



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Lenguaje de Programación Python

Actividad 12:


Diccionario

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 8 de Noviembre del 2023

 **RMBA_PY_ACT12.ipynb** ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)


Comentario Compartir Configuración B

+ Código + Texto Conectar

ACTIVIDAD 12

MENU

- 1.- Agregar (automatico 10)
- 2.- Eliminar {ID}
- 3.- Imprimir lista (tabla)
- 4.- Buscar {ID}
- 5.- Buscar {appat} todas las coincidencias
- 6.- Ordenar {ID}
- 7.- Generar archivo {ID} (preguntar nombre del archivo)
 - a) excel
 - b) txt
 - c) cvs
 - d) Markdown
- 8.- Cargar archivo {ID}
- 9.- Imprimir archivo {ID}
- 10.- Borrar Toda la lista {ID}

 **RMBA_PY_ACT12.ipynb** ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

Comentario Compartir Configuración B

+ Código + Texto Conectar

9.- Imprimir archivo {ID}

10.- Borrar Toda la lista {ID}

0.- SALIR

NOTA: Los datos del diccionario son los Datos básicos de un trabajador de una fabrica.

NOTA 2: VALIDAR EL PROGRAMA 100%

Preguntar siempre si esta seguro eliminar, No ordenar si ya esta Ordenado, decir si lista vacía, no existe o mostrar si estala búsqueda, No se permiten ID repetidos

```
[ ] !pip install pyyaml
```

Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (6.0.1)

```
[ ] import sys
import os
from IPython.display import clear_output
import random
import json
import yaml
import pandas as pd
```

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

CO

RMBA_PY_ACT12.ipynb

☆

Comentario

Compartir

⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

+ Código+ TextoConectar

```
[ ] def valinum_int(msge,ri=0,rf=9999999,msgError1="El numero esta fuera de rango",msgError2="Solo valores numericos"):
    while True:
        try:
            num = int(input(f"{msge}: "))
        except:
            print (msgError2)
            continue
        if num < ri or num > rf:
            print(f"{msgError1} ({ri} a {rf})")
            continue
        else:
            break
    return num

[ ] def agregar_aut(lista_dic):
    clear_output()
    registro = ["ID","Nombre","Ap.Paterno","Ap.Materno","Cargo","Sexo"]
    nombres_m=["Carlos","Francisco","Eduardo","Eynar","Alexis","Abraham","Irving","Andres"]
    nombres_f=["Camila","Nayeli","Gabriela","Samantha","Jennifer","Citlalli","Vanessa","Mayra"]
    apellidos=["Torres","Sanchez","Peredo","Ceseña","Diaz","Perez","Cruz","Ramirez","Garcia","Martinez"]
    puesto=["Supervisor","Encargad@","Conserje","Gerente","Guardia","Empleado Gral","Jefe"]
    sexo=["H","M"]
    for i in range(10):
        list_datos=[]
        repetido = False
        while repetido == False:
            clear_output()
            id = random.randint(300000,400000)
            repetido = verificar_id(lista_dic,id)
```

CO

RMBA_PY_ACT12.ipynb

☆

Comentario

Compartir


⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

+ Código+ TextoConectar

```
list_datos=[]
repetido = False
while repetido == False:
    clear_output()
    id = random.randint(300000,400000)
    repetido = verificar_id(lista_dic,id)
list_datos.append(id)
h_o_m =random.randint(1,2)
cant_nom = random.randint(1,2)
if h_o_m == 1:
    if cant_nom == 1:
        list_datos.append(random.choice(nombres_m))
    else:
        nombre = f"{random.choice(nombres_m)} {random.choice(nombres_m)}"
        list_datos.append(nombre)
else:
    if cant_nom == 1:
        list_datos.append(random.choice(nombres_f))
    else:
        nombre = f"{random.choice(nombres_m)} {random.choice(nombres_m)}"
        list_datos.append(nombre)
list_datos.append(random.choice(apellidos))
list_datos.append(random.choice(apellidos))
list_datos.append(random.choice(puesto))
if h_o_m == 1:
    list_datos.append(sexo[0])
else:
    list_datos.append(sexo[1])
lista_dic.append(dict(zip(registro,list_datos)))
```



RMBA_PY_ACT12.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

Comentario

Compartir

⚙️

B


+ Código+ Texto

Conectar

```
[ ] def eliminar(lista_dic):
    clear_output()
    encontrado = False
    id_eliminar = valinum_int("Introduce el ID a eliminar: ")
    for i, diccionario in enumerate(lista_dic):
        if diccionario.get("ID") == id_eliminar:
            print(f"El ID ha sido eliminado: {id_eliminar}")
            del lista_dic[i]
            encontrado = True
    if not encontrado:
        print(f"ID no encontrado {id_eliminar}")

def verificar_id(lista_dic,id):
    for diccionario in lista_dic:
        if diccionario.get("ID") == id:
            print("ID ya existente")
            return False
    return True

[ ] def imprimir(lista_dic,check_ord):
    clear_output()
    op = random.randint(1,3)
    lista_ord = sorted(lista_dic, key=lambda x: x["ID"])
    if check_ord == 1:
        salida=pd.DataFrame(lista_ord)
        print(salida)
    else:
```



RMBA_PY_ACT12.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

Comentario

Compartir

⚙️

B

+ Código+ Texto

Conectar

```
[ ] lista_ord = sorted(lista_dic, key=lambda x: x["ID"])
if check_ord == 1:
    salida=pd.DataFrame(lista_ord)
    print(salida)
else:
    salida=pd.DataFrame(lista_dic)
    print(salida)

[ ] def ordenar_dic(lista_dic,check_ord):
    clear_output()
    if check_ord == 1:
        print("La lista ya se encuentra ordenada")
    else:
        print("Se ordeno la lista")
    return 1

[ ] def buscar(lista_dic):
    clear_output()
    dic_encontrado = []
    encontrado = False
    id_busq = valinum_int("Ingresa el ID a buscar: ",300000,400000)
    for diccionario in lista_dic:
        if diccionario.get("ID") == id_busq:
            print(f"ID encontrado: {id_busq}")
            dic_encontrado.append(diccionario)
    salida=pd.DataFrame(dic_encontrado)
```

CO

RMBA_PY_ACT12.ipynb

☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Comentario Compartir Configuración B

+ Código + Texto

Conectar

```
[ ] encontrado = False
id_busq = valinum_int("Ingresa el ID a buscar: ",300000,400000)
for diccionario in lista_dic:
    if diccionario.get("ID") == id_busq:
        print(f"ID encontrado: {id_busq}")
        dic_encontrado.append(diccionario)
        salida=pd.DataFrame(dic_encontrado)
        print(salida)
        encontrado = True
if not encontrado:
    print(f"No se encontro el ID {id_busq}")

[ ]
def buscar_appat(lista_dic):
    clear_output()
    dic_encontrado = []
    encontrado = False
    appat_busq = input("Ingresa el apellido a buscar ")
    for diccionario in lista_dic:
        if diccionario.get("Ap.Paterno") == appat_busq:
            dic_encontrado.append(diccionario)
            encontrado = True
    if encontrado == True:
        print(f"Se encontro registros con el apellido: {appat_busq}")
        salida=pd.DataFrame(dic_encontrado)
        print(salida)
    if not encontrado:
        print(f"No se encontro ningun registro con el apellido {appat_busq}")
```

CO

RMBA_PY_ACT12.ipynb

☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Comentario Compartir Configuración B

+ Código + Texto

Conectar

```
[ ] def generar_archivo(lista_dic):
    clear_output()
    nombre_archivo = input("Ingresa el nombre con el que deseas guardar el archivo: ")
    data = pd.DataFrame(lista_dic)
    data.to_excel(f'/content/drive/MyDrive/colab/ACTIVIDAD_12/{nombre_archivo}.xlsx', index=False)
    data.to_csv(f'/content/drive/MyDrive/colab/ACTIVIDAD_12/{nombre_archivo}.csv', index=False)
    data.to_markdown(f'/content/drive/MyDrive/colab/ACTIVIDAD_12/{nombre_archivo}.md', index=False)

[ ] def cargar_archivo():
    clear_output()
    path = "/content/drive/MyDrive/proyecto/holamundo.txt"
    with open(path, "r") as f:
        contenido = f.read()
    print("SE CARGO EL ARCHIVO")
    return contenido

[ ] def imprimir_archivo(contenido):
    clear_output()
    print("CONTENIDO DEL ARCHIVO: ")
    print(contenido)

[ ] def menu():
    print("MENU:")
    print("1.-AGREGAR (AUTOMATICO 10)")
```