



Universidad Nacional Autónoma de  
México

Facultad de Ingeniería

División de Ingeniería Mecánica e Industrial

Laboratorio de Cómputo de Ingeniería Mecatrónica (1964)



*Profesor: Miguel Serrano Reyes*

*Semestre 2021-2*

Práctica No. 3

Nombre de la práctica

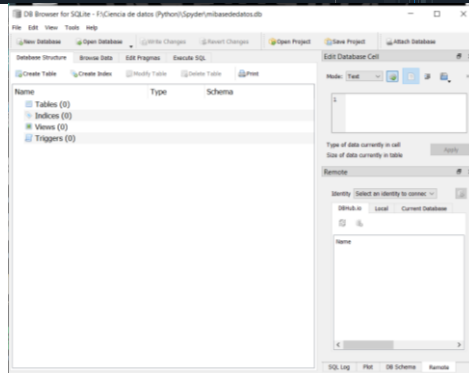
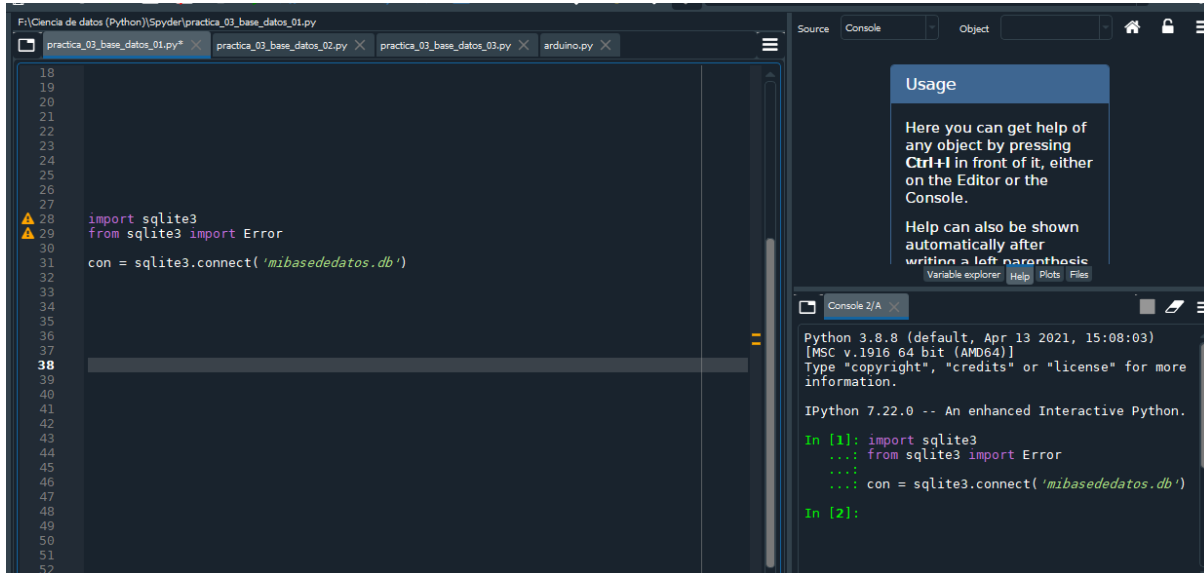
Base de Datos

Nombre del Estudiante:

Dimas Ramírez Luis Daniel

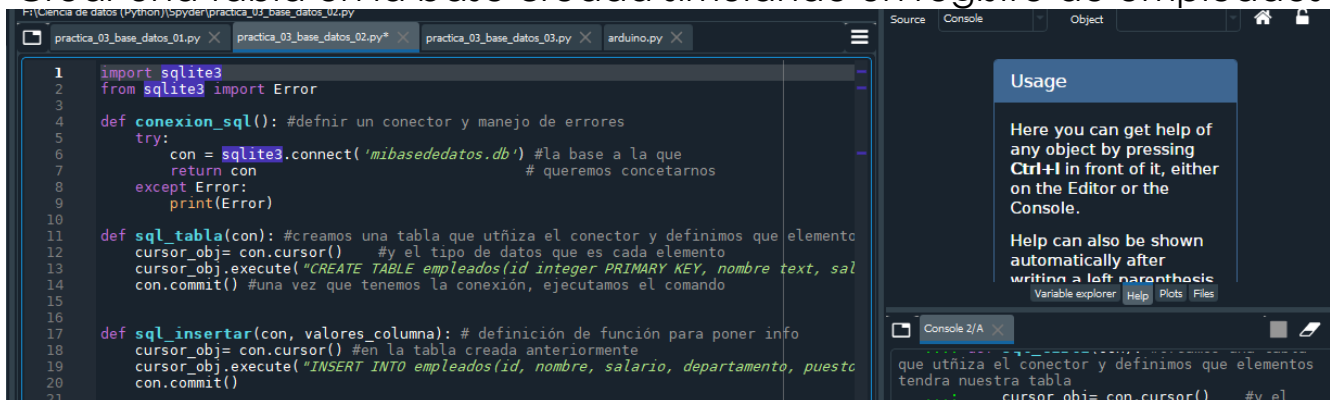
## Actividad 1

### Crear una base de datos



## Actividad 2

### Crear una tabla en la base creada simulando un registro de empleados



DB Browser for SQLite - F:\Ciencia de datos (Python)\Spyder\basededatos.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: empleados

id	nombre	salario	departamento	puesto	fecha_contratacion
Filter	Filter	Filter	Filter	Filter	Filter

## Actividad 3

### Insertar 6 elementos en la tabla

```

1 import sqlite3
2 from sqlite3 import Error
3
4 def conexion_sql(): #definir un conector y manejo de errores
5     try:
6         con = sqlite3.connect('basededatos.db') #la base a la que
7             return con # queremos conectarnos
8     except Error:
9         print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elementos
12     cursor_obj = con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit()) #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna): #definición de función para poner info
17     cursor_obj = con.cursor() #en la tabla creada anteriormente
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit())
20
21 con = conexion_sql() #manda a llamar la función
22
23 # Datos con los que se va a llenar la tabla
24 fila1=(1, 'Luis', 10000, 'Programación', 'Programador Jr', '2010-10-12')
25 fila2=(2, 'Antonio', 40000, 'Programación', 'Programador', '2000-10-12')
26 fila3=(3, 'Davida', 10000, 'Programación', 'Programador Jr', '2010-10-12')
27 fila4=(4, 'Marco', 40000, 'Programación', 'Secretario', '2000-10-12')
28 fila5=(5, 'Alejandra', 1000000, 'Programación', 'Desarrollo de aplicaciones móviles',
29 fila6=(6, 'Daniel', 90000, 'Programación', 'Programador Sr', '2000-10-12')
30
31 sql_insertar(con, fila1)
32 sql_insertar(con, fila2)
33 sql_insertar(con, fila3)
34 sql_insertar(con, fila4)
35 sql_insertar(con, fila5)
36 sql_insertar(con, fila6)

```

Desarrollo

No further documentation available

Variable explorer Help Plots Files

Console 2/A

```

que utiliza el conector y definimos que elementos
tendra nuestra tabla
... cursor_obj = con.cursor() #y el
... cursor_obj.execute("CREATE TABLE
empleados(id integer PRIMARY KEY, nombre text,
salario real, departamento text, puesto text,
fecha_contratacion text)")
... con.commit() #una vez que tenemos la
conexión, ejecutamos el comando
... con = sql_tabla(con)

In [14]: runcell(0, 'F:/Ciencia de datos
(Python)/Spyder/practica_03_base_datos_02.py')

In [15]:

```

DB Browser for SQLite - F:\Ciencia de datos (Python)\Spyder\basededatos.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

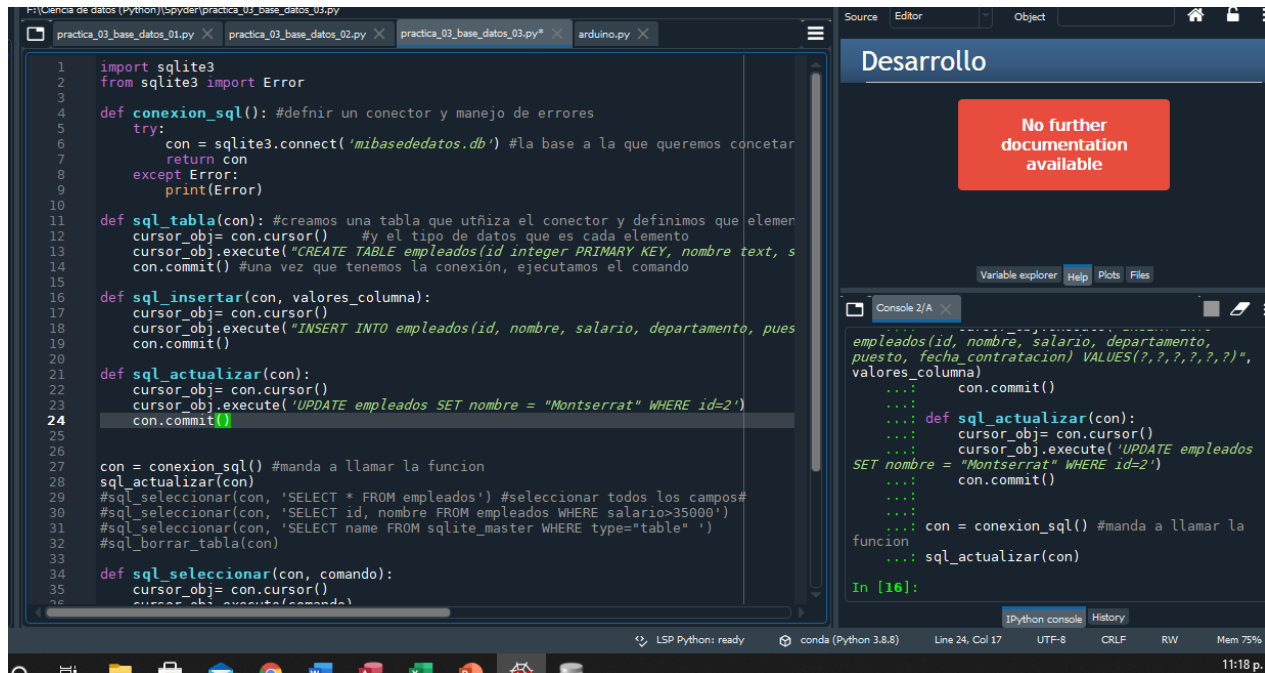
Database Structure Browse Data Edit Pragma Execute SQL

Table: empleados

	id	nombre	salario	departamento	puesto	fecha_contratacion
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Luis	10000.0	Programación	Programador Jr	2010-10-12
2	2	Antonio	40000.0	Programación	Programador	2000-10-12
3	3	Davida	10000.0	Programación	Programador Jr	2010-10-12
4	4	Marco	40000.0	Programación	Secretario	2000-10-12
5	5	Alejandra	1000000.0	Programación	Desarrollo de ...	2010-10-12
6	6	Daniel	90000.0	Programación	Programador Sr	2000-10-12

## Actividad 4

### Modificar un elemento de la tabla mediante algún selector



```
1 import sqlite3
2 from sqlite3 import Error
3
4 def conexion_sql(): #definir un conector y manejo de errores
5     try:
6         con = sqlite3.connect('mibasededatos.db') #la base a la que queremos conectar
7         return con
8     except Error:
9         print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elemento
12     cursor_obj= con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit() #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna):
17     cursor_obj= con.cursor()
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit()
20
21 def sql_actualizar(con):
22     cursor_obj= con.cursor()
23     cursor_obj.execute("UPDATE empleados SET nombre = 'Montserrat' WHERE id=2")
24     con.commit()
25
26
27 con = conexion_sql() #manda a llamar la funcion
28 sql_actualizar(con)
29 #sql_seleccionar(con, 'SELECT * FROM empleados') #seleccionar todos los campos#
30 #sql_seleccionar(con, 'SELECT id, nombre FROM empleados WHERE salario>35000')
31 #sql_seleccionar(con, 'SELECT name FROM sqlite_master WHERE type="table" ')
32 #sql_borrar_tabla(con)
33
34 def sql_seleccionar(con, comando):
35     cursor_obj= con.cursor()
36     cursor_obj.execute(comando)
```

Desarrollo

No further documentation available

Console 2/A

```
empleados(id, nombre, salario, departamento, puesto, fecha_contratacion) VALUES(?, ?, ?, ?, ?, ?)",
...:     con.commit()
...:
...: def sql_actualizar(con):
...:     cursor_obj= con.cursor()
...:     cursor_obj.execute("UPDATE empleados
...:     SET nombre = 'Montserrat' WHERE id=2")
...:     con.commit()
...:
...: con = conexion_sql() #manda a llamar la
...: funcion
...:     sql_actualizar(con)
In [16]:
```

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

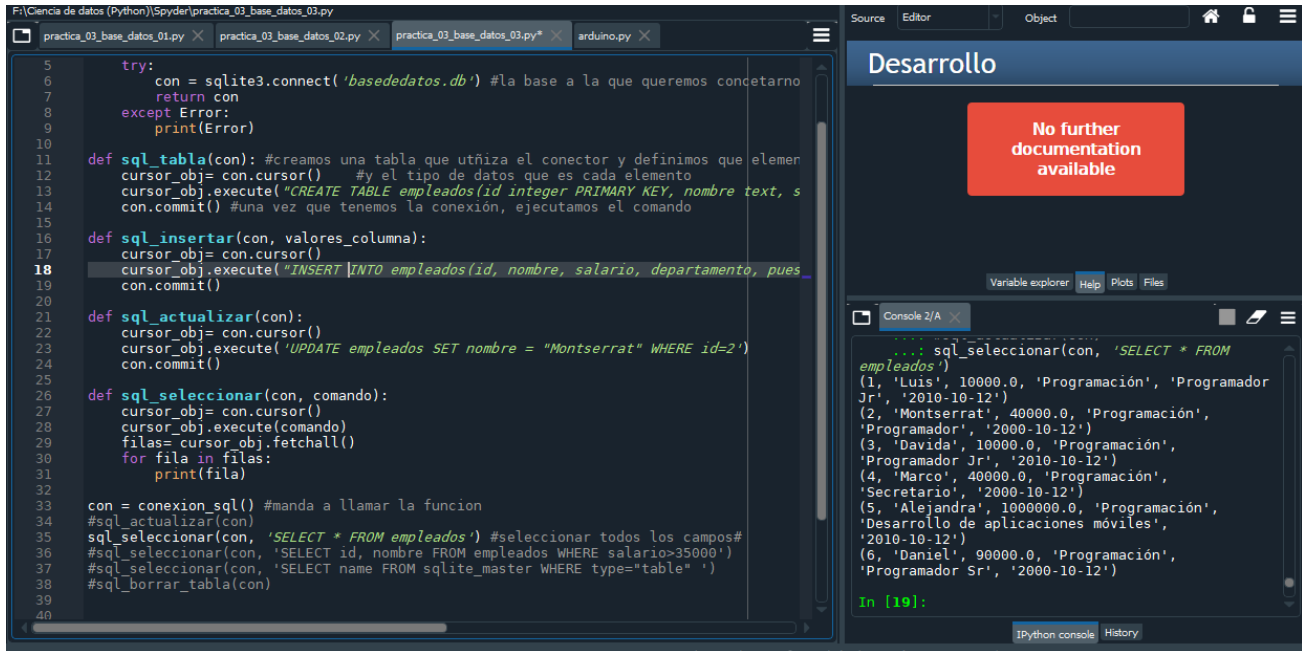
Database Structure Browse Data Edit Pragma Execute SQL

Table: empleados Filter in any column

	id	nombre	salario	departamento	puesto	fecha_contratacion
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Luis	10000.0	Programación	Programador Jr	2010-10-12
2	2	Montserrat	40000.0	Programación	Programador	2000-10-12
3	3	Davida	10000.0	Programación	Programador Jr	2010-10-12
4	4	Marco	40000.0	Programación	Secretario	2000-10-12
5	5	Alejandra	1000000.0	Programación	Desarrollo de aplicaciones móviles	2010-10-12
6	6	Daniel	90000.0	Programación	Programador Sr	2000-10-12

## Actividad 5

### Seleccionar todos los elementos de una tabla



```
5 try:
6     con = sqlite3.connect('basededatos.db') #la base a la que queremos conectarnos
7     return con
8 except Error:
9     print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elemento
12     cursor_obj = con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit() #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna):
17     cursor_obj = con.cursor()
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit()
20
21 def sql_actualizar(con):
22     cursor_obj = con.cursor()
23     cursor_obj.execute("UPDATE empleados SET nombre = 'Montserrat' WHERE id=2")
24     con.commit()
25
26 def sql_seleccionar(con, comando):
27     cursor_obj = con.cursor()
28     cursor_obj.execute(comando)
29     filas = cursor_obj.fetchall()
30     for fila in filas:
31         print(fila)
32
33 con = conexion_sql() #manda a llamar la función
34 #sql_actualizar(con)
35 sql_seleccionar(con, 'SELECT * FROM empleados') #seleccionar todos los campos#
36 #sql_seleccionar(con, 'SELECT id, nombre FROM empleados WHERE salario>35000')
37 #sql_seleccionar(con, 'SELECT name FROM sqlite_master WHERE type="table" ')
38 #sql_borrar_tabla(con)
39
40
```

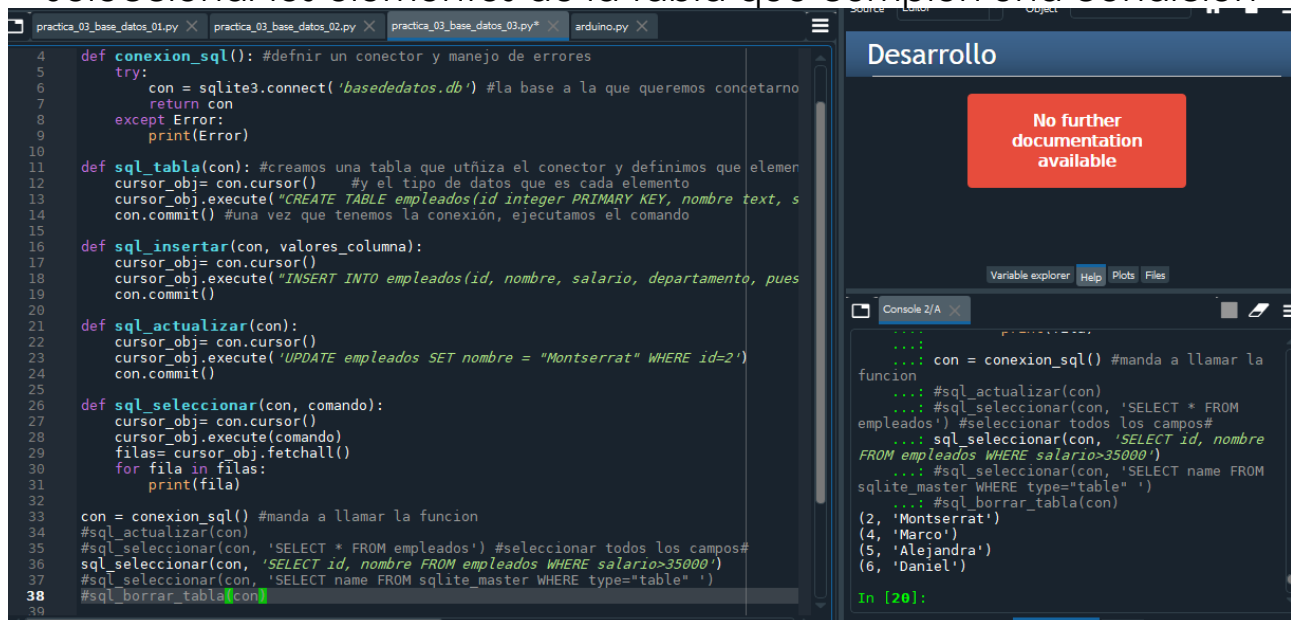
Console 2/A

```
... sql_seleccionar(con, 'SELECT * FROM
empleados')
(1, 'Luis', 10000.0, 'Programación', 'Programador
Jr', '2010-10-12')
(2, 'Montserrat', 40000.0, 'Programación',
'Programador', '2000-10-12')
(3, 'Davida', 10000.0, 'Programación',
'Programador Jr', '2010-10-12')
(4, 'Marco', 40000.0, 'Programación',
'Secretario', '2000-10-12')
(5, 'Alejandra', 100000.0, 'Programación',
'Desarrollo de aplicaciones móviles',
'2010-10-12')
(6, 'Daniel', 90000.0, 'Programación',
'Programador Sr', '2000-10-12')

In [19]:
```

## Actividad 6

### Seleccionar los elementos de la tabla que cumplen una condición



```
4 def conexion_sql(): #definir un conector y manejo de errores
5     try:
6         con = sqlite3.connect('basededatos.db') #la base a la que queremos conectarnos
7         return con
8     except Error:
9         print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elemento
12     cursor_obj = con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit() #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna):
17     cursor_obj = con.cursor()
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit()
20
21 def sql_actualizar(con):
22     cursor_obj = con.cursor()
23     cursor_obj.execute("UPDATE empleados SET nombre = 'Montserrat' WHERE id=2")
24     con.commit()
25
26 def sql_seleccionar(con, comando):
27     cursor_obj = con.cursor()
28     cursor_obj.execute(comando)
29     filas = cursor_obj.fetchall()
30     for fila in filas:
31         print(fila)
32
33 con = conexion_sql() #manda a llamar la función
34 #sql_actualizar(con)
35 #sql_seleccionar(con, 'SELECT * FROM empleados') #seleccionar todos los campos#
36 sql_seleccionar(con, 'SELECT id, nombre FROM empleados WHERE salario>35000')
37 #sql_seleccionar(con, 'SELECT name FROM sqlite_master WHERE type="table" ')
38 #sql_borrar_tabla(con)
39
```

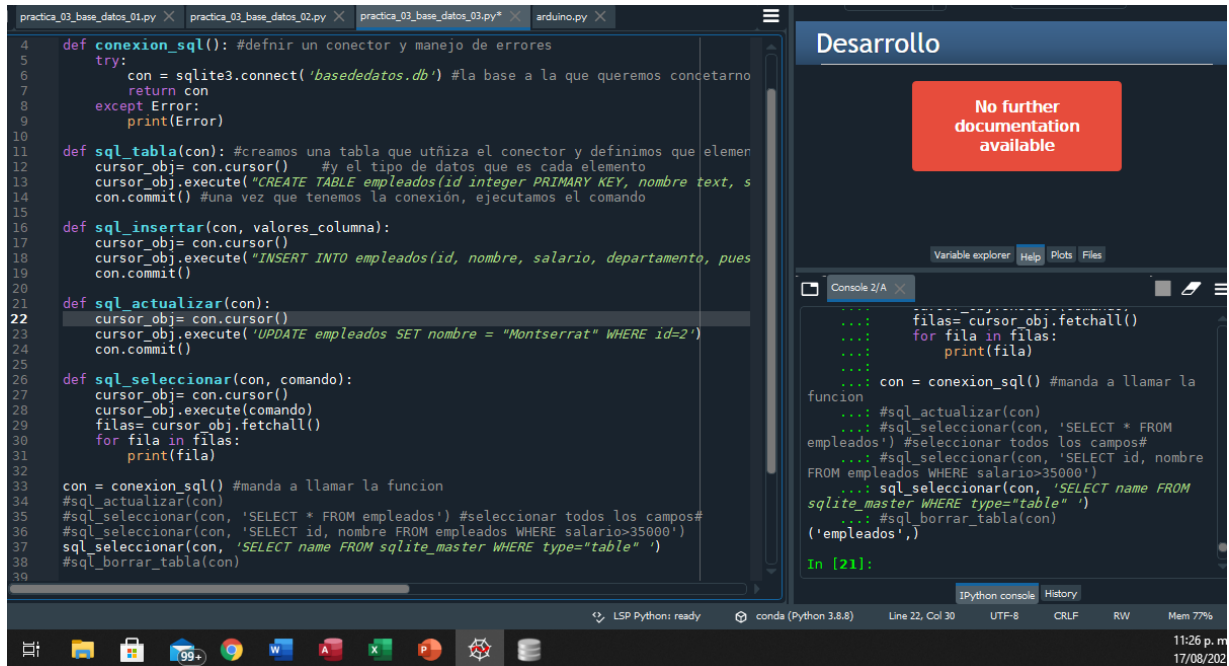
Console 2/A

```
... con = conexion_sql() #manda a llamar la
función
... #sql_actualizar(con)
... #sql_seleccionar(con, 'SELECT * FROM
empleados') #seleccionar todos los campos#
... sql_seleccionar(con, 'SELECT id, nombre
FROM empleados WHERE salario>35000')
... #sql_seleccionar(con, 'SELECT name FROM
sqlite_master WHERE type="table" ')
... #sql_borrar_tabla(con)
(2, 'Montserrat')
(4, 'Marco')
(5, 'Alejandra')
(6, 'Daniel')

In [20]:
```

## Actividad 7

### Obtener todas las tablas que hay en las bases de datos



The screenshot shows a Jupyter Notebook with a dark theme. The code in the cell is as follows:

```
4 def conexion_sql(): #definir un conector y manejo de errores
5     try:
6         con = sqlite3.connect('basededatos.db') #la base a la que queremos conectarnos
7         return con
8     except Error:
9         print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elemento
12     cursor_obj = con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit()) #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna):
17     cursor_obj = con.cursor()
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit()
20
21 def sql_actualizar(con):
22     cursor_obj = con.cursor()
23     cursor_obj.execute("UPDATE empleados SET nombre = 'Montserrat' WHERE id=2")
24     con.commit()
25
26 def sql_seleccionar(con, comando):
27     cursor_obj = con.cursor()
28     cursor_obj.execute(comando)
29     filas = cursor_obj.fetchall()
30     for fila in filas:
31         print(fila)
32
33 con = conexion_sql() #manda a llamar la funcion
34 #sql_actualizar(con)
35 #sql_seleccionar(con, 'SELECT * FROM empleados') #seleccionar todos los campos#
36 #sql_seleccionar(con, 'SELECT id, nombre FROM empleados WHERE salario>35000')
37 #sql_seleccionar(con, 'SELECT name FROM sqlite_master WHERE type="table" ')
38 #sql_borrar_tabla(con)
39
```

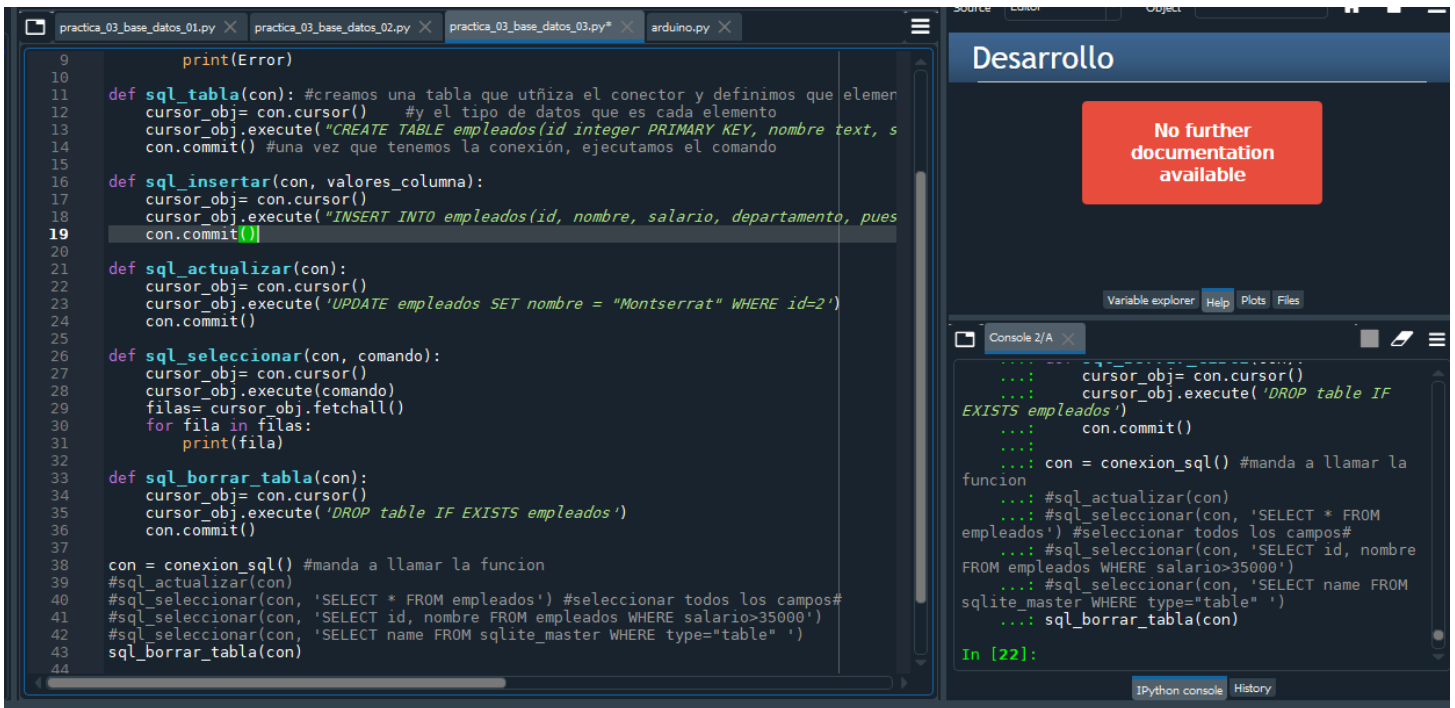
The right sidebar shows the 'Desarrollo' (Development) tab with a message: "No further documentation available". The 'Console 2/A' tab shows the following code:

```
...:     filas = cursor_obj.fetchall()
...:     for fila in filas:
...:         print(fila)
...:
...:     con = conexion_sql() #manda a llamar la
funcion
...:     #sql_actualizar(con)
...:     #sql_seleccionar(con, 'SELECT * FROM
empleados') #seleccionar todos los campos#
...:     #sql_seleccionar(con, 'SELECT id, nombre
FROM empleados WHERE salario>35000')
...:     #sql_seleccionar(con, 'SELECT name FROM
sqlite_master WHERE type="table" ')
...:     #sql_borrar_tabla(con)
('empleados',)

In [21]:
```

## Actividad 8

### Borrar la tabla creada



The screenshot shows a Jupyter Notebook with a dark theme. The code in the cell is as follows:

```
9     print(Error)
10
11 def sql_tabla(con): #creamos una tabla que utiliza el conector y definimos que elemento
12     cursor_obj = con.cursor() #y el tipo de datos que es cada elemento
13     cursor_obj.execute("CREATE TABLE empleados(id integer PRIMARY KEY, nombre text, s
14     con.commit()) #una vez que tenemos la conexión, ejecutamos el comando
15
16 def sql_insertar(con, valores_columna):
17     cursor_obj = con.cursor()
18     cursor_obj.execute("INSERT INTO empleados(id, nombre, salario, departamento, pues
19     con.commit()
20
21 def sql_actualizar(con):
22     cursor_obj = con.cursor()
23     cursor_obj.execute("UPDATE empleados SET nombre = 'Montserrat' WHERE id=2")
24     con.commit()
25
26 def sql_seleccionar(con, comando):
27     cursor_obj = con.cursor()
28     cursor_obj.execute(comando)
29     filas = cursor_obj.fetchall()
30     for fila in filas:
31         print(fila)
32
23 def sql_borrar_tabla(con):
24     cursor_obj = con.cursor()
25     cursor_obj.execute("DROP table IF EXISTS empleados")
26     con.commit()
27
28 con = conexion_sql() #manda a llamar la funcion
29 #sql_actualizar(con)
30 #sql_seleccionar(con, 'SELECT * FROM empleados') #seleccionar todos los campos#
31 #sql_seleccionar(con, 'SELECT id, nombre FROM empleados WHERE salario>35000')
32 #sql_seleccionar(con, 'SELECT name FROM sqlite_master WHERE type="table" ')
33 sql_borrar_tabla(con)
34
```

The right sidebar shows the 'Desarrollo' (Development) tab with a message: "No further documentation available". The 'Console 2/A' tab shows the following code:

```
...:     cursor_obj = con.cursor()
...:     cursor_obj.execute('DROP table IF
EXISTS empleados')
...:     con.commit()
...:
...:     con = conexion_sql() #manda a llamar la
funcion
...:     #sql_actualizar(con)
...:     #sql_seleccionar(con, 'SELECT * FROM
empleados') #seleccionar todos los campos#
...:     #sql_seleccionar(con, 'SELECT id, nombre
FROM empleados WHERE salario>35000')
...:     #sql_seleccionar(con, 'SELECT name FROM
sqlite_master WHERE type="table" ')
...:     sql_borrar_tabla(con)

In [22]:
```

Name	Type	Schema
Tables (0)		
Indices (0)		
Views (0)		
Triggers (0)		

1

Type of data currently in cell  
Size of data currently in table

Apply

Identity Select an identity to connect

DBHub.io Local Current Database

Name