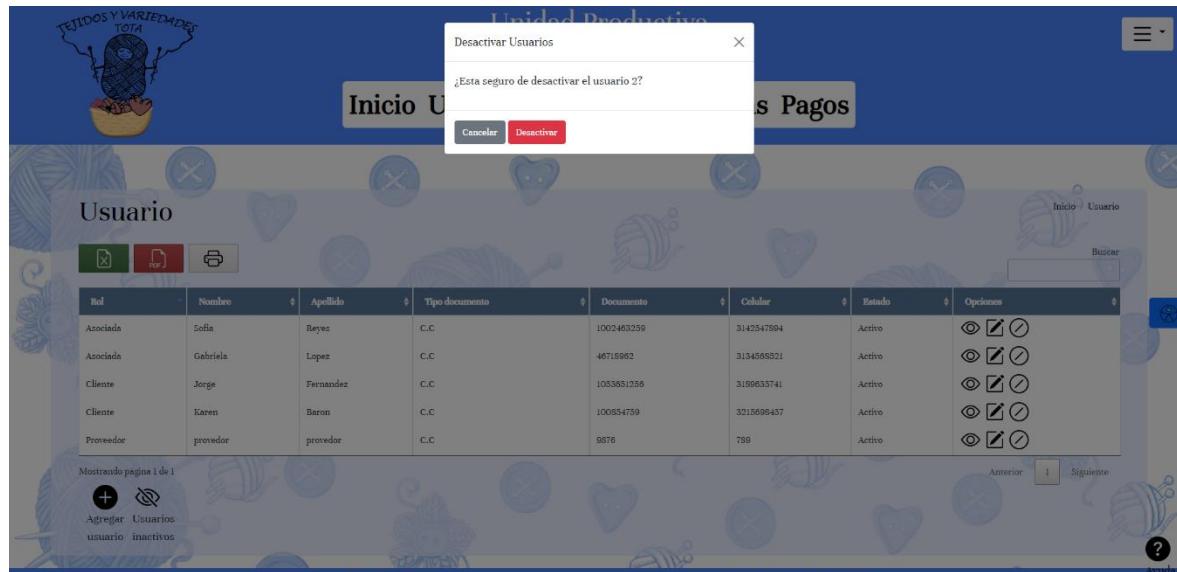
### 5.3 Editar usuario

En el segundo ícono, donde hay un lápiz, este corresponde para editar los datos de un usuario registrado, nos redirigirá a la siguiente interfaz.



### 5.4 Desactivar usuario

En el tercer ícono, donde hay un círculo con una línea enlazada, corresponde para desactivar un usuario, seguido de este aparecerá una ventana emergente para confirmar la decisión.



## 5.5 Crear super usuario

Un super usuario es el encargado de administrar todo el sistema con todos los permisos necesarios para modificar el sistema. Se crea mediante comandos:

**Python manage.py createsuperuser**

Se ingresan los datos necesarios para crear el super usuario: nombre de usuario, correo electrónico y contraseña.

```
German@German MINGW64 ~/Documents/GitHub/Proyecto-tejidos (German)
$ python manage.py createsuperuser
Nombre de usuario (leave blank to use 'german'): admin
Dirección de correo electrónico: admin@gmail.com
Password:
Password (again):
```

```
La contraseña es demasiado similar a la de nombre de usuario.
Esta contraseña es demasiado corta. Debe contener al menos 8 caracteres.
Esta contraseña es demasiado común.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Luego de la confirmación de la contraseña, se creará correctamente el super usuario.

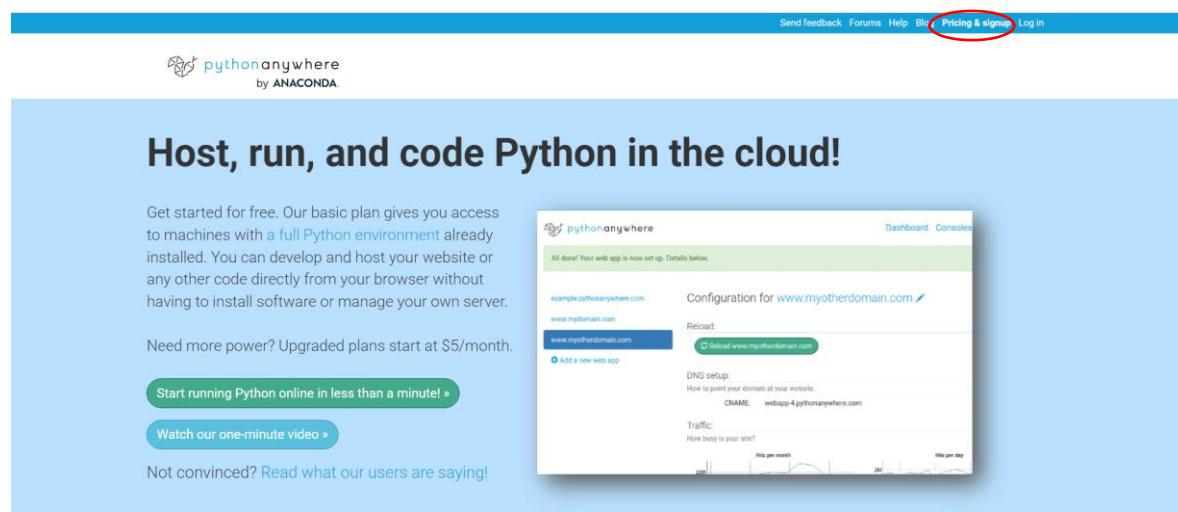
## 6. DESCRIPCIÓN DE LA PLATAFORMA

Unidad Productiva Tejidos Y Variedades Tota es una aplicación web basada en Python con el framework Django.

Su interfaz gráfica está basada en HTML5, CSS, JavaScript, JQuery, Bootstrap y Crispy. Mientras que su parte lógica está totalmente hecha en Python con la ayuda de su framework Django que se encarga de gran parte del desarrollo web haciendo así que el proyecto tenga una mejor estructuración.

### 6.1 Instalación del servidor

Entrar a la siguiente página <https://www.pythonanywhere.com/> para hacer el despliegue del proyecto. Crear una cuenta nueva.



The screenshot shows the PythonAnywhere homepage. At the top, there is a navigation bar with links: Send feedback, Forums, Help, Bio, **Pricing & signup** (which is circled in red), and Log in. Below the navigation bar, the PythonAnywhere logo is displayed, followed by the text "pythonanywhere by ANACONDA". The main heading "Host, run, and code Python in the cloud!" is prominently displayed. To the right of the heading, a screenshot of the PythonAnywhere dashboard is shown, featuring sections for DNS setup and Traffic. At the bottom left, there are two calls-to-action: "Start running Python online in less than a minute!" and "Watch our one-minute video".

Elegir el plan **Beginner**, este es el plan gratuito dentro de la página.

Send feedback | Forums | Help | Blog | Pricing & signup | Log in

 pythonanywhere  
by ANACONDA

### Plans and pricing

**Beginner: Free!**

A limited account with one web app at your-username.pythonanywhere.com, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython/Jupyter notebook support.  
It works and it's a great way to get started!

[Create a Beginner account](#)

**Education accounts**

Are you a teacher looking for a place your students can code Python? You're not alone. Click through to find out more about our Education beta.

All of our paid plans come with a no-quibble 30-day money-back guarantee – you're billed monthly and you can cancel at any time. The minimum contract length is just one month. You get unrestricted Internet access from your applications, unlimited in-browser Python, Bash and database consoles, and full SSH access to your account. All accounts (including free ones) have screen-sharing with other PythonAnywhere accounts, and free SSL support (though you'll need to get a certificate for your own domains).

Hacker	\$5/month	Web dev	\$12/month	Startup	\$99/month	Custom	\$5 to \$500/month
Run your Python code in the cloud from one web app and the console		If you want to host small Python-based websites for you or for your clients		Start a business and don't worry about having to scale to handle traffic spikes		Want a combination that's not on the list? Create your own! All custom plans have:	
A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles	
One web app on a custom domain or your-username.pythonanywhere.com		Up to 2 web apps on custom domains or your-username.pythonanywhere.com		Up to 3 web apps on custom domains or your-username.pythonanywhere.com		Up to 20 web apps on custom domains or your-username.pythonanywhere.com	
Enough power to run a typical 100,000 hit/day website. <a href="#">(more info)</a>		Enough power to run a typical 150,000 hit/day website on each web app. <a href="#">(more info)</a>		Enough power to run a typical 1,000,000 hit/day website on each web app. <a href="#">(more info)</a>		As many web workers as you need to scale your site's capacity. <a href="#">(more info)</a>	
2,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks <a href="#">(more info)</a>		4,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks <a href="#">(more info)</a>		10,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks <a href="#">(more info)</a>		Up to 100,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks <a href="#">(more info)</a>	
iPython/Jupyter notebook support		iPython/Jupyter notebook support		iPython/Jupyter notebook support		iPython/Jupyter notebook support	
1GB disk space		5GB disk space		50GB disk space		As much disk space as you choose	
<a href="#">Create a Hacker account</a>		<a href="#">Create a Web Developer account</a>		<a href="#">Create a Startup account</a>		<a href="#">Create a Custom account</a>	

Rellena los campos correspondientes para crear la cuenta, nombre de usuario, correo electrónico y contraseña.



## Create your account

Username:	<input type="text"/>
Email:	<input type="text"/>
Password:	<input type="password"/>
Password (again):	<input type="password"/>

I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

Luego de haber creado la cuenta esta será la interfaz principal de la página.

here'." data-bbox="131 530 854 807"/&gt;

Hay cuatro secciones dentro de la página, Consoles, Files, Notebook y Web apps.

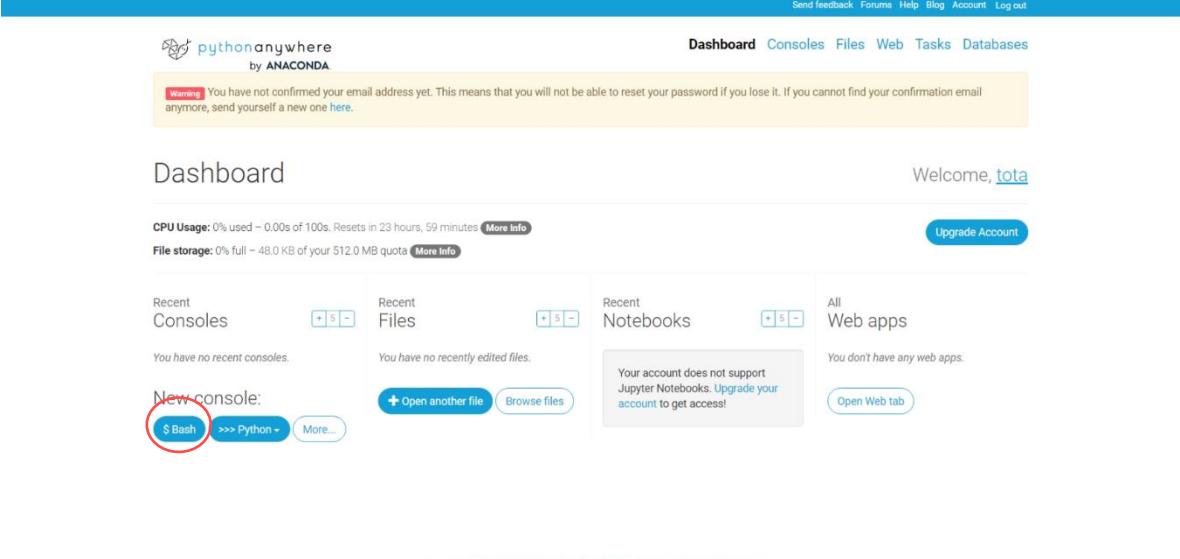
Dentro de consoles, maneja la consola respectiva para la administración del proyecto.

Dentro de files están los archivos guardados como puede ser el entorno virtual y la carpeta del proyecto.

La sección de notebooks no está activa debido al que tipo de cuenta es gratuita.

Dentro de Web apps encontramos la herramienta para administrar la página web.

1. Abrir una nueva consola en **bash** para descargar el proyecto previamente subido en GitHub.



The screenshot shows the PythonAnywhere dashboard. At the top, there's a navigation bar with links for Send feedback, Forums, Help, Blog, Account, and Log out. Below that, the PythonAnywhere logo and 'by ANACONDA' are displayed. A warning message says: 'Warning: You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#)'. The main area is titled 'Dashboard' and includes sections for Consoles, Files, Web, Tasks, and Databases. The 'Consoles' section has tabs for Recent Consoles, Recent Files, Recent Notebooks, and All Web apps. Under 'Recent Consoles', it says 'You have no recent consoles.' and shows a 'New console:' section with buttons for '\$ Bash' (circled in red), '>>> Python >', and 'More...'. Other sections show 'Recent Files' (empty), 'Recent Notebooks' (empty), and 'All Web apps' (empty). At the bottom, there's a copyright notice: 'Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies'.

2. Se abrirá la consola e ingresamos el siguiente comando:

**git clone <https://github.com/Gjaset/Tejidos-tota-proyecto.git>**

Ejecutamos el comando.

3. Cuando el proyecto se ha clonado con éxito aparecerá lo siguiente.



Bash console 25478798

```
14:08 ~ $ git clone https://github.com/Gjaset/Tejidos-tota-proyecto.git
Cloning into 'Tejidos-tota-proyecto'...
remote: Enumerating objects: 537, done.
remote: Counting objects: 100% (537/537), done.
remote: Compressing objects: 100% (347/347), done.
remote: Total 537 (delta 222), reused 494 (delta 181), pack-reused 0
Receiving objects: 100% (537/537), 8.66 MiB | 28.78 MiB/s, done.
Resolving deltas: 100% (222/222), done.
Updating files: 100% (427/427), done.
14:10 ~ $ █
```

Ahora que el proyecto ha sido clonado correctamente debemos crear el entorno virtual, activar e instalar las librerías correspondientes.

4. Para crear el entorno virtual debemos ejecutar el siguiente comando.

python -m venv .virtualenvs/(nombre del entorno).



Bash console 25478798

```
14:20 ~ $ python -m venv .virtualenvs/tejidos
14:21 ~ $ █
```

Luego de haberse creado se podrá seguir escribiendo en la consola, debemos activarlo ejecutando los comandos correspondientes en el mismo orden.

```
cd .virtualenvs
cd tejidos/
cd bin
source activate
```



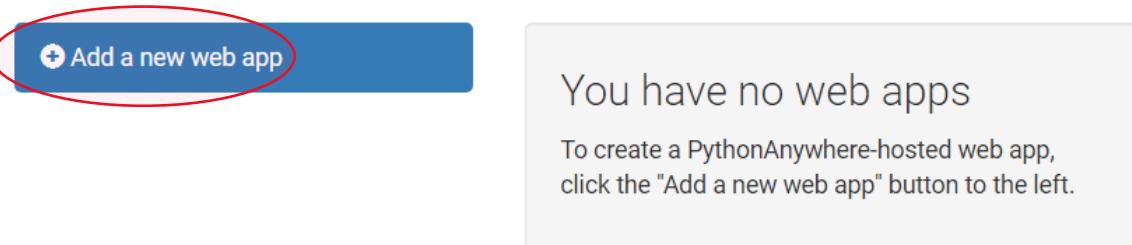
Bash console 25478798

```
14:22 ~ $ cd .virtualenvs
14:22 ~./.virtualenvs $ cd tejidos/
14:23 ~./.virtualenvs/tejidos $ cd bin
14:23 ~./.virtualenvs/tejidos/bin $ source activate
(tejidos) 14:23 ~./.virtualenvs/tejidos/bin $ █
```

5. Volvemos al **dashboard** e ingresamos a la parte Web que está ubicada en la parte derecha de la página.

Dashboard

Welcome, [tota](#)



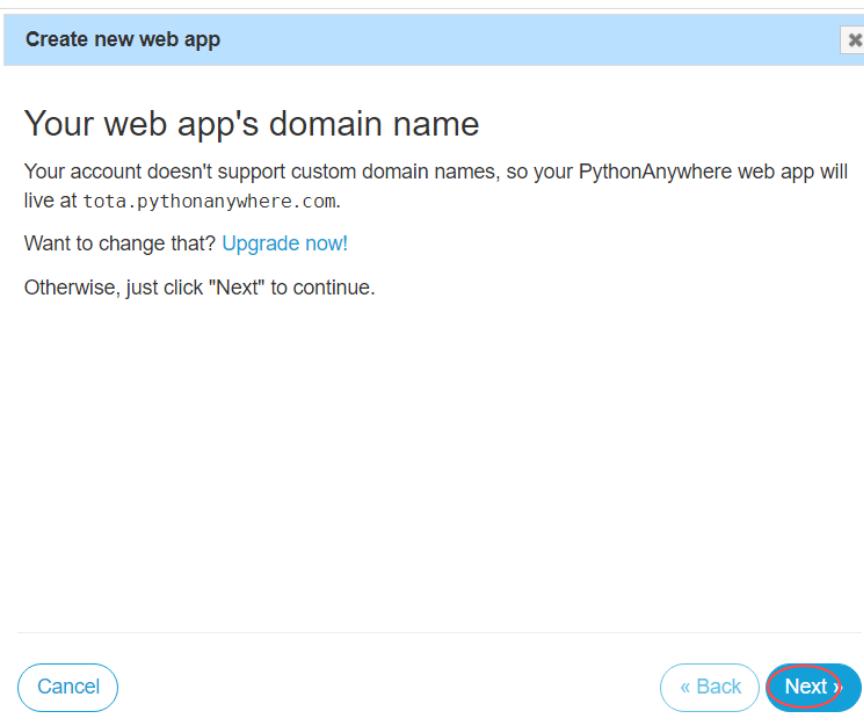
The screenshot shows the PythonAnywhere dashboard. At the top, there's a blue header bar with the text "Welcome, tota". Below it, a large blue button with a white plus sign and the text "Add a new web app" is circled in red. To the right, a white box contains the text "You have no web apps" and "To create a PythonAnywhere-hosted web app, click the 'Add a new web app' button to the left."

[\\$ Bash](#)   [>>> Python](#) [More...](#)

Tendremos la siguiente interfaz y crearemos una nueva aplicación web.

Seguimos el procedimiento para crear una aplicación web.

Elegimos **next** o **siguiente**.



The screenshot shows the "Create new web app" wizard. The title bar says "Create new web app". The main content area is titled "Your web app's domain name" and contains the message: "Your account doesn't support custom domain names, so your PythonAnywhere web app will live at tota.pythonanywhere.com." It also includes links to "Upgrade now!" and "Otherwise, just click 'Next' to continue." At the bottom, there are "Cancel" and "Next" buttons, with "Next" being circled in red.

Seleccionamos **manual configuration** y nuevamente damos en **next**.

Create new web app x

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » Flask
- » Bottle
- » **Manual configuration (including virtualenvs)**

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Cancel « Back Next »

Elegimos la versión de **Python 3.10** y nuevamente damos en **next**.

Create new web app x

Select a Python version

- » Python 2.7
- » Python 3.6
- » Python 3.7
- » Python 3.8
- » Python 3.9
- » **Python 3.10**

Cancel « Back Next »

Por último, daremos **next** para crear la aplicación web.

Create new web app ×

## Manual Configuration

Manual configuration involves editing your own WSGI configuration file in `/var/www/`. Usually this imports a WSGI-compatible application which you've stored elsewhere

When you click "Next", we will create a WSGI file for you, including a simple "Hello World" app which you can use to get started, as well as some comments on how to use other frameworks.

You will also be able to specify a `virtualenv` to use for your app.

[Cancel](#) [« Back](#) [Next »](#)

Esta será la interfaz de la configuración de la aplicación web.

[Send feedback](#) [Forums](#) [Help](#) [Blog](#) [Account](#) [Log out](#)

 pythonanywhere  
by ANACONDA

All done! Your web app is now set up. Details below. ×

**Warning** You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

tota.pythonanywhere.com Configuration for tota.pythonanywhere.com

Add a new web app

Reload: [Reload tota.pythonanywhere.com](#)

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details](#).

This site will be disabled on **Saturday 10 December 2022**

[Run until 3 months from today](#)

Paying users' sites stay up forever without any need to log in to keep them running.

Bajamos hasta encontrar **code** y **virtualenv**.

Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)

Working directory: [/home/tota/](#)

 Go to directory

WSGI configuration file: [/var/www/tota\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.10 

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[Enter path to a virtualenv, if desired](#)

En **source code** y **working directory** seleccionamos el proyecto que previamente clonamos en la consola.

En **virtualenv** seleccionamos el entorno virtual creado previamente.

Code:

What your site is running.

Source code: [/home/tota/Tejidos-tota-proyecto](#)

 Go to directory

Working directory: [/home/tota/Tejidos-tota-proyecto](#)

 Go to directory

WSGI configuration file: [/var/www/tota\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.10 

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/tota/.virtualenvs/tejidos](#)

 Start a console in this virtualenv

En **WSGI configuration file** seleccionamos el enlace correspondiente y aparecerá la siguiente interfaz.



```
/var/www/tota_pythonanywhere_com_wsgi.py

1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://tota.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some
4 # description.
5 #
6
7 # +++++++ GENERAL DEBUGGING TIPS +++++++
8 # getting imports and sys.path right can be fiddly!
9 # We've tried to collect some general tips here:
10 # https://help.pythonanywhere.com/pages/DebuggingImportError
11
12
13 # +++++++ HELLO WORLD ++++++
14 # A little pure-wsgi hello world we've cooked up, just
15 # to prove everything works. You should delete this
16 # code to get your own working.
17
18
19 HELLO_WORLD = """<html>
20 <head>
21   <title>PythonAnywhere hosted web application</title>
22 </head>
23 <body>
24 <h1>Hello, World!</h1>
25 <p>
26   This is the default welcome page for a
27   <a href="https://www.pythonanywhere.com/">PythonAnywhere</a>
28   hosted web application.
29 </p>
30 <p>
31   Find out more about how to configure your own web application
32   by visiting the <a href="https://www.pythonanywhere.com/web_app_setup/">web app setup</a> page
33 </p>
34 </body>
35 </html>"""
--
```

Debemos comentar el siguiente código, desde la línea 38 hasta la línea 47.

Comentamos con el #.

```
38 #def application(environ, start_response):
39 #    if environ.get('PATH_INFO') == '/':
40 #        status = '200 OK'
41 #        content = HELLO_WORLD
42 #    else:
43 #        status = '404 NOT FOUND'
44 #        content = 'Page not found.'
45 #    response_headers = [('Content-Type', 'text/html'), ('Content-Length', str(len(content)))]
46 #    start_response(status, response_headers)
47 #    yield content.encode('utf8')
```

Bajar hasta la línea 74 y debemos comentar desde la línea 76 hasta la línea 89. Además de editar la línea 85, `os.environ['DJANGO_SETTINGS_MODULE'] = 'gestion.settings'`, esto se hace para que el servidor reconozca los ajustes correspondientes del proyecto.

Editar la línea 81 también `path = '/home/tota/Tejidos-tota-proyecto'`

```
76 import os
77 import sys
78 #
79 ## assuming your django settings file is at '/home/tota/mysite/mysite/settings.py'
80 ## and your manage.py is is at '/home/tota/mysite/manage.py'
81 path = '/home/tota/Tejidos-tota-proyecto'
82 if path not in sys.path:
83     sys.path.append(path)
84 #
85 os.environ['DJANGO_SETTINGS_MODULE'] = 'gestion.settings'
86 #
87 #then:
88 from django.core.wsgi import get_wsgi_application
89 application = get_wsgi_application()
90
```

Guardamos y volvemos a la configuración de aplicación web.

Ubicamos **Static files**.

En **url** debemos colocar la carpeta static y en **directory** debemos colocar la dirección de la carpeta static.

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/static/	/home/tota/Tejidos-tota-proyecto/static	
Enter URL	Enter path	

Volvemos al **dashboard**, en la parte superior derecha encontramos **databases** y seleccionamos.

## Dashboard Consoles Files Web Tasks Databases

Tendremos esta interfaz, elegimos una contraseña para la base de datos, luego de eso seleccionamos **Initialize MySQL**.

Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

New password:

.....

Confirm password:

.....

• Initialize MySQL

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

Estaremos en los ajustes de MySQL, ahora debemos crear una base de datos.

### Create a database

Your database names always start with your username + "\$". There's no need to type that prefix in below, though: PythonAnywhere will automatically add it.

Database name:

septiembre

Create

Elegimos el nombre correspondiente, en este caso será **septiembre**, se selecciona **create** y se creará la base de datos.

Aquí podemos ver las bases de datos creadas en el servidor.

### Your databases:

---

Click a database's name to start a MySQL console logged in to it.

Start a console on: tota\$default

Start a console on: tota\$septiembre

Guardaremos el nombre de usuario y Database host adres para luego colocarlo en settings.py.

### Connecting:

---

Use these settings in your web applications.

Database host address: tota.mysql.pythonanywhere-services.com

Username: tota

Nuevamente debemos escribir la contraseña para la base de datos.

### MySQL password:

---

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

#### New password:

.....

#### Confirm password:

.....

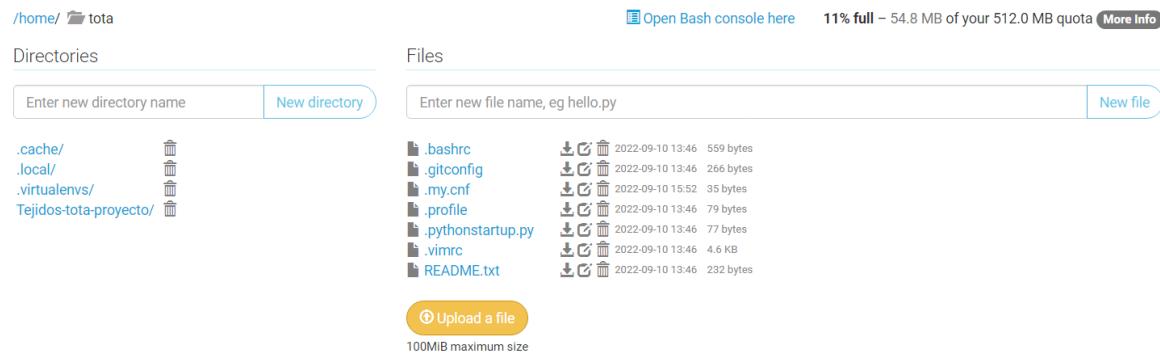
**Set MySQL password**

Damos en **Set MySQL password** para confirmar la contraseña.

Debemos dirigirnos a **files**.

Dashboard Consoles **Files** Web Tasks Databases

Encontraremos esto.



The screenshot shows a file manager interface with the following details:

- Directories:** .cache/, .local/, .virtualenvs/, Tejidos-tota-proyecto/
- Files:**
  - .bashrc (2022-09-10 13:46, 559 bytes)
  - .gitconfig (2022-09-10 13:46, 266 bytes)
  - .my.cnf (2022-09-10 15:52, 35 bytes)
  - .profile (2022-09-10 13:46, 79 bytes)
  - .pythonstartup.py (2022-09-10 13:46, 77 bytes)
  - .vimrc (2022-09-10 13:46, 4.6 KB)
  - README.txt (2022-09-10 13:46, 232 bytes)
- Actions:** Open Bash console here, 11% full – 54.8 MB of your 512.0 MB quota, More Info, New directory, New file, Upload a file (100MB maximum size).

Nos dirigimos a la siguiente ruta /Tejidos-tota-proyecto/gestión-settings.py y abrimos el archivo.

Al abrir nos dirigimos a la línea 32 y debemos poner [tota.pythonanywhere.com](http://tota.pythonanywhere.com)

```
12
13 from pathlib import Path
14 import mimetypes
15 import os
16
17 ROOT_PATH = os.path.dirname(__file__)
18
19 # Build paths inside the project like this: BASE_DIR / 'subdir'.
20 BASE_DIR = Path(__file__).resolve().parent.parent
21
22
23 # Quick-start development settings - unsuitable for production
24 # See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/
25
26 # SECURITY WARNING: keep the secret key used in production secret!
27 SECRET_KEY = 'django-insecure-8z$-bw4$yma-pu^9$0$p4@tzipjqp9v_c*)=*z7ze3pg1h8c8*'
28
29 # SECURITY WARNING: don't run with debug turned on in production!
30 DEBUG = True
31
32 ALLOWED_HOSTS = ['tota.pythonanywhere.com']
33
```

Bajar a la línea 90 y comentar hasta la línea 99. Desde la línea 101 debemos descomentar hasta la línea 110.

```
90 #DATABASES = {
91 #     'default': {
92 #         'ENGINE': 'django.db.backends.mysql',
93 #         'NAME': 'pruebasep',
94 #         'USER': 'root',
95 #         'PASSWORD': 'admin',
96 #         'HOST': 'localhost',
97 #         'PORT': '3306',
98 #     }
99 #}
100
101 DATABASES = {
102     'default': {
103         'ENGINE': 'django.db.backends.mysql',
104         'NAME': 'tota$septiembre',
105         'USER': 'tota',
106         'PASSWORD': 'mysqldjango',
107         'HOST': 'tota.mysql.pythonanywhere-services.com',
108         'PORT': '3306',
109     }
110 }
```

Guardamos y abrimos una consola dentro del proyecto.

Dentro de Web apps, bajamos hasta Virtualenv y damos clic en **Start a console in this virtualenv**.

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/tota/.virtualenvs/tejidos](#)

 [Start a console in this virtualenv](#)

Ahora debemos instalar las librerías correspondientes para la ejecución del proyecto. Se instala con **pip install (nombre de la librería)**.

 Bash console 25478798

```
(tejidos) 16:20 ~/Tejidos-tota-proyecto (main)$ pip install django
Looking in links: /usr/share/pip-wheels
Collecting django
  Downloading Django-4.1.1-py3-none-any.whl (8.1 MB)
    8.1/8.1 MB 40.6 MB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    42.3/42.3 KB 819.0 kB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.1 sqlparse-0.4.2
(tejidos) 16:22 ~/Tejidos-tota-proyecto (main)$
```

Luego ejecutamos el comando **pip list** para ver las librerías instaladas en el entorno virtual.

```
(tejidos) 16:23 ~/Tejidos-tota-proyecto (main)$ pip list
Package           Version
-----
asgiref          3.5.2
crispy-bootstrap5 0.6
Django            4.1.1
django-crispy-forms 1.14.0
mysqlclient       2.1.1
pip               22.0.4
PyMySQL           1.0.2
setuptools        58.1.0
sqlparse          0.4.2
tzdata            2022.2
(tejidos) 16:23 ~/Tejidos-tota-proyecto (main)$
```

Por último, debemos hacer la migración correspondiente de la base de datos, con el comando **python manage.py migrate**.

Así se vería la migración completa.

```
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, cuenta, facturas, gestion, pago, productos, sessions, usuarios
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying cuenta.0001_initial... OK
  Applying usuarios.0001_initial... OK
  Applying productos.0001_initial... OK
  Applying facturas.0001_initial... OK
  Applying facturas.0002_detalle_preciox... OK
  Applying gestion.0001_initial... OK
  Applying pago.0001_initial... OK
  Applying pago.0002_alter_detalle_pago_total_alter_detalle_pago_valor... OK
  Applying productos.0002_alter_producto_precio_venta... OK
  Applying productos.0003_alter_producto_porcentaje... OK
  Applying sessions.0001_initial... OK
  Applying usuarios.0002_alter_usuario_documento... OK
(tejidos) 16:28 ~/Tejidos-tota-proyecto (main)$
```

Luego de migrar la base de datos, nos dirigimos a Web apps nuevamente.

## Configuration for tota.pythonanywhere.com

Reload:

 Reload tota.pythonanywhere.com

Damos en el botón **Reload tota.pythonanywhere.com** y por último damos clic en el enlace de arriba para abrir la aplicación web.

Despliegue correcto, esta es la interfaz del proyecto dentro del servidor.



## 6.2 Mantenimiento del software

Es recomendable actualizar las librerías en un periodo de 6 meses a 1 año.

IDE Visual studio code

Python 3.10 o superior

crispy-bootstrap 5 0.6

Django 4.1

django-crispy-forms 1.14.0

mysqlclient 2.1.1

PyMySQL 1.0.2

## 7. DOCUMENTACIÓN DEL CODIGO FUENTE

```
✓ cuenta
  • __init__.py
  • admin.py
  • apps.py
  • forms.py
  • models.py
  • tests.py
  • urls.py
  • views.py

✓ facturas
  • __init__.py
  • admin.py
  • apps.py
  • forms.py
  • models.py
  • tests.py
  • urls.py
  • views.py
```

```
✓ gestion
  ✓ templates
    > ayuda
    > factura
    > modals
    > pago
    > partials
    > producto
    > registration
    > user
    > usuario
    <> backup.html
    <> index.html
    <> inicio.html
  • __init__.py
  • asgi.py
  • forms.py
  • models.py
  • settings.py
  • urls.py
  • views.py
  • wsgi.py
```

```
✓ pago
  • __init__.py
  • admin.py
  • apps.py
  • forms.py
  • models.py
  • tests.py
  • urls.py
  • views.py

✓ productos
  • __init__.py
  • admin.py
  • apps.py
  • forms.py
  • models.py
  • tests.py
  • urls.py
  • views.py

✓ static
  > backup
  > css
  > fontawesomefree
  > fonts
  > img
  > js
```

```
✓ usuarios
  • __init__.py
  • admin.py
  • apps.py
  • forms.py
  • models.py
  • tests.py
  • urls.py
  • views.py
  • manage.py
```

## 8. ESTRUCTURA DE DIRECTORIOS

Directorio	Descripcion
<b>Gestion</b>	Es la carpeta principal, contiene los ajustes correspondientes del framework, los enlaces para cada modelo y plantilla.
<b>Facturas</b>	Este modelo contiene las clases, logica de la creacion de facturas.
<b>Cuenta</b>	Este modelo contiene la creacion de cuenta, inicio de sesion y restablecimiento de contraseña.
<b>Pago</b>	Este modelo contiene la logica de los pagos que se afectua hacia el rol "asociada"
<b>Productos</b>	Este modelo contiene las clases, la parte logica de la creacion de productos.
<b>Usuarios</b>	Este modelo contiene las clases, la parte logica de la gestion de usuarios, es diferente al modelo de cuenta.
<b>Static</b>	Contiene las imágenes, los estilos en cascada, archivos JavaScript y fuentes del proyecto.
<b>Templates</b>	Contiene las plantillas en HTML5 del proyecto.

## 9. RELACIÓN DE PROGRAMAS

Programa	Lenguaje	Descripción
Gestion	Python	Implementa la clase backup,
Pago	Python	Implementa las clases Pago y Detalle_pago.
Prodctos	Python	Implementa las clases Talla, Marca, Color y Prodcuto .
Facturas	Python	Implementa las clases Factura y Detalle.
Usuarios	Python	Implementa las clases Rol y Usuario.
Style.css	CSS	Contiene los estilos correspondientes para cada una de las interfaces HTML.

## 10. DESCRIPCIÓN DE LOS ACUERDOS DE NIVELES DE SERVICIOS O ANS

De acuerdo con los lineamientos establecidos entre la mesa de ayuda y el cliente se determinó que estará presta a responder cualquier tipo de inconveniente que se presente en la aplicación esta dará solución al cliente en un límite de tiempo máximo de 7 días hábiles, siguiendo el protocolo de identificación y solución al problema.

## 11. GLOSARIO

**Clic:** Es la acción de pulsar cualquier botón o tecla del dispositivo apuntador de la computadora.

**Chrome:** es un navegador web de código cerrado desarrollado por Google.

**GB:** Un gigabyte (GB) es una unidad estándar usada con mucha frecuencia en informática. Aporta información sobre la capacidad de almacenamiento de un medio o el tamaño de un archivo.

**Windows:** es un sistema operativo, es decir, un programa de software que admite funciones básicas, como la administración de archivos y la ejecución de aplicaciones, y que usa dispositivos periféricos, como la impresora, el monitor, el teclado y el ratón.

**Ventana emergente:** es una ventana superpuesta a la de la página web activa.

**Icono:** un pictograma que es utilizado para representar archivos, carpetas, programas, unidades de almacenamiento, etc. en un sistema operativo gráfico.

**Filtrar:** Los Filtros son unos mecanismos que, sea en el campo de la informática o en el de las páginas web, se utilizan para determinar qué tipo de contenidos se deben visualizar o mostrar por pantalla, o cuáles no deben aparecer. Es algo que se emplea en numerosos lugares con fines muy diferentes, pero que siempre tiene como objetivo perfilar la búsqueda para acertar mejor con los resultados.

**Rol:** función que alguien o algo desempeña

**Stock:** Stock, término anglosajón el cual indica la cantidad de productos o materias primas que posee un comercio en su almacén a la espera de su venta o comercialización.

**Interfaz:** Como interfaz designamos, en informática, la conexión física y funcional que se establece entre dos aparatos, dispositivos o sistemas que funcionan independientemente uno del otro. En este sentido, la comunicación entre un ser humano y una computadora se realiza por medio de una interfaz.

**CRUD:** es una agrupación de cuatro acciones que sirven para gestionar la información que se almacena en diferentes tipos de desarrollo. Se usa para comprobar que los desarrollos sean completamente funcionales, sus siglas (Create: crear, Read: leer, Update: actualizar, Delete: eliminar).

**Python:** es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros.

**HTML:** Es el código que se utiliza para estructurar y desplegar una página web y sus contenidos.

**Software :** es el conjunto de programas o aplicaciones, instrucciones y reglas informáticas que hacen posible el funcionamiento del equipo.

**Framework:** Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software

**ANS:** Es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado mediante indicadores y métricas asociadas para la calidad de dicho servicio.

**Backup:** se trata de guardar datos en un soporte de almacenamiento adecuado (como un disco duro externo, por ejemplo). Al finalizar el backup, se habrá creado una copia de seguridad que contendrá los datos importantes de forma redundante, es decir, en una copia o repetición de los mismos.

**Django:** Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC).

**MySQL:** es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo.

**Laragon:** es una herramienta para equipos técnicos que permite crear diferentes entornos de desarrollo, facilitando el trabajo con las aplicaciones.

**CMD:** Command Prompt es el intérprete de comandos y sistemas basados en Windows.

**Visual Studio Code:** es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

**Entorno virtual:** es un entorno de Python parcialmente aislado que permite instalar paquetes para que los use una aplicación en particular, en lugar de instalarlos en todo el sistema.

**Workbench:** es un entorno gráfico de diseño de bases de datos, servidores, administración y mantenimiento para el sistema MySQL. Además, esta herramienta gráfica fue desarrollada y distribuida por la compañía de desarrollo de nube y locales Oracle Corporation.

**GitHub:** es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarla la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

**Patrón MTV:** (modelo-vista-plantilla) donde el modelo maneja todo lo relacionado con la información, la vista es un enlace entre el modelo y la plantilla, y la plantilla decide cómo se mostrará la información.

**Bootstrap:** Es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegables, etc. También existe la posibilidad de utilizar extensiones de JavaScript adicionales.

**JavaScript:** Es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el nombre, no está relacionado con Java.

## 12. REFERENCIAS

- [https://www.amaq.edu.pe/public\\_html/Docs/files/mensaje-20-cierre-de-aplicaciones-instructivo.pdf7](https://www.amaq.edu.pe/public_html/Docs/files/mensaje-20-cierre-de-aplicaciones-instructivo.pdf7)
- <https://www.mindomo.com/es/mindmap/mentefanto-de-navegadores-e9298629b1d4419bb49832ab40a0a21f#:~:text=GOOGLE%20CHROME,Est%C3%A1%20disponible%20gratuitamente.>
- <https://neoattack.com/neowiki/filtros/>
- [https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-un-gigabyte/#:~:text=Un%20gigabyte%20\(GB\)%20es%20una,est%C3%A1%20incluida%20en%20el%20contarato.](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-un-gigabyte/#:~:text=Un%20gigabyte%20(GB)%20es%20una,est%C3%A1%20incluida%20en%20el%20contarato.)
- <https://privacy.microsoft.com/es-mx/windows10privacy>
- [https://es.wikipedia.org/wiki/Ventana\\_emergente](https://es.wikipedia.org/wiki/Ventana_emergente)
- <https://www.studocu.com/co/document/servicio-nacional-de-aprendizaje/fundamentos-marketing/html/17104433>
- <https://es.wikipedia.org/wiki/Python#:~:text=Python%20es%20un%20lenguaje%20de,programaci%C3%B3n%20multiparadigma%2C%20ya%20que%20soporta>
- <https://www.crehana.com/blog/desarrollo-web/que-es-crud/>
- <https://prezi.com/wrraqsgmbsyj/un-pictograma-que-es-utilizado-para-representar-archivos-c/>
- <https://dle.rae.es/rol?m=form>
- <https://www.significados.com/stock/>
- <https://www.significados.com/interfaz/>
- [https://es.wikipedia.org/wiki/Django\\_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework))
- <https://es.wikipedia.org/wiki/MySQL>
- <https://www.arsys.es/blog/introduccion-a-laragon>
- [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code)
- <https://omes-va.com/virtualenv-python/>
- <https://keepcoding.io/blog/que-es-mysql-workbench/>
- <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- <https://espifreelancer.com/mtv-django.html>
- <https://www.arimetrics.com/glosario-digital/bootstrap>
- <https://www.arimetrics.com/glosario-digital/javascript>