

**EVIDENCIA CALIFICABLE #3**

Nombre de la Unidad de Aprendizaje : Etapa 3: Construir soluciones de información que permitan la serialización y la recuperación de elementos de su estado a formatos de intercambio de datos.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nombre del proyecto: Evidencia3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Semestre: 3\_\_\_\_ Grupo: \_\_32\_\_\_\_

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

Nombre de los integrantes del equipo:

TORRES LONGORIA XIMENA ABIGAIL #2011987

|  |  |
| --- | --- |
| **Contenido mínimo a evaluar** | **Cumplimiento** |
| Índice |  |
| Introducción .- incluye valores UANL aplicados |  |
| Análisis y emisión de juicio |  |
| Conclusiones individuales |  |
| Conclusión del equipo |  |
| Identificación de sub resultados de aprendizaje ANECA. | 3 |
| Calificación Evidencia: |  |
| Firma del maestro | |

San Nicolás de los Garza, ciudad universitaria a (02/11/2022)

TORRES LONGORIA XIMENA ABIGAIL #2011987

Mujer de cabello largo con la boca abierta

Descripción generada automáticamente con confianza media

## INDICE

Instrucciones……………………………………………………………………………….. 3

Código……………………………………………………………………………………………6

Imágenes de la ejecución………………………………………………………………..

Estructura de datos y su procesamiento.

Evidencia 3

Especificaciones del caso a desarrollar:

Un negocio dedicado a la renta de espacios de coworking desea automatizar su sistema de reserva de salas de reuniones atendiendo las siguientes indicaciones:

1) Supuestos:

a) Se cuenta con una cantidad determinada de salas con cupo variado dependiendo de la sala.

b) La renta de la sala se hace por turnos consistentes en: mañana, tarde y noche incluso en días festivos.

2) Condiciones:

a) La reserva de la sala se debe hacer, por lo menos, dos días antes y debe indicarse el nombre del evento al momento de realizar la reservación.

b) Solamente pueden reservar una sala aquellos que son clientes registrados.

c) No puede reservarse una sala para dos eventos al mismo tiempo.

d) Para cada reservación se debe generar un folio único.

3) Funcionalidad que se debe cumplir:

a) Se presentará un sistema de menú principal y submenús apegado a la siguiente estructura:

i) Reservaciones

(1) Debe poder registrarse la reservación de una sala existente, para un cliente específico en un turno específico bajo las condiciones señaladas.

(2) Una vez hecha la reservación se puede llegar a editar la descripción de la misma.

(3) Se debe poder consultar la disponibilidad de salas para una fecha especificada por el usuario, informando mediante un informe tabular que presente los siguientes datos de las opciones disponibles para esa fecha: clave de la sala, nombre de la sala y turno

(4) Para eliminar una reservación se debe identificar mediante su folio y, en caso de que dicho folio exista, se le deben mostrar los datos de ella al usuario para que confirme que desea eliminarla avisándole que este procedimiento no puede ser deshecho. Considere también que solamente pueden eliminarse reservaciones con tres días de anticipación.

(5) Volver al menú principal mostrará dicho elemento para que el usuario continúe con su trabajo.

ii) Reportes:

(1) Se debe poder obtener un reporte tabular de las reservaciones de salas para una fecha que indique el usuario bajo un formato similar al siguiente ejemplo:

Ejemplo del reporte a obtener para las reservaciones en una fecha específica

(2) Se debe poder exportar el mismo contenido del reporte tabular a formato de MsExcel (xlsx), en dicho reporte se puede omitir el formato, pero los datos a presentar deben estar completos con sus encabezados

(3) Volver al menú principal mostrará dicho elemento para que el usuario continúe con su trabajo.

iii) Registrar una sala con los siguientes datos:

(1) Una clave única, generada automáticamente, para la sala.

(2) Nombre de la sala (No puede omitirse).

(3) Cupo de la sala.

(a) No puede omitirse

(b) Debe ser mayor que cero

iv) Registrar a un nuevo cliente con los siguientes datos:

(1) Una clave única, generada automáticamente, para el cliente.

(2) Nombre del cliente (No puede omitirse)

v) Salir

b) Después de realizar cada acción, se deberá volver al menú en el cual se ofreció realizarla.

c) Al momento de iniciar el programa, se intentará detectar si es la primera vez que el programa se ejecuta, considere los siguientes casos:

i) Se encuentra el archivo de la base de datos, se asume entonces que ya hay datos previos del estado.

ii) No se encuentra el archivo de la base de datos, se asume entonces que es la primera ejecución del programa por lo que se avisa al usuario de tal situación y se procede a crear la base de datos con las estructuras necesarias para la operación de la solución

4) Entregables:

a) Cree un repositorio en plataforma Git para esta evidencia que contendrá el código desarrollado con acceso público.

b) Genere un documento de Word (.docx) que contendrá un reporte técnico de esta actividad con las siguientes características:

c) Utilizar formato Word (.docx)

d) Presentar portada oficial que incluya nombre y matrícula de cada integrante del equipo.

e) Incluir una segunda portada con el nombre completo de cada integrante ordenado alfabéticamente por sus apellidos y que incluya una fotografía a color en la que se aprecie claramente el rostro de cada miembro del equipo.

f) Tipografía Calibri 11

g) Tamaño carta

h) A espacio sencillo

i) Deberá presentar los siguientes contenidos: i) Un listado de objetivos específicos del problema (requerimientos) a cumplir por parte del proyecto.

ii) Código Python de la solución que incluya números de línea.

iii) Explicación textual del aprovechamiento del almacén de datos en SQLite 3 para proveer la funcionalidad solicitada refiriendo por números de línea las secciones de código que la soportan refiriendo las capturas de pantalla que lo demuestran, cada una de ellas con un título alusivo a lo que está demostrando. iv) Capturas de pantalla de la interfaz de usuario para el script y su secuencia de uso para demostrar el cumplimiento de la funcionalidad indicada.

v) Una sección donde indique la aportación específica de cada integrante del equipo en cuanto al proyecto y/o documento.

vi) Una sección de conclusiones donde se presente la conclusión grupal después de realizar el proyecto y una reflexión acerca de mejoras futuras que se le pudieran realizar.

vii) URL sintética para acceder al repositorio GIT en caso de ser necesaria su validación.

j) No deberá presentar faltas de ortografía.

k) Toda imagen o captura de pantalla deberá estar identificada de acuerdo con el formato APA 7 (https://tinyurl.com/3dpwnek5)

l) Deberá incluir un índice que omita las portadas y el índice en sí, debiendo iniciar la numeración en la primera página del contenido especificado (La portada y el índice no cuentan para la numeración de las páginas del documento).

Lista de cotejo

FACTOR\* APORTACIÓN

Permite el registro de las reservaciones bajo las condiciones señaladas

Solamente registra salas existentes 5

Solamente registra reservaciones para clientes existentes 5

No permite el registro de eventos simultáneos en la misma sala 15

Solamente permite el registro de salas con, al menos, dos días de

anticipación 5

Se debe generar un folio único para cada reservación 5

Permite editar el nombre del evento para una reservación ya hecha 5

Obtiene el reporte de reservaciones para una fecha con los datos especificados 15

Exporta el reporte de reservaciones a formato MsExcel 5

Obtiene el reporte de disponibilidad de salas para una fecha específica 15

Permite cancelar reservaciones con un mínimo de anticipación de tres días 5

Permite registrar nuevos clientes con los datos indicados 5

Permite registrar nuevas salas con los datos indicados 5

El entregable cumple con las indicaciones de formato, incluyendo formato APA para

las figuras y capturas de pantalla 5

El entregable cumple con las indicaciones de contenido 5

Total 100

\*No se calificarán los puntos incompletos o no demostrados en el entregable; así mismo el manejo del estado de la solución debe realizarse mediante interacción con una base de datos de SQLite3, de no cumplir con este último punto el trabajo se considerará como no entregado.

Menu principal

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Registrando un cliente y una sala

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Sala ya registrada

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Reservando una sala

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Reportar en pantalla una fecha determinada

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Tabulación en Excel

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Se crea el archivo csv correctamente

Texto

Descripción generada automáticamente

Pantalla de juego de computadora

Descripción generada automáticamente con confianza media

documentación de SQL LITE

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

CODIGO

from ast import While

from curses.ascii import isspace

from ast import Break, If, Return

from configparser import LegacyInterpolation

from csv import reader

import random

from re import T

from time import sleep

import datetime

import re

import sys

import os

usuario1=[]

sala1=[]

turno1=[]

evento=[]

fecha1=[]

diccionario\_cambiar={}

diccionario\_notas={}

sala\_registro={}

diccionario\_salas={}

csv={}

turno=['Matutino','Vespertino','Nocturno']

lvacia=[]

reservacion=[]

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

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

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

sleep(2)

while True:

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

print("1:RESERVACIONES")

print("2:REPORTES")

print("3:REGISTRAR UNA SALA")

print("4:REGISTRAR UN CLIENTE")

print("5:SALIR")

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

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

if opcion == 1:

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

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

opcion\_reservacion=int(input('¿QUE OPCION DESEA UTILIZAR? '))

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

if opcion\_reservacion==1:

reserva =[]

print("\*\*\*\*\*\*\*PARA RESERVAR UNA SALA TIENE QUE ESTAR REGISTRADO EL USUARIO\*\*\*\*\*\*\*")

nom\_registro = input("Ingrese su Nombre: ")

if nom\_registro in usuario1:

num\_registro=int(input("Ingrese su numero de registro: "))

fecha\_str=input("Fecha del evento dd/mm/aaaa: ")

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

fecha\_actual=datetime.date.today()

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

fecha1.append(fecha\_str)

if fecha<=dia\_siguiente:

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

else:

print('NO SE PUEDE CREAR UNA RESERVACION EN UN APARTADO EXISTENTE')

turnos=["Matutino", "Vespertino","Nocturno"]

print(turnos)

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

turno1.append(turnos\_elegir)

nom\_evento= input("Cual es el nombre de tu evento:")

if sala\_registro == None:

print('NO HAY SALAS DISPONIBLES')

continue

else:

print('LAS SALAS DISPONIBLES SON: ')

print(sala\_registro)

sala\_elegir=int(input('¿Que numeo de sala desea?: '))

sala1.append(sala\_elegir)

if fecha\_str in fecha1:

if turnos\_elegir in turno1:

if sala\_elegir in sala1:

diccionario\_notas[fecha\_str]=[(sala\_elegir),(nom\_registro),(nom\_evento),(turnos\_elegir)]

diccionario\_cambiar[fecha\_str]=(nom\_evento)

print(diccionario\_notas)

else:

print('ESTA FECHA Y TURNO NO ESTAN DISPONIBLES EN LA SALA DESEADA, FAVOR DE INTENTAR NUEVAMETE')

continue

else:

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

print("EL USUARIO TIENE QUE ESTAR REGISTRADO")

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

elif opcion\_reservacion==2:

print(list(diccionario\_cambiar.items()))

cambiar=input('Fecha a cambiar en formato (dd/mm/aaaa): ')

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

folio = input("Ingrese Reservacion: ")

if folio == "":

print("ERROR, Debe Ingresar Reservacion.")

continue

if folio not in reserva.keys():

print( "No Existe Una Reservacion Con ese Folio")

break

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

while True:

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

if nuevo\_nombre == "":

print("Debe Ingresar Un Nombre Del Evento.")

continue

nombre\_antiguo = reserva[folio][3]

reserva[folio][3] = nuevo\_nombre

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

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

break

elif opcion\_reservacion ==3:

if not sala:

print("No Existe Ninguna Sala.Por Favor Registra Una.")

break

while True:

fecha\_consulta = input('¿Qué fecha desea consultar?: ')

if fecha\_consulta == "":

print("Debe Ingresar una Fecha.")

continue

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

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

continue

try:

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

except:

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

continue

lista\_encontrados = []

reservaciones\_posibles = []

for clave,datos in reserva.items():

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

if fecha == fecha\_consulta:

lista\_encontrados.append((sala, turno))

reservas\_encontradas = set(lista\_encontrados)

for sala in sala.keys():

for turno in turnos\_elegir.values():

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

combinaciones\_reservas\_posibles = set((reservaciones\_posibles))

salas\_turnos\_posibles = set((combinaciones\_reservas\_posibles - reservas\_encontradas))

print(f"Las salas disponibles {fecha\_consulta.strftime('%d/%m/%Y')} \n")

print('ID SALA TURNO')

for sala in salas\_turnos\_posibles:

print(f'{sala}')

break

elif opcion\_reservacion == 4:

print("Saliendo........")

break

elif opcion == 2:

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

print('MENU DE OPCIONES PARA EL REPORTE')

print('1.REPORTE EN PANTALLA PARA RESERVACIONES DE UNA FECHA')

print('2.EXPORTAR REPORTE TABULAR EN EXCEL')

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

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

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

if opcion\_reporte==1:

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

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

if fecha\_reporte in diccionario\_notas:

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

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

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

print( " SALA CLIENTE EVENTO TURNO ")

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

print(diccionario\_notas.get(fecha\_reporte))

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

else:

print("NO HAY RESERVACIONES PARA ESE DIA" )

if opcion\_reporte==2:

print("EXPORTAR REPORTE TABULAR EN EXCEL")

print("EXPORTAR REPORTE TABULAR EN EXCEL")

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

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

if fecha\_reporte in diccionario\_notas:

libro = openpyxl.Workbook()

hoja = libro["Sheet"]

hoja.title = "Registro"

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

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

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

hoja["B2"].value=(nom\_registro)

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

hoja["C2"].value=(nom\_evento)

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

hoja["D2"].value=(turnos\_elegir)

libro.save("MiExcelDesdePython.xlsx")

else:

print("NO HAY RESERVACIONES PARA ESE DIA")

elif opcion == 3:

import random

cupo=[]

clave\_unica={}

num\_sala=int(input("Coloque el numero de la sala: "))

nombre\_sala=input("Cual es el nombre de tu sala?: ")

sala\_registro[(num\_sala)]=(nombre\_sala)

cupo1=int(input("De cuanto es el cupo de tu sala?: "))

cupo.append(cupo1)

print("La clave unica de la sala es: ", max(clave\_unica.keys(),default=0)+1)

print(sala\_registro)

for sala in range(0,3):

lvacia.append(num\_sala)

lvacia.append(nombre\_sala)

lvacia.append(turno[sala])

print(lvacia)

elif opcion == 4:

import random

clave=[]

nombre=input(f"Coloque su nombre de usuario: ")

clave\_unica=(print(f"La clave unica del cliente es: {random.randrange(20)}"))

usuario1.append(nombre)

clave.append(clave\_unica)

elif opcion ==5:

import csv

datos\_grabar=(diccionario\_notas)

with open("Reporte.csv","w", newline="") as archivo:

grabador=csv.writer(archivo)

grabador.writerow(('FECHA','SALA','NOMUSUARIO','NOMEVENTO','TURNO'))

grabador.writerows([(clave, datos[0], datos[1], datos[2], datos[3]) for clave, datos in datos\_grabar.items()])

print('ARCHIVO CSV CREADO CORRECTAMENTE')

import sys

import sqlite3

from sqlite3 import Error

try:

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

mi\_cursor = conn.cursor()

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

print("Tabla creada exitosamente")

except Error as e:

print(e)

except:

print(f"Se produjo el siguiente error: {sys.exc\_info()[0]}")

try:

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

mi\_cursor = conn.cursor()

mi\_cursor.execute("INSERT INTO proyecto VALUES(1,'FECHA')")

mi\_cursor.execute("INSERT INTO proyecto VALUES(2,'SALA')")

mi\_cursor.execute("INSERT INTO proyecto VALUES(3,'USUARIO')")

mi\_cursor.execute("INSERT INTO proyecto VALUES(4,'EVENTO')")

mi\_cursor.execute("INSERT INTO proyecto VALUES(5,'TURNO')")

print("Registro agregado exitosamente")

except Error as e:

print(e)

except:

print(f"Se produjo el siguiente error: {sys.exc\_info()[0]}")

finally:

if (conn):

conn.close()

GITHUB

<https://github.com/XimenaLongoria01/EVIDENCIA3.git>

XIMENA ABIGAIL TORRES LONGORIA

En este código primero solicitamos la fecha en la deseamos tabular en Excel , y si en esa fecha hay registro creamos la hoja de excel ,utilizando diccionario damos el nombre del libro de Excel , damos a cada celda asignada un valor y en seguida el dato que fue registrado. Damos acceso al excel con un libro.save

PARA CVS

Para el archivo csv dimos a guardar en un diccionario los datos que fueron almacenados en el código ejecutado, poniendo los datos en un archivo csv conforme al orden más legible posible