**UNIVERSIDAD IBEROAMERICANA PUEBLA**

*DEPARTAMENTO DE CIENCIAS E INGENIERÍAS*

LICENCIATURA EN INGENIERIA MECATRÓNICA



*“Proyecto final”*

Presenta:

AGUILAR CRISTIANI JOSE PABLO

OROPEZA GONZALEZ ADRIAN

GARCIA RANGEL ORLANDO MISAEL

***Arquitecturas aplicadas***

Profesor: Ing. Rafael Pérez Aguirre

OTOÑO 2019

**Introducción**

La programación hace referencia al efecto de programar, es decir, de organizar una secuencia de pasos ordenados a seguir para hacer cierta cosa.

En el ámbito de la informática, la programación refiere a la acción de crear programas o aplicaciones, a través del desarrollo de un código fuente, el cual se basa en el conjunto de instrucciones que sigue el ordenador para ejecutar un programa.

Estas instrucciones se encuentran escritas en lenguaje de programación que luego son traducidas a un lenguaje de máquina, que puede ser interpretado y ejecutado por el hardware del equipo (parte física del equipo). Dicho código fuente es creado, diseñado, codificado, mantenido y depurado a través de la programación, donde el principal objetivo a lograr es el desarrollo de sistemas que sean eficaces, accesibles y agradables o amigables para el usuario.

Los programas informáticos suelen seguir algoritmos, que son el conjunto de instrucciones que se encuentran organizadas y relacionadas entre sí de cierta manera y que permiten llegar a la solución de un problema, y que a su vez contienen módulos más pequeños que le aportan detalles más finos. Todo esto a escala de gran complejidad forma parte de la programación informática que permite trabajar al software de los equipos de computación y la cual suele ser llevada a cabo por técnicos o ingenieros en sistemas.

En los comienzos del desarrollo de la programación informática, se utilizaban lenguajes máquina muy básicos y limitados como el sistema binario (uso de los números 0 y 1 en distintas combinaciones); más tarde comenzaron a surgir lenguajes que hacían uso de códigos de palabras, y luego, conjuntos de algoritmos mucho más complejos que se denominaron lenguajes de alto nivel.

**Marco teórico**

**Raspberry Pi**

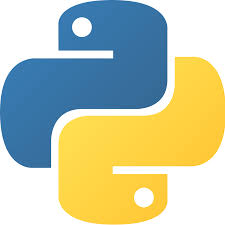
Es un ordenador del tamaño de una tarjeta de crédito. Consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Fue lanzado en 2006 por la Fundación Raspberry Pi con el objeto de estimular la enseñanza de informática en las escuelas de todo el mundo.

El procesador funciona a 700 Mhz. y puede acelerar gráficos 3D por hardware. Más o menos como el ordenador que tenías en 2003, con la salvedad de que puedes ver películas en alta definición.

Se pueden instalar un buen puñado de ellos, la mayoría basados en el kernel de Linux. Algunos de los más conocidos son Android, Firefox OS, Raspbian, OpenWebOS o Unix. También se pueden cargar interfaces gráficas similares a Windows, de modo que la curva de aprendizaje del sistema no es demasiado pronunciada.

Otra de las diferencias con el viejo ordenador es que Raspberry Pi cuenta con una floreciente comunidad de desarrolladores, siempre dispuesta a compartir paso a paso las nuevas características que van encontrándole a la placa. Estos son algunos de los avances más reseñables que se han efectuado hasta la fecha.

**Phyton**

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License,2​ que es compatible con la Licencia pública general de GNU.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos). Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable. Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

**Sensor**

Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas.

Un sensor en la industria es un objeto capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica, etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la magnitud que la condiciona o variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Por ejemplo, el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

**Base de datos**

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Hay programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

**GitHub**

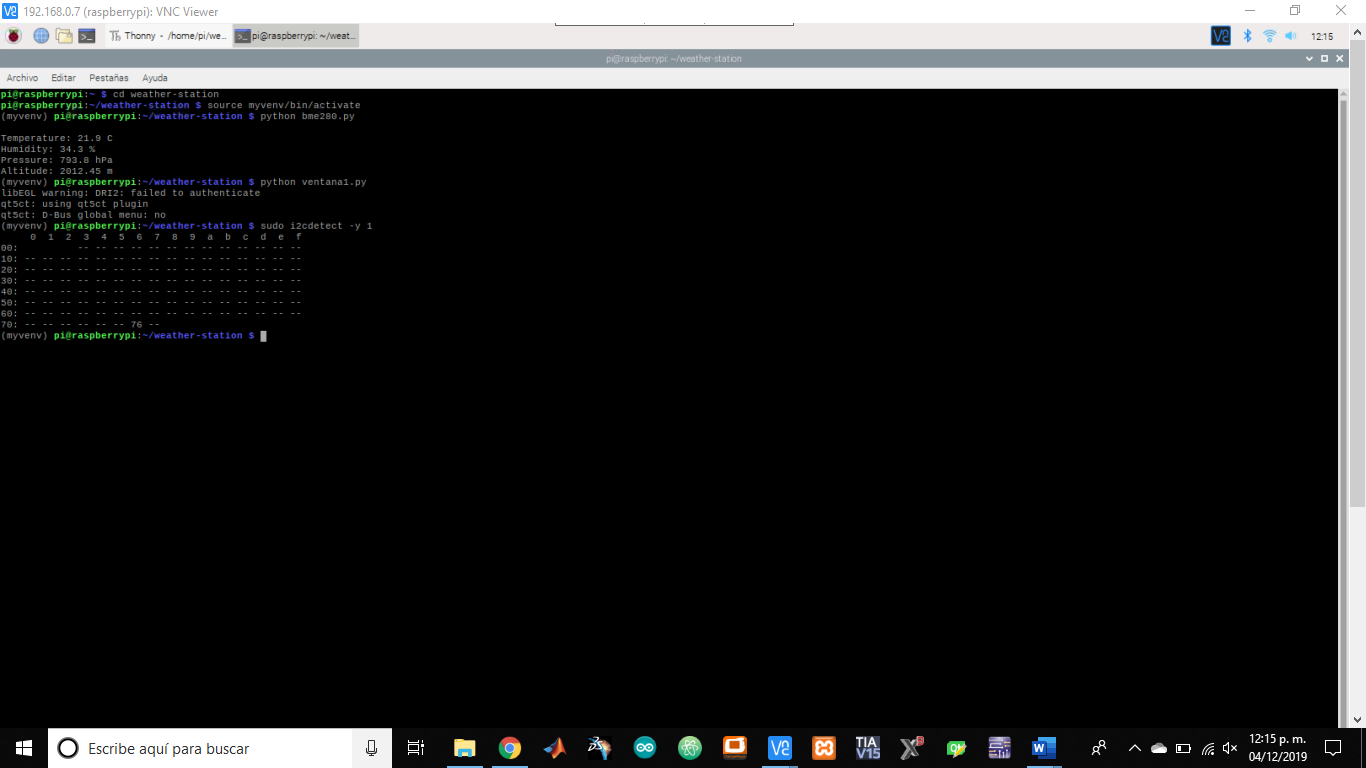
GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls,

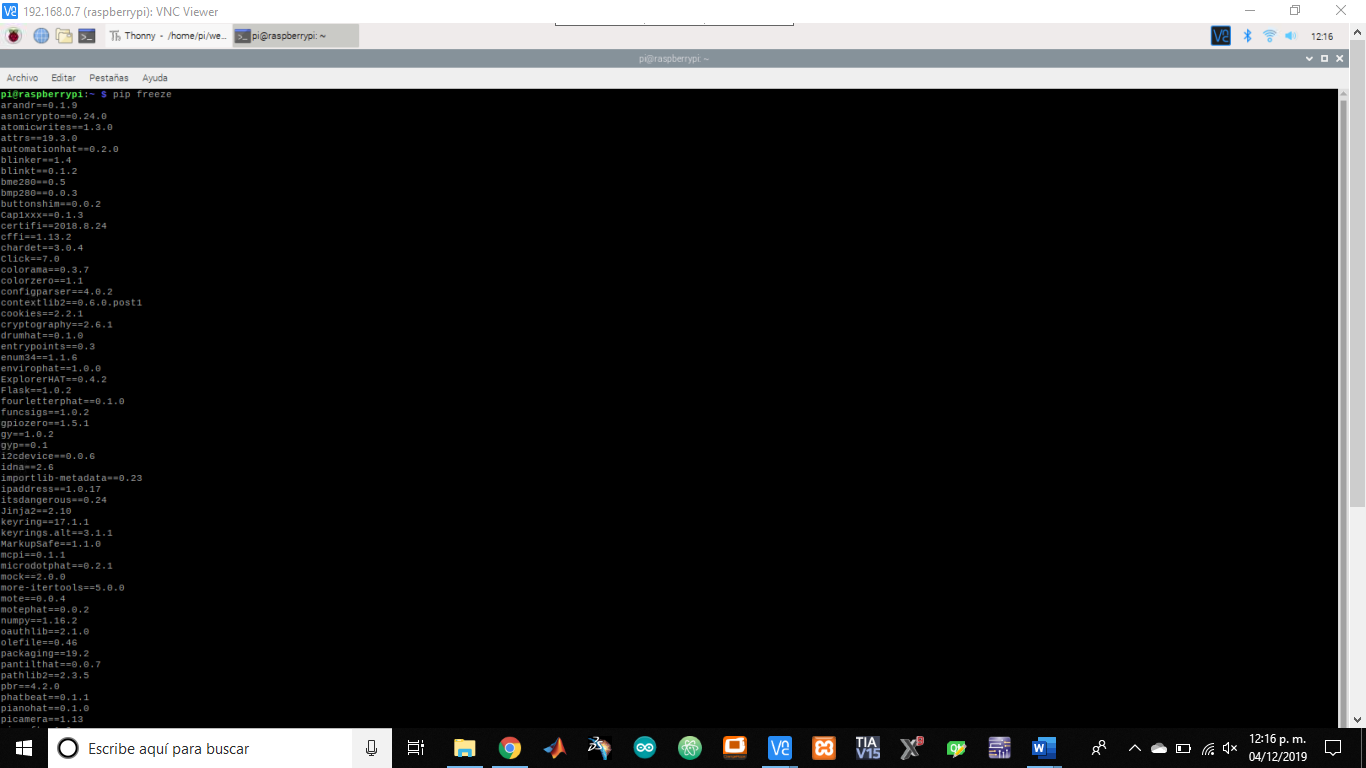
En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo.

**Desarrollo**

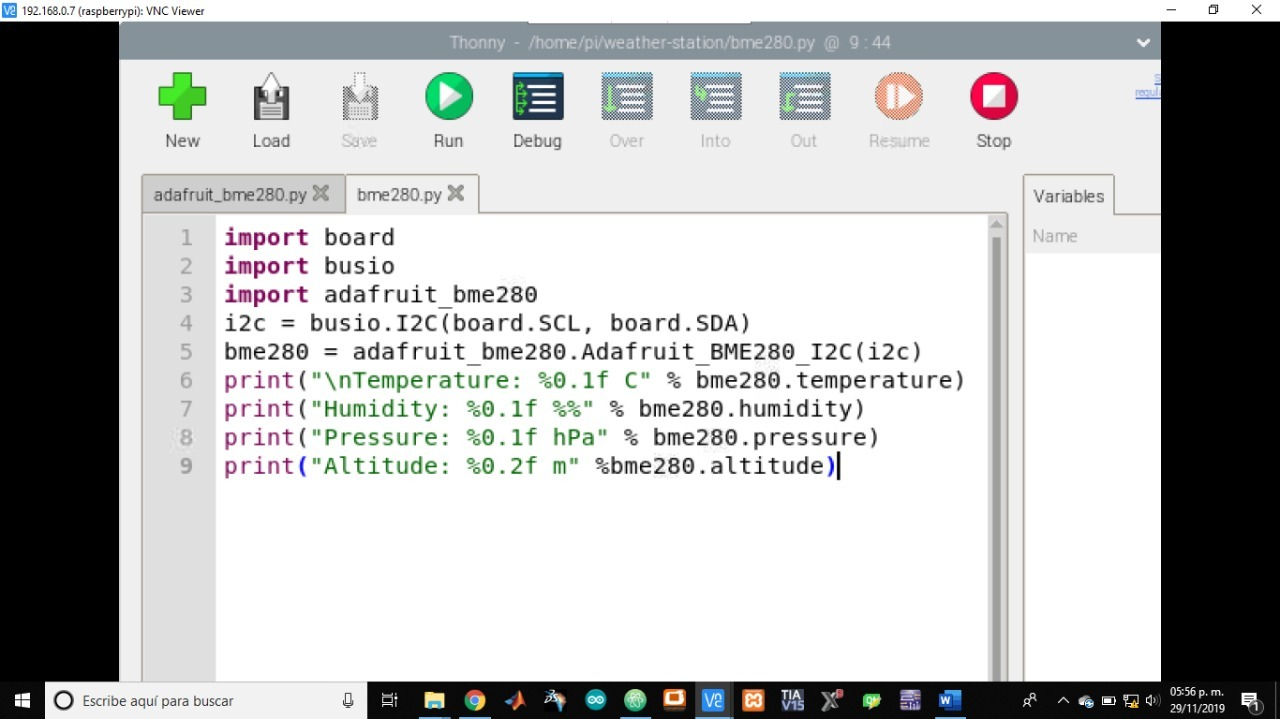
Como primer paso se empieza con el codigo que ejecute la raspberry

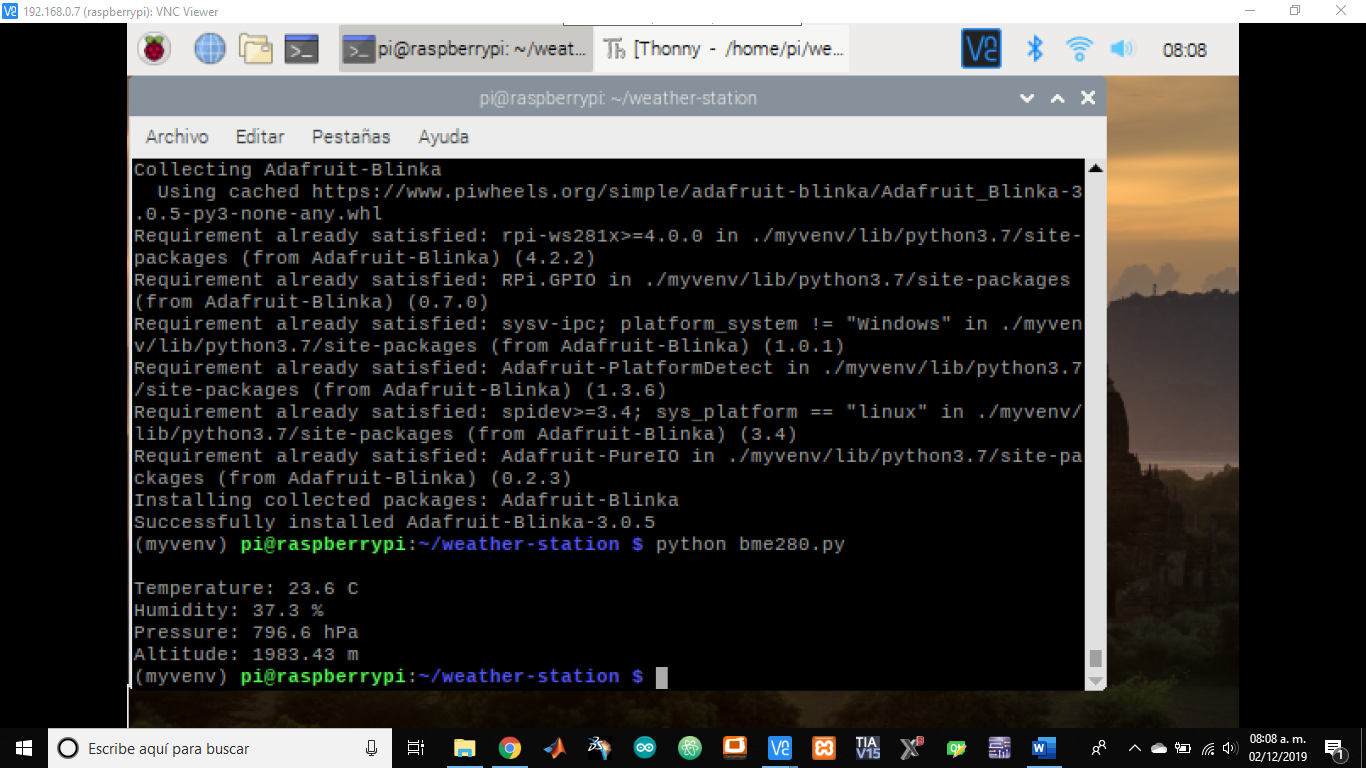


Detectamos la direccion del sensor mediante i2c

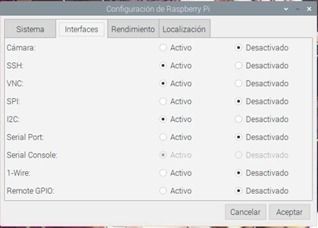


Librerias usadas para el proyecto

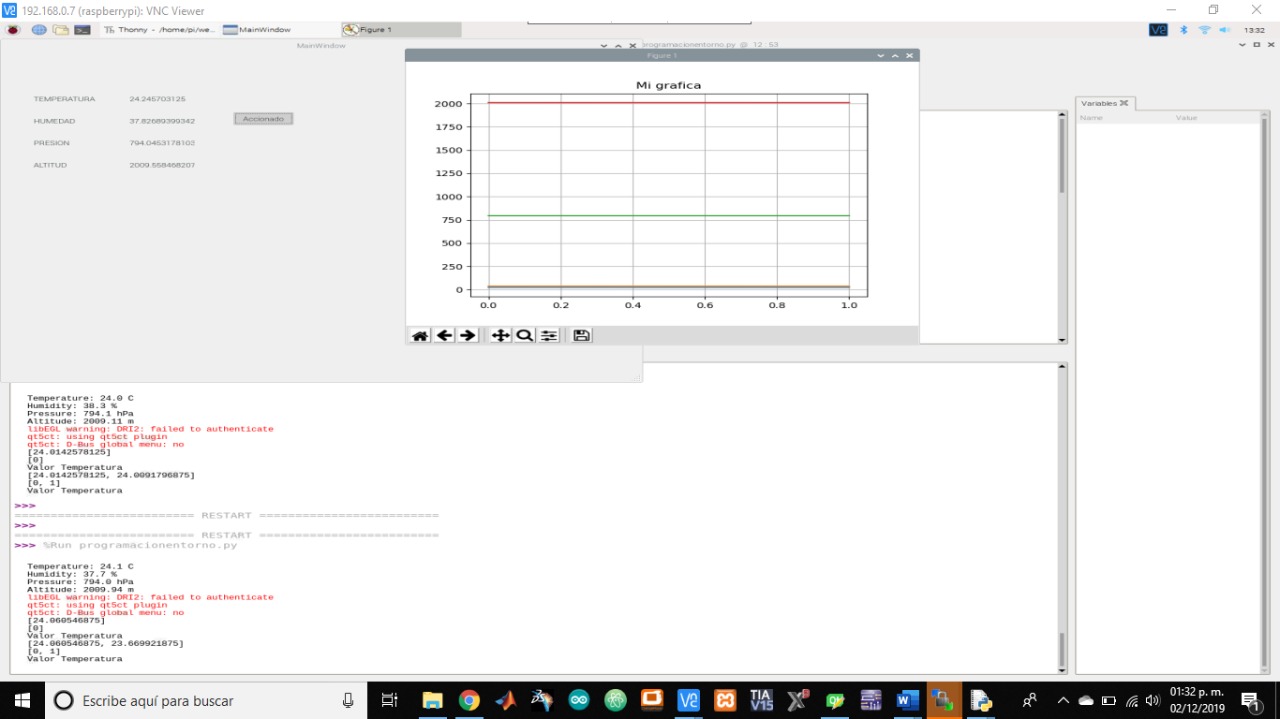




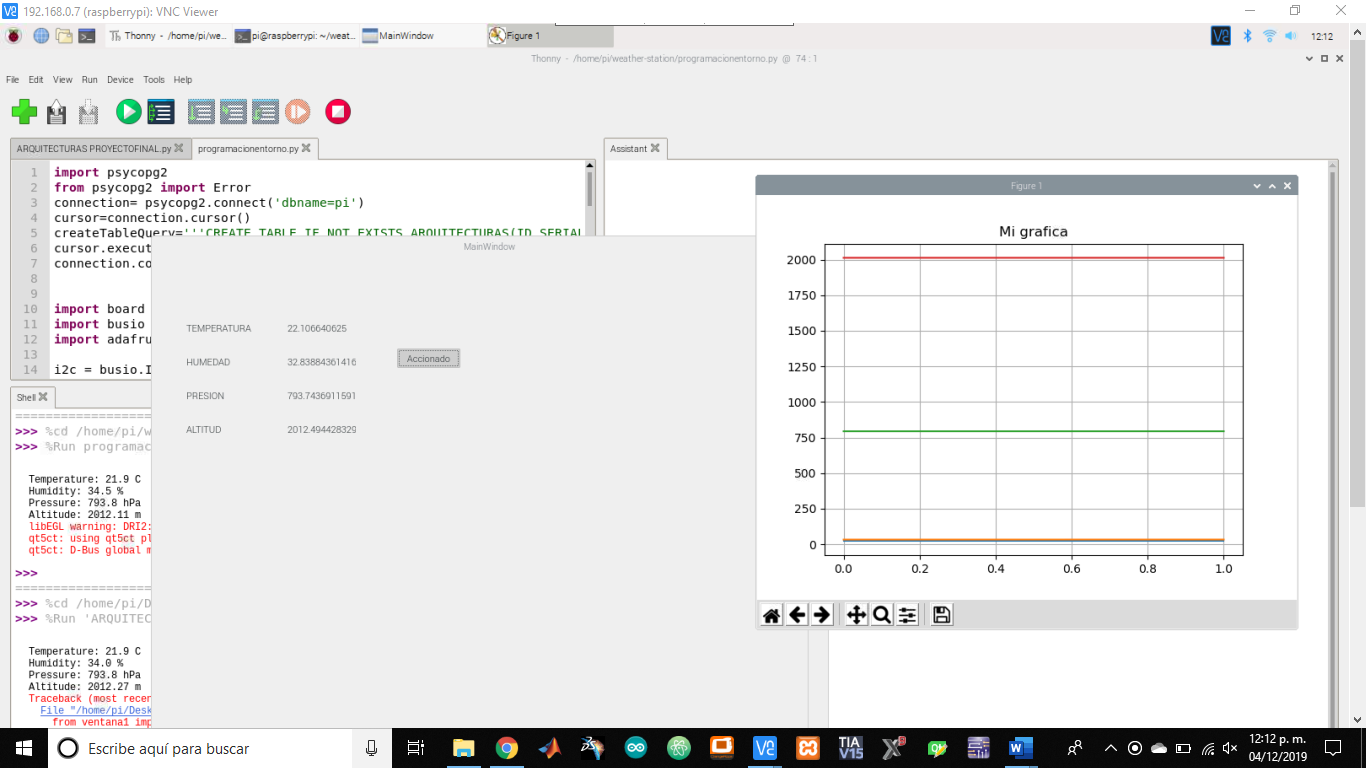
Abrimos la terminal estando en el entorno virtual, escribimos el comando para ver el funcionamiento de nuestro sensor

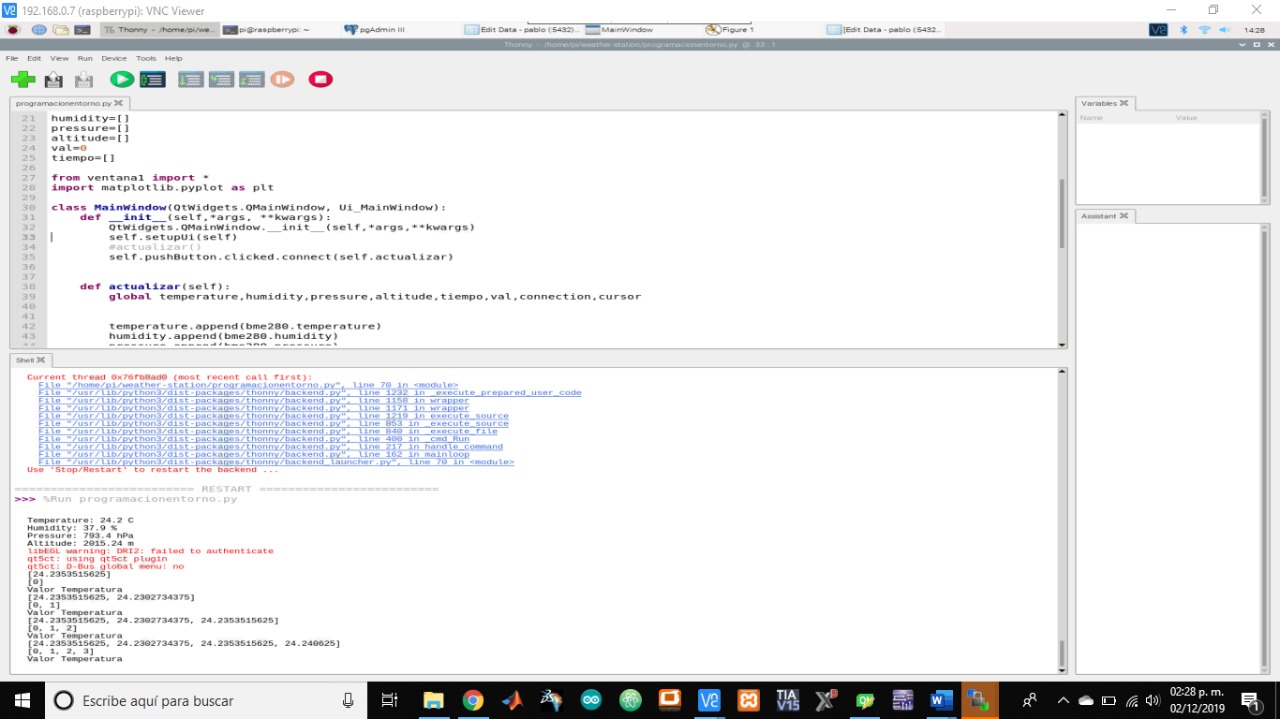


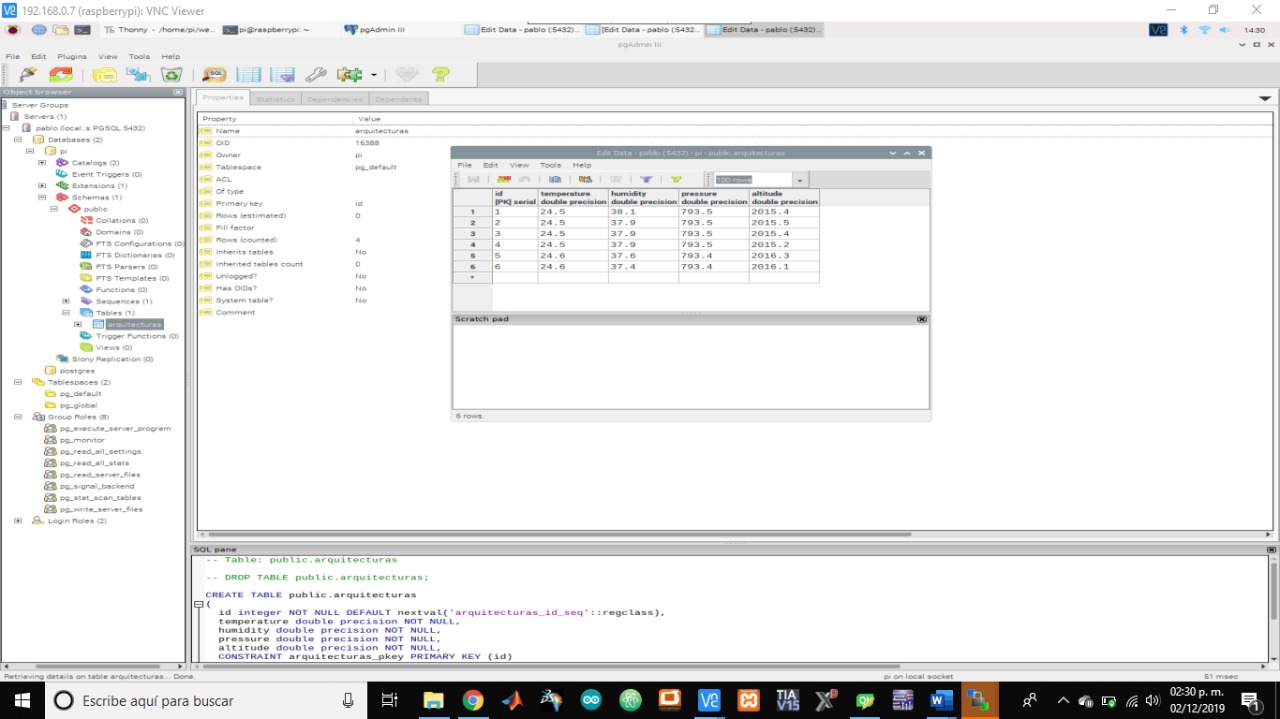
Activacion del i2c y comunicación del SSH



Gráfica el cual nos muestra los valores del sensor.







Base de datos del proyecto usando PGadmin, el cual nos muestra la tabla con los datos del sensor, en este caso Temperatura, Humedad, Presion, Altitud.

**Código usado para el proyecto**

import psycopg2

from psycopg2 import Error

connection= psycopg2.connect(user="pi",password="123456",host="127.0.0.1",port="5432",database="pi")

cursor=connection.cursor()

import board

import busio

import adafruit\_bme280

i2c = busio.I2C(board.SCL, board.SDA)

bme280 = adafruit\_bme280.Adafruit\_BME280\_I2C(i2c)

print("\nTemperature: %0.1f C" % bme280.temperature)

print("Humidity: %0.1f %%" % bme280.humidity)

print("Pressure: %0.1f hPa" % bme280.pressure)

print("Altitude: %0.2f m" %bme280.altitude)

temperature=[]

humidity=[]

pressure=[]

altitude=[]

val=0

tiempo=[]

from ventana1 import \*

import matplotlib.pyplot as plt

class MainWindow(QtWidgets.QMainWindow, Ui\_MainWindow):

def \_\_init\_\_(self,\*args, \*\*kwargs):

QtWidgets.QMainWindow.\_\_init\_\_(self,\*args,\*\*kwargs)

self.setupUi(self)

#actualizar()

self.pushButton.clicked.connect(self.actualizar)

def actualizar(self):

global temperature,humidity,pressure,altitude,tiempo,val

temperature.append(bme280.temperature)

humidity.append(bme280.humidity)

pressure.append(bme280.pressure)

altitude.append(bme280.altitude)

tiempo.append(val)

val+=1

print(temperature)

print (tiempo)

print("Valor Temperatura")

self.label\_8.setText(str(float(bme280.pressure)))

self.label\_3.setText(str(float(bme280.temperature)))

self.label\_5.setText(str(float(bme280.humidity)))

self.label\_7.setText(str(float(bme280.altitude)))

self.pushButton.setText("Accionado")

plt.title('Mi grafica')

plt.plot(tiempo,temperature,tiempo,humidity,tiempo,pressure,tiempo,altitude)

plt.grid(True)

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

app = QtWidgets.QApplication([])

window = MainWindow()

window.show()

app.exec\_()

**GIT HUB**

<https://github.com/PabloCristiani/proyectoarquitecturas.git>

liga para ingresar el repositorio de Git Hub

**Conclusión**

La programacion es de suma importancia, ya que con esta podemos tomar ventaja de la internet y sus diferentes campos, asi como de mismo modo, la repesentacion de valores a largas distancias.

Con referencia al proyecto, es grato ver como el envio de datos esta mas cerca cada vez, lo importante es la potencia de señal que uno tenga, siendo sinceros, el entorno de python se hizo un poco complicado ya que habia que buscar las librerias de cada cosa, asi de mismo modo la interaccion no es la mas grata.

**Referencias**

¿Qué es Raspberry PI y para que sirve?. Consultado el 24 de septiembre de 2019. Disponible en: (https://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html)

Introducción - Conociendo GitHub. (2012). Consultado el día 26 de noviembre de 2019. Disponible en: (<https://conociendogithub.readthedocs.io/en/latest/data/introduccion/>)

Base de datos – Concepto, Tipo y Elementos. Consultado el día 26 de noviembre de 2019. Disponible en: (<https://concepto.de/base-de-datos/#ixzz66PI2Hflx>)