

# **UNIVERSIDAD NACIONAL DE TRUJILLO**

**FACULTAD DE INGENIERIA**

**ESCUELA DE INGENIERÍA MECATRÓNICA**



## **LABORATORIO N°8 – GRUPOS IMPARES**

**CURSO:** PROGRAMACIÓN I

**INTEGRANTES:**

VERA NORIEGA, BRAYAN STALIN

VÁSQUEZ SILVA, AUGUSTO MARTIN

VENTURA SERRANO, SEBASTIAN

**DOCENTES:**

ASTO RODRIGUEZ EMERSON MAXIMO

**CICLO:** III CICLO

**TRUJILLO – PERÚ**

**Agosto del 2022**

# Índice

|   |          |
|---|----------|
| <b>RESUMEN .....</b>                          | <b>3</b> |
| <b>DESARROLLO DEL LABORATORIO .....</b>       | <b>4</b> |
| <b>1.1 Resultados de la experiencia .....</b> | <b>4</b> |
| <b>Resultado 1: .....</b>                     | <b>4</b> |
| <b>Resultado 2: .....</b>                     | <b>5</b> |
| <b>Resultado 3: .....</b>                     | <b>6</b> |
| <b>Resultado 4: .....</b>                     | <b>6</b> |
| <b>TEST DE COMPROBACIÓN .....</b>             | <b>7</b> |
| <b>BIBLIOGRAFÍA .....</b>                     | <b>8</b> |
| <b>ANEXOS .....</b>                           | <b>9</b> |

## RESUMEN

Para el presente laboratorio se hará uso de los conocimientos adquiridos en las clases teóricas de la semana 10 de los temas “Archivos-Módulos-Librerías diversas” y haciendo uso anaconda y Python para añadir las librerías necesarias guiándonos así de las grabaciones y explicaciones de internet para poder desarrollar el contenido de cada una de las cuestiones planteadas y adquirir mayores conocimientos en el uso del nuevo programa utilizado, después de analizar las distintas cuestiones hemos resuelto cada una de estas mismas con cada una de las peticiones que se nos menciona; Además logramos desarrollar con éxito el test de comprobación, explicando que es un `archivo_git_init.py`, explicando qué significa la `w`, `r`, `a`, `t`, `b`, en el manejo de archivos y explicando que son los decoradores en Python, logrando así absorber cada uno de estos conocimientos para el progreso en el curso.

## DESARROLLO DEL LABORATORIO

### 1.1 Resultados de la experiencia

#### Resultado 1:

Escribir un programa que mediante voz pregunte los datos de una persona, para luego crear un archivo csv con los datos consultados. Si una nueva persona quiere registrar sus datos, el programa debe anexar estos en el mismo archivo csv.

```
from ast import Str
from asyncio import write
import py_compile
from numpy import source
import pyttsx3
import speech_recognition
engine = pyttsx3.init()
recognizer = speech_recognition.Recognizer()
peticion=input('¿quiere ingresar sus datos?. (y/n)')
if 'y' in peticion:
    engine.say('Dime tu nombre')
    engine.runAndWait()
    with speech_recognition.Microphone() as source:
        audio=recognizer.listen(source)
        NOMBRE=recognizer.recognize_google(audio,Language='es')
    engine.say('Dime tu edad')
    engine.runAndWait()
    with speech_recognition.Microphone() as source:
        audio=recognizer.listen(source)
        EDAD=recognizer.recognize_google(audio,Language='es')
    engine.say('Dime tu profesion')
    engine.runAndWait()
    with speech_recognition.Microphone() as source:
        audio=recognizer.listen(source)
        PROF=recognizer.recognize_google(audio,Language='es')
    Str(NOMBRE)
    Str(EDAD)
    Str(PROF)
    engine.say('OK.TU NOMBRE ES:')
    engine.say(NOMBRE)
    engine.say('TIENES')
    engine.say(EDAD)
    engine.say('AÑOS DE EDAD, Y ERES')
    engine.say(PROF)
    engine.say('ESO ES TODO, tus datos ya fueron ingresados, GRACIAS')
    engine.runAndWait()
import csv
import random
with open('Datos_personales.csv','a', newline = "") as file:
    writer=csv.writer(file, delimiter=";",)
```

```
writer.writerow([NOMBRE, EDAD, PROF])
```

¿quiere ingresar sus datos?. (y/n)y

|   | A                 | B       | C           | D |
|---|-------------------|---------|-------------|---|
| 1 | Augusto Vázquez   | 20 años | mecatrónica |   |
| 2 | Sebastián Ventura | 20 años | mecatrónico |   |
| 3 | Brian Vera        | 19 años | mecatrónico |   |
| 4 |                   |         |             |   |

## Resultado 2:

Escribir un programa que consulte si uno quiere saber sobre algún tema en específico, y luego responda sobre lo consultado usando información de wikipedia. El programa debe responder usando voz.

```
import wikipedia as wiki
import speech_recognition as sr
import pyttsx3 as pyt
recognizer=sr.Recognizer()
engine = pyt.init()
def Hablar(enunciado):
    engine.say(enunciado)
    engine.runAndWait()
def escuchar():
    try:
        with sr.Microphone() as source:
            print("...")
            audio= recognizer.listen(source)
            Mivoz= recognizer.recognize_google(audio, Language='es')
            Mivoz= Mivoz.lower(0)
    except:
        pass
    return Mivoz
def atender():
    Hablar("¿Qué deseas saber?")
    print("Qué deseas saber?")
    Mivoz = escuchar()
    pregunta = Mivoz
    wiki.set_lang("es")
    respuesta = wiki.summary(pregunta, sentences = 2)
    print(respuesta)
    Hablar(respuesta)
atender()
```

¿Que desea saber?:  
 Pokemon  
 Wikipedia: Pokémon (ポケモン, Pokemon?) es una franquicia de medios que originalmente comenzó como un videojuego RPG, pero debido a su popularidad ha logrado expandirse a otros medios de entretenimiento como series de televisión, películas, juegos de cartas, ropa, entre otros, convirtiéndose en una marca que es reconocida en el mercado mundial.

### Resultado 3:

Escribir un programa que que lea un archivo csv e imprima la información de forma agradable en la consola.

```
import csv

with open('Datos.csv') as file:
    reader=csv.reader(file)
    for row in reader:
        print("Apellido: {0}, Nombre: {1}, Género: {2}, Edad: {3},
Universidad: {4}, Carrera: {5} ".format(row[0], row[1], row[2], row[3],
row[4], row[5]))
```

```
PS D:\Bash\python> & C:/Users/HP/anaconda3/envs/programacionUNT/python.exe d:/Bash/python/lector_csv.py
Apellido: Vasquez Silva, Nombre: Augusto Martin, Género: Hombre, Edad: 20, Universidad: UNT, Carrera: Ingenieria mecatronica
Apellido: Ventura Serrano, Nombre: Sebastian, Género: Hombre, Edad: 20, Universidad: UNT, Carrera: Ingenieria mecatronica
Apellido: Vera Noriega, Nombre: Brayan Stalin, Género: Hombre, Edad: 19, Universidad: UNT, Carrera: Ingenieria mecatronica
PS D:\Bash\python> █
```

|   | A   | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Vasquez Silva,Augusto Martin,Hombre,20,UNT,Ingenieria mecatronica |   |   |   |   |   |
| 2 | Ventura Serrano,Sebastian,Hombre,20,UNT,Ingenieria mecatronica    |   |   |   |   |   |
| 3 | Vera Noriega,Brayan Stalin,Hombre,19,UNT,Ingenieria mecatronica   |   |   |   |   |   |
| 4 |   |   |   |   |   |   |
| 5 |   |   |   |   |   |   |

### Resultado 4:

Escribir un programa en el que el usuario pueda establecer una hora y minuto del día, y sea alertado mediante una notificación ó sonido 5 minutos antes.

```
from datetime import datetime, timedelta
import pytz
import os
dt = datetime.now()
print("Hora actual")
print("{}:{}".format(dt.hour, dt.minute))
from win10toast import ToastNotifier
print("Estableza una hora y minuto")
Hora= int(input("Hora: "))
Minuto= int(input("Minuto: "))

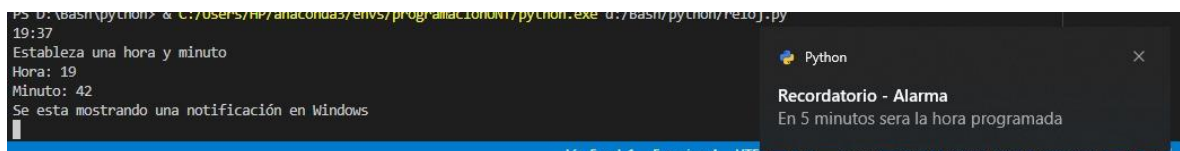
def notificacion():
    os.chdir(os.path.dirname(os.path.realpath(__file__)))
    toast=ToastNotifier()
```

```

titulo="Recordatorio - Alarma"
mensaje="En 5 minutos sera la hora programada"
time=10
icon=None

toast.show_toast(titulo, mensaje, icon_path=icon, duration=time,
threaded=True)
print ("Se esta mostrando una notificación en Windows")
while True:
    import datetime
    tiempo=datetime.datetime.now()
    tiempo_5_minutos_antes= tiempo+timedelta(minutes=5)
    if Hora==datetime.datetime.now().hour and
Minuto==tiempo_5_minutos_antes.minute:
        notificacion()
        break

```



## TEST DE COMPROBACIÓN

### i) Explique que es el archivo `__init__.py`

En el desarrollo de proyectos nos encontramos frecuentemente con la necesidad de importar variables, funciones, clases de otros archivos localizados en una subcarpeta dentro del directorio, en el entorno de Python si a este punto importamos dicho archivo, nos genera un error, dado que al empezar una importación el programa va a la carpeta y localiza al archivo `__init__.py` y luego ejecuta los demás archivos. En resumen, el archivo `__init__.py` convierte a una carpeta en un modulo que contiene mas módulos para importarlos.

### ii) Explique qué significa la w, r, a, t, b, en el manejo de archivos.

Estos códigos son las funciones que podemos realizar en un proyecto para el control de archivos:

Estos códigos se utilizan con la función

```
open("documentos.txt", "comando")
```

(w) SOBREScribir-este comando es usado para ingresar texto dentro del archivo abierto, desde el principio sobre escribiendo todo; crea el archivo si este no existe:

```
f=open("documentos.txt","w")
f.write("ingresando datos")
f.close()
```

(r) LEER-este comando abre el archivo para lectura, permitiéndonos usar su contenido. Este nos da error cuando el archivo no existe:

```
f=open("documentos.txt","r")
print(f.read())
```

(a)AÑADIR-viene de append, y cumple con agregar los datos ingresados al final sin sobrescribir el archivo:

```
f=open("documentos.txt","a")
f.write("ingresando datos")
f.close()
```

(t) MODO TEXTO-indica que solo se ingresa texto

(b)BINARIO-nos permite guardar el archivo como una imagen u otro tipo que se puede codificar de esta manera.

### iii) Investigue y explique que son los decoradores en Python.

Estos son llamados también patrones de diseño que cambia dinámicamente la funcionalidad de un método, clase o función, ayudándonos a tener un código mas limpio, mejor autodocumentado, volverlo mucho más legible, permitiéndonos también agregar un control a nuestras funciones con posibilidades infinitas.

Se conocen dos tipos de decoradores:

Por parámetros:

- No admiten parámetros
- Sí admiten parámetros

Preservación de la firma o signature:

- Decoradores no que preservan la firma
- Decoradores que sí la preservan

## BIBLIOGRAFÍA

Decoradores en Python. (2009). aps. Recuperado 29 de julio de 2022, de <https://www.apsl.net/blog/2009/08/31/decoradores-en-python/>

G. P.; E. I. (s. f.).(2020) Decoradores en Python. Easycode. Recuperado 29 de julio de 2022, de <https://codigofacilito.com/articulos/decoradores-python/>



## **ANEXOS**

[https://github.com/BRAYAN-VERA/Laboratorio-8-GRUPO\\_3-PI-UNT-2022..git](https://github.com/BRAYAN-VERA/Laboratorio-8-GRUPO_3-PI-UNT-2022..git)