Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas Distribuidos

Equipo

Díaz Ramírez Yoeli

Jiménez Ramírez Lizeth

Mayo Ruiz Jesús Daniel

Proyecto final:

14 de Diciembre de 2021

Semestre 2022-1

Ing. Guadalupe Lizeth Parrales Romay

Aplicación:

Durante el presente trabajo se desarrollara, una tienda de boletos para conciertos, donde se podrá apreciar los datos que se piden comúnmente en una tienda en línea, como nombre completo, tarjeta, dirección, edad, etc; consideramos que es muy importante que el cliente al generar un boleto,contenga los datos más importantes por ejemplo el monto, asiento, lugar,artista, etc.

Justificación del proyecto:

Elegimos un sistema de venta de boletos de concierto ya que nos pareció una idea interesante, creemos que este tipo de sistema es algo que actualmente se requiere mucho. Cuando compras un boleto como cliente en realidad no tienes idea de todo el proceso que conlleva y los tipos de "obstáculos" que se puede encontrar en el camino tu solicitud, por ello es bueno tener un buen sistema de soporte.

Justificación de herramientas:

Como equipo nos pareció pertinente utilizar el lenguaje de programación Python, ya que es un lenguaje de alto nivel, actualizado y que se suele utilizar mucho en la industria. Además utilizamos Postgres como sistema de gestión de bases de datos debido a que se tenía previa experiencia y no es difícil de manejar. Como de editores de texto se utilizó Visual studio code y Sublime text, por la facilidad que manejan para hacer la conexión con la base de datos, de editar en el lenguaje que elegimos y de hacer a la vista más agradable todo el código, por lo que consideramos que ambos editores son prácticos.

Diseño del sistema:

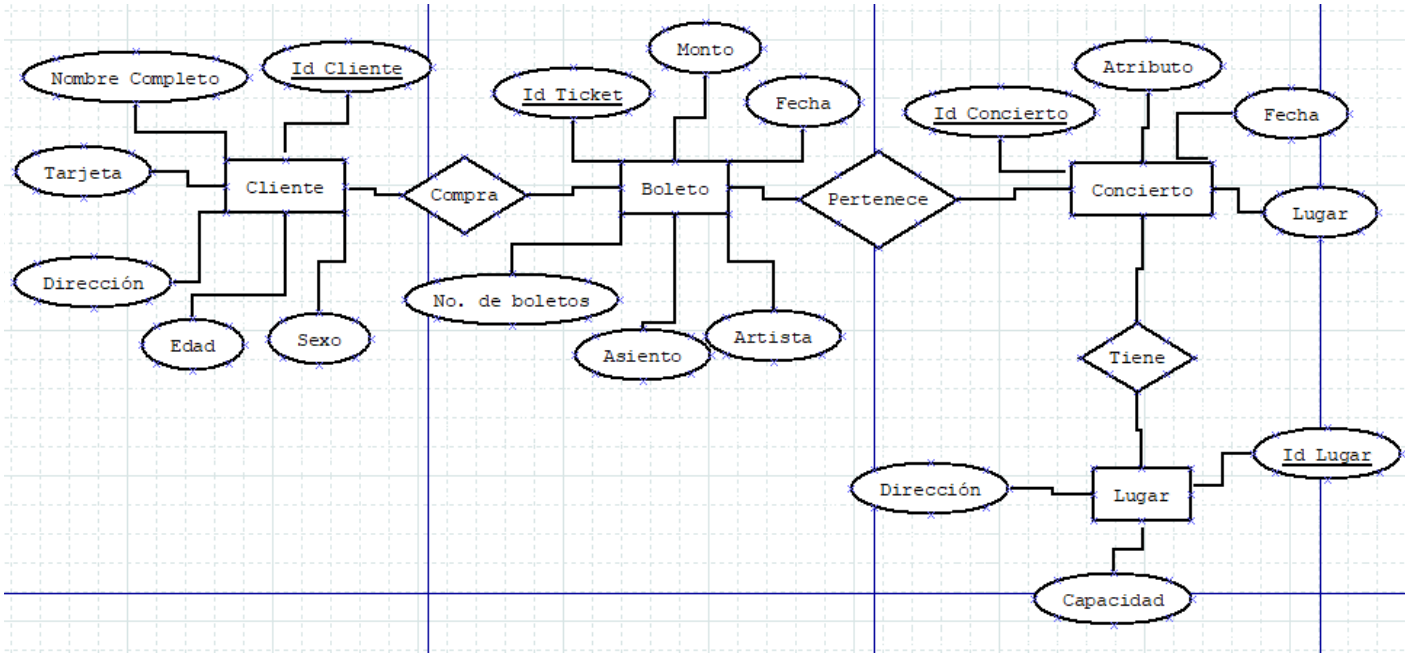
Tomamos como punto de referencia sistemas de empresas grandes y reconocidas que utilizan sistema en línea de venta de boletos, para así guiarnos como manejan su información ya que consideramos que tienen experiencia en eso y podrían ser una buena referencia.

Para llevar a cabo nuestro proyecto empezamos por plantear la base de datos;
lo primero en lo que nos enfocamos fue en seleccionar el lenguaje que se utilizará, para implementar la base de datos y lo más importante, que haría y tendría nuestro programa, en esta última parte fue importante ponernos desde una perspectiva de cliente para saber qué es lo que quería, qué es lo que necesitas para comprar un boleto de un concierto. Definimos opciones de registrarse como usuario, consultar un boleto, concierto o usuario, comprar boleto y cerrar las opciones, de los cuales se fueron desarrollando funciones para llevar a cabo lo que se solicitaba dependiendo del caso.

En nuestro caso decidimos utilizar el id de usuario, como log para poder acceder a la función de compra, en caso de no tener uno se puede crear en el menú principal usando la opción de registro, la cual corresponde al número 1.

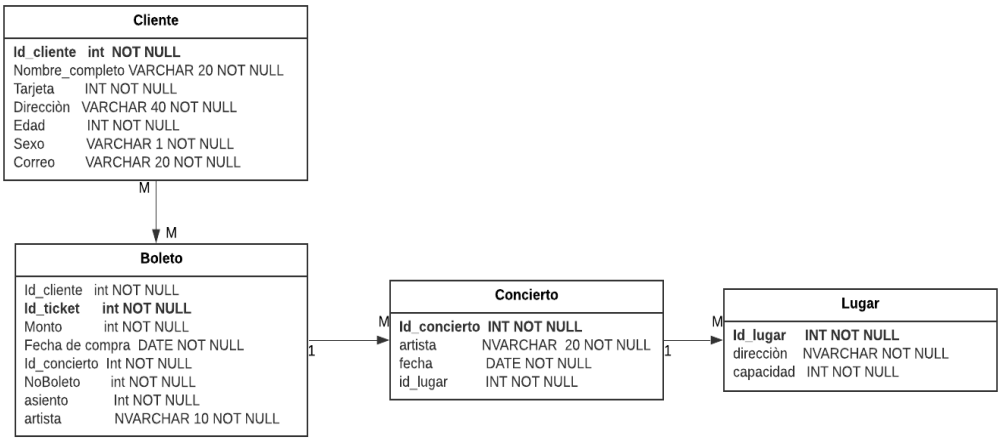
Además elegimos Ordenamiento por marcas de tiempo ya que cada operación en una transacción es validada cuando es llevada a cabo y si la operación no puede ser validada, la transacción es abortada.

Diagrama Entidad-Relación:



UML:

Para desarrollar el diagrama UML fue necesario segmentar cada una de las tablas para que se pueda quitar la redundancia, además se puede apreciar la correspondencia de cada una de las tablas, con las que nos guiamos para poder programar la base de datos, por último se incorporaron los rubros necesarios para poder hacer la base lo más completo y real posible, debido a que nos basamos en la boletera de ticketmaster.



Desarrollo:

En la imagen podemos apreciar la creación de la tabla cliente, donde se usó el id_cliente como identificador de la tabla, además en dicha tabla se integró el nombre, tarjeta, dirección, edad, sexo y correo electronico, utilizado una llave primaria para poder acceder a la tabla.

```
create database Tienda_Boletos

Use Tienda_Boletos;
Create database  Tienda_Boletos
/*TABLA 1*/
CREATE TABLE CLIENTE(
id_cliente INT IDENTITY (1,1) NOT NULL,
nombre NVARCHAR(100) NOT NULL,
tarjeta INT NOT NULL,
direccion NVARCHAR(40) NOT NULL,
edad INT NOT NULL,
sexo CHAR NOT NULL,
correo NVARCHAR(30) NOT NULL,
CONSTRAINT CLIENT_PK PRIMARY KEY (id_cliente),
)
```

La segunda tabla en ser creada en la base de datos, fue lugar debido a que esta era la única tabla que no requería una llave foránea, esta tabla incluye la llave primaria del id_lugar, además de utilizar la dirección y capacidad del lugar donde se conformará el concierto.

```
/*TABLA 2*/
Create TABLE Lugar(
id_lugar INT IDENTITY (1,1) NOT NULL,
direccion NVARCHAR(40) NOT NULL,
capacidad INT NOT NULL,
CONSTRAINT Lugar_PK PRIMARY KEY (id_lugar)
)
```

En la siguiente tabla CONCIERTO tenemos el identificador id_concierto, además el artista, fecha del concierto y la llave foránea id_lugar.

```
/*TABLA 3*/
Create TABLE CONCIERTO(
id_concierto INT IDENTITY (1,1) NOT NULL,
artista NVARCHAR(40) NOT NULL,
fecha DATE NOT NULL,
id_lugar INT NOT NULL,
CONSTRAINT Concierto_PK PRIMARY KEY (id_concierto),
CONSTRAINT LUGAR_FK FOREIGN KEY (id_lugar) REFERENCES Lugar (id_lugar)
)
```

En la tabla 4 BOLETO, está el id_tiket como llave primaria y se agregó el monto, fecha, número de boleto, asiento, artista y id_tiket con id_cliente como llaves foráneas, esta tabla tiene relevancia debido a que el cliente puede generar un nuevo boleto al realizar una compra.

```
/*TABLA 4*/
CREATE TABLE BOLETO(
id_tiket INT IDENTITY (1,1) NOT NULL,
monto INT NOT NULL,
fecha DATE NOT NULL,
no_boleto INT NOT NULL,
asiento INT NOT NULL,
artista NVARCHAR(15) NOT NULL,
id_cliente INT NOT NULL,
id_concierto INT NOT NULL,
CONSTRAINT Tirket_PK PRIMARY KEY (id_tiket),
CONSTRAINT CLIENTE_FK FOREIGN KEY (id_cliente) REFERENCES CLIENTE (id_cliente),
CONSTRAINT Concierto_FK FOREIGN KEY (id_concierto) REFERENCES CONCIERTO (id_concierto),
```

Aquí se agregó información para cada variable de la tabla CLIENTE.

```
)
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Juan Daniel Perez Prado', 12458752,'enrrique segobiano m35', 20,'M','jesusdmr87@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Enrique Mayo Ruiz', 25486592,'ecatepec mz 35 calle benito', 26,'M','enri@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Julia camacho rivera', 52496520,'naucalpan edo mex 55', 19,'F','camacho@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Abigail Torres Chavez', 68524750,'tecamac bosques3 ojo de adua', 28,'F','silvartorres@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('erick Doninguez Hernandez', 35254785,'el rosario planetas36', 25,'M','dominero@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Monserrat Resendiz Diaz', 25478410,'Benito juerez palonas', 23,'F','monseDiaz@gmail.com' )
Insert into CLIENTE (nombre,tarjeta,direccion,edad,sexo,correo)values ('Veronica Delgado Rodriguez', 35214785,'coyoacan mz 35', 23,'F','veroDelga@gmail.com' )
```

Se agregó información para la tabla Lugar.

```
Insert into Lugar (direccion,capacidad) values ('paseo de la reforma',500)
Insert into Lugar (direccion,capacidad) values ('Santa Fe',350)
Insert into Lugar (direccion,capacidad) values ('Naucalpan',100)
Insert into Lugar (direccion,capacidad) values ('Ecatepec',150)
Insert into Lugar (direccion,capacidad) values ('Lomas verdes',410)
Insert into Lugar (direccion,capacidad) values ('Coyoacan',50)
Insert into Lugar (direccion,capacidad) values ('Coapa',100)
```

Se agregó información a la tabla CONCIERTO.

```
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Maluma','2021/12/26',1)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Metallica','2022/05/26',2)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Matisse','2022/04/15',1)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('CNCO','2022/06/5',3)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Camila','2022/04/16',4)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Reik','2022/04/12',5)
Insert into CONCIERTO(artista,fecha,id_lugar) values ('Morat','2022/03/06',6)
```

Se agregó información a la tabla BOLETO.

```
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (500,'2021/12/25',5,10,'CNCO',3,4)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (1100,'2022/01/10',6,20,'Metallica',1,2)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (250,'2022/02/01',5,10,'Morat',4,7)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (500,'2022/03/06',7,12,'Reik',2,6)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (2500,'2022/02/15',9,23,'Camila',5,5)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (1620,'2022/03/20',10,12,'Maluma',6,1)
Insert into BOLETO(monto,fecha,no_boleto,asiento,artista,id_cliente,id_concierto)values (2100,'2022/04/12',11,25,'Matisse',7,3)
```

Para seleccionar las tablas que se requieran observar:

```
Select * from CLIENTE
Select * from Lugar
Select * from CONCIERTO
Select * from BOLETO
--Drop table BOLETO
--Borrar dato de la base de datos
--DELETE FROM CLIENTE WHERE id_cliente = 5
```

Aquí se puede observar el contenido de las tablas Cliente con su información.

	id_cliente	nombre	tarjeta	direccion	edad	sexo	correo
1	1	Juan Daniel Perez Prado	12458752	enrique segobiano m35	20	M	jesusdmr87@gmail.com
2	2	Enrique Mayo Ruiz	25486592	ecatepec mz 35 calle benito	26	M	enri@gmail.com
3	3	Julia camacho rivera	52496520	naucalpan edo mex 55	19	F	camacho@gmail.com
4	4	Abigail Torres Chavez	68524750	tecamac bosques3 ojo de adua	28	F	silvartorres@gmail.com
5	6	erick Doninguez Hernandez	35254785	el rosario planetas36	25	M	dominero@gmail.com
6	7	Monserrat Resendiz Diaz	25478410	Benito juerez palonas	23	F	monseDiaz@gmail.com
7	8	Veronica Delgado Rodriguez	35214785	coyoacan mz 35	23	F	veroDelga@gmail.com

Tabla de Lugar con su información.

	id_lugar	direccion	capacidad
1	1	paseo de la reforma	500
2	2	Santa Fe	350
3	3	Naucalpan	100
4	4	Ecatepec	150
5	5	Lomas verdes	410
6	6	Coyoacan	50
7	7	Coapa	100

Tabla de concierto con los rubros establecidos.

	id_concierto	artista	fecha	id_lugar
1	1	Maluma	2021-12-26	1
2	2	Metallica	2022-05-26	2
3	3	Matisse	2022-04-15	1
4	4	CNCO	2022-06-05	3
5	5	Camila	2022-04-16	4
6	6	Reik	2022-04-12	5
7	7	Morat	2022-03-06	6

Tabla de boletos con su información.

	id_tiket	monto	fecha	no_boleto	asiento	artista	id_cliente	id_concierto
1	1	500	2021-12-25	5	10	CNCO	3	4
2	2	1100	2022-01-10	6	20	Metallica	1	2
3	3	250	2022-02-01	5	10	Morat	4	7
4	4	500	2022-03-06	7	12	Reik	2	6
5	6	1620	2022-03-20	10	12	Maluma	6	1
6	7	2100	2022-04-12	11	25	Matisse	7	3

Creación del menú que el cliente podrá observar en pantalla, con las diferentes opciones que podrá llevar a cabo.

```
#Creación menu
print('HOLA BIEMVENIDO A BOLETILANDIA')
print('Opciones de la pagina')
print('Opción 1:Registrarse' )
print('Opción 2:Consiltar (Boleto,conciertos,usuarios)')
print('Opción 3:Comprar (solo si ya estas registrado)')
print('Opción 4:Cerrar')
seleccion=int( input('Selecciona con numero la opción que deseas: '))
menu(seleccion)
```

Dependiendo del número de opción elegida se desplegará la información, si se elige un número de opción que no está se arrojará el mensaje "opción inválida".

```
def menu(a):
    if a==1:
        print('opción1')
        registro1()
    elif a==2:
        print('opcion2')
        menu_consulta()
    elif a==3:
        print('opción3')
    elif a==4:
        print('opción4')
        exit()
    else:
        print('opción invalida')
    return(a)
```

```
def menu_consulta():
    print("Opción 1:Boleto")
    print("Opción 2:Conciertos Disponibles")
    print("Opción 3:Datos del usuario")
    print("Opción 4:Volver al menu anterior")
    sel=int( input('Selecciona con numero la opción que deseas: '))
    if sel==1:
        con_boleto()
    if sel==2:
        print('opción2')
        con_concierto()
    if sel==3:
        print('opción2')
        con_usuario()
    if sel==4:
        print('-------------------------------')
        print('Opción 1:Registrarse' )
        print('Opción 2:Consiltar (Boleto,conciertos,usuarios)')
        print('Opción 3:Comprar (solo si ya estas registrado)')
        print('Opción 4:Cerrar')
        seleccion=int( input('Selecciona con numero la opción que deseas: '))
        menu(seleccion)
    return()
```

Función que despliega la información del concierto la cual despliega la información en consola utilizando lo guardado en la base de datos.

```
def con_concierto():
    print("Id conciert__________artista_____________________fecha_________________dirección")
    cursor.execute('select id_concierto,artista,fecha,direccion from concierto B,lugar V where B.id_lugar = V.id_lugar')
    consulta=cursor.fetchall()
    for fila in consulta:
        print(fila)
```

En esta función se pide al usuario el id del ticket y se despliega la información de ese boleto, como la es el monto, el asiento y el artista que se eligió, todo extraído de la base de datos.

```
def con_boleto():
    tiket= input('Proporciona id del tiket: ')
    cursor.execute('select id_tiket from boleto where id_tiket ='+tiket)
    consulta=cursor.fetchone()
    print('id:',consulta)
    cursor.execute('select monto from boleto where id_tiket ='+tiket)
    consulta=cursor.fetchone()
    print('monto:',consulta)
    cursor.execute('select asiento from boleto where id_tiket ='+tiket)
    consulta=cursor.fetchone()
    print('asiento:',consulta)
    cursor.execute('select artista from boleto where id_tiket ='+tiket)
    consulta=cursor.fetchone()
    print('artista:',consulta)
    cursor.execute('select id_concierto from boleto where id_tiket ='+tiket)
    consulta=cursor.fetchone()
    print('Id concierto:',consulta)
    return()
```

Esta función es para desplegar información de un cliente, de igual manera se extrae de la base de datos, y se requiere el id del cliente.

```
def con_usuario():
    usuario= input('Proporciona id del cliente: ')
    cursor.execute('select id_cliente from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('id cliente:',consulta)
    cursor.execute('select nombre from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('nombre:',consulta)
    cursor.execute('select tarjeta from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('tarjeta:',consulta)
    cursor.execute('select direccion from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('direccion:',consulta)
    cursor.execute('select edad from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('edad:',consulta)
    cursor.execute('select sexo from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('sexo:',consulta)
    cursor.execute('select correo from cliente where id_cliente ='+usuario)
    consulta=cursor.fetchone()
    print('correo:',consulta)
    return
```

La siguiente función llamada registro1, es para que el usuario pueda llevar a cabo su registro en el sistema. En primera instancia se requiere el nombre completo, despues el numero de tarjeta con el que hará su compra, su dirección, su sexo y su correo en electronico, en cada opción se manejo excepciones en dado caso de que lo ingresado fuera incorrecto.

```
def registro1():
    ID=[]
    tiempos=[]
    decision = True
    transaccion = True
    validación1 = False
    validación2 = False
    validación3 = False
    validación4 = False
    validación5 = False

    while transaccion == True:
        print("\n¡Bienvenido! Para llevar a cabo tu registro es necesario que proporciones algunos datos. Co
        id_transaction = randint(1000,9999)
        while validación1 == False:
            nombre = input("\nIngresa tu nombre completo:")
            if(any(chr.isdigit() for chr in nombre) == True):
                print("Nombre inválido, intente de nuevo.")
            else:
                validación1 = True

        while validación2 == False:
            tarjeta = int(input("\nIngresa el número de tu tarjeta:"))
            if(len(str(tarjeta))!=16):
                print("Número de tarjeta inválido, intente de nuevo. Debe ser de 16 dígitos")
            else:
                validación2 = True

        dirección = input("\nIngresa la dirección de tu domicilio:")

        while validación3 == False:
            edad = int(input("\nIngresa tu edad:"))
            if ((edad<18) or (edad>99)):
                print("Eres menor de edad o tu edad está fuera del rango. Vuelve a intentarlo")
            else:
                validación3 = True

        while validación4 == False:
            sexo = input("\nIngresa tu sexo (ATENCIÓN: Sólo coloca F o M):")
            if((len(sexo)>1) or (any(chr.isdigit() for chr in sexo) == True) or (any((cr == 'F' or cr == 'M'
                print("Respuesta no válida, sigue las indicaciones.")
            else:
                validación4 = True

        while validación5 == False:
            correo = input("\nIngresa tu correo electrónico:")
            if(any(c == '@' for c in correo)==False):
                print("Correo no válido.")
            else:
                validación5 = True
```

En esta sección se muestra en pantalla toda la información que se había solicitado previamente y la opción de finalizar el registro. También muestra la opción de volver al menú principal.

```
time_transaction = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
print("El ID de tu registro es: ",id_transaction)
ID.append(id_transaction)
tiempos.append(time_transaction)
print("Los datos que has proporcionado son:\n")
print("\t\t|ID\t\t|",id_transaction)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Nombre\t\t|", nombre)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Tarjeta\t|", tarjeta)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Dirección\t|",dirección)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Edad\t\t|",edad)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Sexo\t\t|",sexo)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Correo\t\t|", correo)
print("\t\t-------------------------------------------------------")

validación1 = False
validación2 = False
validación3 = False
validación4 = False
validación5 = False
registro=input("\n¿Desea finalizar su registro?(1 = SI o 2 = NO):")
if(registro == '1'):
    sql="INSERT INTO CLIENTE (id_cliente, nombre, tarjeta, direccion, edad, sexo, correo) VALUES (%s,%s,%s,%
    val=(id_transaction, nombre, tarjeta, dirección, edad, sexo, correo)
    cur.execute(sql, val)
    conn.commit()

while decision == True:
    print("\n¿Quieres Volver al menu principal?:\n\t1:SI\t2:NO")
    answer = int(input("\nRespuesta:"))
    if(answer == 2):
        decision = False
        transaccion = False
    elif(answer == 1):
        print('Opción 1:Registrarse' )
        print('Opción 2:Consiltar (Boleto,conciertos,usuarios)')
        print('Opción 3:Comprar (solo si a estas registrado)')
        print('Opción 4:Cerrar')
        seleccion=int( input('Selecciona con numero la opción que deseas: '))
        menu(seleccion)
    else:
        print("\nOpción invalida. Try again")
nt("\nLos registros realizados fueron:\n")
nt ("{:<8} {:<7} {:<1}".format('Order','ID','Date'))
 indice in range(0, len(ID)):
 print(indice, "|\t", ID[indice], "|\t", tiempos[indice])
 print("---------------------------------------")
nt("\n¡Gracias por tu registro!\n")
```

La siguiente función llamada Compra, es para que el usuario pueda llevar a cabo la compra de su boleto en el sistema. Para llevar esto a cabo primero se debía comprobar que el usuario estuviera previamente registrado, de lo contrario nunca accedería al sistema de compra. Posteriormente se le pidió al usuario que escogiera el concierto al cual quería asistir ya que las variables de número de boleto, asiento, monto, artista y capacidad del lugar del concierto dependían de esta decisión, además en caso de escoger un concierto que no se tiene contemplado no te permitirá seguir hasta que hayas seleccionado una opción válida.

```
def Compra():
    validación1 = True
    validación2 = True
    validación3 = True
    validación4 = True
    respuesta = True
    respuesta2 = True
    boleto = 0

    print("\n------------------------------***¡Bienvenido!***---------------------------------- \n\tPara co

    #Validar ID del cliente
    while validación1 == True:
        ID_Cliente = input("\nProporciona tu ID del registro:")
        cur.execute('select id_cliente from cliente where id_cliente ='+ID_Cliente)
        consulta1 = cur.fetchone()
        if(consulta1 == None):
            print("El ID no existe, vuelve a intentarlo")
        else:
            print("ID identificado puede continuar")
            validación1 = False

    print("\nLos conciertos disponibles son:")
    print("Id__________artista_____________fecha_____________________dirección___")
    cursor.execute('select id_concierto,artista,fecha,direccion from concierto B,lugar V where B.id_lugar =
    consulta=cursor.fetchall()
    for fila in consulta:
        print(fila)
    #Seleccionar el ID del concierto
    while(validación2 == True):
        ID_Concierto = int(input("\nSelecciona el concierto al que deseas asistir según su número(ID):"))
        if((ID_Concierto>7) or (ID_Concierto<1)):
            print("\n\tOpción no válida intenta de nuevo\n")
        else:
            #Asignando monto y artista en cada ID
            if(ID_Concierto==1):
                Monto=1620
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            elif(ID_Concierto==2):
                Monto=1100
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            elif(ID_Concierto==3):
                Monto=2100
                cur.execute('select capacidad from lugar where id_lugar='+str(1))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            elif(ID_Concierto==4):
                Monto=500
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto-1))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            elif(ID_Concierto==5):
                Monto=2500
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto-1))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            elif(ID_Concierto==6):
                Monto=500
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto-1))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()
            else:
                Monto=250
                cur.execute('select capacidad from lugar where id_lugar='+str(ID_Concierto-1))
                capacidad = cur.fetchone()
                cur.execute('select artista from concierto where id_concierto='+str(ID_Concierto))
                artista = cur.fetchone()

            validación2 = False
```

Posteriormente se realizó la parte de elección del número de asiento por parte del cliente, para ello le dimos a conocer al usuario la capacidad del lugar donde se llevaría a cabo el concierto elegido y los lugares ya ocupados para que pueda elegir otro. De todas maneras se implementó lo necesario para que fuera comparando la elección del cliente, por si su elección fuera de un asiento ocupado, permitiéndole escoger otro.

```
#Generar el número de asiento
print("\nHay un total de", capacidad,"asientos.")
cur.execute('select asiento from boleto where id_concierto='+str(ID_Concierto))
asientos_oc = cur.fetchall()
cantidad = len(asientos_oc)
if (cantidad == 0):
    print("Todos los asientos están disponibles")
else:
    print("\nLos asientos no disponibles son:")
    print(asientos_oc)
#Comprobar asientos ocupados
while(validación3 == True):
    asientos = int(input("\nSelecciona el número de asiento:"))
    res = int(''.join(map(str, capacidad)))
    if((asientos>res) or (asientos<1)):
        print("\tOpción no válida intenta de nuevo")
    else:
        for i in range (0,cantidad):
            res2 = int(''.join(map(str, asientos_oc[i])))
            if(asientos == res2):
                respuesta = False

        if respuesta == False:
            print("Asiento ocupado intenta de nuevo")
            respuesta = True
        elif respuesta == True:
            print("Has registrado tu asiento de manera exitosa")
            validación3 = False
```

A continuación se le asignó un número de boleto al cliente, asignándole uno no registrado anteriormente. Se generó el id del ticket de 4 dígitos de manera aleatoria y se recuperó la fecha en que se realizó dicho registro.

```
#Generar el número de boleto
cur.execute('select no_boleto from boleto where id_concierto='+str(ID_Concierto))
boletos_com = cur.fetchall()
cantidad2 = len(boletos_com)

#Comprobar boletos ocupados
while(validación4 == True):
    for j in range (0,cantidad2):
        res2 = int(''.join(map(str, boletos_com[j])))
        if(boleto == res2):
            respuesta2 = False

    if respuesta2 == False:
        respuesta2 = True
        boleto = boleto + 1
    elif respuesta2 == True:
        boleto = boleto + 1
        validación4 = False


#Generar el número de su ticket
id_ticket = randint(1000,9999)

#Generar la fecha de la compra
fecha = datetime.now().strftime('%Y-%m-%d')
```

A continuación se le presenta al cliente los datos que aparecerán en su boleto, que se fueron asignando durante el código.

```
print("\nLos datos de tu boleto son:\n")
print("\t\t|ID ticket\t\t|",id_ticket)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Monto\t\t\t|", Monto)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Fecha\t\t\t|", fecha)
print("\t\t-------------------------------------------------------")
print("\n\t\t|N° de boleto\t\t|",boleto)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Asiento\t\t|",asientos)
print("\t\t-------------------------------------------------------")
print("\n\t\t|Artista\t\t|",artista)
print("\t\t-------------------------------------------------------")
print("\n\t\t|ID_Cliente\t\t|", ID_Cliente)
print("\t\t-------------------------------------------------------")
print("\n\t\t|ID_Concierto\t\t|", ID_Concierto)
print("\t\t-------------------------------------------------------")
```

Por último, sabemos que el usuario puede arrepentirse de su compra por lo que le damos la elección de guardar o no su registro. Si no desea guardar su registro le damos la opción de regresar al menú principal o de salir del sistema. Si desea llevar a cabo su compra los datos son guardados en la base de datos, por lo que únicamente si está de acuerdo se aplicará la instrucción commit, esto para evitar datos no deseados en nuestra base. Además al principio de la imagen se ven unas asignaciones, esto nos permitirá reiniciar nuestros valores por si se vuelve a ingresar a la función compra.

```
validación1 = True
validación2 = True
validación3 = True
validación4 = True
respuesta = True
respuesta2 = True

compra=input("\n¿Desea finalizar su registro?(1 = SI o 2 = NO):")
if(compra == '1'):
    sql="INSERT INTO boleto (id_tiket, monto, fecha, no_boleto, asiento, artista, id_cliente, id_concier
    val2=(id_ticket, Monto, fecha, boleto, asientos, artista, ID_Cliente, ID_Concierto)
    cur.execute(sql, val2)
    conn.commit()
    boleto = 0
else:
    boleto = 0
    print("\nSi desea regresar al menú principal presiona 1")
    print("Si desea salir del sistema presiona cualquier tecla")
    final=input("Respuesta:")

    if(final == '1'):
        print('-------------------------------')
        print('Opción 1:Registrarse' )
        print('Opción 2:Consiltar (Boleto,conciertos,usuarios)')
        print('Opción 3:Comprar (solo si ya estas registrado)')
        print('Opción 4:Cerrar')
        seleccion=int( input('Selecciona con numero la opción que deseas: '))
        menu(seleccion)
    else:
        exit()

print("\n¡Gracias por tu compra!\n")
```

Para poder evitar la colisión de 2 personas que están comprando al mismo tiempo utilizamos ordenamiento por marcas de tiempo, para con esto hacer un uso eficiente de los recursos, además podemos evitar que 2 o más usuarios compren al mismo tiempo el mismo boleto, debido a que en la imagen anterior donde se revisa que no esté vendido el boleto, por lo que si ya se vendió el boleto no se podrá ocupar de nuevo.

```
time_transaction1 = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
time_transaction2 = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

if(time_transaction1<time_transaction2):
    print("La transacción 1 ocurrió en el tiempo", time_transaction1)
    Compra()
else:
    print("La transacción 2 ocurrió en el tiempo", time_transaction2)
    Compra()
```

El código final cuando minimizamos las funciones queda de la siguiente forma, se puede apreciar un código muy limpio y fácil de interpretar.

```
from random import randint
from datetime import datetime

import psycopg2
#poso1 crear conección


conn = psycopg2.connect(
    host='localhost',
    database='tienda_boletos',
    user ='postgres',
    password = '1234'
    )

print('conección creada')
cursor=conn.cursor()
cur = conn.cursor()
#cursor.execute('select * from lugar\n')
#consulta=cursor.fetchall()
#cursor.execute('select * from lugar where (id_lugar = 3)')
#consulta=cursor.fetchone()
#print(consulta)

> def Compra():···

> def registro1():···

> def con_usuario():···

> def con_boleto():···

> def con_concierto():···

> def menu_consulta():···

> def menu(a):···


#Creación menu
print('HOLA BIEMVENIDO A BOLETILANDIA')
print('Opciones de la pagina')
print('Opción 1:Registrarse' )
print('Opción 2:Consiltar (Boleto,conciertos,usuarios)')
print('Opción 3:Comprar (solo si ya estas registrado)')
print('Opción 4:Cerrar')
seleccion=int( input('Selecciona con numero la opción que deseas: '))
menu(seleccion)



#paso3 final
conn.close()
```

Pruebas funcionamiento:

En la imagen siguiente se puede apreciar el funcionamiento del menú principal, el cual se puede apreciar cada uno de los casos que se desarrollaron en este proyecto final, el primero es el registro, el segundo está la consulta del boleto que se generó, los conciertos disponibles y la información de tu usuario, en tercer opción tenemos la compra de boleto, por último tenemos la opción de cerrar el programa.

```
HOLA BIEMVENIDO A BOLETILANDIA
Opciones de la pagina
Opción 1:Registrarse
Opción 2:Consiltar (Boleto,conciertos,usuarios)
Opción 3:Comprar (solo si ya estas registrado)
Opción 4:Cerrar
Selecciona con numero la opción que deseas:
```

En caso de seleccionar la primera opción, se le solicitará al usuario, su nombre completo, número de su tarjeta, la dirección, edad, sexo y su correo electronico, esto te generara un id automaticamente para meterlo en la base de datos con los sus datos pertinentes, además se debe estar registrado para poder generar un boleto.

```
Selecciona con numero la opción que deseas: 1
opción1

¡Bienvenido! Para llevar a cabo tu registro es necesario que proporciones algunos datos. Comenzemos.

Ingresa tu nombre completo:Daniel

Ingresa el número de tu tarjeta:5698547856985478

Ingresa la dirección de tu domicilio:tacuba

Ingresa tu edad:25

Ingresa tu sexo (ATENCIÓN: Sólo coloca F o M):M

Ingresa tu correo electrónico:juanperez@gmail.com
El ID de tu registro es:  9336
Los datos que has proporcionado son:

                |ID             | 9336
                ------------------------------------------------------

                |Nombre         | Daniel
                ------------------------------------------------------

                |Tarjeta        | 5698547856985478
                ------------------------------------------------------

                |Dirección      | tacuba
                ------------------------------------------------------

                |Edad           | 25
                ------------------------------------------------------

                |Sexo           | M
                ------------------------------------------------------

                |Correo         | juanperez@gmail.com
                ------------------------------------------------------
```

Durante esta imagen se puede apreciar, que en caso de haber seleccionado en el menú principal, la opción número 2, podemos acceder al submenú de consultas, donde se podrá consultar la información de boletos, los conciertos que tenemos disponibles, la información del usuario que metió en el sistema, por último existe la opción de regresar al menú anterior.

```
Selecciona con numero la opción que deseas: 2
opcion2
Opción 1:Boleto
Opción 2:Conciertos Disponibles
Opción 3:Datos del usuario
Opción 4:Volver al menu anterior
Selecciona con numero la opción que deseas:
```

En caso de haber seleccionado en el submenú la opción número 1 (boleto), además te solicita el número de tiket que tienes, para que puedas obtener la información correspondiente, por ejemplo el id, monto, asiento,artista y el id del concierto.

```
Selecciona con numero la opción que deseas: 1
Proporciona id del tiket: 5
id: (5,)
monto: (2500,)
asiento: (23,)
artista: ('Camila         ',)
Id concierto: (5,)
PS C:\Users\chabo\Desktop\pfnal>
```

Al seleccionar la opción 2 en el submenú de consulta, se puede apreciar todos los conciertos que se tienen disponibles en la base de datos, además podemos apreciar que contiene el id, nombre del artista, la fecha del evento, por último se puede apreciar gracias al uso de un join la dirección del evento.

```
Selecciona con numero la opción que deseas: 2
opción2
Id__________artista_____________fecha_____________________dirección___
(1, 'Maluma                ', datetime.date(2021, 12, 26), 'paseo de la reforma                  ')
(2, 'Metallica             ', datetime.date(2022, 5, 26), 'Santa Fe                             ')
(3, 'Matisse               ', datetime.date(2022, 4, 15), 'paseo de la reforma                  ')
(4, 'CNCO                  ', datetime.date(2022, 6, 5), 'Naucalpan                            ')
(5, 'Camila                ', datetime.date(2022, 4, 16), 'Ecatepec                             ')
(6, 'Reik                  ', datetime.date(2022, 4, 12), 'Lomas verdes                         ')
(7, 'Morat                 ', datetime.date(2022, 3, 6), 'Coyoacan                             ')
PS C:\Users\chabo\Desktop\pfnal>
```

En caso de haber seleccionado la opción número 3 se le mostrará la información del usuario, siempre y cuando se le proporcione un número válido del id cliente, con lo que podrá apreciar el nombre, la tarjeta, la dirección, edad, el sexo y por ultimo su correo electronico.

```
Selecciona con numero la opción que deseas: 3
opción2
Proporciona id del cliente: 6
id cliente: (6,)
nombre: ('Monserrat Resendiz Diaz                ',)
tarjeta: (25478410,)
direccion: ('Benito juerez palonas                  ',)
edad: (23,)
sexo: ('F',)
correo: ('monseDiaz@gmail.com     _       ',)
```

Si se seleccionó el último caso para regresar al menú principal, este llama a la función menú para poder seleccionar la opción que desea ingresar.

```
Selecciona con numero la opción que deseas: 4
------------------------------
Opción 1:Registrarse
Opción 2:Consiltar (Boleto,conciertos,usuarios)
Opción 3:Comprar (solo si ya estas registrado)
Opción 4:Cerrar
Selecciona con numero la opción que deseas:
```

En la imagen siguiente podemos apreciar la implementación del caso compra en el menú principal; al seleccionar este campo, podemos apreciar que nos solicita el número de id como login, no obstante se muestra los conciertos que tenemos disponibles en la base de datos, asimismo nos solicita el número de id del concierto que se desea comprar un boleto, después se le muestra al usuario los asientos que ya no se encuentran disponibles y se le solicita que elija el número de asiento que quiere (lugar), en caso de no dar un número válido se le advertirá al usuario que el asiento no está disponible y se le dará una nueva oportunidad de seleccionar el número válido, por último cuando se proporciona un número de asiento valido se le genera el boleto con un número de id aleatorio, donde se le mostrará toda la información del evento y los datos más relevantes para el usuario.

Una vez generado el boleto se le preguntará si desea finalizar y en caso de dar una respuesta afirmativa, se generará el boleto en la base de datos, por el contrario se podrá regresar al menú y no se generará el boleto en la base de datos.

```
Proporciona tu ID del registro:5
ID identificado puede continuar

Los conciertos disponibles son:
Id__________artista_____________fecha_____________________dirección___
(1, 'Maluma                   ', datetime.date(2021, 12, 26), 'paseo de la reforma                  ')
(2, 'Metallica                ', datetime.date(2022, 5, 26), 'Santa Fe                             ')
(3, 'Matisse                  ', datetime.date(2022, 4, 15), 'paseo de la reforma                  ')
(4, 'CNCO                     ', datetime.date(2022, 6, 5), 'Naucalpan                            ')
(5, 'Camila                   ', datetime.date(2022, 4, 16), 'Ecatepec                             ')
(6, 'Reik                     ', datetime.date(2022, 4, 12), 'Lomas verdes                         ')
(7, 'Morat                    ', datetime.date(2022, 3, 6), 'Coyoacan                             ')

Selecciona el concierto al que deseas asistir según su número(ID):2

Hay un total de (350,) asientos.

Los asientos no disponibles son:
[(20,)]

Selecciona el número de asiento:20
Asiento ocupado intenta de nuevo

Selecciona el número de asiento:25
Has registrado tu asiento de manera exitosa

Los datos de tu boleto son:

                |ID ticket              | 3958
                ------------------------------------------------------------

                |Monto                  | 1100
                ------------------------------------------------------------

                |Fecha                  | 2021-12-16
                ------------------------------------------------------------

                |N° de boleto           | 1
                ------------------------------------------------------------

                |Asiento                | 25
                ------------------------------------------------------------

                |Artista                | ('Metallica                ',)
                ------------------------------------------------------------

                |ID_Cliente             | 5
                ------------------------------------------------------------

                |ID_Concierto           | 2
                ------------------------------------------------------------

¿Desea finalizar su registro?(1 = SI o 2 = NO):
```

Conclusiones:

Díaz Ramírez Yoeli:

Las transacciones están presentes en muchos aspectos de nuestra vida cotidiana y sin duda nos han ayudado de manera inimaginable, ya que podemos hacer compras en línea, pago de facturas, enviar dinero, entre muchas cosas más; todo desde un clic y desde cualquier lugar. Sin embargo, el saber que hay detrás de ellas es algo de lo que nos percatamos al implementar este proyecto, ya que cuando nosotros realizamos una transacción otra persona más lo está haciendo, incluso al mismo tiempo, entonces el sistema podría saturarse. Comprendimos que los modelos de control de concurrencia son de vital importancia para evitar está saturación o colapso de una plataforma y que en lo personal considero que son un poco difícil de implementar. Por último realizar el presente proyecto me aportó muchísimo, ya que no sabía cómo conectarme a una base de datos y modificarla a través de Python, además me sirvió mucho para reforzar mis habilidades en programación ya que consideraba que no era buena programando.

Jiménez Ramírez Lizeth:

En el proyecto realizado pudimos reforzar conocimientos adquiridos durante la materia, fue importante tener un conocimiento previo de las bases de datos ya que es donde se iban a realizar las transacciones, las cuales como sabemos es una forma de interactuar con la base, además aprendimos más acerca de ellas y el cómo funcionan de verdad ya en un sistema, es decir verlo de forma práctica, pudimos notar su importancia y también la importancia de tener un modelo de control de la concurrencia, el cual es un método que nos ayuda a determinar ciertos permisos para realizar determinadas acciones en la base. Pienso que con lo que implementamos nos pudimos dar una idea de como trabajan sistemas que hemos ocupado en varias ocasiones pero que nunca nos habíamos puesto a pensar en cómo trabajan realmente y todo lo que hay detrás para poder darle funcionamiento, ya que podría verse “simple” de alguna forma, pero al hacer realmente la codificación notas todo aquello que es fundamental, por último puedo agregar que la participación en equipo también fue importante dado que en el mundo laboral será un habilidad que tendremos que aplicar y que en nuestro caso no todos lograron desarrollarla.

Mayo Ruiz Jesús Daniel:

Al desarrollar este proyecto se necesito implementar lo visto en clases, en especial los modelos transaccionales, los cuales nos fueron de mucha ayuda para poder plantear el proyecto, asimismo decidimos iniciar por la creación de la base de datos para poder segmentar cada una de las tablas, con la información necesaria para poder realizar transacciones desde el código, al estar creando la base de datos, fue necesario utilizar las llaves primarias y foráneas para poder interconectar las tablas y que no fuera una base redundante, por último le asignamos la correspondencia pertinente a cada una de las tablas. Cuando nos enfocamos en la creación del código decidimos utilizar postgresql para poder comunicarnos de manera sencilla y eficaz con la base de datos, lo cual nos permitió poder dividir el código en diferentes procesos, los cuales fueron: compra, registro, consulta (el cual tenía subprocesos) y salir del programa; En particular los puntos más difíciles de implementar fueron el registro y la compra de boletos, puesto que se implementaron muchos condicionales para evitar que hubiera una colisión de datos, por ejemplo para evitar que 2 usuarios compraran el mismo asiento al mismo tiempo, fue necesario evaluar la hora minuto y segundo para asignar el asiento al que lo haya adquirido primero y al otro pedirle que seleccione otro. Dicho lo anterior podemos concluir que se cumplieron los objetivos planeados por la profesora, aplicando el Ordenamiento por marcas de tiempo, abrir transacción, cerrar transacción y abortar transacción. En lo personal elaborar este Proyecto final me ayudó a

mejorar mis conocimientos de python, manejo de base de datos y la forma de conectar una base de datos con python.

Bibliografía:

V.C. (2007). *Técnicas de control de concurrencia* (1.ª ed., Vol. 1). n/a. http://informatica.uv.es/iiguia/2000/BD2/4_0_BD2Tema4_06.pdf

Instalación y Configuración de Base de Datos (postgres) con Python. (2020, 19 junio). [Vídeo]. YouTube. https://www.youtube.com/watch?v=fh5D1M0rGp0&t=1316s

Enlace del código:

https://drive.google.com/drive/folders/1Wy5ol6YX1YZnvxlhSPpgCRqBty0uKeNJ?usp=sharing