**Introducción a la programación en Python**

**Caso Práctico 4.**

**Temas 14-15**

Contenido

[Ejercicio Tema 14 – Bases de Datos 3](#_Toc81938859)

[Ejercicio Tema 15 - Documentación 4](#_Toc81938860)

## Ejercicio Tema 14 – Bases de Datos

Para afianzar los conceptos vistos en este tema se propone al alumno el siguiente ejercicio:

* Vamos a crear primero una nueva tabla que se llamará “Tabla2” que contendrá los campos DNI, número de despacho y número de teléfono. Todos los campos son de tipo numérico.
* Una vez creada esta tabla, vamos a rellenarla con los datos contenidos en el documento de texto DatosBBDD2.txt de manera análoga a como lo hemos hecho en el apartado 2.2.2 del Tema 14.

Mostrar una captura de pantalla de la “Tabla2” en DB Browser en la que se observen los datos que hemos introducido en ella (5 puntos).

**Practica14.py**

import sqlite3

dir = r'C:\BDSql\BBDD\_EJER\_14.db'

conn = sqlite3.connect(dir)

cursor = conn.cursor()

cursor.execute("DROP TABLE IF EXISTS Tabla2")

cursor.execute("CREATE TABLE Tabla2 (DNI char(10), Despacho char(25), Telefono char(25))")

archivo = open(r'C:\python\T14-T15\DatosBBDD2.txt' )

lineas = archivo.readlines()

## print(lineas)

## print(len(lineas))

for i in range(len(lineas) -1):

##print ( i)

dni =lineas[i+1].split("\t")[0]

numeroDespacho = lineas[i+1].split("\t")[1]

telf = lineas[i+1].split("\t")[2]

cadenaDatos= "'"+dni+"',"+"'"+numeroDespacho+"',"+"'"+telf+"'"

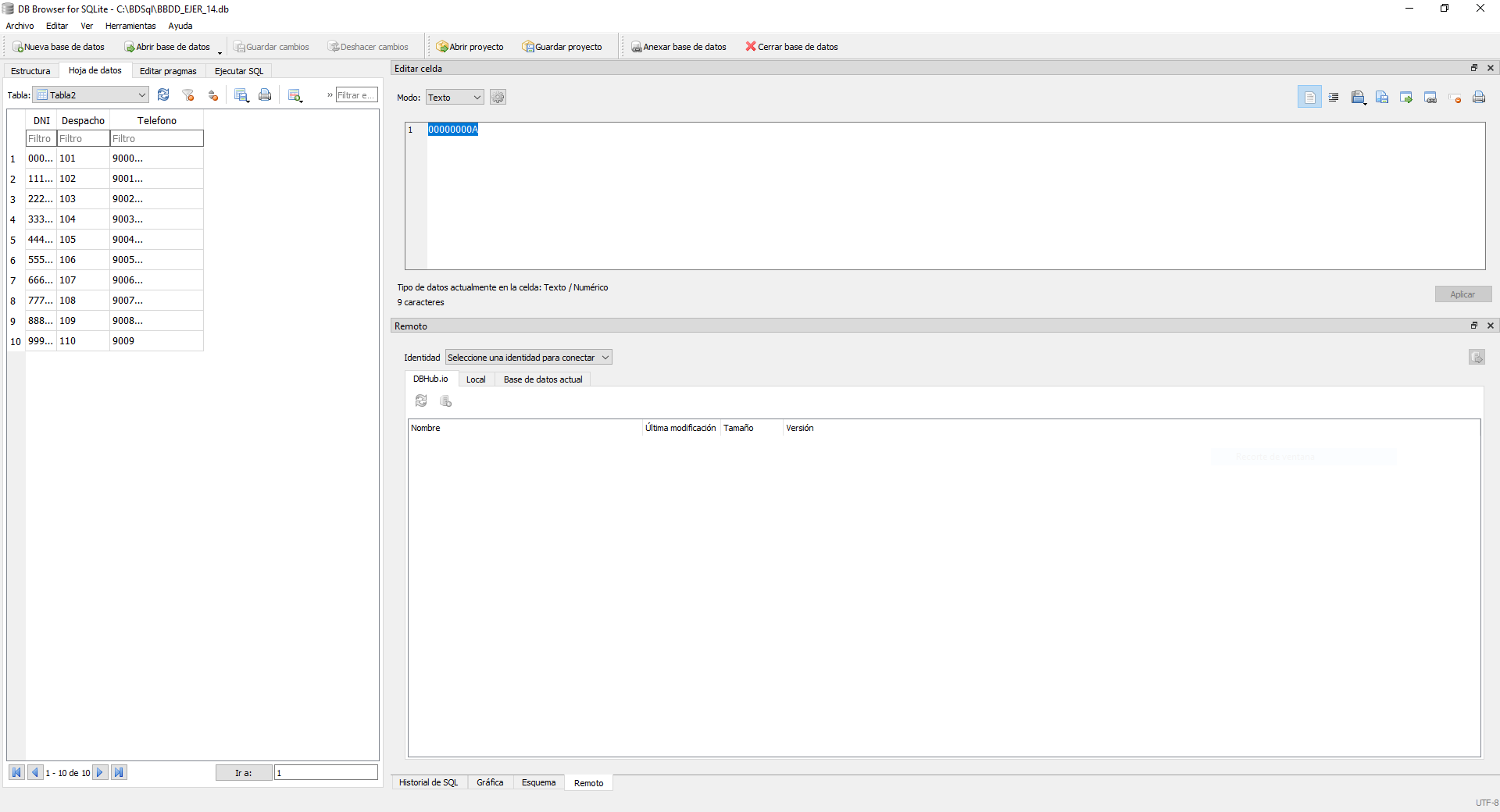
##print("cadena"+ cadenaDatos)

##print("INSERT INTO Tabla2 (DNI, Despacho, Telefono) VALUES (" + cadenaDatos +")")

cursor.execute("INSERT INTO Tabla2 (DNI, Despacho, Telefono) VALUES ("+ cadenaDatos +")")

conn.commit()

print ("Hecho")



* Una vez creada la Tabla2 y debidamente rellenada con la información en el archivo de texto, vamos a realizar una consulta conjunta a esta tabla y a la Tabla1, creada en el Tema 14. Ya que ambas tablas tienen el campo “DNI”, único para cada persona de la lista, vamos a hacer una consulta para obtener el *nombre, apellidos y número de despacho* de la persona con el DNI 22222222.

Mostrar el código con el que se ha hecho la consulta y el resultado de la misma (5 puntos).

import sqlite3

dir = r'C:\BDSql\BBDD\_Prueba.db'

conn = sqlite3.connect(dir)

cursor = conn.cursor()

cursor.execute("select tb1.nombre, tb1.ape1, tb1.ape2, tb2.numeroDespacho from Tabla1 as tb1, Tabla2 as tb2 where tb1.dni = '22222222C' and tb1.dni = tb2.dni")

results = cursor.fetchall()

for i in range(len(results)):

print (results[i])

conn.commit()

print ("Hecho")

Pista: la consulta de dos tablas es muy similar a la que hemos visto en el ejercicio anterior, pero en este caso debemos dar un nombre identificativo para cada una de las tablas, de una manera análoga a esta:

cursor.execute("SELECT a.nombre, a.ape1, a.ape2, b.telefono FROM Tabla1 as a, Tabla2 as b WHERE...)

## Ejercicio Tema 15 - Documentación

Dada la siguiente función que permite saber si un número es primo:

def es\_primo(num):

for n in range(2, num):

if num % n == 0:

print("No es primo", n, "es divisor")

return False

print("Es primo")

return True

Se pide al alumno:

* Documentar brevemente la función explicando el proceso que en ella se realiza.

Mostrar el código de la función con su correspondiente documentación (5 puntos).

def es\_primo(num): ## Declaramos la funcion es primo de un nuemero que nos facilitan por parametro

for n in range(2, num): ## recorremnos con un bucle for un rango de numeeros desde 2 hasta el número que nos pasan por parametro

if num % n == 0: ## optenemos el resto del numero divido entre 2, tantas veces que es posible dividir el numero y verifacamos que el resto sea diferente a 2

print("No es primo", n, "es divisor") ## si el resto no es 0 no es Prîmo

return False

print("Es primo") ##Si es divisible por 0 es primo

return True

* Generar un archivo HTML con la documentación y mostrar este fichero en un navegador Adjuntar pantallazo del fichero HTML en el navegador (5 puntos)