**import** pymysql

*#----------------Ejercicio1----------------*

**class** ConectarBd ():

**def** conectar(self):

**try**:

**global** cur,conn

d=dict()

d[**'host'**]=**'localhost'**

d[**'port'**]=3306

d[**'user'**]=**'root'**

d[**'passwd'**]=**'root'**

d[**'db'**]=**'proyectosocio'**

self.conn= pymysql.connect(\*\*d) *#Con self las transformo en propiedades de la clase conectarBD*

self.cur = self.conn.cursor()

**return** 0

**except** Exception **as** f:

**return** 1 *#Si no conecta retorna 1 para que otro metodo sepa que la base de datos no conecto*

*#--------------------------------------------*

*#----------------Ejercicio2------------------*

**class** Socio ():

**def** \_\_init\_\_(self,idSocio,dni,nombre,apellido):

self.idSocio=idSocio

self.dni=dni

self.nombre=nombre

self.apellido=apellido

*#--------------------------------------------*

*#----------------Ejercicio3------------------*

**class** CDSocio ():

**def** \_\_init\_\_(self): *#Instancio una conexion*

self.conexion = ConectarBd()

a = self.conexion.conectar()

*#print(a)*

**def** alta(self,socioAlta):

consul1=**"insert into socio (dni,nombre,apellido) values({0},{1},{2})"**.format(repr(socioAlta.dni), repr(socioAlta.nombre),repr(socioAlta.apellido))

self.conexion.cur.execute(consul1)

self.conexion.conn.commit()

a = self.conexion.cur.lastrowid

print(**"El socio fue agregado correctamente y su Id es: {0}"**.format(a))

**return** self.conexion.cur.lastrowid *#DEVULVE EL ULTIMO ID, NO UN OBJETO socioAlta!!!*

**def** borrar(self,socioBorrar):

consul2=**"delete from socio where idsocio = %d"** %socioBorrar.idSocio

self.conexion.cur.execute(consul2)

self.conexion.conn.commit()

**if** self.conexion.cur.rowcount == 0:

**return False**

**else**:

**return True**

**def** modificar(self,socioModificar): *#Revisar que pasa cuando no se modifico*

consul3=**'update socio set dni={0}, nombre={1}, apellido={2} where idsocio = {3}'**.format(repr(socioModificar.dni), repr(socioModificar.nombre), repr(socioModificar.apellido), socioModificar.idSocio)

self.conexion.cur.execute(consul3)

self.conexion.conn.commit()

**return** socioModificar

**def** buscarxId(self,idSocio):

consul4=**'select \* from socio where idsocio=%d'** %idSocio

self.conexion.cur.execute(consul4)

socioEnco= self.conexion.cur.fetchone() *#Ojo con este self*

**return** socioEnco

print (socioEnco[0],socioEnco[1],socioEnco[2], socioEnco[3])

**def** todos(self):

consul5 = **"SELECT \* FROM socio"**

self.conexion.cur.execute(consul5) *#Ojo con este self*

socios = self.conexion.cur.fetchall()

**for** i **in** socios:

print (i[0],i[1],i[2],i[3])

*#--------------------------------------------*

*#----------------Ejercicio4------------------*

**def** anterior(self, id):

consul1 = **"SELECT \* FROM socio WHERE idsocio=(SELECT MAX(idSocio) FROM socio WHERE idsocio < {0})"**.format(id)

self.conexion.cur.execute(consul1)

socEnco = self.conexion.cur.fetchone()

**if** self.conexion.cur.rowcount == 0:

print(**"No existen un socio anterior"**)

**else**:

print(**"El socio anterior es: \n""ID:"**, socEnco[0], **"\nDNI:"**, socEnco[1], **"\nNOMBRE:"**, socEnco[2], **"\nAPELLIDO:"**, socEnco[3])

**def** proximo(self, id):

consul2 = **"SELECT \* FROM socio WHERE idsocio=(SELECT MIN(idSocio) FROM socio WHERE idsocio > {0})"**.format(id)

self.conexion.cur.execute(consul2)

socEnco = self.conexion.cur.fetchone()

**if** self.conexion.cur.rowcount == 0:

print(**"No existen un socio posterior"**)

**else**:

print(**"El socio siguiente es: \n""ID:"**, socEnco[0], **"\nDNI:"**, socEnco[1], **"\nNOMBRE:"**, socEnco[2], **"\nAPELLIDO:"**, socEnco[3])

**def** primero(self):

consul3 = **"SELECT \* FROM socio WHERE idsocio=(SELECT MIN(idsocio) FROM socio)"**.format(id)

self.conexion.cur.execute(consul3)

socEnco = self.conexion.cur.fetchone()

**if** self.conexion.cur.rowcount == 0:

print(**"No existen socios"**)

**else**:

print(**"El primer socio es: \n""ID:"**, socEnco[0], **"\nDNI:"**, socEnco[1], **"\nNOMBRE:"**, socEnco[2], **"\nAPELLIDO:"**, socEnco[3])

**def** ultimo(self):

consul4 = **"SELECT \* FROM socio WHERE idsocio=(SELECT MAX(idsocio) FROM socio)"**.format(id)

self.conexion.cur.execute(consul4)

socEnco = self.conexion.cur.fetchone()

**if** self.conexion.cur.rowcount == 0:

print(**"No existen socios"**)

**else**:

print(**"El ultimo socio es: \n""ID:"**, socEnco[0], **"\nDNI:"**, socEnco[1], **"\nNOMBRE:"**, socEnco[2], **"\nAPELLIDO:"**, socEnco[3])

*#----------------------------------------*

*#------------Ejercicio 5 ----------------*

**def** buscaDNI(self, dni):

consul1 = **"SELECT COUNT(\*) FROM socio WHERE dni={0}"**.format(repr(dni))

self.conexion.cur.execute(consul1)

cant = self.conexion.cur.fetchone()

print(**"La cantidad de socios con el DNI {0} es {1}"**.format(dni, cant[0]))

*#----------------------------------------*

capaDatSoc = CDSocio()

socionuevo = Socio(1,**'39370797'**,**'Juan'**,**'Helbert'**)

id = 29

*#capaDatSoc.alta(socionuevo)*

*#capaDatSoc.borrar(socionuevo)*

*#capaDatSoc.modificar(socionuevo)*

*#capaDatSoc.buscarxId(id)*

*#capaDatSoc.todos()*

*#capaDatSoc.anterior(id)*

*#capaDatSoc.proximo(id)*

*#capaDatSoc.primero()*

*#capaDatSoc.ultimo()*

*#capaDatSoc.buscaDNI(122333)*