**Manual de Usuario**

# Instalación de los equipos

PASO 1: Sacar los equipos de sus respectivos empaques, los cuales son:

-2 Cámaras web   
-2 Servomotores  
-Equipo Controlador

PASO 2: Colocar una camara en la entrada y otra en la salida del local, si solo tiene una vía de acceso, se recomienda ubicarlos de una manera que no dificulte el ingreso.

NOTA: las cámaras deben esta ubicadas de preferencia de tal forma que la cámara no tenga visión a la fila del local dado que puede reconocer los rostros de los clientes y dañaría el sistema.

PASO 3: Preparar las puertas del local para que puedan ser accionadas por los servomotores, los cuales serán ubicados en estas.

PASO 4: Colocar el equipo controlador a una distancia en la cual los cables que lo conectan no lleguen a tensarse, para evitar un mal funcionamiento del sistema.

Paso 5: Conectar el equipo controlador al equipo en donde se realizará las pertinentes configuraciones.

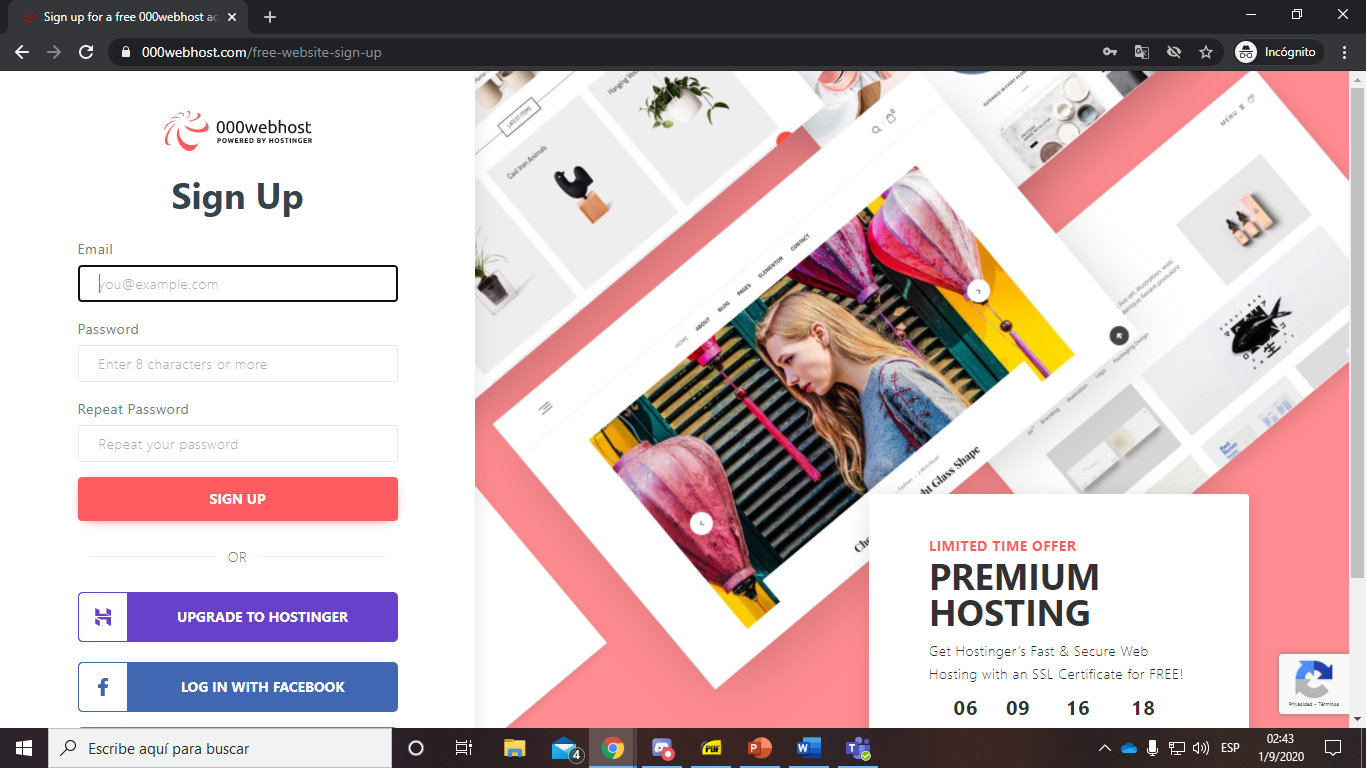
# Instalar IDE Arduino en el sistema operativo

PASO 1: ir a la página oficial de arduino en la seccion de software descargar el instalador del IDE <https://www.arduino.cc/en/Main/Software>

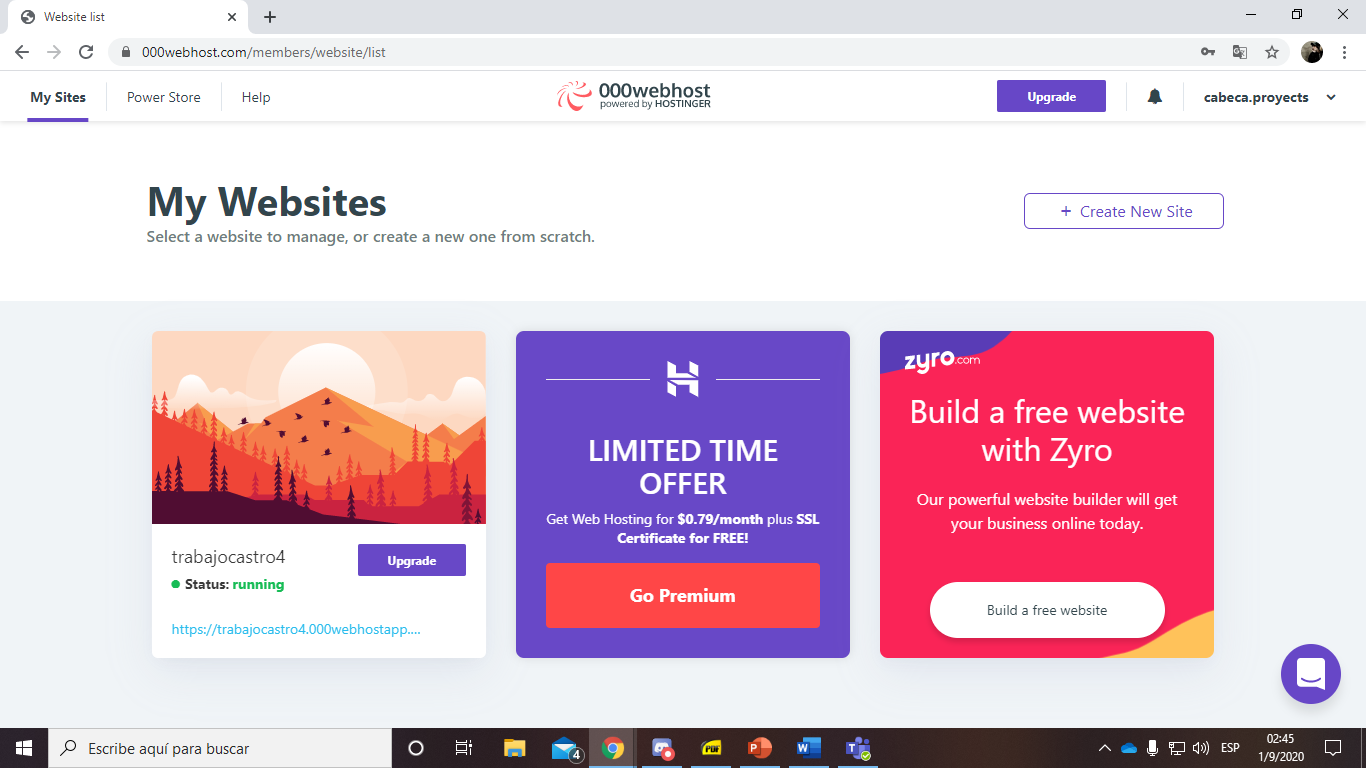


# Manejo sistema de administración

PASO 1: Ir a la pagina <https://www.000webhost.com/> y crearse una cuenta

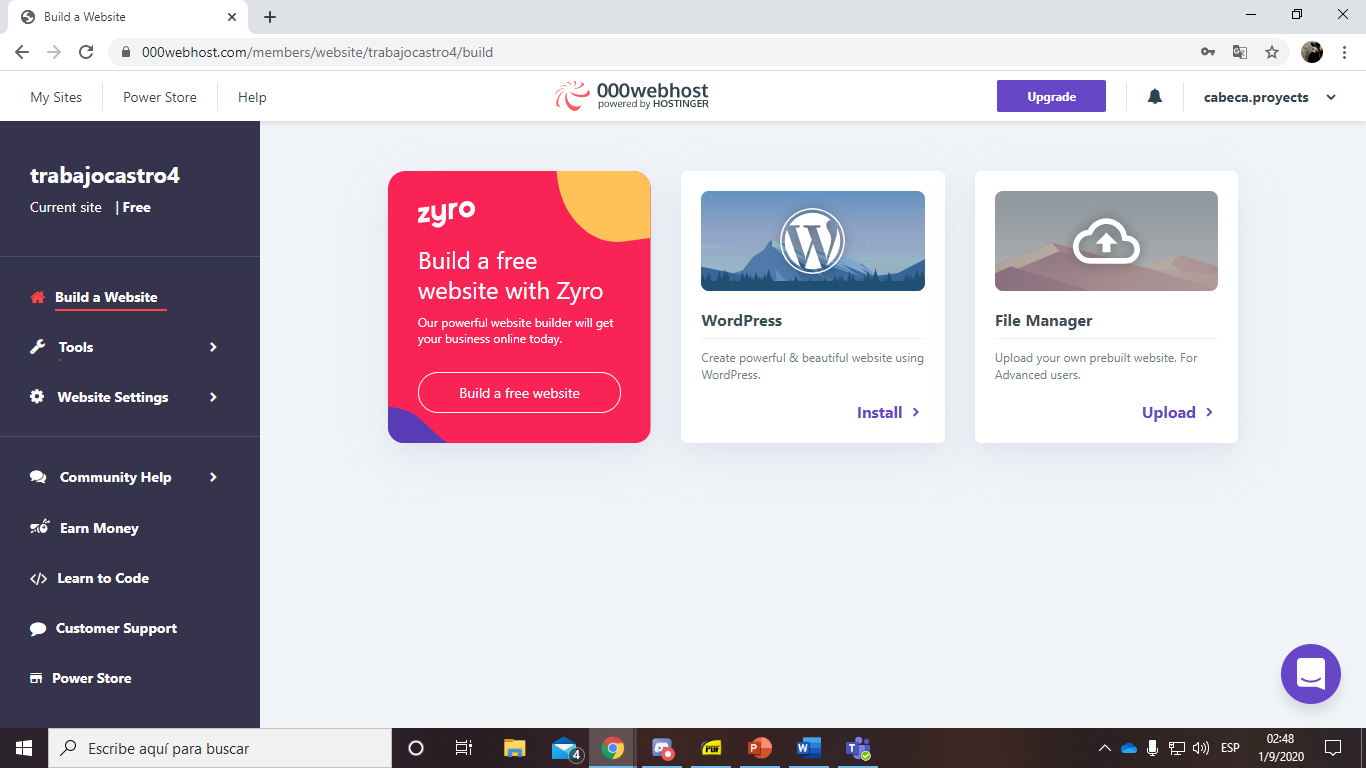


PASO 2 : Ir a la opción de “ + Create a new site”   
Recomendación: Asegurarse de guardar los datos como nombres y contraseñas para futuras referencias.



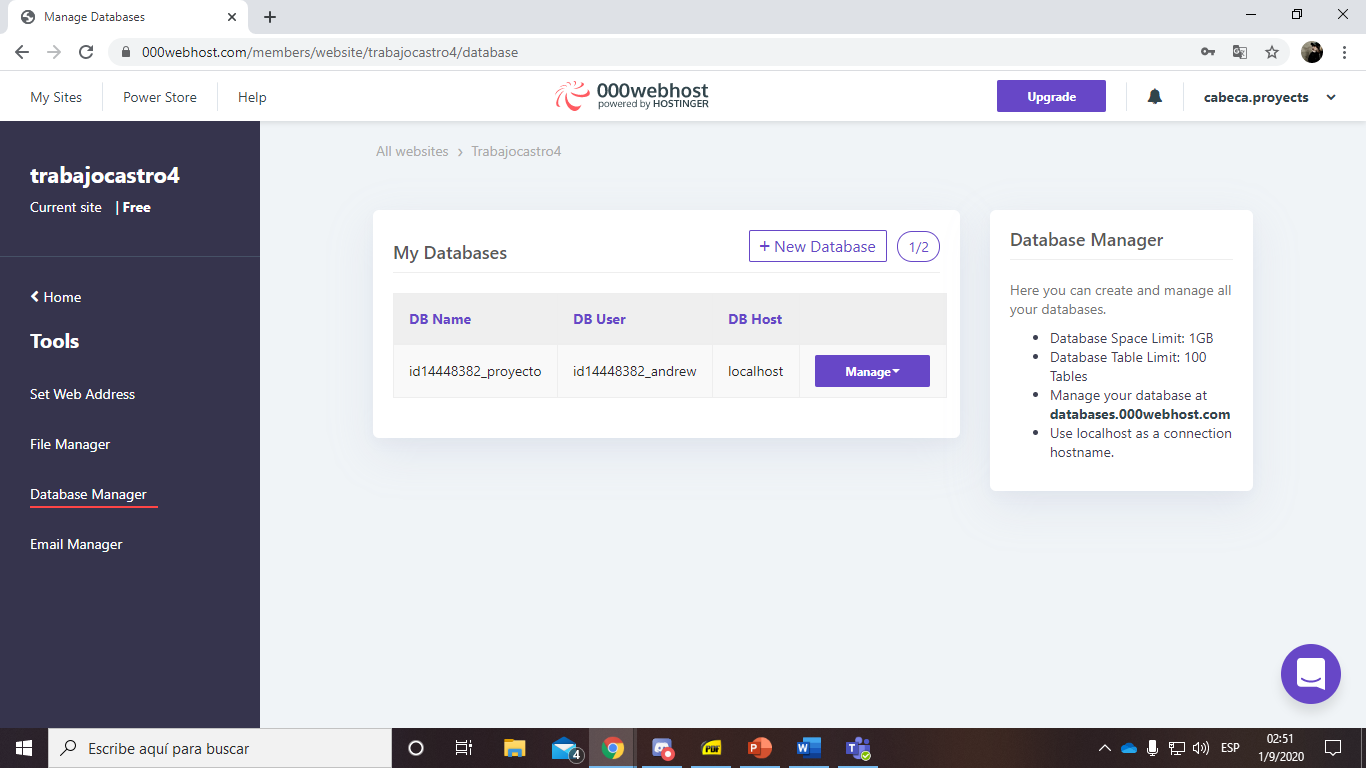
PASO 3: Después de la creación, ponemos el cursor encima de la pagina creada, y le damos click a “Manage Website”

Se desplegará la siguiente ventana



Le damos a la opción de “ Tools “ y le damos a la opción de “ Database managear”

PASO 4: Le damos a la opción “ + New database” Y le asignaremos los valores como nombre y contraseña  
Importante: Tener un registro de todos estos valores ya que serán las credenciales de acceso para la pagina web.



PASO 5: Para acceder a la base de datos le damos click a la opción que dice “Manage” de la cual se desplegara un menú, seleccionamos “PhPMyAdmin”

PASO 6: Para crear las tablas vaya a la seccion de “ SQL “ e inserte los códigos a continuación.

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `informe`

--

CREATE TABLE `informe` (

`id\_informe` int(11) NOT NULL,

`fecha` datetime NOT NULL,

`conteo\_diario` int(11) NOT NULL,

`id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `local`

--

CREATE TABLE `local` (

`id` int(11) NOT NULL,

`ubicacion` varchar(50) COLLATE utf8\_unicode\_ci NOT NULL,

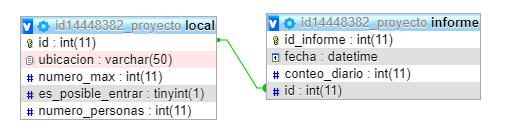
`numero\_max` int(11) NOT NULL,

`es\_posible\_entrar` tinyint(1) NOT NULL,

`numero\_personas` int(11) NOT NULL

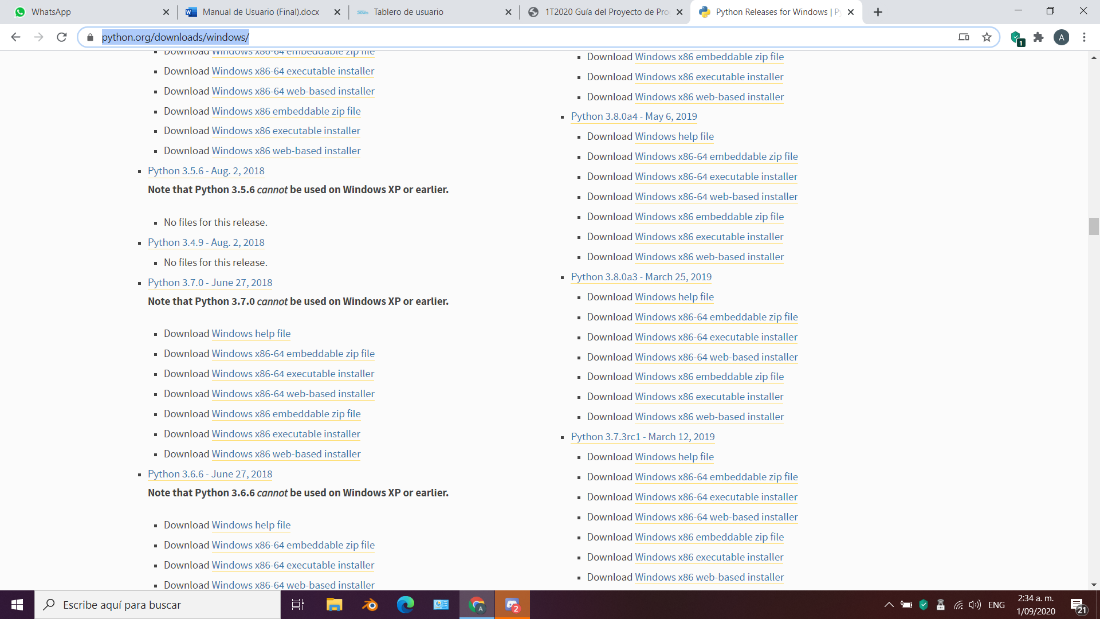
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

PASO 7: Revisar que sus tablas cumplen el siguiente esquema.



# Instalar Python en sistema operativo

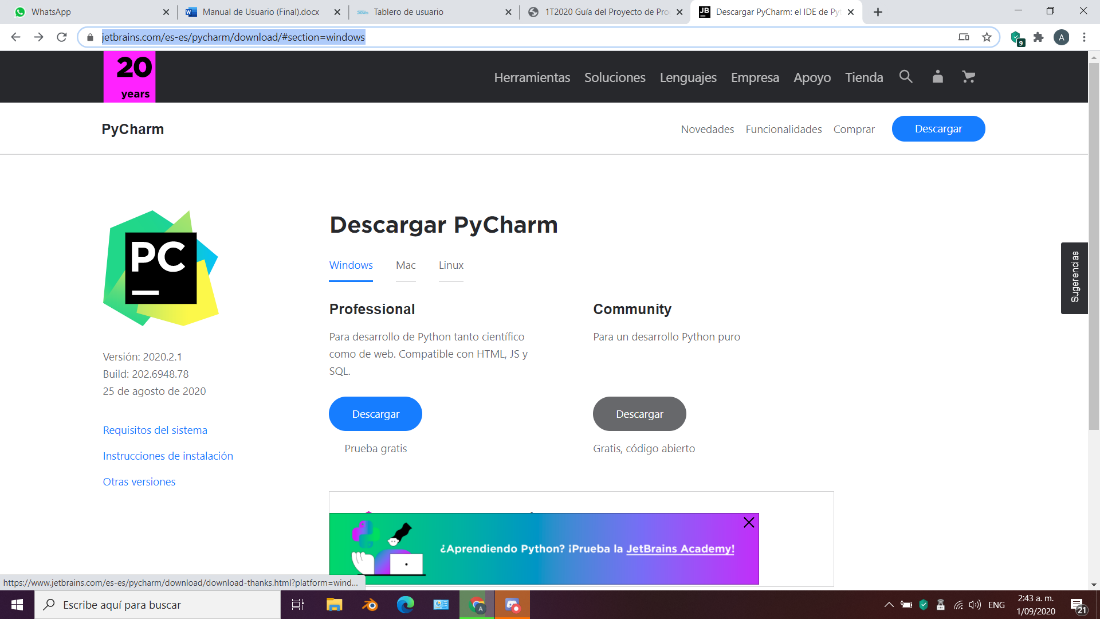
PASO 1 : ir a python.org y en la seccion de descargas seleccione el sistema operativo que posee y descargar la version 3.7.0 de Python <https://www.python.org/downloads/windows/>



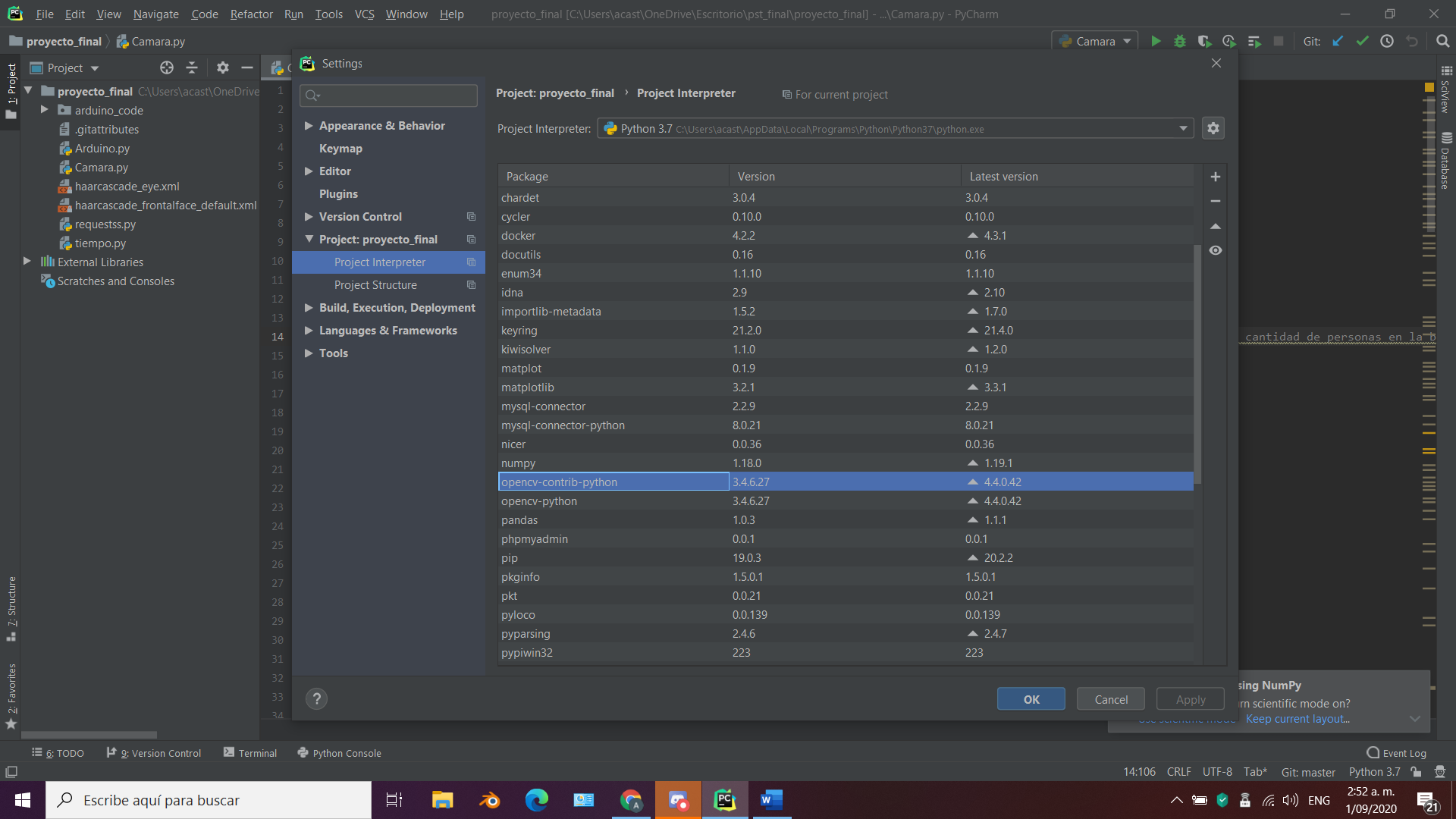
Paso 2: ir a la página de jetbrains.com y en la seccion de herramientas seleccionar pycharm

Paso 3: seleccionar descargar e indicar la version de comunity

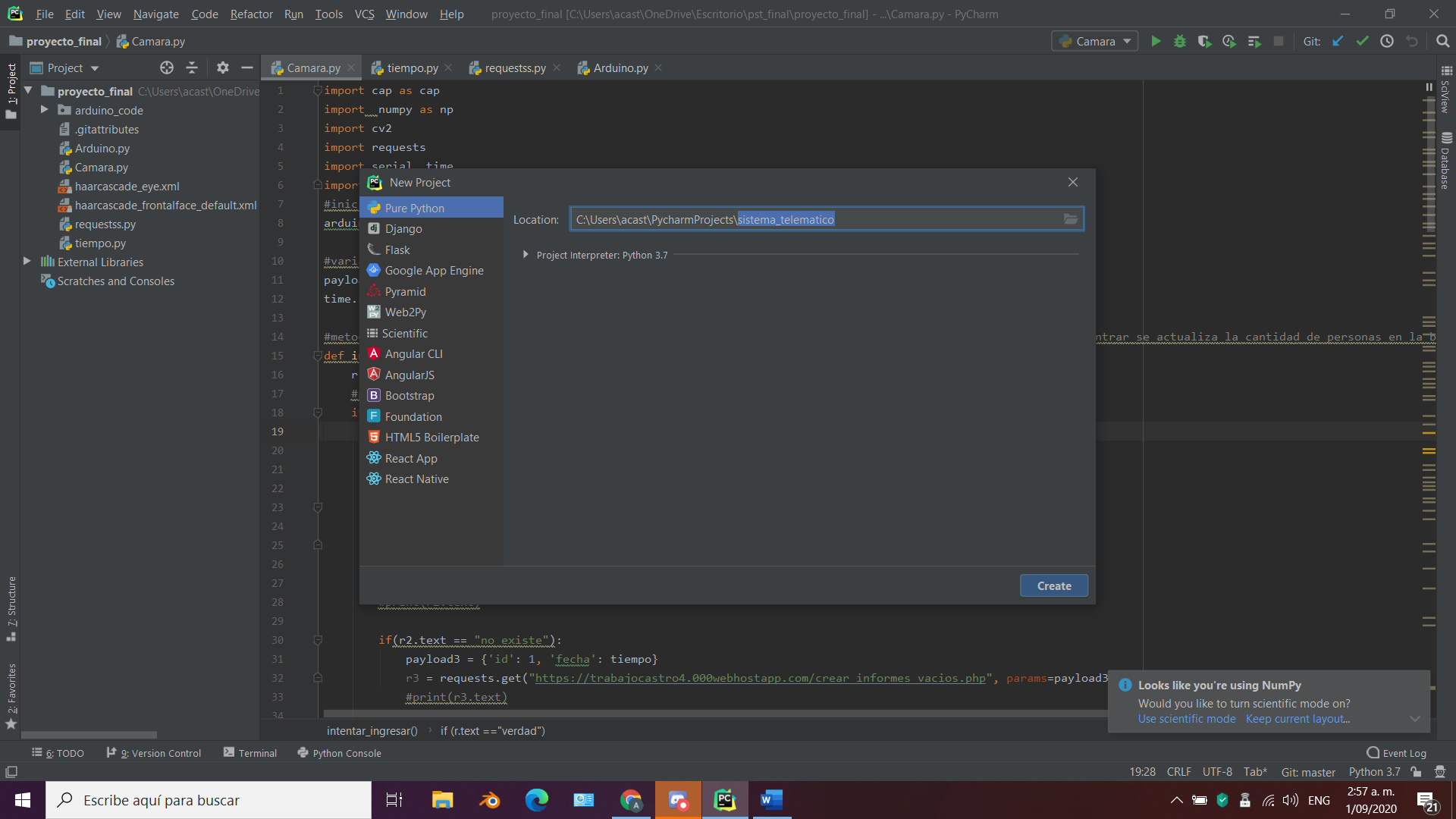
<https://www.jetbrains.com/es-es/pycharm/download/#section=windows>



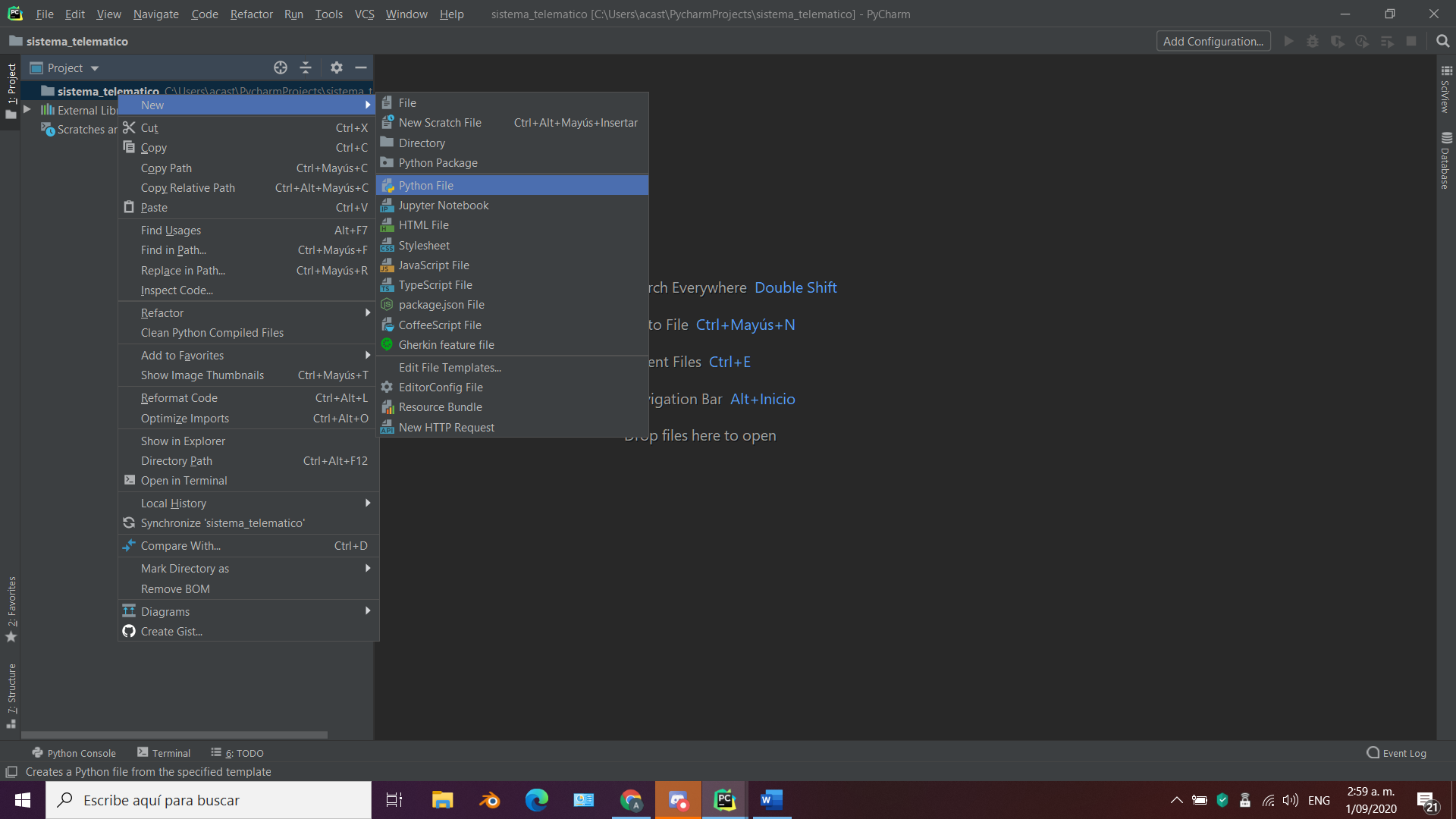
Paso 4: abrir pycharm e ir a file->settings->project interpreter y pulsar el boton del signo de suma para instalar numpy,opencv-python y opencv-contrib-python(versión 3.4.6.27).



Paso5 : crear un nuevo proyecto (file->new Project->pure python ) y seleccionar donde y que nombre desea crear el proyecto.



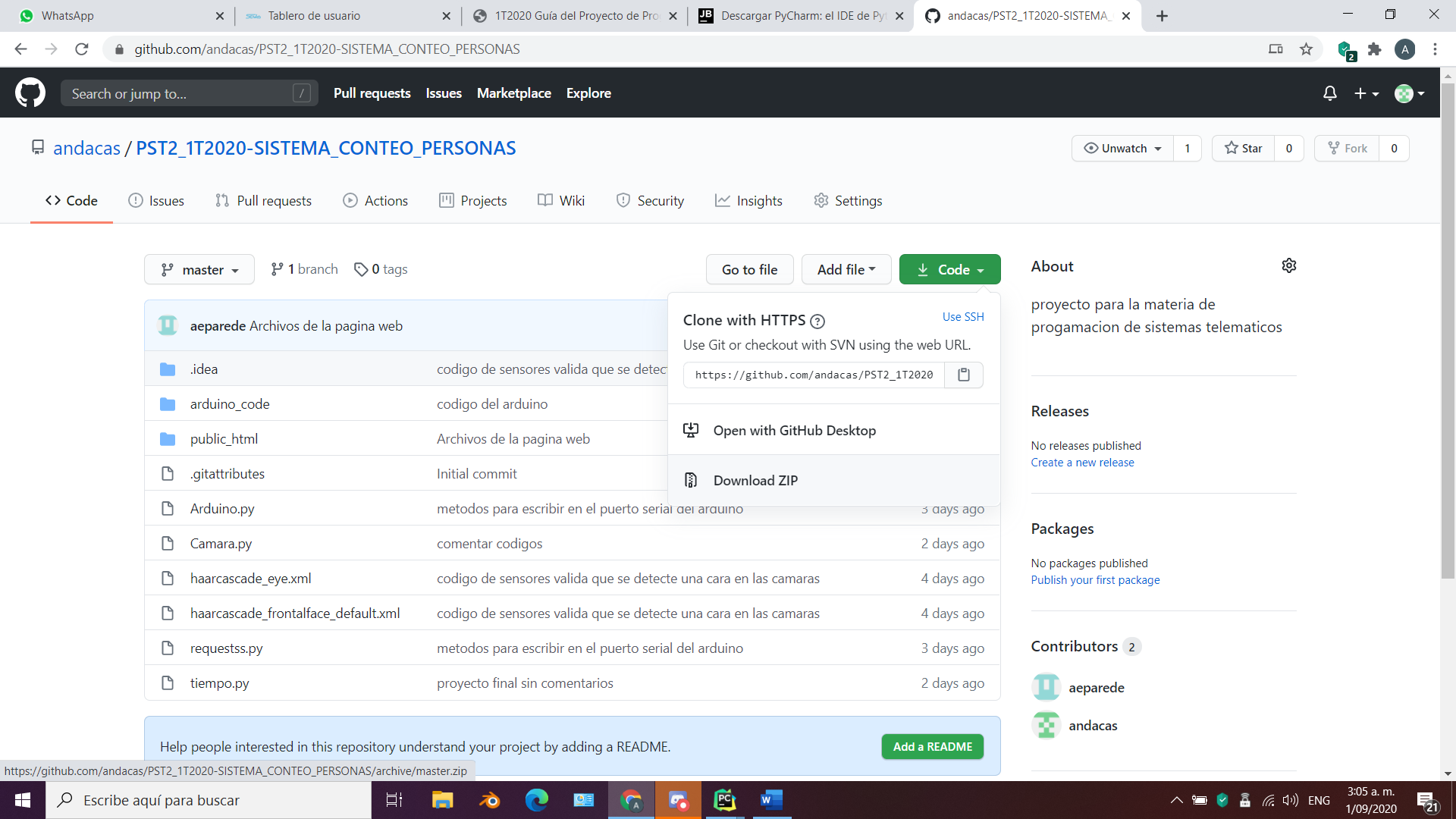
Paso6: crear un archivo de Python haciendo click derecho en la carpeta del proyecto-> new ->Python file



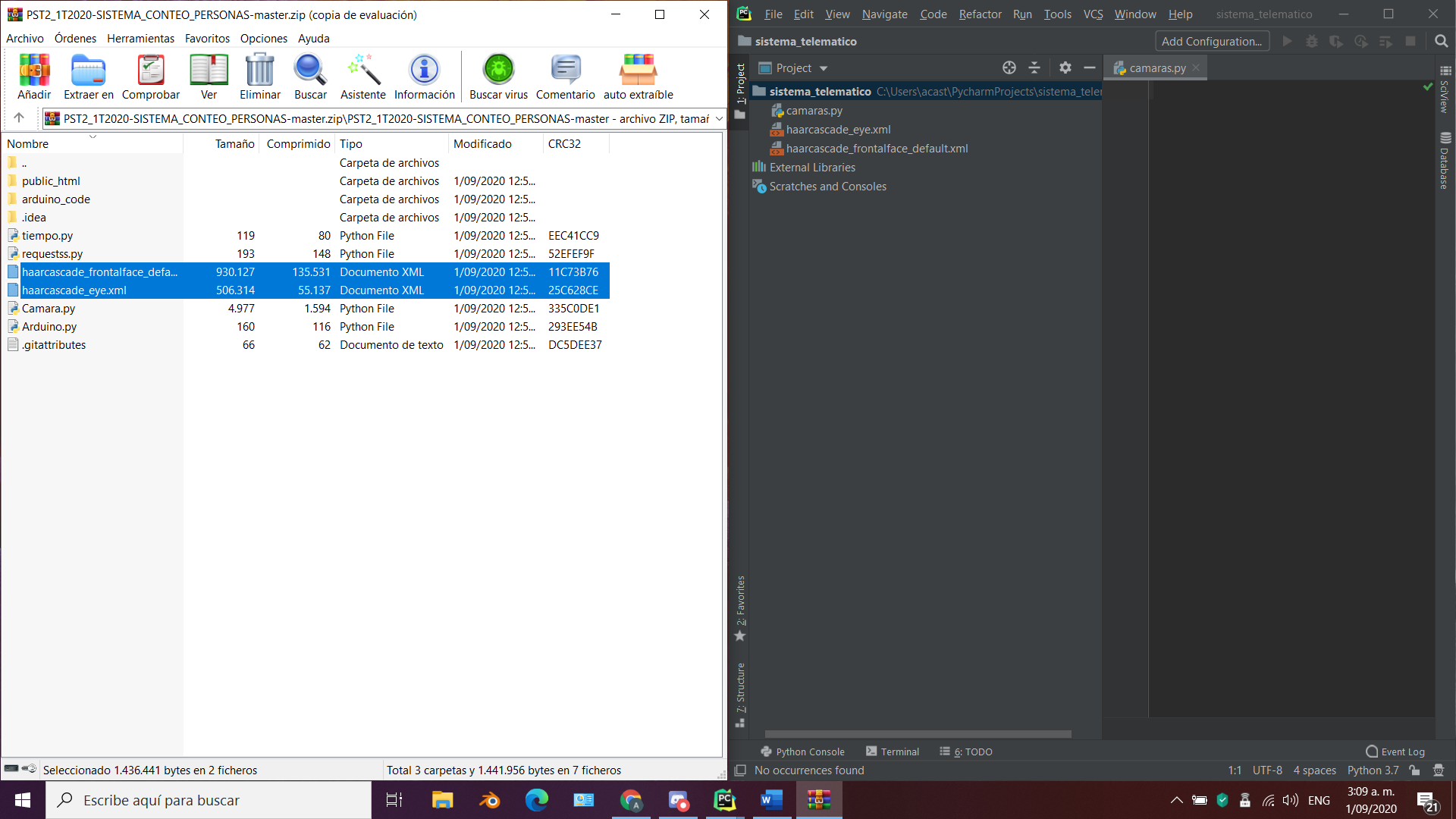
Paso7 : copiar y pegar el código

import cap as cap  
import numpy as np  
import cv2  
import requests  
import serial, time  
import datetime  
#inicializar arduino  
arduino = serial.Serial("COM3", 9600)  
  
#variable para hacer el query  
payload ={'id':1}  
time.sleep(2)  
  
#metodo que intenta acceder a la base de datos validando que se puede ingresar al local especifico: si se puede entrar se actualiza la cantidad de personas en la base de datos  
def intentar\_ingresar():  
 r = requests.get("https://trabajocastro4.000webhostapp.com/cantidad\_python.php", params=payload)  
 #print(r.text)  
 if(r.text =="verdad"):  
 arduino.write(b'I')  
 arduino.write(b'D')  
 tiempo = datetime.datetime.now()  
 tiempo = tiempo.replace(minute=0, second=0, microsecond=0)  
 #print(tiempo)  
  
 #preguntar si en la base de datos existe el informe a esa hora , en caso de no existir se crea uno  
 payload2 = {'id': 1,'fecha':tiempo}  
 r2 = requests.get("https://trabajocastro4.000webhostapp.com/revisar\_informe.php", params=payload2)  
 #print(r2.text)  
  
 if(r2.text == "no existe"):  
 payload3 = {'id': 1, 'fecha': tiempo}  
 r3 = requests.get("https://trabajocastro4.000webhostapp.com/crear\_informes\_vacios.php", params=payload3)  
 #print(r3.text)  
  
 #incrementar el número de personas que han ingresado  
 payload4 = {'id': 1, 'fecha': tiempo}  
 r4 = requests.get("https://trabajocastro4.000webhostapp.com/ingreso\_por\_id.php", params=payload4)  
 #print(r4.text)  
  
 #en caso de que no se puede ingresar por numero de personas se cambia estado del arduino  
 else:  
 arduino.write(b'F')  
  
 #se vuelve a preguntar si se puede ingresar para actualizar el estado del arduino  
 r = requests.get("https://trabajocastro4.000webhostapp.com/cantidad\_python.php", params=payload)  
 #print(r.text)  
 if (r.text == "verdad"):  
 arduino.write(b'D')  
 else:  
 arduino.write(b'F')  
  
  
  
  
#metodo que intenta acceder a la base de datos validando que existe a esa hora un informe para un local especifico: si existe actualiza la cantidad de personas en la base de datos  
def intentar\_salir():  
 #cambiar estado del arduino para salir y que esta disponible para ingreso  
 arduino.write(b'S')  
 arduino.write(b'D')  
 tiempo = datetime.datetime.now()  
 tiempo = tiempo.replace(minute=0, second=0, microsecond=0)  
 #print(tiempo)  
 #preguntar si existe informe a esa hora para el local  
 payload2 = {'id': 1, 'fecha': tiempo}  
 r2 = requests.get("https://trabajocastro4.000webhostapp.com/revisar\_informe.php", params=payload2)  
 #print(r2.text)  
 #si no existe se crea un informe a esa hora  
 if (r2.text == "no existe"):  
 payload3 = {'id': 1, 'fecha': tiempo}  
 r3 = requests.get("https://trabajocastro4.000webhostapp.com/crear\_informes\_vacios.php", params=payload3)  
 #print(r3.text)  
 #se reduce la cantidad que esta actualmente en el local  
 payload4 = {'id': 1, 'fecha': tiempo}  
 r4 = requests.get("https://trabajocastro4.000webhostapp.com/salida\_por\_id.php", params=payload4)  
 #print(r4.text)  
  
  
  
#se hace uso de haarcascade para detectar una cara de frente con los ojos si estan los unos un tiempo definido por los contadores, cuando se cumple ese tiempo se  
# ejecutan las funciones de intento entrar y de intento salir dependiendo de la camara que está reconociendo la cara  
face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')  
eye\_cascade = cv2.CascadeClassifier('haarcascade\_eye.xml')  
cap = cv2.VideoCapture(1)  
cap2 = cv2.VideoCapture(0)  
hay\_elementos\_1 = False  
hay\_elementos\_2 = False  
contar\_cuadros\_1=0;  
contar\_cuadros\_2 = 0;  
while True:  
 ret, img = cap.read()  
 gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)  
 faces = face\_cascade.detectMultiScale(gray,1.3,5)  
 hay\_elementos\_1 = False  
 for (x,y,w,h) in faces:  
 cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)  
 roi\_gray = gray[y:y+h , x:x+w]  
 roi\_color = img[y:y+h , x:x+w]  
 eyes = eye\_cascade.detectMultiScale(roi\_gray)  
 for(ex,ey,ew,eh) in eyes:  
 cv2.rectangle(roi\_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)  
 hay\_elementos\_1 = True  
 if(hay\_elementos\_1):  
 contar\_cuadros\_1 += 1  
 if(contar\_cuadros\_1>=30):  
 print("entrar.....")  
 intentar\_ingresar()  
 contar\_cuadros\_1 = 0  
 else:  
 contar\_cuadros\_1 = 0  
 cv2.imshow('img',img)  
 k = cv2.waitKey(30) & 0xff  
 if k ==27:  
 break  
  
  
 ret2, img2 = cap2.read()  
 gray2 = cv2.cvtColor(img2, cv2.COLOR\_BGR2GRAY)  
 faces2 = face\_cascade.detectMultiScale(gray2, 1.3, 5)  
 hay\_elementos\_2 = False  
 for (x, y, w, h) in faces2:  
 cv2.rectangle(img2, (x, y), (x + w, y + h), (255, 0, 0), 2)  
 roi\_gray2 = gray2[y:y + h, x:x + w]  
 roi\_color2 = img2[y:y + h, x:x + w]  
 eyes2 = eye\_cascade.detectMultiScale(roi\_gray2)  
 for (ex, ey, ew, eh) in eyes2:  
 cv2.rectangle(roi\_color2, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)  
 hay\_elementos\_2 = True  
 if (hay\_elementos\_2):  
 contar\_cuadros\_2 += 1  
 if (contar\_cuadros\_2 >= 30):  
 print("salir.....")  
 intentar\_salir()  
 contar\_cuadros\_2 = 0  
 else:  
 contar\_cuadros\_2 = 0  
 cv2.imshow('img2', img2)  
 k = cv2.waitKey(30) & 0xff  
 if k == 27:  
 break  
  
cap.release()  
cap2.release()  
cv2.destroyAllWindows()

paso8: ir a la pagina de git <https://github.com/andacas/PST2_1T2020-SISTEMA_CONTEO_PERSONAS> y descargar el zip del proyecto pulsando el botón de code



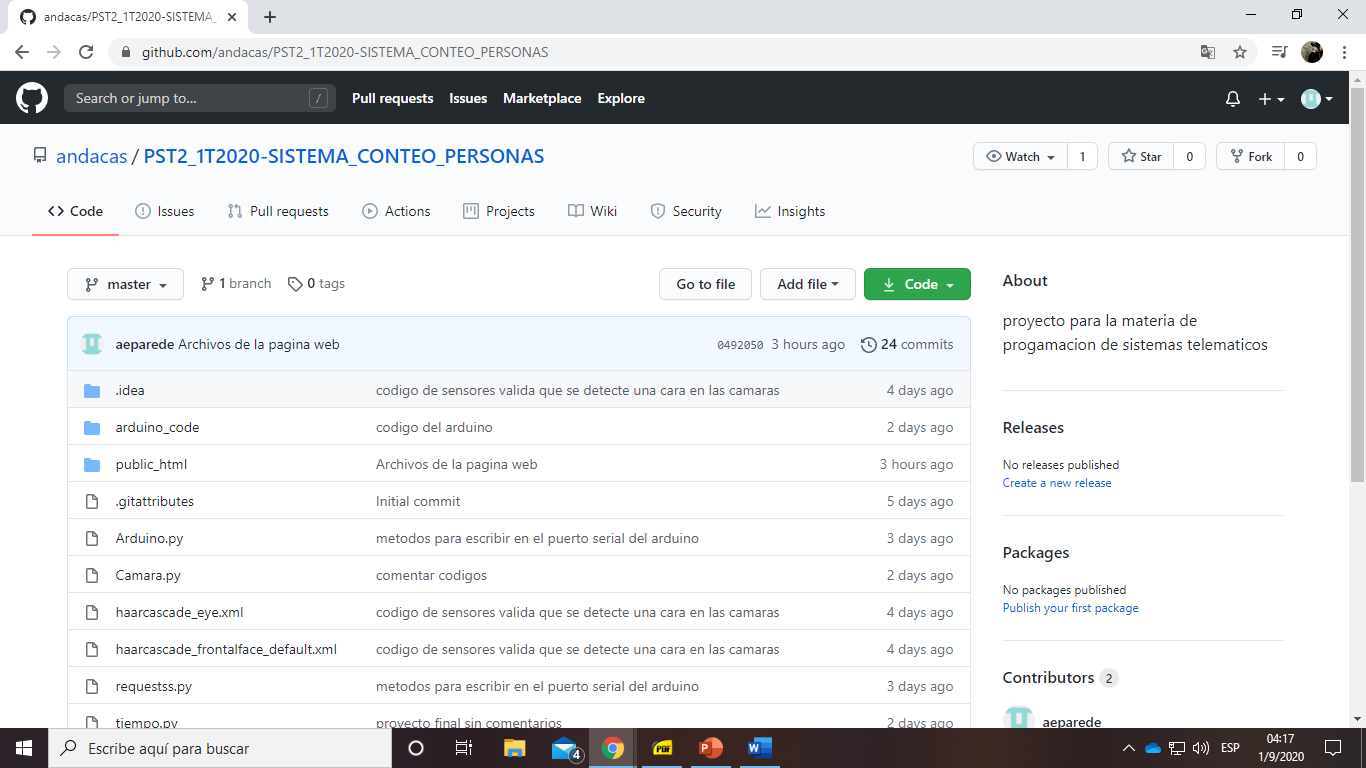
Paso9: mover los archivos .xml del archivo comprimido a la carpeta del proyecto



Paso10: una vez este todo lo demás listo y conectado ejecutar el script pulsando ALT+MAYUS+F10

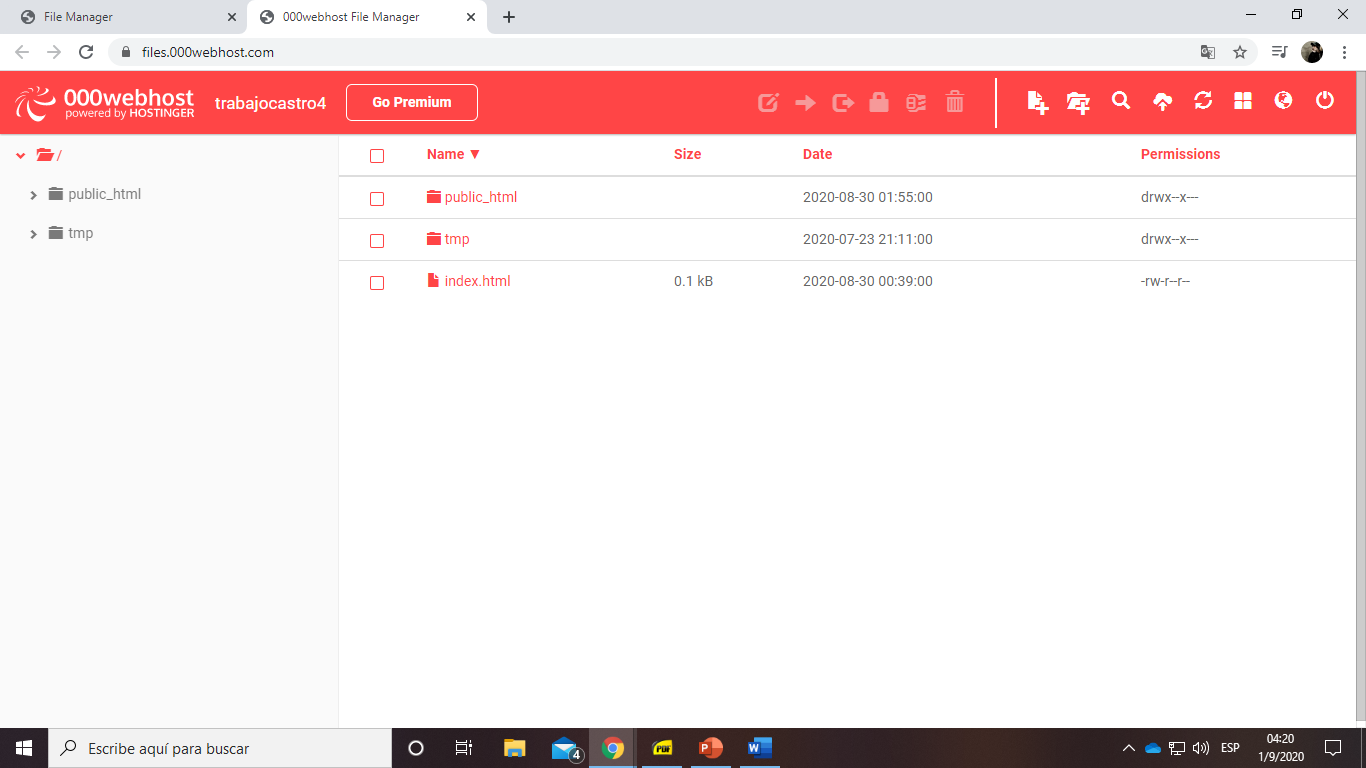
# Mantenimiento de página web

PASO 1: entrar a <https://github.com/andacas/PST2_1T2020-SISTEMA_CONTEO_PERSONAS> y descargar la carpeta llamada “public\_html”

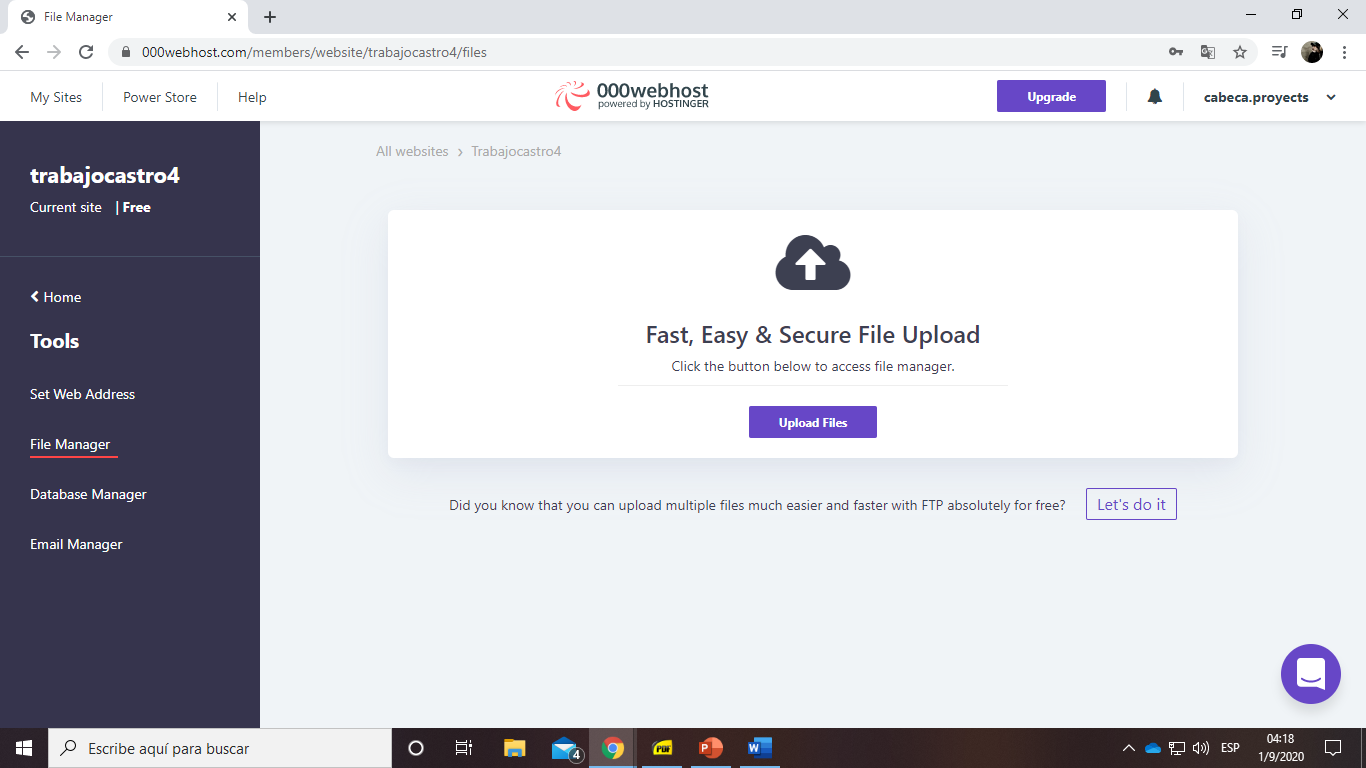


PASO 2: Regresar a la pagina de web host pero ahora entraremos a la seccion de “File Manager”

PASO 3: Darle click a “Upload Files”  
Que nos llevara a la página de administración de la página web  
Aquí reemplazaremos la carpeta “public\_html” con la que descargamos previamente.

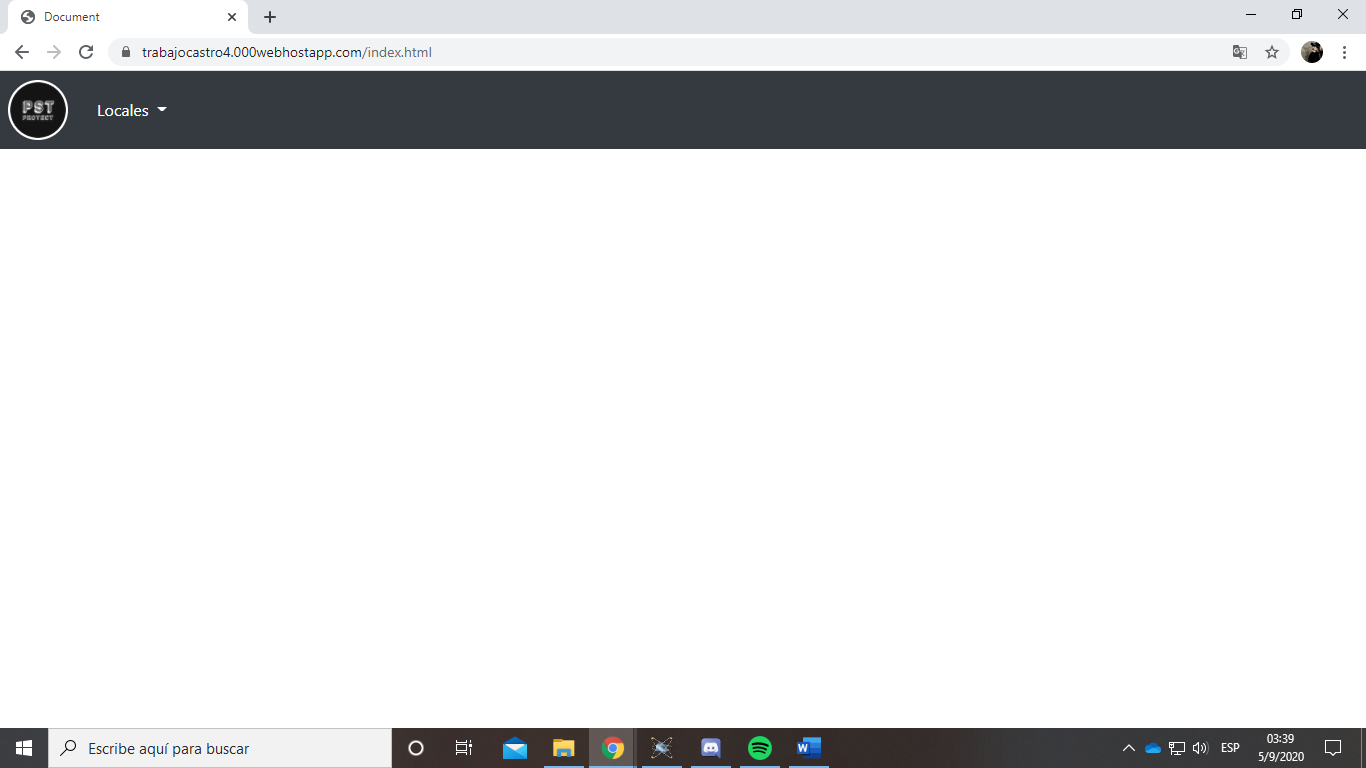


PASO 4: Entrar a la carpeta “public\_html” y editar todos los archivos con extensión .php   
ya que en ellos se deberán poner las credenciales de la pagina web para que se pueda realizar la conexión.

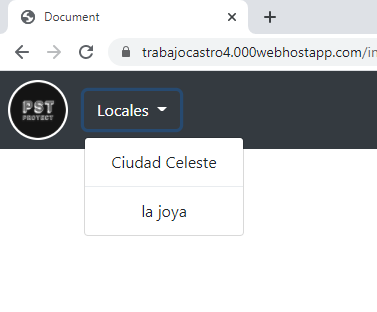


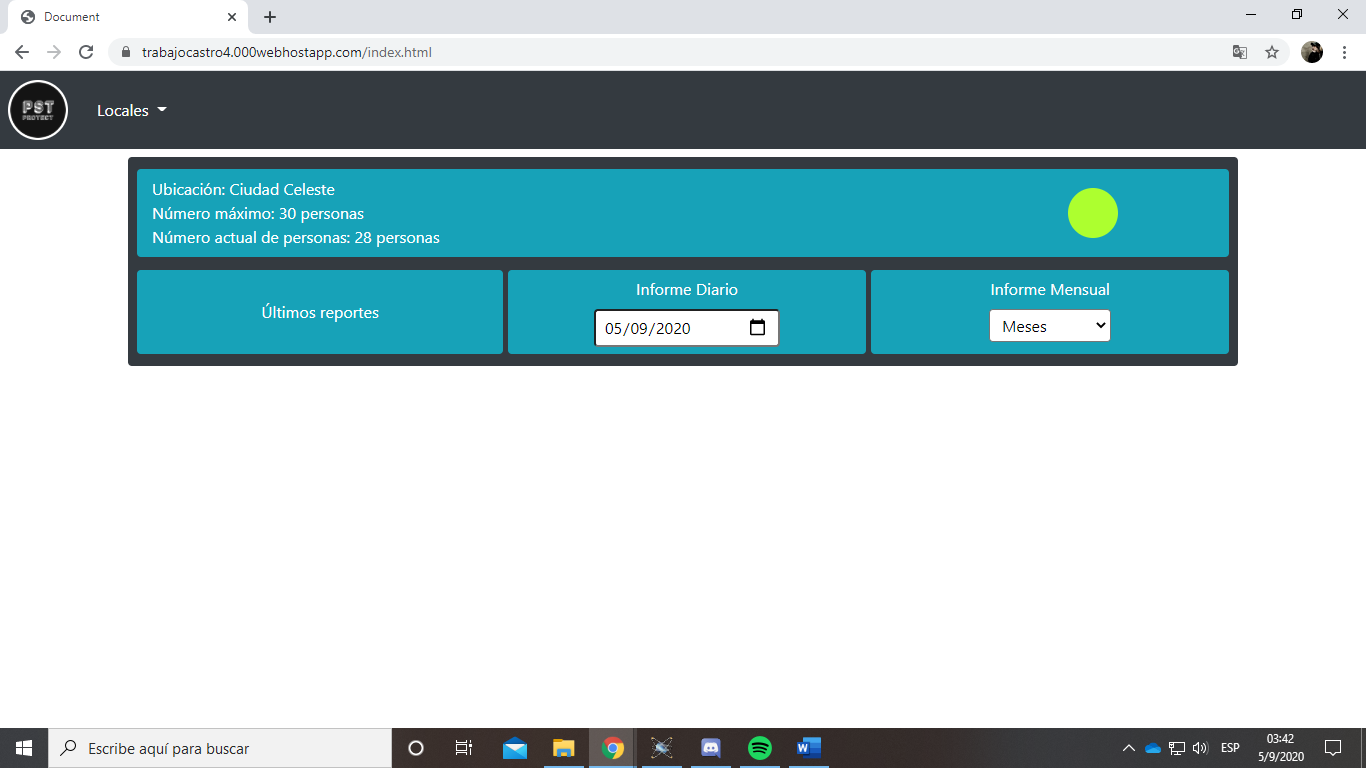
# Guía de uso de la página del sistema

Al entrar en la página se encontrará con la ventana principal



Para ver el registro de los locales le da click a la pestaña que dice “locales” en la cual se desplegara una pestaña con los nombres de los locales asociados al sistema.

  
Al elegir una de las opciones, pasara a la ventana de presentación

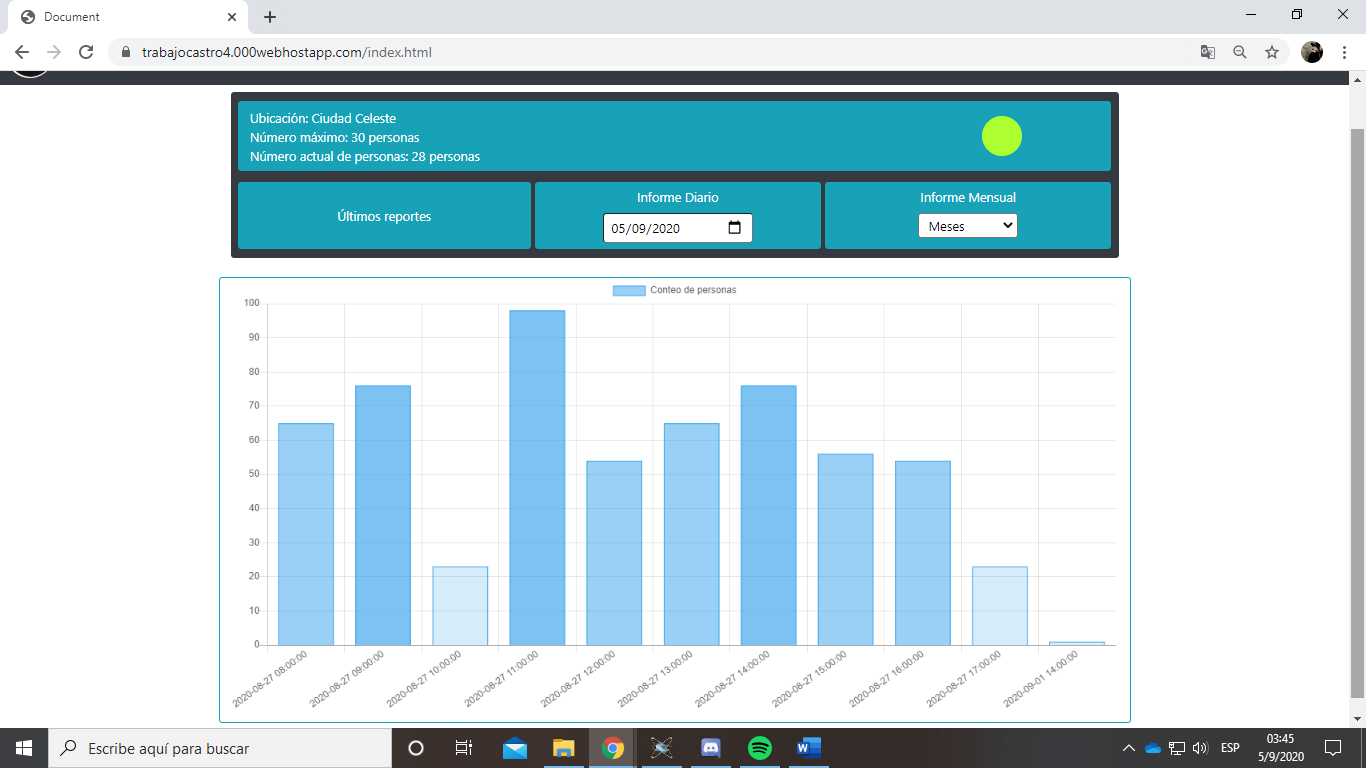


En donde vemos distintos campos, tanto como la “ubicación” del local, el “número máximo de personas” y el “numero actual de personas” estos datos variaran según el local.

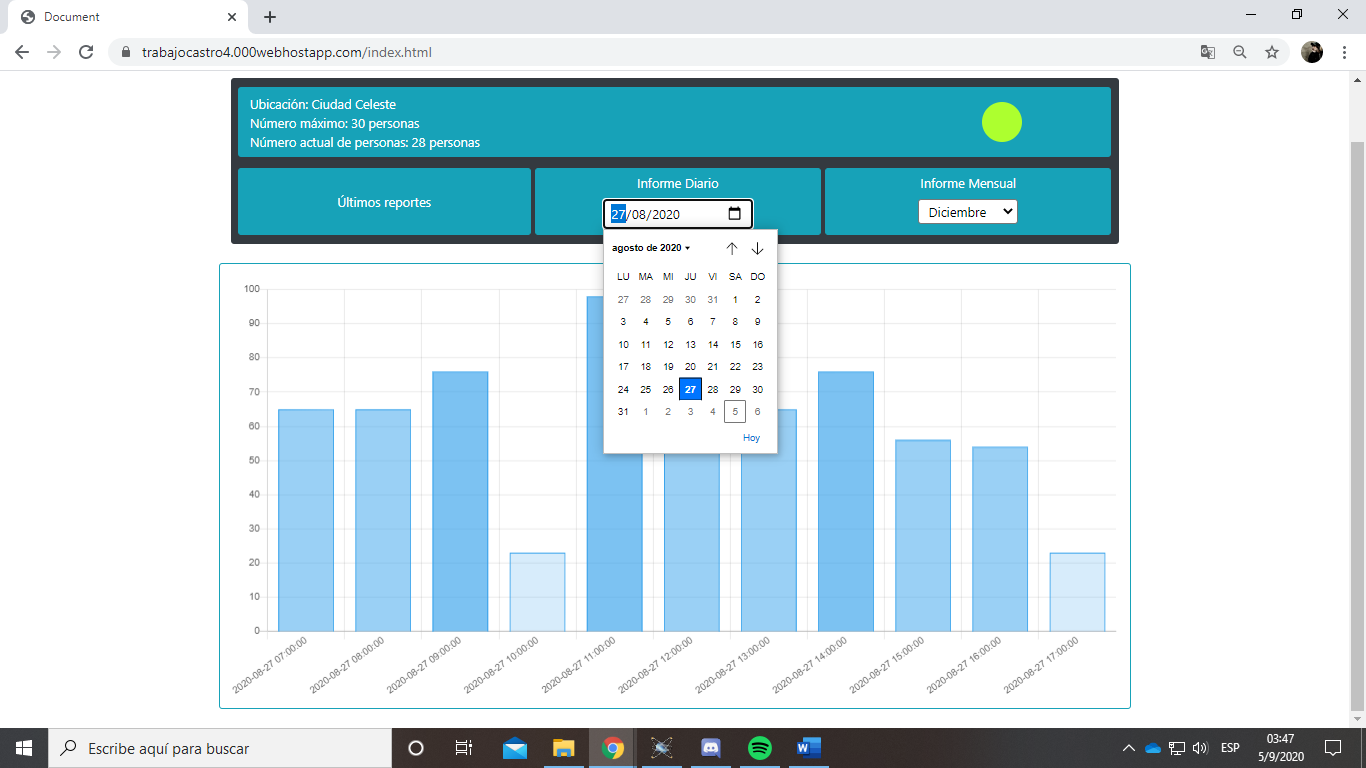
Del lado derecho vemos un circulo verde que señala que se puede acceder al local por que no a sobrepasado el limite de personas dentro.

Debajo de los datos del local encontramos 3 apartados:

-El primero llamado “últimos reportes” muestra en unas graficas cuando fue la ultima entrada de datos hecha para este local.



-En el segundo apartado, “informe diario” encontramos una pestaña que nos despliega un calendario en el cual se puede elegir tanto como mes día y año para ver su registro.



-En el ultimo apartado “informe mensual” encontramos una pestaña desplegable en la cual estan listados todos los meses, al seleccionar uno muestra un informa de todos los días en los cuales se realizo un registro.

