UNIVERSIDAD AUTONOMA DE TAMAULIPAS

FACULTAD DE INGENIERÍA

“ARTURO NARRO SILLER”

INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROGRAMACION DE INTERFACES Y PUERTOS

6 I

Portafolio de Evidencias

Equipo:

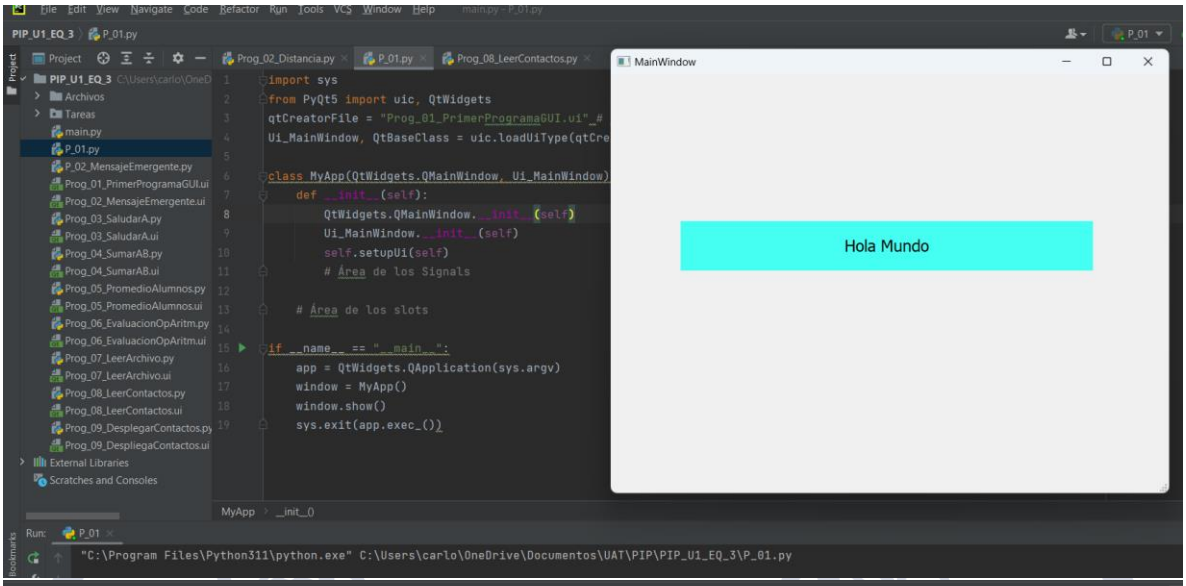
Integrantes:

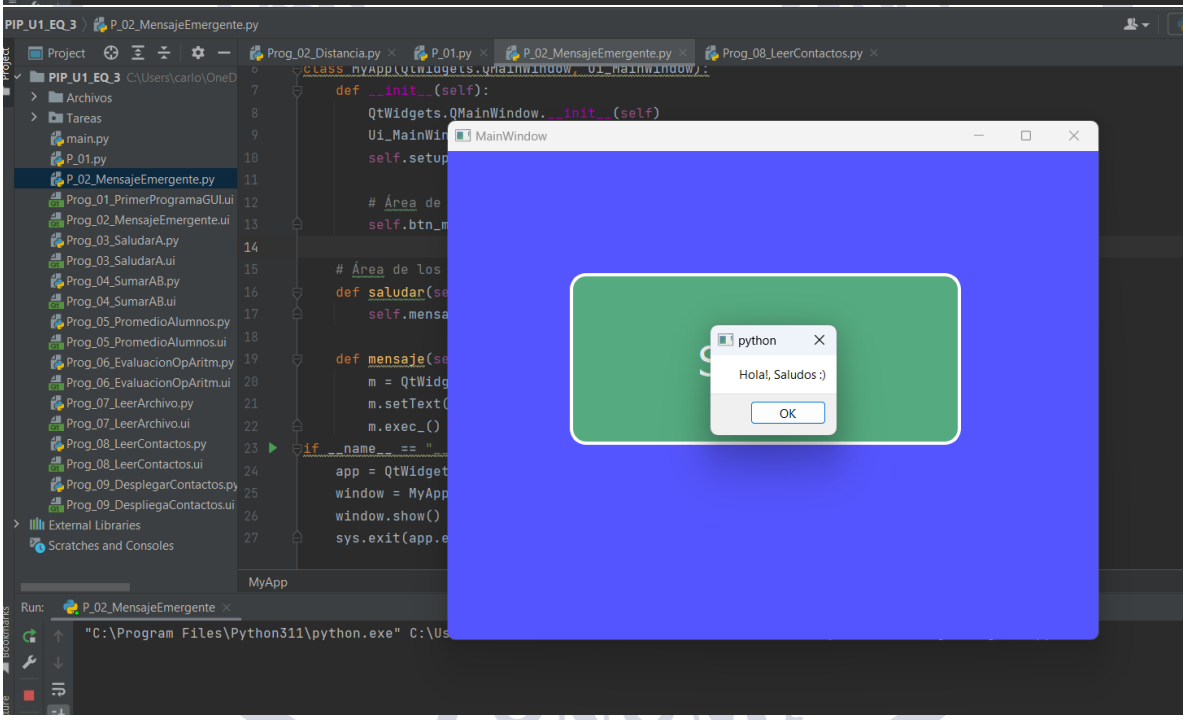
- Gonzalez Santiago Jose Alonso
- Moreno Wilches Fernando
- Verástegui Cruz Carlos

Fecha de entrega: 15 de febrero del 2023

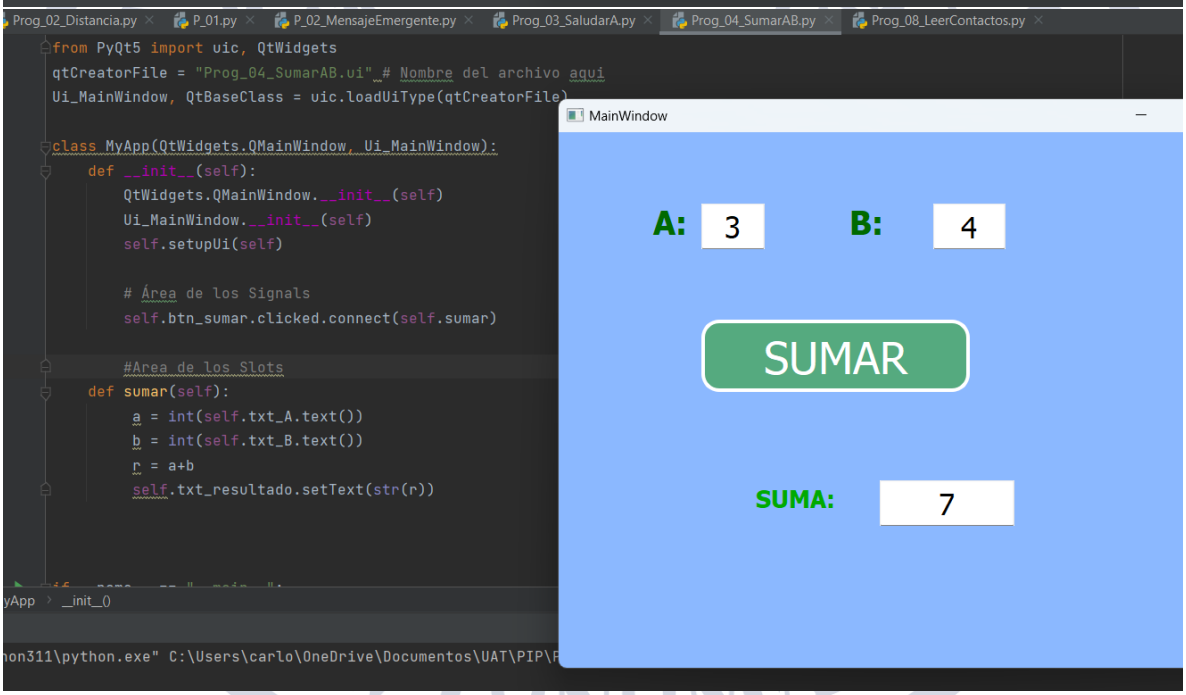
Contenido

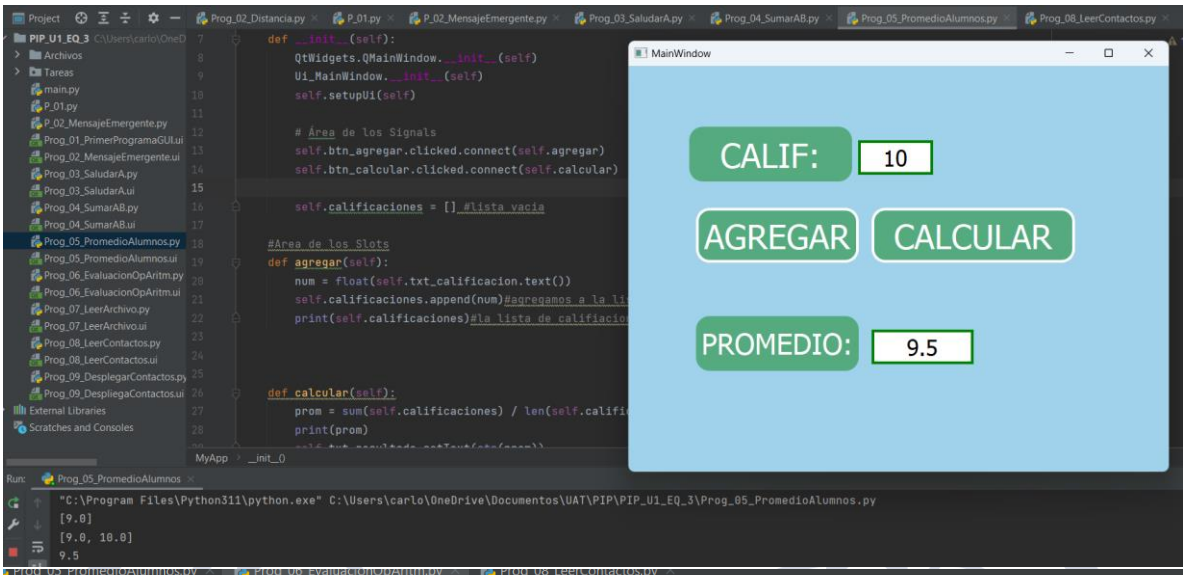
Actividades Realizadas en clase:



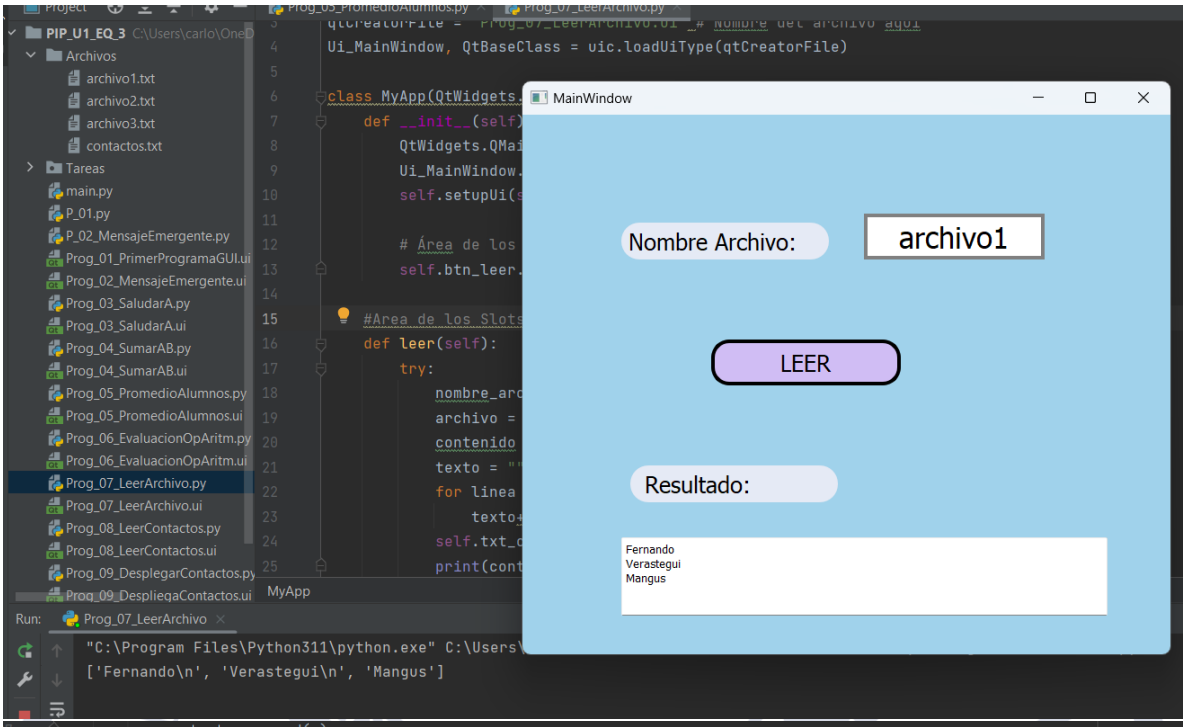


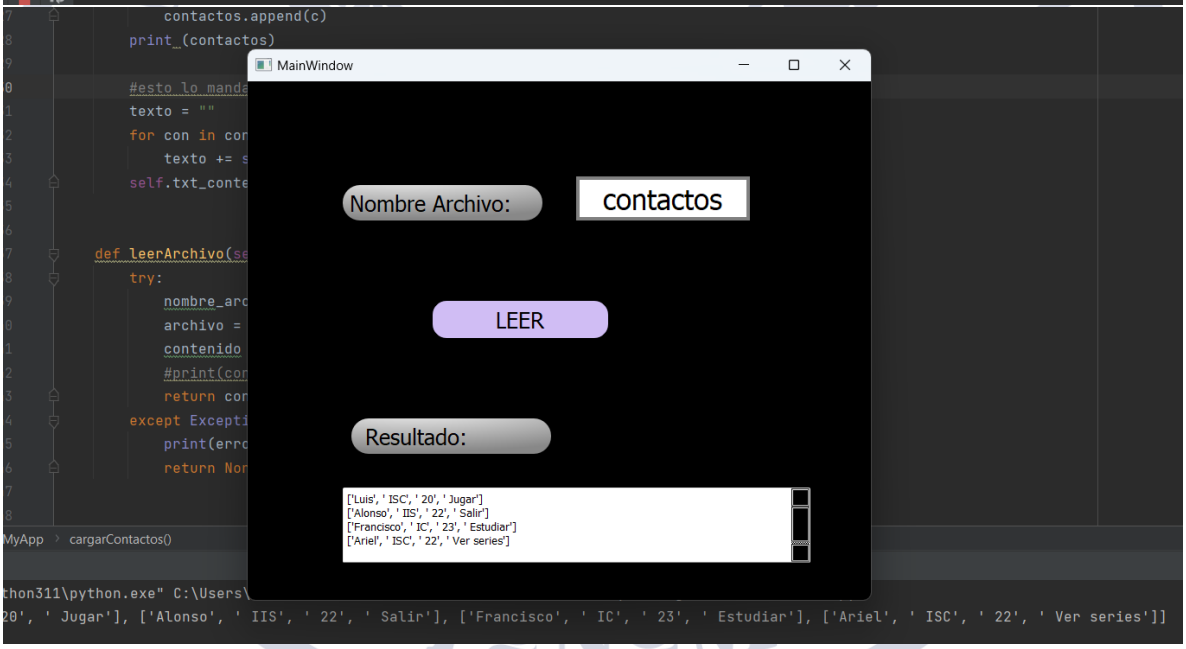

Prog_02_Distancia.py
P_01.py
P_02_MensajeEmergente.py
Prog_03_SaludarA.py
Prog_08_LeerContactos.py
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
def __init__(self):
QtWidgets.QMainWindow.__init__(self)
Ui_MainWindow.__init__(self)
self.setupUi(self)
# Área de los Signals
self.btn_saludar.clicked.connect(self.saluda
# Área de los slots
def saludar(self):
n = self.txt_nombre.text()
self.mensaje("Hola! "+ n +", Saludos :)")
def mensaje(self, msj):
m = QtWidgets.QMessageBox()
m.setText(msj)
m.exec_()
if __name__ == "__main__":
MyApp › __init__()
MainWindow
Nombre:
car
python
Hola! car, Saludos :)
OK
Saludar


Prog_02_Distancia.py
P_01.py
P_02_MensajeEmergente.py
Prog_03_SaludarA.py
Prog_04_SumarAB.py
Prog_08_LeerContactos.py
from PyQt5 import uic, QtWidgets
qtCreatorFile = "Prog_04_SumarAB.ui" # Nombre del archivo aquí
Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
def __init__(self):
QtWidgets.QMainWindow.__init__(self)
Ui_MainWindow.__init__(self)
self.setupUi(self)
# Área de los Signals
self.btn_sumar.clicked.connect(self.sumar)
#Area de los Slots
def sumar(self):
a = int(self.txt_A.text())
b = int(self.txt_B.text())
r = a+b
self.txt_resultado.setText(str(r))
yApp › __init__()
non311\python.exe" C:\Users\carlo\OneDrive\Documentos\UAT\PIP\P
MainWindow
A:
3
B:
4
SUMAR
SUMA:
7


MainWindow
CALIF:
10
AGREGAR
CALCULAR
PROMEDIO:
9.5
"C:\Program Files\Python311\python.exe" C:\Users\carlo\OneDrive\Documentos\UAT\PIP\PIP_U1_EQ_3\Prog_05_PromedioAlumnos.py
[9.0]
[9.0, 10.0]
9.5


MainWindow
Expresión Aritm:
3+6-2
CALCULAR
Resultado:
7


MainWindow
Nombre Archivo:
archivo1
LEER
Resultado:
Fernando
Verastegui
Mangus
Run: Prog_07_LeerArchivo
"C:\Program Files\Python311\python.exe" C:\Users\
['Fernando\n', 'Verastegui\n', 'Mangus']


MainWindow
Nombre Archivo:
contactos
LEER
Resultado:
['Luis', ' ISC', ' 20', ' Jugar']
['Alonso', ' IIS', ' 22', ' Salir']
['Francisco', ' IC', ' 23', ' Estudiar']
['Ariel', ' ISC', ' 22', ' Ver series']
MyApp > cargarContactos()


self.btn_adelante.clicked.connect(self.adelante)
#Area de los Slots
def atras(self):
def adelante(self)
def cargarContacto
info_contactos
self.contactos
for linea in
MyApp
atras()
MainWindow
Nombre Archivo:
contactos
LEER
Datos:
Adelante
Nombre:
Francisco
Atras
Carrera:
IC
Edad:
23

Tareas e Investigaciones.

1.- ¿Qué es comunicación y cuáles son sus componentes?

La comunicación es un proceso clave para la vida humana y para la sociedad en su conjunto. Es un medio a través del cual se comparte información, se transmiten ideas, se establecen relaciones y se crean vínculos sociales y emocionales. La comunicación es esencial para la resolución de conflictos, la toma de decisiones, la solución de problemas y el crecimiento personal y profesional.

Puede ser verbal o no verbal y tomar lugar tanto en persona como a través de medios tecnológicos. Además, puede ser formal o informal y tener distintos objetivos, como informar, persuadir, motivar o entretener. La comunicación es un proceso dinámico y complejo que involucra la interacción de muchos factores, como la cultura, la edad, el género, la educación y las emociones. Por lo tanto, es importante ser consciente de estos factores y trabajar en mejorar las habilidades de comunicación y comprender las perspectivas y necesidades de las personas con las que se comunica.

Los componentes de la comunicación incluyen:

El emisor, que es la persona o entidad que origina el mensaje y lo envía.

El mensaje, que es la información que se desea transmitir.

El canal, que es el medio por el cual se transmite el mensaje.

El receptor, que es la persona o entidad que recibe el mensaje.

La retroalimentación, que es la respuesta o reacción del receptor al mensaje recibido. La retroalimentación permite al emisor conocer si el mensaje ha sido comprendido y recibido de la manera deseada.

2.- ¿En qué consiste la interacción Hombre-Máquina?

La interacción hombre-máquina se refiere a la forma en que los seres humanos interactúan y trabajan con las máquinas. Incluye la comunicación y la colaboración entre los usuarios y las tecnologías, como computadoras, robots, sistemas de control, entre otros.

Esto se logra a través del uso de interfaces que permiten a los usuarios enviar instrucciones a las máquinas y recibir información de ellas. La interacción hombre-máquina es crucial en muchas áreas de la vida moderna, desde la automatización industrial hasta el uso diario de dispositivos electrónicos.

3.- ¿En qué consiste la interacción Máquina-Máquina?

El término "interacción máquina-máquina" se refiere a cómo las máquinas interactúan entre sí sin intervención humana directa. Esto se logra mediante el uso de tecnologías como Internet de las cosas (IoT), que permite que las máquinas intercambien datos y tomen decisiones automáticamente. La interacción máquina-máquina se utiliza en aplicaciones como la automatización de procesos industriales, la gestión de redes, la supervisión y el control de sistemas y el análisis de datos. La capacidad de las máquinas para interactuar entre sí de forma autónoma puede aumentar la eficacia y la eficiencia de muchos sistemas y procesos.

4.- Arquitectura de Software y Hardware: Centralizada

El término "arquitectura centralizada de software y hardware" se refiere a un sistema en el que un solo dispositivo o servidor sirve como punto principal de control y administración de recursos para una red. Todas las tareas y procesos se llevan a cabo en el servidor central de esta arquitectura, mientras que los dispositivos periféricos se conectan y acceden a los recursos a través de él.

Los beneficios de la arquitectura centralizada incluyen una administración y un control más sencillos, así como una mayor seguridad de los datos debido al almacenamiento de toda la información en un solo lugar. Sin embargo, también tiene inconvenientes, como un mayor riesgo de fallas y una mayor sobrecarga del servidor, lo que puede tener un impacto en el rendimiento de la red y la disponibilidad de recursos.

En conclusión, la arquitectura centralizada puede ser útil en entornos de red de pequeña escala y en situaciones que requieren un mayor control y seguridad, pero puede no ser adecuada en entornos de red más grandes y complejos.

5.- Arquitectura de Software y Hardware: Distribuida

La arquitectura distribuida de software y hardware se refiere a un método de diseño de sistemas de información en el que los recursos se distribuyen entre varios componentes o nodos en lugar de estar centralizados en una ubicación. Cada componente de una arquitectura distribuida tiene la capacidad de funcionar de forma independiente o en conjunto con otros componentes para llevar a cabo una tarea específica. Estos componentes se pueden ubicar en varias ubicaciones geográficas y conectarse a través de una red.

En comparación con la arquitectura ubicada centralmente, la arquitectura distribuida tiene como objetivo lograr una mayor escalabilidad, disponibilidad, confiabilidad y seguridad.

Permite la distribución de la carga de trabajo entre los componentes y el uso eficiente de los recursos. También permite una mayor flexibilidad en la toma de decisiones y la resolución de problemas porque los componentes pueden operar de forma independiente entre sí y colaborar para lograr un objetivo compartido. Sin embargo, debido a que administrar la comunicación y la

coordinación entre componentes distribuidos es tan complejo, también puede ser más difícil de desarrollar y mantener.

6.- Arquitectura de Software y Hardware: Hibrida

La arquitectura híbrida es un enfoque que combina distintos componentes de hardware y software para crear un sistema más completo y equilibrado. Este enfoque busca obtener lo mejor de ambos mundos, es decir, la potencia de procesamiento del hardware y la flexibilidad y escalabilidad del software.

Por ejemplo, pueden utilizarse componentes de hardware especializados, como GPU o FPGA, para realizar tareas intensivas de procesamiento de datos, mientras que el software se utiliza para coordinar y distribuir estas tareas a través de una red. Además, el software también puede proporcionar funcionalidades adicionales, como la gestión de datos, la toma de decisiones y la integración con otras aplicaciones.

En resumen, la arquitectura híbrida combina componentes de hardware y software para ofrecer un sistema más completo, escalable y flexible. Esto permite a los usuarios obtener el mejor rendimiento y rentabilidad en comparación con los sistemas basados únicamente en hardware o software.

7.- Normativas vigentes para la implementación de sistemas en casas, edificios o ciudades inteligentes

ISO 37120: Esta norma establece estándares para la medición de los servicios urbanos y el desarrollo sostenible de las ciudades.

EN 15232: Este estándar europeo especifica los requisitos para la integración de sistemas en edificios.

IEEE 2030.5: Este estándar establece los requisitos técnicos para la integración de sistemas de energía y gestión de la demanda en edificios y ciudades inteligentes.

BACnet: Este estándar de protocolo de comunicación se utiliza para la integración de sistemas de control de edificios, como sistemas de climatización, iluminación y seguridad.

KNX: Este estándar de protocolo de comunicación se utiliza para la integración de sistemas de automatización de edificios, como sistemas de climatización, iluminación y seguridad.

8.- ¿Qué es un Sistema Electrónico y cuáles son sus componentes?

Un sistema electrónico es un conjunto de componentes electrónicos interconectados que trabajan juntos para realizar una tarea específica. Estos componentes pueden incluir circuitos integrados, dispositivos de entrada y salida, dispositivos de almacenamiento de datos, dispositivos de transmisión de señales y dispositivos de energía.

Algunos de los componentes comunes de un sistema electrónico incluyen:

Microprocesadores o microcontroladores: Estos son los componentes centrales de un sistema electrónico y se encargan de ejecutar las tareas y las instrucciones de un programa.

Circuitos integrados: Estos son componentes electrónicos integrados en un chip que realizan funciones específicas, como amplificación, filtrado y conmutación.

Dispositivos de entrada: Estos son componentes que reciben información del exterior, como sensores, teclados y dispositivos de seguimiento.

Dispositivos de salida: Estos son componentes que transmiten información hacia el exterior, como pantallas, altavoces y dispositivos de impresión.

Dispositivos de almacenamiento de datos: Estos son componentes que almacenan información, como discos duros, memoria RAM y dispositivos de memoria flash.

Dispositivos de transmisión de señales: Estos son componentes que permiten la transmisión de señales y datos a través de un sistema, como módems y dispositivos de red.

Dispositivos de energía: Estos son componentes que proporcionan energía a un sistema electrónico, como baterías y adaptadores de corriente.

9.- ¿Qué es una Interfaz de Software y cuáles son sus tipos?

Una interfaz de software es un medio a través del cual un usuario interactúa con un programa informático. Es la parte visible de un programa que permite a los usuarios realizar tareas, ingresar datos y recibir resultados.

Existen dos tipos principales de interfaces de software:

Interfaces de línea de comandos: Son interfaces que requieren que el usuario ingrese comandos mediante el teclado. Estas interfaces son más adecuadas para usuarios experimentados y no suelen ser amigables para usuarios no técnicos.

Interfaces gráficas de usuario (GUI): Son interfaces que utilizan elementos gráficos, como botones, menús y ventanas, para interactuar con un programa. Estas interfaces son más intuitivas y fáciles de usar que las interfaces de línea de comandos.

Además de estos dos tipos principales, también existen otras interfaces de software especializadas, como interfaces basadas en web, interfaces de programación de aplicaciones (API) y interfaces de programación de aplicaciones de usuario (UAPI). Cada tipo de interfaz de software tiene sus propias características y se utiliza para diferentes aplicaciones y situaciones.

10.- ¿Qué es una Interfaz de Hardware y cuáles son sus tipos?

Una interfaz de hardware es un conjunto de componentes y conectores que permiten la conexión y la comunicación entre diferentes dispositivos y componentes electrónicos. Las interfaces de hardware facilitan la transmisión de datos, energía y señales entre los diferentes componentes de un sistema electrónico.

Algunos de los tipos de interfaces de hardware incluyen:

Interfaz de bus: Es un canal de comunicación que permite la transmisión de datos entre diferentes componentes de un sistema. Ejemplos incluyen el bus PCI, el bus USB y el bus SCSI.

Interfaz de serie: Es un canal de comunicación que transmite datos un bit a la vez. Ejemplos incluyen el puerto RS-232 y el puerto RS-485.

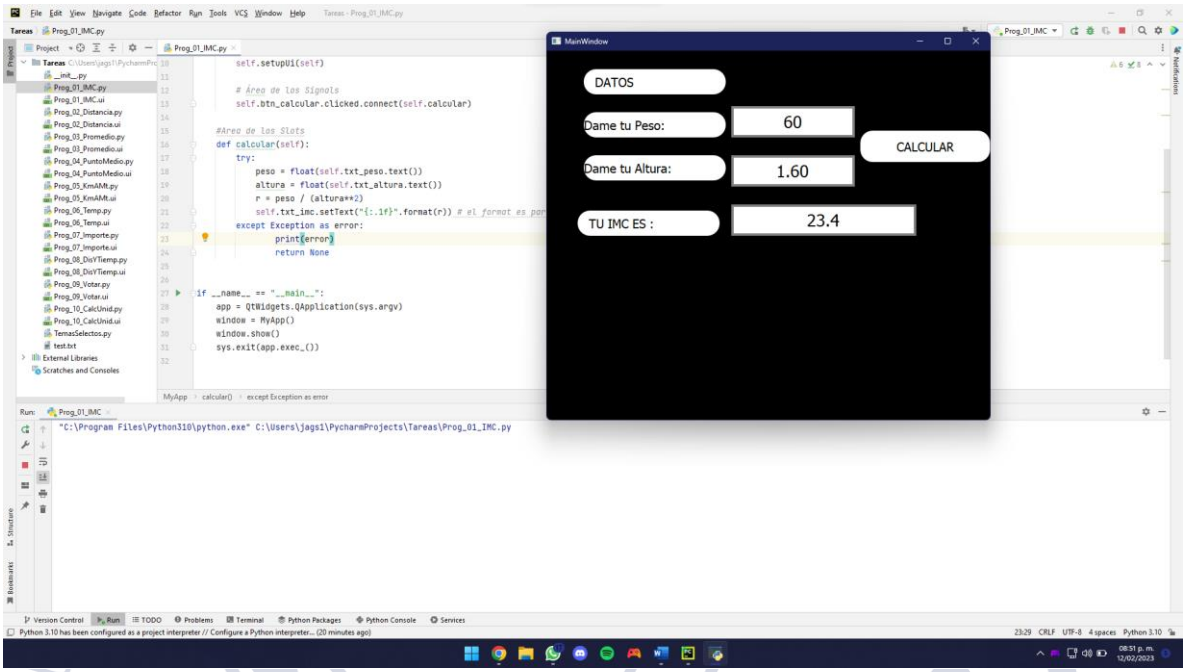
Interfaz paralela: Es un canal de comunicación que transmite datos en bloques. Ejemplos incluyen el puerto paralelo LPT.

Interfaz de red: Es un canal de comunicación que permite la transmisión de datos entre diferentes dispositivos a través de una red. Ejemplos incluyen Ethernet y Wi-Fi.

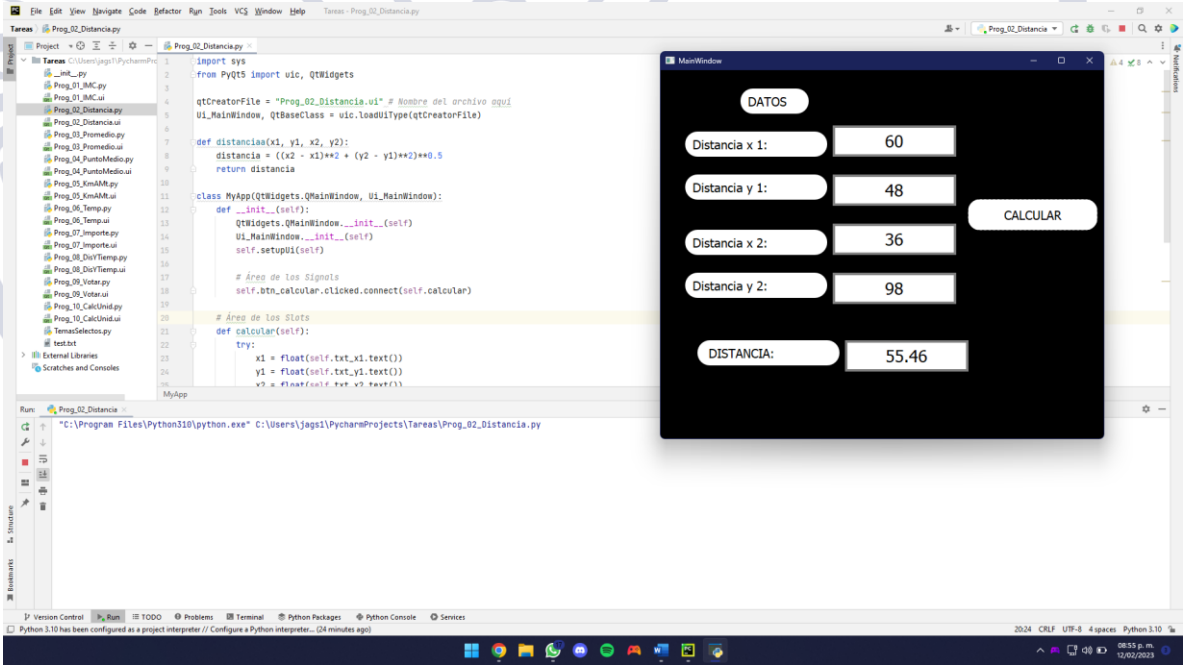
Interfaz de almacenamiento: Es un canal de comunicación que permite la conexión de dispositivos de almacenamiento, como discos duros y unidades flash USB, a un sistema electrónico.

Programas / ejercicios.

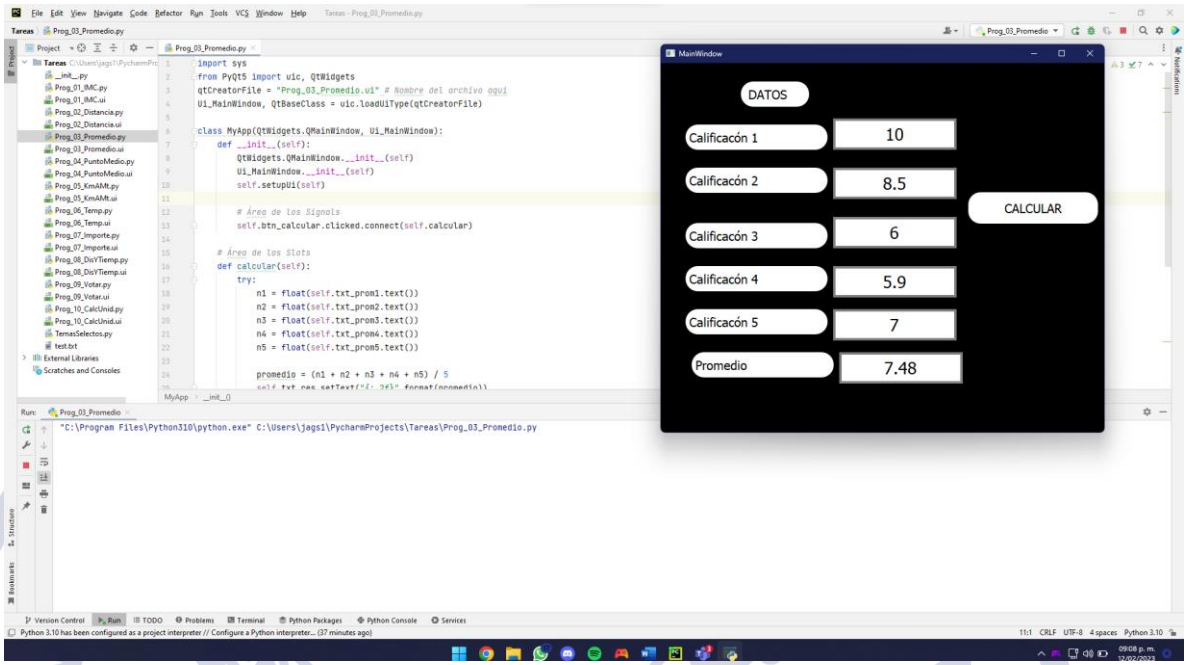
1. Cálculo del IMC



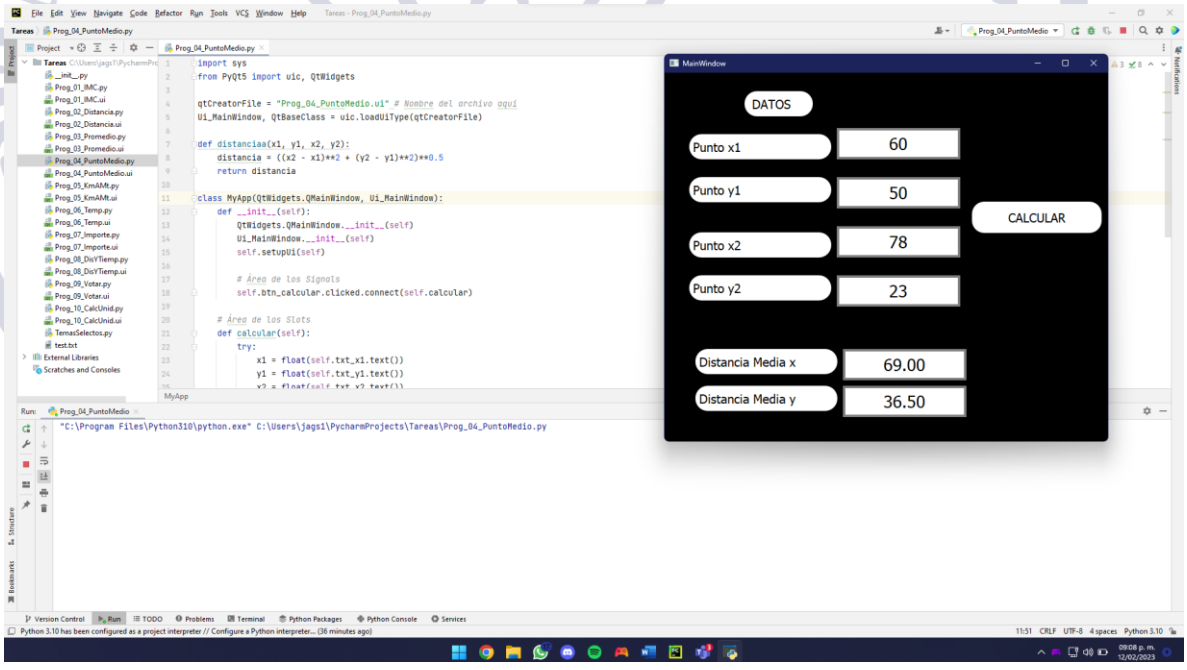
2. Distancia entre dos puntos



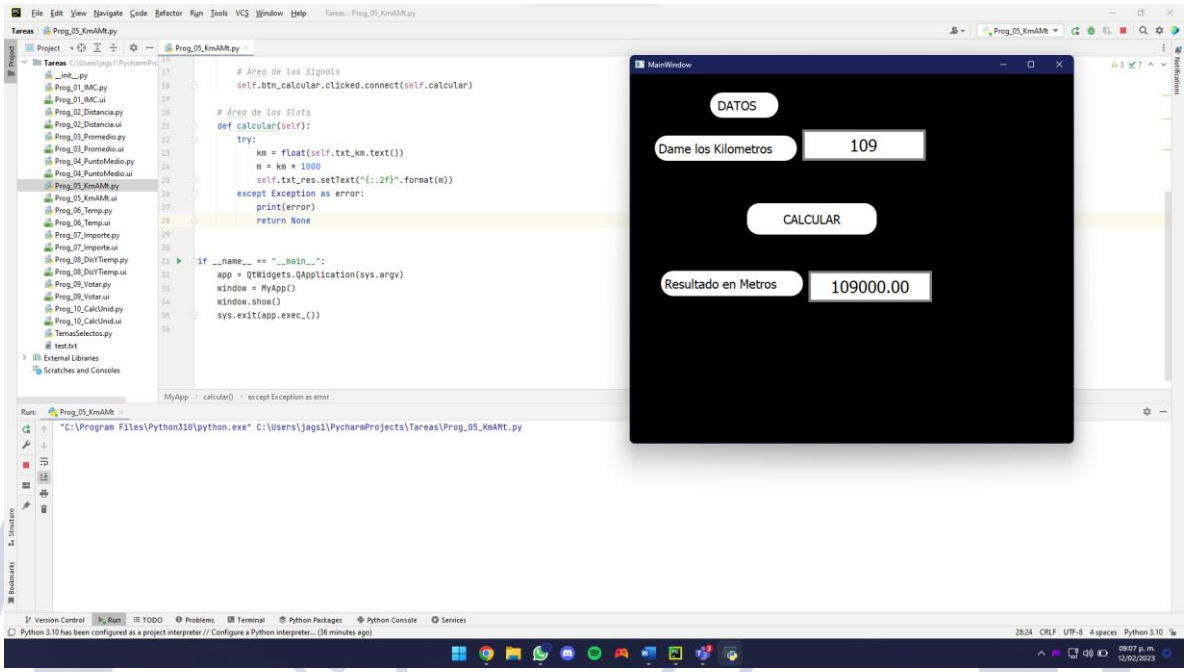
Promedio calificaciones



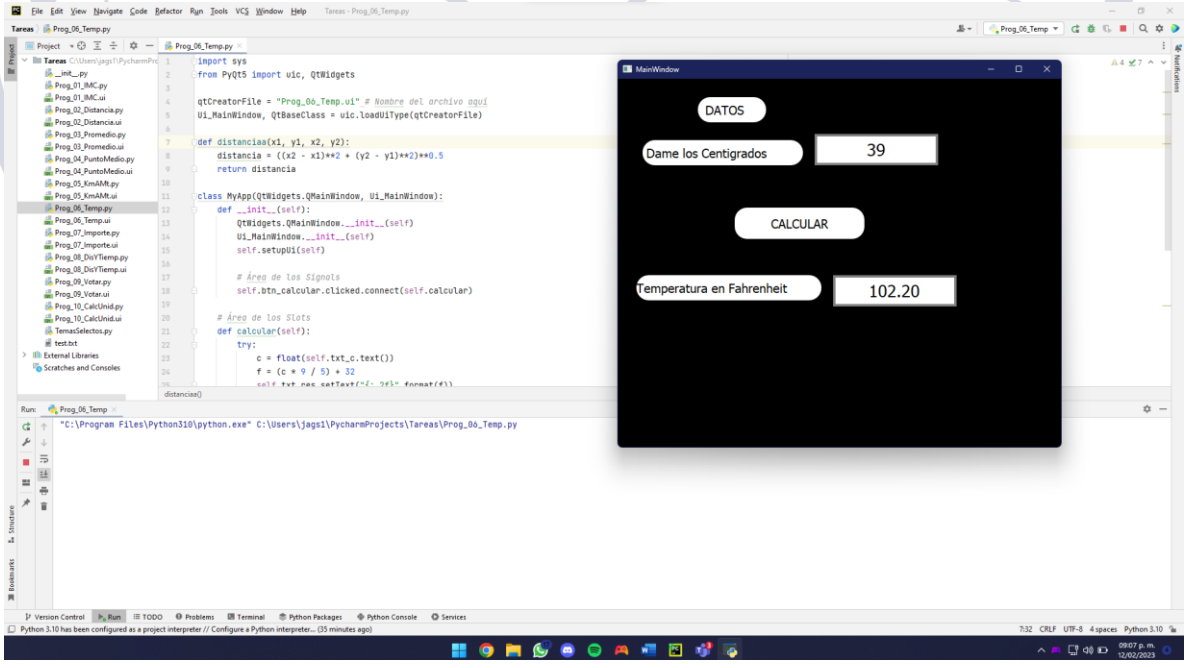
Punto medio entre dos puntos



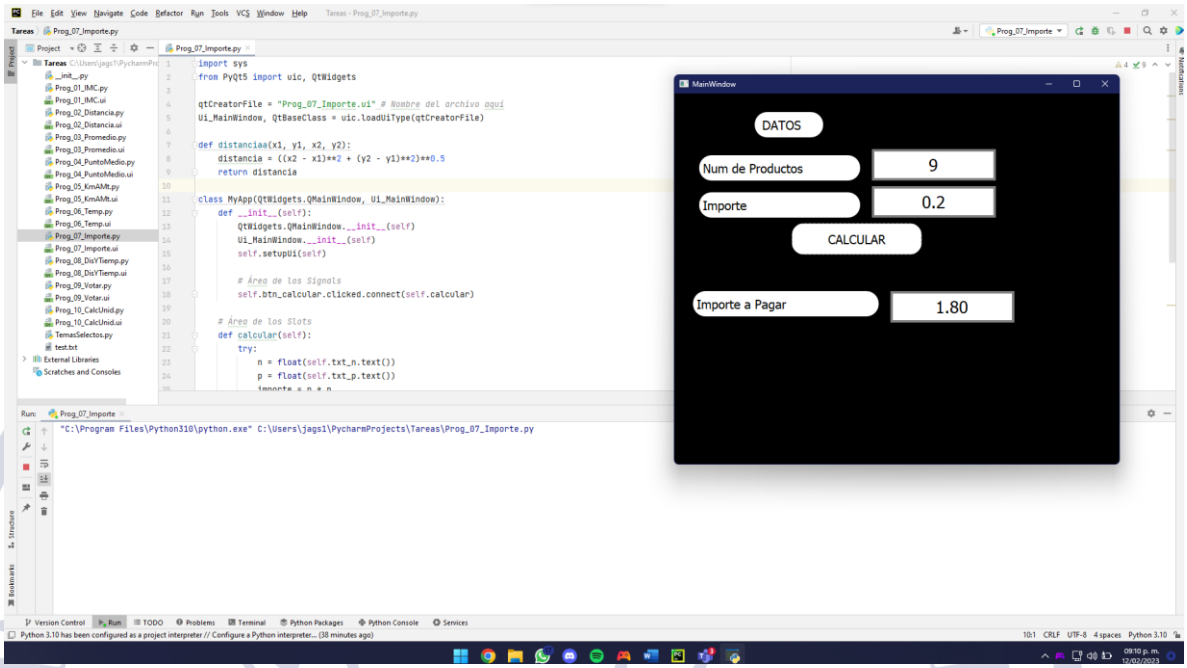
Conversión de Kilometro a Metros



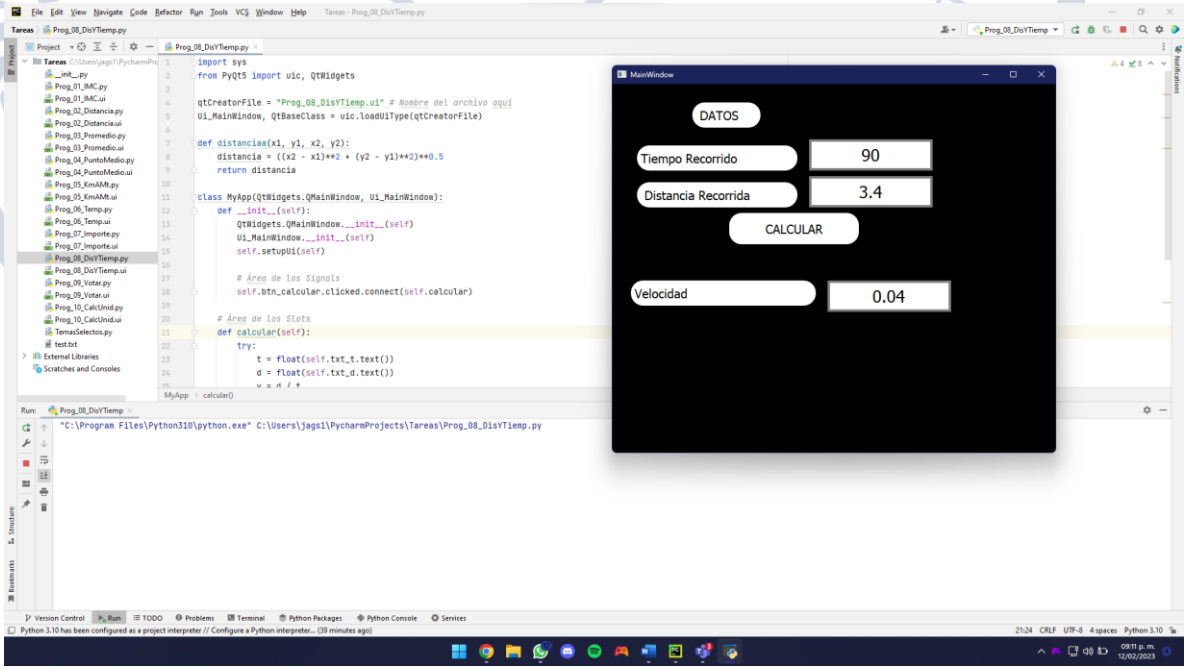
Cambio de Centígrados a Fahrenheit



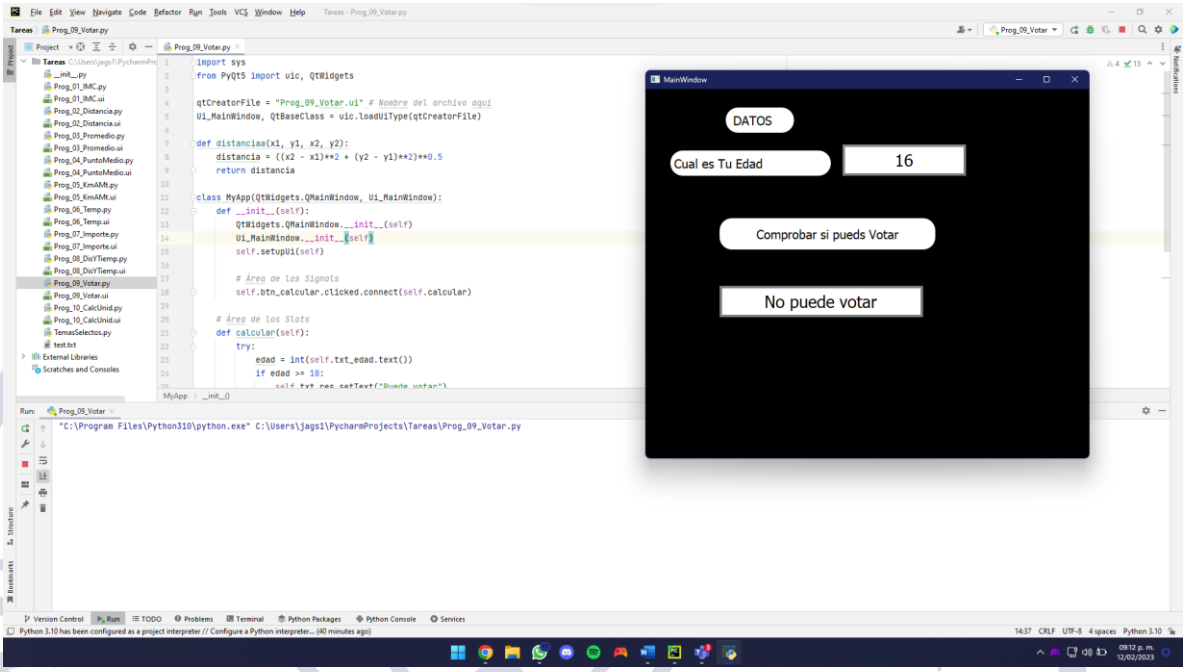
Importe a pagar por **n** productos con un mismo precio **p**



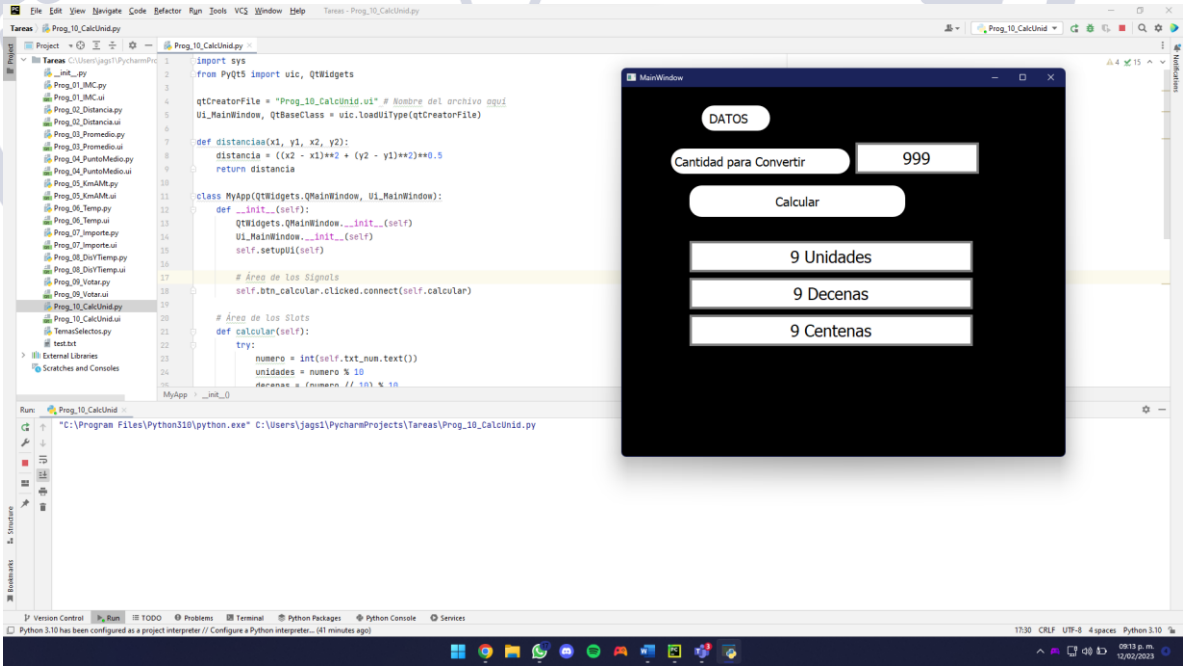
Velocidad de un objeto dada su distancia recorrida y un tiempo



Determinar si una persona puede votar o no, dada su edad



Calcular la cantidad de unidades, decenas y centenas de un número



Enlace al repositorio o repositorios de la Unidad (Códigos y/o Complementos).

Gonzalez Santiago Jose Alonso: https://github.com/Lolo192O/Programacion-Interfaces-y-Puertos

Moreno Wilches Fernando: https://github.com/ferchinwill/Programacion-Interfaces-y-Puertos

Verástegui Cruz Carlos: https://github.com/CarlosVerasteguii/Programacion-Interfaces-y-Puertos

Enlace de los trabajos en equipo: https://github.com/CarlosVerasteguii/Programacion-Interfaces-y-Puertos/tree/main/Portafolio%20de%20Evidencias%20Equipo%204