

**PRODUCTO INTEGRADOR DE APRENDIZAJE**

Nombre de la Unidad de Aprendizaje : Producto Integrador de Aprendizaje\_\_\_\_\_\_\_\_\_

Nombre del proyecto: Programar una solución de desarrollo de software mediante Python que demuestre la aplicación de los conceptos de estructuras de datos multidimensionales, su serialización y su almacenamiento y recuperación en sistemas gestores de bases de datos.

Programa educativo: Licenciado en Tecnologías de Información\_\_\_\_\_\_\_\_\_\_

Semestre: \_3\_\_\_ Grupo: \_\_32\_\_\_\_

Nombre del maestro: ALVARO FRANCISCO SALAZAR GONZALEZ \_\_\_\_\_\_\_\_\_

Nombre de los integrantes del equipo:

TORRES LONGORIA XIMENA ABIGAIL #2011987

San Nicolás de los Garza, ciudad universitaria a (día-mes-año)

### INDICE

[Introducción . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_INTRODUCCION.)

[Especificaciones y Explicación del código . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2.](#_Especificaciones_y_explicación)

[Importaciones y listas . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . .. . . . . . . . . . . . . . . . .2](#_Librería_de_importaciones.)

[descripción del entorno tecnológico de trabajo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. 3](#_Origen_de_Python)

[Introducciones de uso del script . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_SCRIPT_DE_PYTHON) 4

[Ejecución del código y cumplimiento de objetivos . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_Ejecución_del_código) 5

[Conclusión . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_Conclusion.) 31

[Referencias en formato APA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_Blibliografias_en_formato)

[Valores UANL . . . . . .. . . .. . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_Valores_de_la)31

### Introducción.

A continuación vamos a hablar y explicar un poco sobre lo que fue este producto integrador de aprendizaje hablaremos sobre la descripción del entorno tecnológico de trabajo y veremos un poco del Uso del script veremos nuestro programa y conforme vayamos viendo lo vamos a ir explicando paso a paso lo que contiene este código y trabajamos en el sql ya en el cual documentamos los dichos clientes y sus reservaciones tenemos la opción de eliminar , modificar datos fechas o nombre de cualquier cliente utilizando su clave id , es muy interesante este proyecto recomiendo poner mucha atención en cada código , ya que es muy legible y bien tradado , también damos explicación de lo que es el SCRIPTS Y el uso de Python en el entorno de la informática.

GITHUB

<https://github.com/XimenaLongoria01/PIA_PROGRAMACION>

# Librería de importaciones.

En la librería de importaciones damos el uso dependiendo de que códigos vamos a utilizar ya sea Datetime, Random, csv , etc, pero es importante ponerlos ya que cada uno da esa importación de usar el método deseado

### Captura de pantalla de computadora Descripción generada automáticamente

### Listas y diccionarios.

Las listas en este caso las utilice para identificar los datos que utilice durante el código para después pasarlos directamente a la base de datos que creamos para documentar dicho programa.

Los diccionarios los utilice para modificar los datos que deseábamos.

Texto

Descripción generada automáticamente

# SCRIPT DE PYTHON

El recurso de scripts para IBM® SPSS® Statistics proporciona la capacidad de crear **Python** ® **scripts** que operan en la interfaz de usuario de IBM SPSS Statistics , manipulan objetos de salida y ejecutan la sintaxis del mandato. Esta característica requiere IBM SPSS Statistics -Complemento de integración para Python, que se instala de forma predeterminada con Producto IBM SPSS Statistics .

Hay disponible una interfaz complementaria para crear **programasPython** que le permiten controlar el flujo de trabajos de sintaxis de mandatos, leer y escribir datos y crear procedimientos personalizados. Consulte el tema [Introducción a los programas de Python](https://www.ibm.com/docs/es/SSLVMB_29.0.0/statistics_python_plugin_uber_project_ddita/spss/programmability_option/python_package_intro.html#python_package_intro) para obtener más información.

## **Ámbito**

Puede ejecutar scripts Python directamente desde dentro de IBM SPSS Statistics, desde dentro de programas Python o desde un proceso Python externo, como por ejemplo como un IDE de Python o el intérprete de Python .

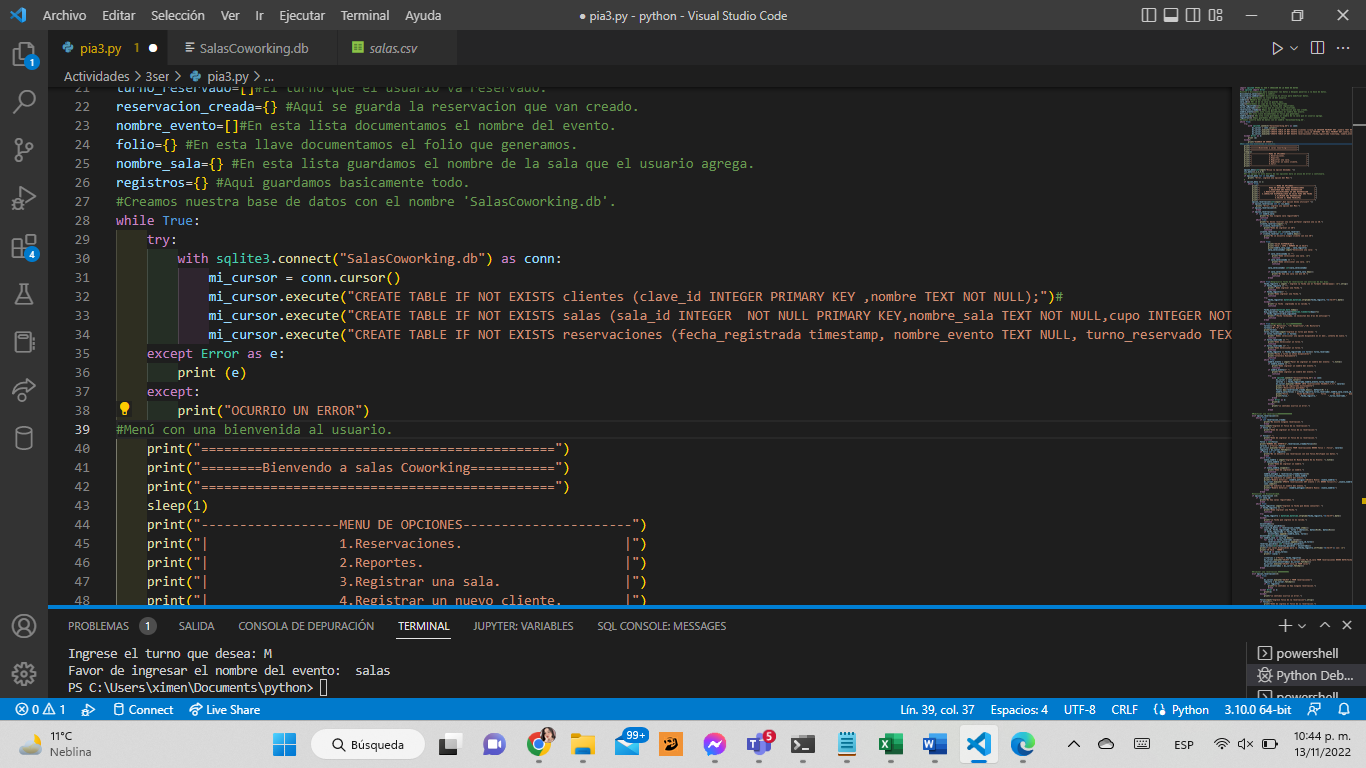
# Origen de Python

Dada la popularidad que ha adquirido en la última década, cabría esperar que se trate de un lenguaje nacido a comienzos de este siglo. Nada más lejos de la realidad: su origen se remonta a finales de los años 80 y principios de los 90. Su implementación comenzó en diciembre de 1989 cuando Guido van Rossum, trabajador del CWI (un centro de investigación holandés de carácter oficial) decidió empezar el proyecto como un pasatiempo dándole continuidad al lenguaje de programación ABC desarrollado por el equipo del que había formado parte en el CWI.

Su nombre se debe a la afición de Van Rossum al grupo Monty Python y su concepción se enfocaba en que fuera fácil de usar y aprender sin que esto penalizara sus capacidades. La causa de que no llegara a adquirir la suficiente importancia en su momento fue la falta de recursos en el hardware de la época.

El avance en las tecnologías de hardware ha sido una condición necesaria para el repunte de su popularidad. No obstante, la generalización del big data en los últimos años, seguida de la explosión de la inteligencia artificial, el machine learning, el deep learning y el surgimiento de la ciencia de datos o data science como una nueva área de trabajo con especialistas propios han revolucionado el panorama. Y es que muchas de las nuevas herramientas que han surgido, y que son explotadas por los ingenieros de datos y los científicos de datos, han sido desarrolladas en Python o nos ofrecen Python como la forma predilecta de interactuar con ellas.

### Ejecución del código y cumplimiento de objetivos.

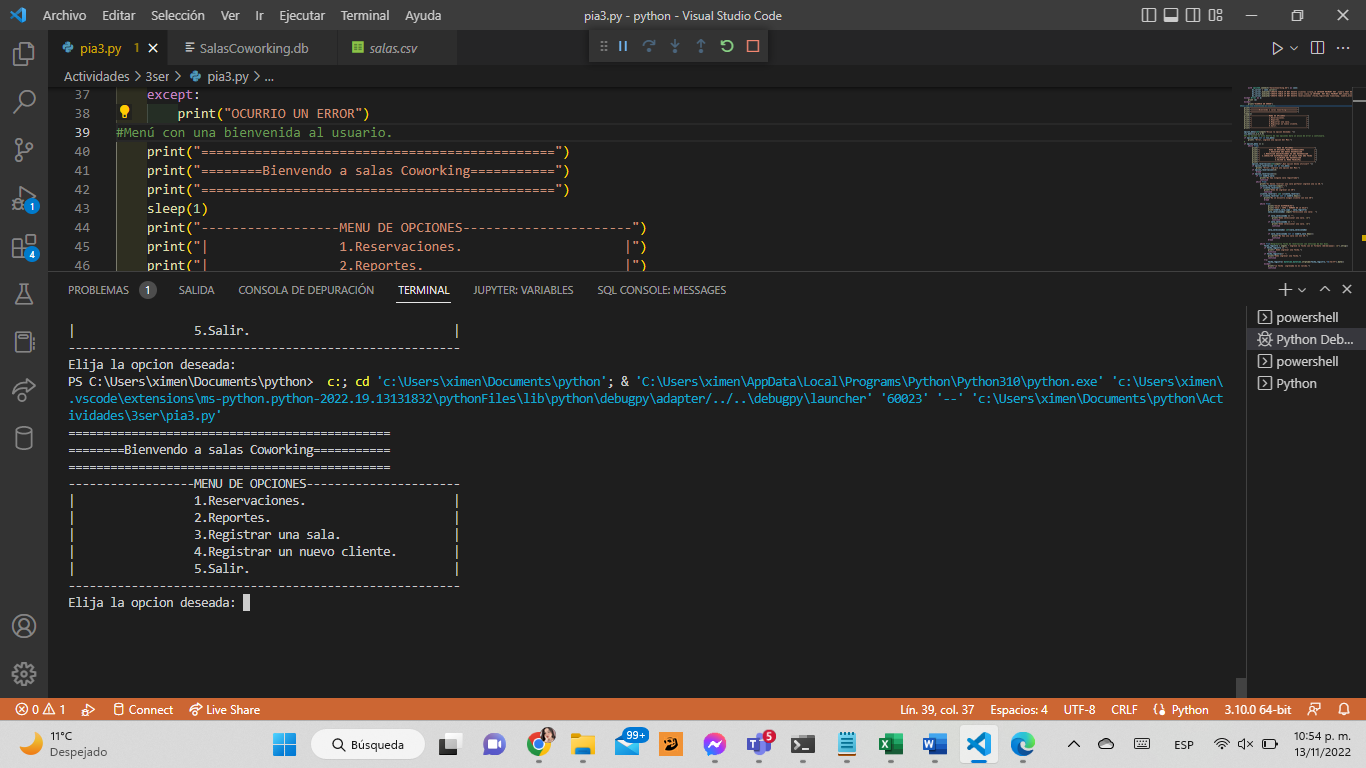
Creación de las 3 tablas llamadas Clientes , Salas, Reservaciones, Con cada valor asignado.

Documentación de SQL lite

Imagen que contiene Texto

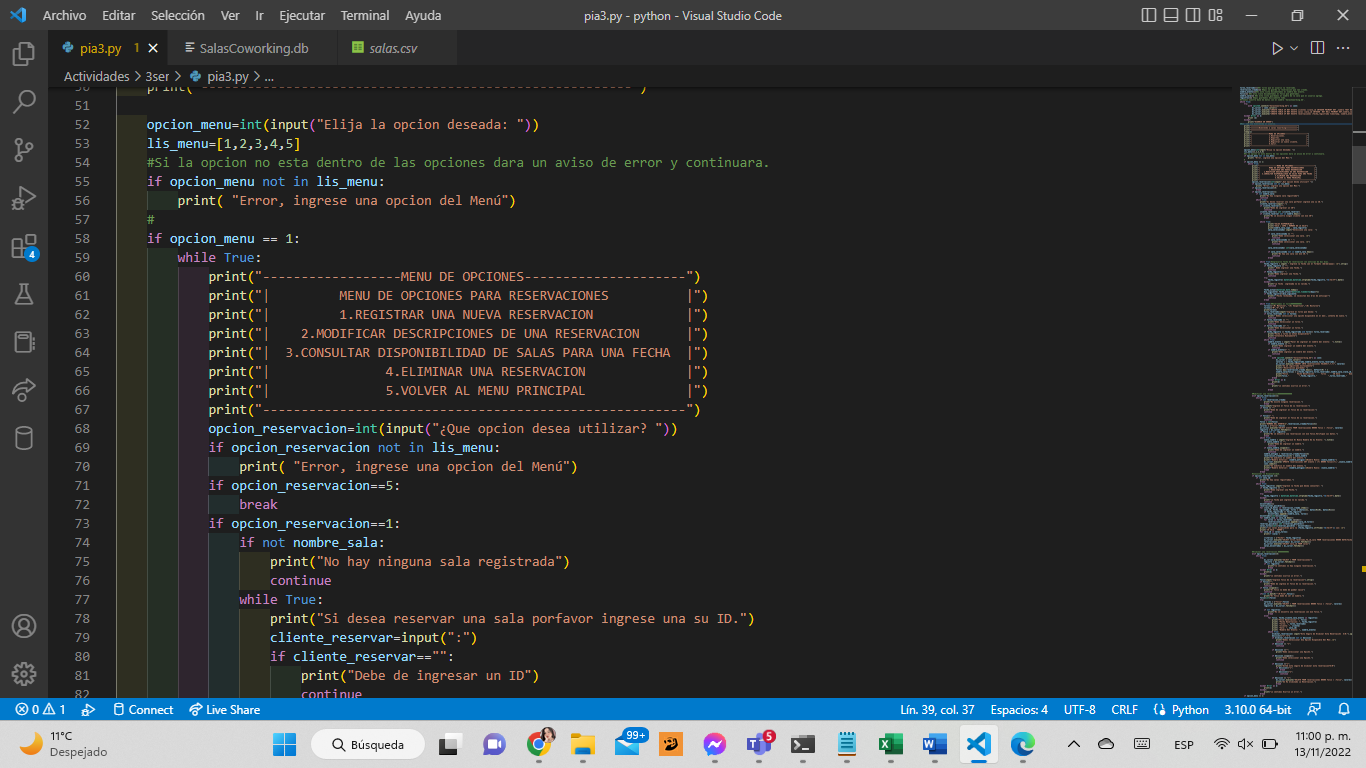
Descripción generada automáticamente

Menú de opciones.



Captura de pantalla de computadora

Descripción generada automáticamenteEl código sleep es para dar unos segundos antes de imprimir el siguiente código.

Si la opción es 1 te mandara este menú de opciones donde podrás hacer lo que gustes indicado en el programa.

En la opción 1 reservamos una sala ya existente y si no hay en existencia te pedirá que registres una.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

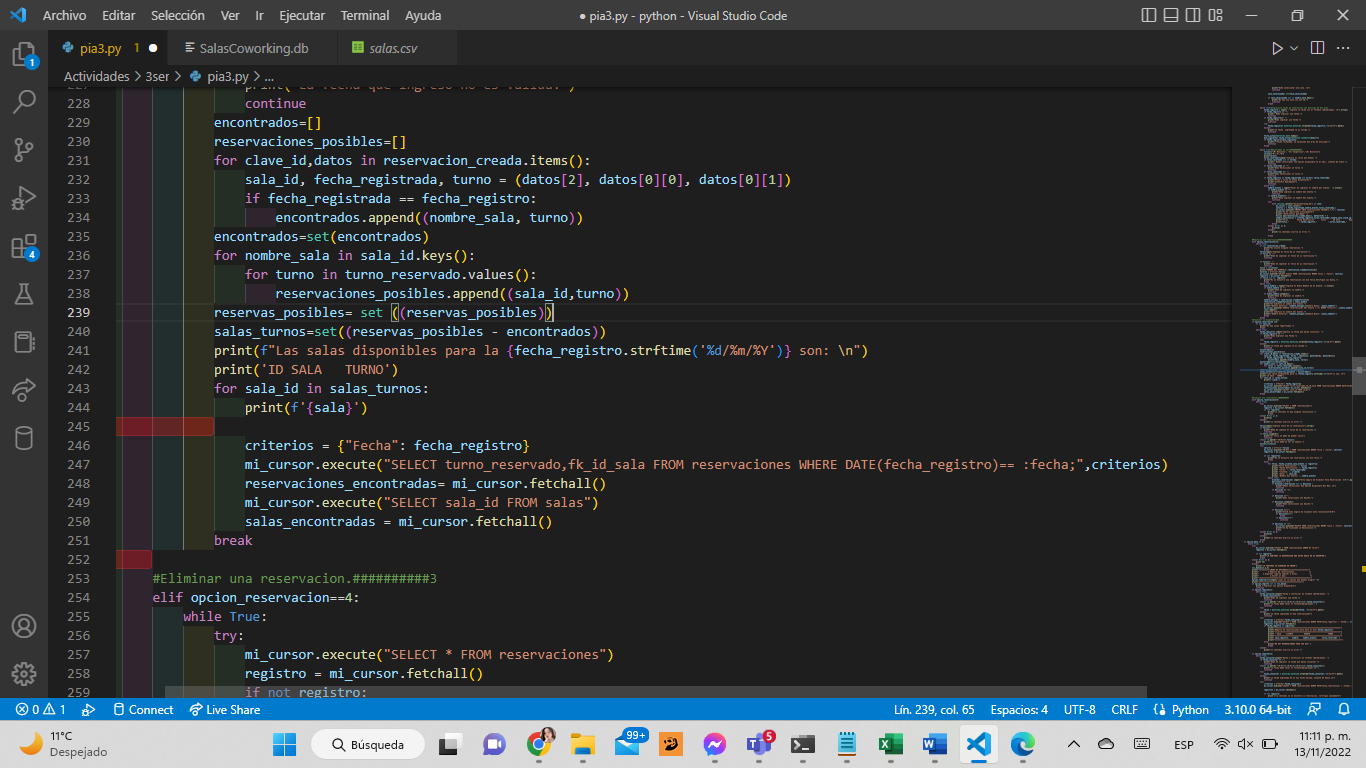
Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

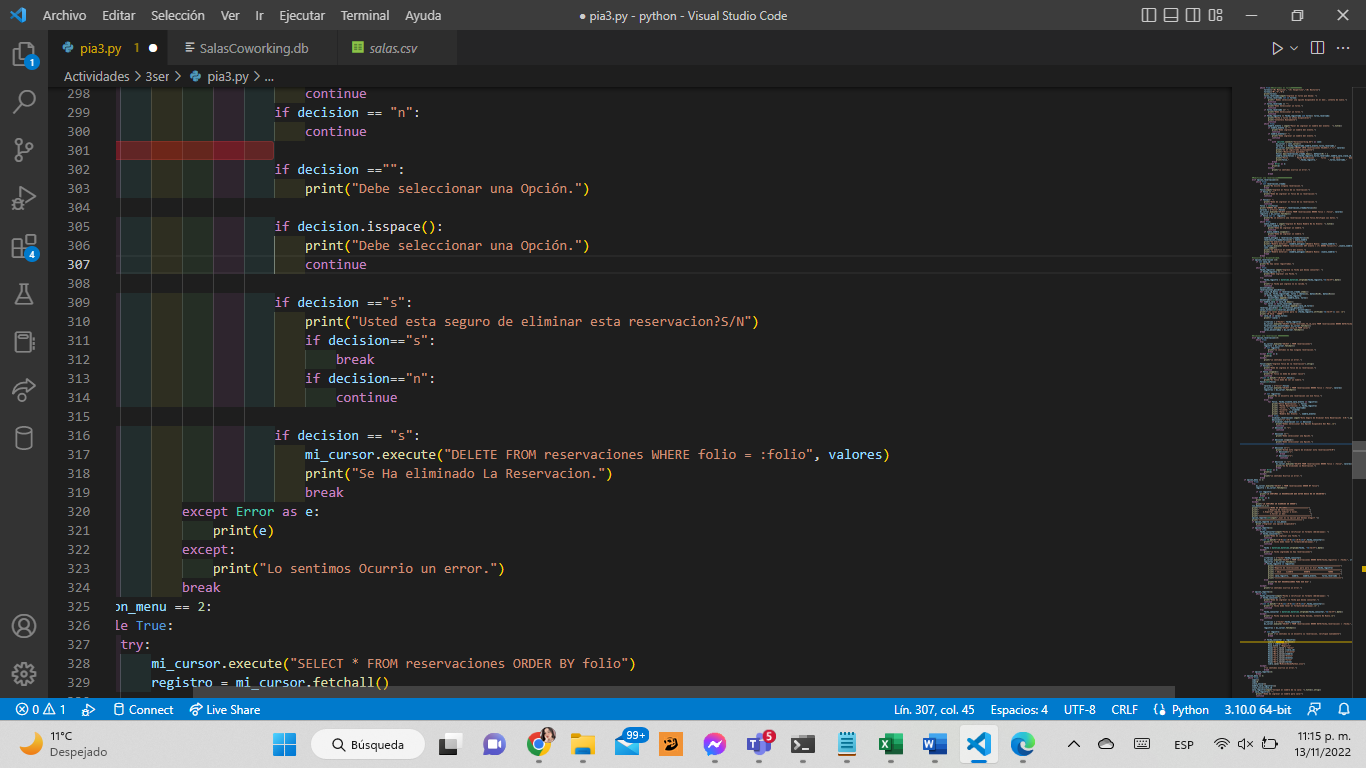


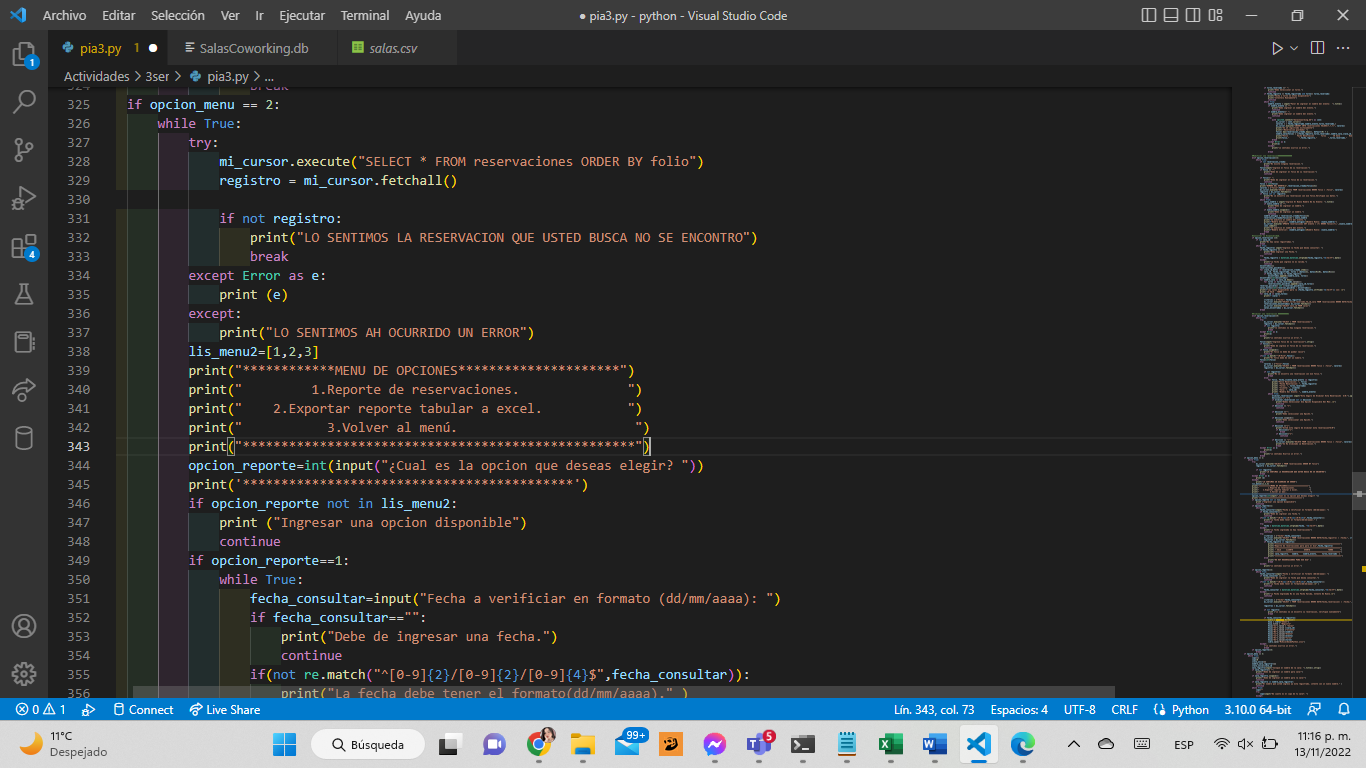
Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente





Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

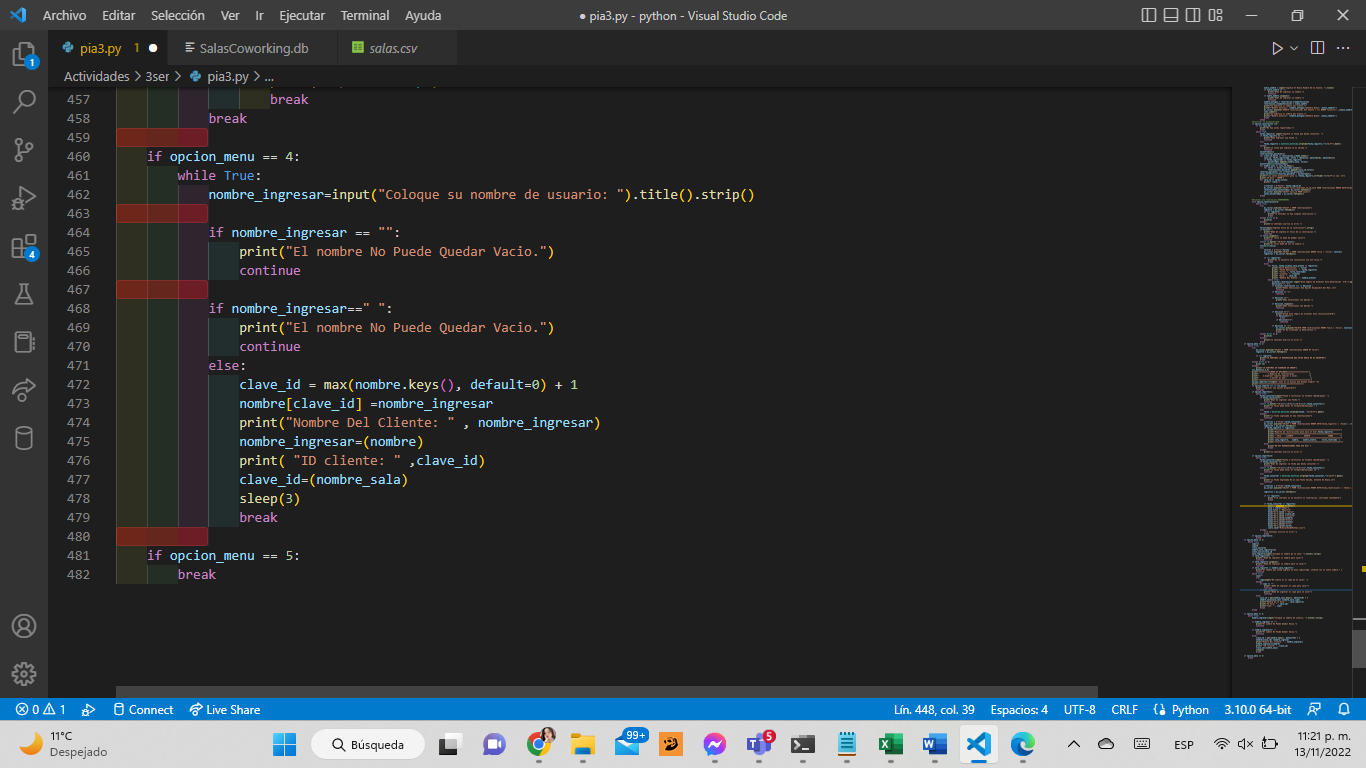
Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente



# CODIGO.

#LIBRERIA DE IMPORTACIONES

from csv import reader #CSV PARA EL USO DE ARCHIVOS

import random #PARA EL USO DE NUMERO O PALABRAS ALAZAR

from time import sleep #LO UTILIZAMOS PARA DAR UN TIEMPO ESTIMADO AL PROGRAMA

import datetime #PARA REALIZAR VALIDACION DE FECHAS

import re # RE PARA EL USO DE VALIDACIONES

import sys #

import os #PARA EL USO DE LOS ARCHIVOS CSV

#PARA EL USO DE LOS ARCHIVOS DE EXCEL.

import sqlite3 #PARA EL USO Y CREACION DE LA BASE DE DATOS

from sqlite3 import Error

#Listas y diccionarios para alamacenar los datos y despues pasarlos a la base de datos.

diccionario\_salas={}#Sala a cambiar.

diccionario\_cambiar={}#El diccionario se utliza para modificar datos.

clave\_id={} #Guardamos la clave id del usuario.

nombre={} #Nombre del usuario.

sala\_id={} #El id de la sala se guarda aqui.

cupo={}#Guardamos el cupo de la sala registrada

fecha\_registrada=[]#guardamos la fecha que registramos.

turno\_reservado=[]#El turno que el usuario va reservado.

reservacion\_creada={} #Aqui se guarda la reservacion que van creado.

nombre\_evento=[]#En esta lista documentamos el nombre del evento.

folio={} #En esta llave documentamos el folio que generamos.

nombre\_sala={} #En esta lista guardamos el nombre de la sala que el usuario agrega.

registros={} #Aqui guardamos basicamente todo.

#Creamos nuestra base de datos con el nombre 'SalasCoworking.db'.

while True:

try:

with sqlite3.connect("SalasCoworking.db") as conn:

mi\_cursor = conn.cursor()

mi\_cursor.execute("CREATE TABLE IF NOT EXISTS clientes (clave\_id INTEGER PRIMARY KEY ,nombre TEXT NOT NULL);")#

mi\_cursor.execute("CREATE TABLE IF NOT EXISTS salas (sala\_id INTEGER NOT NULL PRIMARY KEY,nombre\_sala TEXT NOT NULL,cupo INTEGER NOT NULL);")

mi\_cursor.execute("CREATE TABLE IF NOT EXISTS reservaciones (fecha\_registrada timestamp, nombre\_evento TEXT NULL, turno\_reservado TEXT NOT NULL);")

except Error as e:

print (e)

except:

print("OCURRIO UN ERROR")

#Menú con una bienvenida al usuario.

print("==============================================")

print("========Bienvendo a salas Coworking===========")

print("==============================================")

sleep(1)

print("------------------MENU DE OPCIONES----------------------")

print("| 1.Reservaciones. |")

print("| 2.Reportes. |")

print("| 3.Registrar una sala. |")

print("| 4.Registrar un nuevo cliente. |")

print("| 5.Salir. |")

print("--------------------------------------------------------")

opcion\_menu=int(input("Elija la opcion deseada: "))

lis\_menu=[1,2,3,4,5]

#Si la opcion no esta dentro de las opciones dara un aviso de error y continuara.

if opcion\_menu not in lis\_menu:

print( "Error, ingrese una opcion del Menú")

#

if opcion\_menu == 1:

while True:

print("------------------MENU DE OPCIONES---------------------")

print("| MENU DE OPCIONES PARA RESERVACIONES |")

print("| 1.REGISTRAR UNA NUEVA RESERVACION |")

print("| 2.MODIFICAR DESCRIPCIONES DE UNA RESERVACION |")

print("| 3.CONSULTAR DISPONIBILIDAD DE SALAS PARA UNA FECHA |")

print("| 4.ELIMINAR UNA RESERVACION |")

print("| 5.VOLVER AL MENU PRINCIPAL |")

print("-------------------------------------------------------")

opcion\_reservacion=int(input("¿Que opcion desea utilizar? "))

if opcion\_reservacion not in lis\_menu:

print( "Error, ingrese una opcion del Menú")

if opcion\_reservacion==5:

break

if opcion\_reservacion==1:

if not nombre\_sala:

print("No hay ninguna sala registrada")

continue

while True:

print("Si desea reservar una sala porfavor ingrese una su ID.")

cliente\_reservar=input(":")

if cliente\_reservar=="":

print("Debe de ingresar un ID")

continue

cliente\_reservar= int (cliente\_reservar)

if cliente\_reservar not in nombre.keys():

print("No se encontro ningun cliente con ese ID")

break

while True:

print("SALAS DISPONIBLES")

print("SALA | CUPO | NOMBRE DE LA SALA")

print(nombre\_sala,cupo , sala\_registro)

sala\_selecionada= input("Seleccione una sala: ")

if sala\_selecionada == "":

print("Debe seleccionar una sala. \n")

continue

if sala\_selecionada == " ":

print("Debe seleccionar una sala. \n")

continue

sala\_selecionada= int(sala\_selecionada)

if sala\_selecionada not in nombre\_sala.keys():

print("No hay una sala con ese ID.")

continue

break

while True:#Ingresarla fecha de reservacion con anticipo de dos dias.

fecha\_registro = input( " Ingrese la fecha con el formato (dd/mm/aaaa): \n").strip()

if fecha\_registro == "":

print(" Debe ingresar una fecha.")

continue

if fecha\_registro==" ":

print("Debe ingresar una fecha.")

continue

try:

fecha\_registro= datetime.datetime.strptime(fecha\_registro,"%d/%m/%Y").date()

except:

print("La fecha ingresada no es valida.")

continue

fecha\_actual=datetime.date.today()

dia\_siguiente= fecha\_actual+datetime.timedelta(days=+1)

if fecha\_registro<=dia\_siguiente:

print(f"!Fecha rechazada, se necesitan dos días de anticipo!")

continue

break

while True:#Reservamos el turno##########3

turnos=["(M) Matutino", "(V) Vespertino","(N) Nocturno"]

turnoo=["M","V","N"]

print(turnos)

turno\_reservado=input("Ingrese el turno que desea: ")

if turno\_reservado not in turnoo:

print(" Debes seleccionar una opción disponible en el menú, intenta de nuevo.")

continue

if turno\_reservado == "":

print("Debe Seleccionar un turno.")

continue

if turno\_reservado ==" ":

print("Debe Seleccionar un turno.")

continue

if fecha\_registro == fecha\_registrada and turno== turno\_reservado:

print("Fecha y turno no estan disponibles")

print("Intentelo Nuevamente")

continue

while True:

nombre\_evento = input("Favor de ingresar el nombre del evento: ").title()

if nombre\_evento == "":

print("Debe ingresar un nombre del evento.")

continue

if nombre\_evento==" ":

print("Debe ingresar un nombre del evento.")

continue

try:

with sqlite3.connect("SalasCoworking.db") as conn:

mi\_cursor = conn.cursor()

valores = ( fecha\_registrada,nombre\_evento,turno\_reservado,)

mi\_cursor.execute("INSERT INTO reservaciones VALUES(?,?,?)", valores)

print("Se ah registrado exitosamente")

print("!Reservacion guardada! ")

folio= max(reservacion\_creada.keys(), default=0) + 1

nombre\_sala[folio] = [(fecha\_registro,turno\_reservado),nombre\_sala,clave\_id,nombre\_evento]

print("Folio Fecha Reservacion Turno ID Sala Nombre Del Evento")

print(folio," ",fecha\_registro," ",turno\_reservado," ",sala\_id," ",nombre\_evento)

break

except Error as e:

print(e)

except:

print("Lo sentimos ocurrio un error.")

break

#Modificar una reservacion##############

elif opcion\_reservacion==2:

while True:

if not reservacion\_creada:

print("No existe ninguna reservacion.")

break

folio=input("Ingrese el folio de su reservacion:")

if folio == "":

print("Debe de ingresar el folio de su reservacion:")

continue

if folio==" ":

print("Debe de ingresar el folio de su reservacion.")

continue

folio = int(folio)

print("NOMBRE DEL EVENTO\n",reservacion\_creada[folio][3])

valores = {"folio":folio}

mi\_cursor.execute("SELECT evento FROM reservaciones WHERE folio = :folio", valores)

registro = mi\_cursor.fetchall()

if folio not in registro:

print("No se encontro una reservacion con ese folio.Verifique sus datos.")

break

while True:

nuevo\_nombre = input("Ingrese El Nuevo Nombre De Su Evento: ").title()

if nuevo\_nombre == "":

print("Debe de ingresar un nombre.")

continue

if nuevo\_nombre.isspace():

print("Debe de ingresar un nombre.")

continue

nombre\_antiguo = reservacion\_creada[folio][3]

reservacion\_creada[folio][3] = nuevo\_nombre

print("Se modifico el nombre del evento.")

print(f"Nombre Anterior: {nombre\_antiguo}\nNombre Nuevo: {nuevo\_nombre}")

mi\_cursor.execute('UPDATE reservaciones SET evento = (?) WHERE folio=(?);',[nuevo\_nombre,folio])

conn.commit()

print("Se modifico el nombre del evento.")

print(f"Nombre Anterior: {nombre\_antiguo}\nNombre Nuevo: {nuevo\_nombre}")

break

break

#Consultar la Disponiblidad.

if opcion\_reservacion ==3:

if not sala\_id:

print("No hay salas registradas.")

break

while True:

fecha\_registro= input("Ingrese la fecha que desea consultar: ")

if fecha\_registro == "":

print("Debe Ingresar una Fecha.")

continue

try:

fecha\_registro = datetime.datetime.strptime(fecha\_registro,"%d/%m/%Y").date()

except:

print("La fecha que ingreso no es valida.")

continue

encontrados=[]

reservaciones\_posibles=[]

for clave\_id,datos in reservacion\_creada.items():

sala\_id, fecha\_registrada, turno = (datos[2], datos[0][0], datos[0][1])

if fecha\_registrada == fecha\_registro:

encontrados.append((nombre\_sala, turno))

encontrados=set(encontrados)

for nombre\_sala in sala\_id.keys():

for turno in turno\_reservado.values():

reservaciones\_posibles.append((sala\_id,turno))

reservas\_posibles= set ((reservas\_posibles))

salas\_turnos=set((reservas\_posibles - encontrados))

print(f"Las salas disponibles para la {fecha\_registro.strftime('%d/%m/%Y')} son: \n")

print('ID SALA TURNO')

for sala\_id in salas\_turnos:

print(f'{sala}')

criterios = {"Fecha": fecha\_registro}

mi\_cursor.execute("SELECT turno\_reservado,fk\_id\_sala FROM reservaciones WHERE DATE(fecha\_registro)== :fecha;",criterios)

reservaciones\_encontradas= mi\_cursor.fetchall()

mi\_cursor.execute("SELECT sala\_id FROM salas")

salas\_encontradas = mi\_cursor.fetchall()

break

#Eliminar una reservacion.##########3

elif opcion\_reservacion==4:

while True:

try:

mi\_cursor.execute("SELECT \* FROM reservaciones")

registro = mi\_cursor.fetchall()

if not registro:

print("lo sentimos no hay ninguna reservacion.")

break

except Error as e:

print(e)

except:

print("Lo sentimos ocurrio un error.")

folio=input("Ingrese folio de la reservacion").strip()

if folio=="":

print("Debe de ingresa el folio de su reservacion.")

continue

if folio.isspace():

print("El folio no debe de quedar vacio")

continue

if(not re.match("^[0-9]\*$",folio)):

print("El folio debe de ser un numero.")

folio=int(folio)

try:

valores = {"folio":folio}

mi\_cursor.execute("SELECT \* FROM reservaciones WHERE folio = :folio", valores)

registros = mi\_cursor.fetchall()

if not registros:

print("No se encontro una reservacion con ese folio.")

break

else:

for folio, fecha,cliente,sala,evento in registros:

print("Folio Reservacion: ", folio)

print( "Fecha Reservacion: ", fecha\_registro)

print( "Turno: ", turno\_reservado)

print( "Cliente: ", cliente)

print( "Sala: ", sala\_id)

print( "Nombre Del Evento: ", nombre\_evento)

while True:

eliminar\_reservacion= input("Esta Seguro de Eliminar Esta Reservación S/N:").upper().strip()

decision=["s","n"]

if eliminar\_reservacion not in decision :

print("Debes Seleccionar Una Opción Disponible Del Menú.\n")

continue

if decision == "n":

continue

if decision =="":

print("Debe seleccionar una Opción.")

if decision.isspace():

print("Debe seleccionar una Opción.")

continue

if decision =="s":

print("Usted esta seguro de eliminar esta reservacion?S/N")

if decision=="s":

break

if decision=="n":

continue

if decision == "s":

mi\_cursor.execute("DELETE FROM reservaciones WHERE folio = :folio", valores)

print("Se Ha eliminado La Reservacion.")

break

except Error as e:

print(e)

except:

print("Lo sentimos Ocurrio un error.")

break

if opcion\_menu == 2:

while True:

try:

mi\_cursor.execute("SELECT \* FROM reservaciones ORDER BY folio")

registro = mi\_cursor.fetchall()

if not registro:

print("LO SENTIMOS LA RESERVACION QUE USTED BUSCA NO SE ENCONTRO")

break

except Error as e:

print (e)

except:

print("LO SENTIMOS AH OCURRIDO UN ERROR")

lis\_menu2=[1,2,3]

print("\*\*\*\*\*\*\*\*\*\*\*\*MENU DE OPCIONES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" 1.Reporte de reservaciones. ")

print(" 2.Exportar reporte tabular a excel. ")

print(" 3.Volver al menú. ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

opcion\_reporte=int(input("¿Cual es la opcion que deseas elegir? "))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

if opcion\_reporte not in lis\_menu2:

print ("Ingresar una opcion disponible")

continue

if opcion\_reporte==1:

while True:

fecha\_consultar=input("Fecha a verificiar en formato (dd/mm/aaaa): ")

if fecha\_consultar=="":

print("Debe de ingresar una fecha.")

continue

if(not re.match("^[0-9]{2}/[0-9]{2}/[0-9]{4}$",fecha\_consultar)):

print("La fecha debe tener el formato(dd/mm/aaaa)." )

continue

try:

fecha = datetime.datetime.strptime(fecha, "%d/%m/%Y").date()

except:

print("La fecha ingresada no hay reservaciones")

continue

try:

criterios = {"fecha":fecha\_consultar}

mi\_cursor.execute("SELECT \* FROM reservaciones WHERE DATE(fecha\_registro) = :fecha;", criterios)

registros = mi\_cursor.fetchall()

if fecha\_registro in registros:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Reporte de reservaciones para para el dia",fecha\_registro)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print( " SALA CLIENTE EVENTO TURNO ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print( sala\_registro, nombre, nombre\_evento, turno\_reservado )

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

print("NO HAY RESERVACIONES PARA ESE DIA" )

break

except:

print("Lo sentimos ocurrio un error.")

if opcion\_reporte==2:

while True:

fecha\_consultar=input("Fecha a verificiar en formato (dd/mm/aaaa): ")

if fecha\_consultar == "":

print("Debe de ingresar la fecha que desea consultar.")

continue

if(not re.match("^[0-9]{2}/[0-9]{2}/[0-9]{4}$",fecha\_consultar)):

print("La fecha debe tener el formato(dd/mm/aaaa).\n" )

continue

try:

fecha\_consultar = datetime.datetime.strptime(fecha\_consultar,"%d/%m/%Y").date()

except:

print("La Fecha Ingresada No es una Fecha Valida, Intente De Nuevo.\n")

continue

try:

criterios = {"fecha":fecha\_consultar}

mi\_cursor.execute("SELECT \* FROM reservaciones WHERE DATE(fecha\_reservacion) = :fecha;", criterios)

registros = mi\_cursor.fetchall()

if not registro:

print ("Lo sentimos no se encontro su reservacion, verifique nuevamente")

break

if fecha\_consultar in registros:

libro = openpyxl.Workbook()

hoja = libro["Sheet"]

hoja.title = "Registro"

hoja["A1"].value = "Salas"

hoja["A2"].value =(sala\_id)

hoja["B1"].value ="Cliente"

hoja["B2"].value=(nombre)

hoja["C1"].value="Evento"

hoja["C2"].value=(evento)

hoja["D1"].value="Turno"

hoja["D2"].value=(turno)

libro.save("MiExcelDesdePython.xlsx")

except:

("Lo sentimos ocurrio un error.")

break

if opcion\_reporte==3:

break

if opcion\_menu == 3:

while True:

cupo=[]

cupo=0

clave\_unica={}

nombre\_sala\_registrar={}

clave\_unica=clave\_id

sala\_registro=input("Coloque el nombre de la sala: ").title().strip()

if sala\_registro== "":

print("!Debe de ingresar un nombre para sala!")

continue

if sala\_registro.isspace():

print("!Debe de ingresar un nombre para la sala!")

continue

if sala\_registro in nombre\_sala\_registrar:

print("El nombre que usted ingreso ya esta registrado, intente con un nuevo nombre." )

continue

while True:

cupo=""

try:

cupo=input("De cuanto es el cupo de tu sala?: ")

except:

if cupo == "":

print("!Debe de ingresar el cupo para sala!")

continue

if cupo.isspace():

print("!Debe de ingresar el cupo para la sala!")

continue

else:

sala\_id = max(nombre\_sala.keys(), default=0) + 1

nombre\_sala[sala\_id]= [nombre\_sala,cupo]

print("Nombre De La Sala: " , sala\_registro)

print("Id Sala: " , sala\_id)

print("Cupo: " , cupo)

break

break

if opcion\_menu == 4:

while True:

nombre\_ingresar=input("Coloque su nombre de usuario: ").title().strip()

if nombre\_ingresar == "":

print("El nombre No Puede Quedar Vacio.")

continue

if nombre\_ingresar==" ":

print("El nombre No Puede Quedar Vacio.")

continue

else:

clave\_id = max(nombre.keys(), default=0) + 1

nombre[clave\_id] =nombre\_ingresar

print("Nombre Del Cliente: " , nombre\_ingresar)

nombre\_ingresar=(nombre)

print( "ID cliente: " ,clave\_id)

clave\_id=(nombre\_sala)

sleep(3)

break

if opcion\_menu == 5:

break

Al tabular en cvs o en Excel te pedirá fecha la cual si no hay reservaciones entonces:

Texto

Descripción generada automáticamente

Y si no ingresa nada entonces:

Texto

Descripción generada automáticamente

Reporte de excel

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

# Blibliografias en formato APA

table [X] has no column named [Y]. (2021, 7 septiembre). Stack Overflow en español. <https://es.stackoverflow.com/questions/482691/table-x-has-no-column-named-y>

<https://colab.research.google.com/drive/1SCb2fU9tUdda4l2voD1lAYJws9zU9XaH?usp=sharing>

### Conclusion.

En este gran trabajo proyecto integrador de aprendizaje fue tan interesante ya que conforme fue pasando el semestre fuimos desarrollando mas y mas el programa cada etapa íbamos agregando un poco y un poco mas a este programa, fue muy interesante ver dos de mis materias mas importantes en una sola el que estuviera aquí una base de datos fue una sorpresa ya que jamás me lo imagine pero realmente fue de mi agrado, y lo que aprendimos para llevar a cabo este proyecto fue todo lo que esperaba para poder entender un poco mas de esta gran materia, claro que falta muchismo por aprender pero

### Valores de la UANL Misión, Visión, Valores y Política de Calidad

Nuestra política se establece sustentada en nuestra Misión, Visión y Valores, orientada en la búsqueda constante de la satisfacción con calidad de nuestros usuarios. Política de Calidad En el Sistema Integral de Bibliotecas de la Universidad Autónoma de Nuevo León tenemos el compromiso de brindar servicios de calidad, basados en nuestros valores y en la mejora continua para garantizar la satisfacción de los usuarios. Misión Proveer servicios bibliotecarios y de información con calidad a la comunidad universitaria y otras instituciones de Educación Media y Superior, a Nivel Nacional e Internacional. Visión El SIBUANL será un modelo de red de Bibliotecas Universitarias digitales con un alto prestigio que establecerá alianzas con redes afines y con otras organizaciones Nacionales e Internacionales. Valores La Dirección de Bibliotecas de la Universidad Autónoma de Nuevo León hace suyos los valores que la Universidad Autónoma de Nuevo León ha enunciado como representativos de la institución, los cuales son: actitud de servicio, ética, honestidad, integridad, justicia, responsabilidad, respeto, solidaridad, trabajo en equipo y verdad