



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

UNIDAD DIDÁCTICA VI
DIPLOMATURA EN PYTHON

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning

Módulo II – Nivel Inicial II

Unidad VI – Formatos de archivos y bases de datos I.



Presentación:

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En el pasado las bases de datos estaban conformadas por ficheros que guardaban documentos en papel, ordenados de forma de poder recuperar su contenido rápidamente. En la actualidad gracias al avance de la tecnología las bases de datos han pasado al plano digital pudiéndose migrar rápidamente de una parte del mundo a otra.

Existen varios tipos diferentes de bases de datos, algunas de las más conocidas son:

- Las relacionales del tipo sql.
- Las no relacionales como Mongo.
- Las orientadas a objetos como zodb.

Dependiendo de la aplicación es conveniente un tipo u otro, en esta unidad veremos cómo persistir nuestros datos y recuperarlos posteriormente utilizando la base MySQL, pero comenzaremos viendo cómo persistir datos en un archivo simple lo cual nos puede ser útil si queremos dejar archivos específicos que persistan los datos de lo que sucede en nuestras aplicaciones y puedan ser consultados rápidamente con algún editor de texto.



Objetivos:

Que los participantes:

Aprendan a persistir datos tanto en archivos como en la base de datos MySQL.



Bloques temáticos:

1.- Archivos.

2.- Bases de datos - relacionales



Consignas para el aprendizaje colaborativo

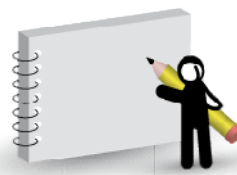
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



Tomen nota

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.



1. Archivos

En esta sección aprenderemos a trabajar con archivos de lectura o para guardar información en ellos, utilizaremos varios métodos que podemos encontrar en la siguiente tabla:

Operación	Interpretación
<code>output = open(r'C:\span', 'w')</code>	Crea un archivo de salida ('w' significa escritura)
<code>input = open('data', 'r')</code>	Crea un archivo de entrada ('r' significa lectura)
<code>aString = input.read()</code>	Lee el archivo como si fuera un string.
<code>aString = input.read(N)</code>	Lee hasta los siguientes N caracteres (o bytes) en un string.
<code>aString = input.readline()</code>	Lea la siguiente línea (incluyendo \n nueva línea) en una cadena
<code>aList = input.readlines()</code>	Lea el archivo completo en una lista de líneas de string (con \n)
<code>output.write(aString)</code>	Escribe una cadena de caracteres (o bytes) en el archivo
<code>output.writelines(aList)</code>	Escribe todas las cadenas de líneas en una lista en un archivo
<code>output.close()</code>	Cierre manual
<code>output.flush()</code>	Vaciar el búfer de salida al disco sin cerrar
<code>for line in open('data'): use line</code>	Los iteradores de archivos leen línea por línea
<code>open('f.txt', encoding='latin-1')</code>	Archivos de texto Python 3.X Unicode (cadenas de caracteres)
<code>open('f.bin', 'rb')</code>	Archivos de bytes Python 3.X (cadenas de bytes)

La función de apertura `open()` crea un objeto de archivo Python, que sirve como enlace a un archivo que reside en nuestra máquina. Los objetos de archivo son algo inusuales, se crean mediante una función incorporada, pero no son números, secuencias o asignaciones, y no responden a los operadores de expresión; solo exportan métodos para tareas comunes de procesamiento de archivos. La mayoría de los métodos de archivo se ocupan de realizar entradas y salidas en el archivo externo asociado con un objeto de archivo, pero otros métodos de archivo nos permiten buscar una nueva posición en el archivo, vaciar los buffers de salida, etc.



Crear y escribir en archivo

Para crear un objeto de archivo, podemos realizarlo llamando a la función de apertura `open()` y pasarle un nombre de archivo y el modo 'w' correspondiente a la escritura de datos.

```
archivo = open('archivo1.txt', 'w')
```

Con esta simple línea, si el archivo no existe lo crea y ya podemos operar sobre él, por ejemplo agregando texto utilizando el método `write()`. Una vez finalizada la operación cerramos el archivo con el método `close()`.

archivo1.py

```
archivo = open('archivo1.txt', 'w')  
archivo.write('Texto agregado desde script\n')  
archivo.close()
```

Una vez ejecutado podemos ingresar al archivo y verificar que su contenido ha sido modificado.

Leer archivo

Contamos con varios métodos para analizar el contenido de los archivos, por ejemplo si creamos un archivo de texto con algunas líneas de código

archivo2.txt

```
Contenido de línea 1  
Contenido de línea 2  
Contenido de línea 3  
Contenido de línea 4  
Contenido de línea 5
```

Y ejecutamos el siguiente script

archivo2.py

```
archivo = open('archivo2.txt', 'r', encoding='utf-8')
```

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



```
print('----- Imprimo líneas en lista -----')
print(archivo.readlines(), end='\n')
print('----- Imprimo líneas -----')
archivo.seek(0)
for x in archivo:
    print(x, end="")
archivo.close()
```

En donde:

- Hemos establecido que vamos a leer información del archivo mediante el uso del parámetro 'r',
- Determinamos que el código que vamos a leer se encuentra en español mediante el parámetro (encoding='utf-8').
- La primer impresión la realizamos guardando cada línea con su salto de línea dentro de los elementos de una lista, mientras que en la segunda impresión de datos nos valemos del método seek(0) para pararnos en la primera línea y mediante el bucle "for" imprimimos línea por línea.

Retorna:

```
----- Imprimo líneas en lista -----
'Contenido de línea 1\n', 'Contenido de línea 2\n', 'Contenido de línea 3\n', 'Contenido de
línea 4\n', 'Contenido de línea 5\n', '\n']
----- Imprimo líneas -----
Contenido de línea 1
Contenido de línea 2
Contenido de línea 3
Contenido de línea 4
Contenido de línea 5
```

Nota: La segunda forma de lectura es más eficiente ya que no lee todo el archivo de una por lo que ocupa menos espacio de memoria.

Opciones de apertura de archivos

Modo	Descripción
'r'	Abre un archivo para lectura.
'w'	Abre un archivo para escritura, si el archivo no existe lo crea.
'x'	Crea y abre un archivo, si éste ya existe la operación falla.
'a'	Abre un archivo para agregar al final del mismo sin truncar, si el archivo no existe lo crea.
't'	Abre en modo texto.
'b'	Abre en modo binario.
'+'	Abre un archivo para actualizar (leer y escribir)

2. Base de datos - Relacional

Trabajar con archivos es muy útil para registrar información, por ejemplo para guardar datos de los procesos que se van realizando en una aplicación a lo largo del tiempo, sin embargo existen muchas ventajas al utilizar bases de datos. Existen muchos tipos de bases de datos diferentes, aquí vamos a iniciar viendo las bases de datos del tipo relacionales, y en particular iniciaremos trabajando con MySQL la cual es muy utilizada en aplicaciones web.

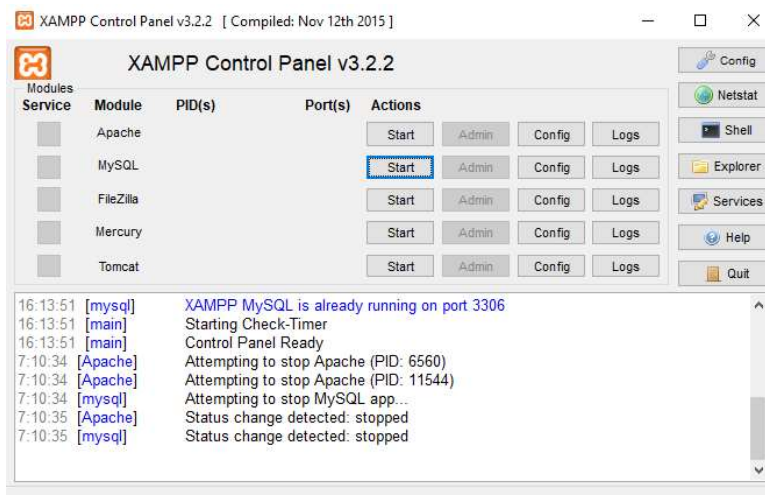
Paso 1: Instalación de MySQL

Podemos contar con varias opciones de mercado de servidores locales que ya traen incorporada la base de datos MySQL, una opción que, como dijimos, ya trae configurada la base de datos pero también un servidor del tipo Apache es xampp. Las versiones de los lenguajes que utiliza se actualizan todo el tiempo y tenemos la posibilidad de elegir el sistema operativo. Yo en particular he descargado la versión para Windows del siguiente sitio:

<https://www.apachefriends.org/es/index.html>



Para instalarlo es muy fácil, simplemente lo descargamos y damos “next” hasta finalizar la instalación eligiendo en el proceso el directorio en el cual se va a instalar, el cual por defecto en Windows es el disco raíz. De esta forma nos aparece una vez finalizada la instalación, un directorio llamado xampp en la carpeta raíz de nuestro equipo. Dentro de este directorio nos vamos a encontrar un sistema de carpetas y un ejecutable con el nombre de “xamppControl”. Si hacemos doble click sobre el mismo se nos va a desplegar la siguiente ventana:



En la misma podemos ver varios botones “Start” que son los que encienden los servicios. Dado que vamos a utilizar también una interfaz gráfica para manipular la base de datos de forma gráfica, iniciaremos los servicios de Apache y MySQL. Cada servicio funciona como un semáforo el cual si todo está bien debe presentar un color verde e indicar el puerto de trabajo. Por defecto, el puerto del servidor es el 80 y el de MySQL el 3306.

Paso 2 – Instalamos conector para python - MySQL

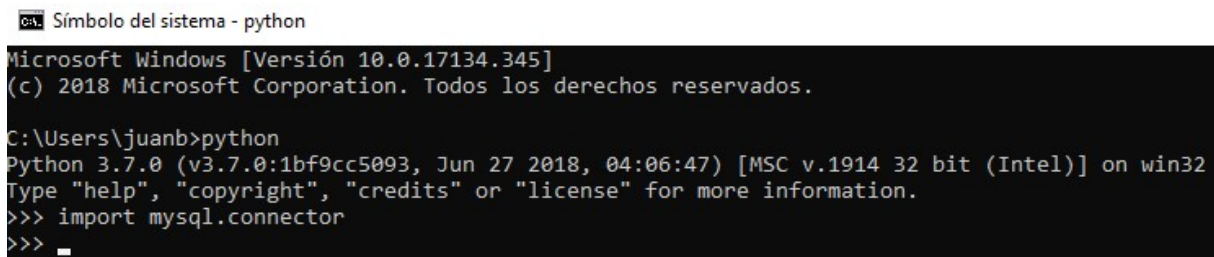
Ingresamos al cmd con permisos de administrador y ejecutamos:

```
pip install mysql-connector
pip install pymysql
pip install mysqlclient
pip install mysql-connector-python
```

Si todos está bien podemos ingresar al intérprete de python desde el cmd (escribiendo: python) y ejecutar:

```
import mysql.connector
```

No debería retornarnos ningún error como se muestra en la siguiente imagen.



```
C:\> Símbolo del sistema - python
Microsoft Windows [Versión 10.0.17134.345]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

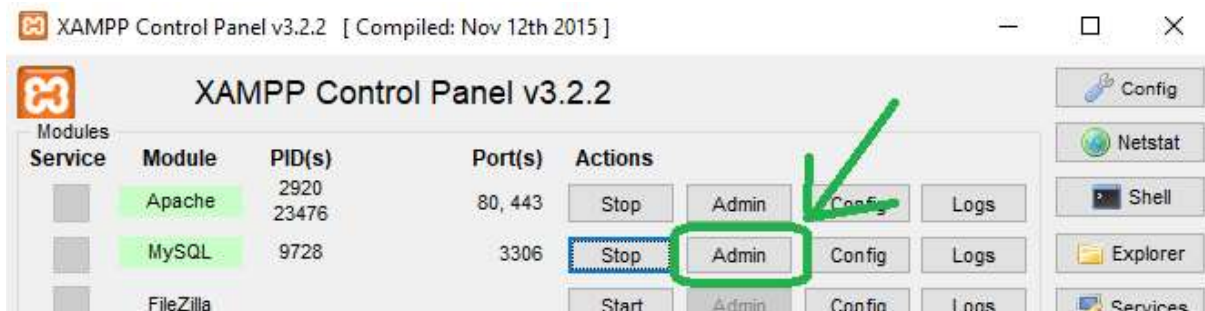
C:\Users\juanb>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import mysql.connector
>>> _
```

Paso 3 – Creamos una base de datos.

Nuestro siguiente paso es crear una base de datos con la cual interactuar, para ello contamos con dos opciones

Opción 1 - desde phpmyadmin.

La primera y más fácil es presionar el botón “Admin”, el cual se encuentra en la línea del servicio MySQL para ingresar a la interfaz gráfica phpmyadmin. La aplicación se lanza en el explorador que tengamos configurado por defecto.



Una vez ahí vamos a “Bases de datos” en las solapas que se encuentran en la cabecera.



Solo nos resta en esta parte darle un nombre cualquiera (yo le he llamado “mi_plantilla”) y elegir el tipo de caracteres que utilizaremos en español (puede ser “utf8_spanish2_ci”). Finalmente presionamos en crear.



Nuestra base ya aparece listada en el menú de la izquierda.



Nota: Por defecto la conexión de la base de datos viene configurada como:

Servidor = localhost

usuario = root

password = ""

El usuario y el password puede ser modificado directamente desde la interfaz gráfica, sin embargo modificar el nombre del servidor requiere alterar algunas líneas en el archivo de configuración de apache.

Opción 2 – Desde Python

La otra opción con la cual contamos es desde la consola de python estando el servidor de la base de datos encendido, ejecutar el siguiente script:

crearbd.py

```
import mysql.connector

mibase = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd=""
)
micursor = mibase.cursor()

micursor.execute("CREATE DATABASE mi_plantilla2")
```

Aquí "mysql.connector" es un objeto que podemos utilizar para conectarnos a la base de datos y mediante notación de punto (agregando un punto al final y a continuación el método que queremos aplicar) le aplicamos el método connect() el cual recibe los parámetros de servidor, usuario y password para poder abrir la conexión. Una vez abierta

Centro de e-Learning SCEU UTN - BA.

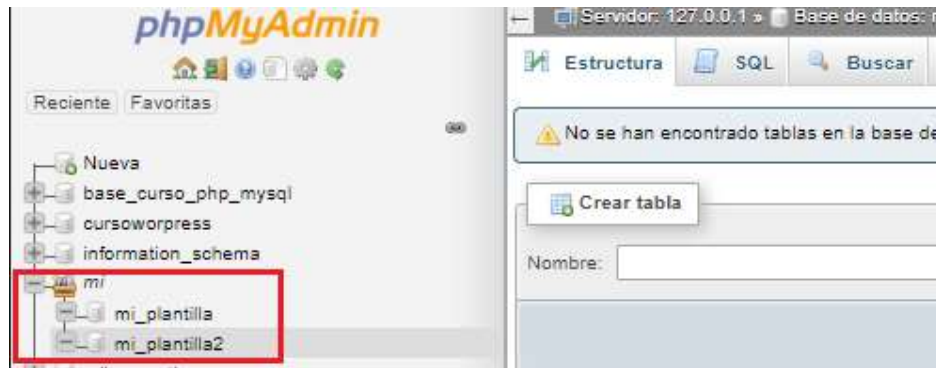
Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



la conexión, podemos nuevamente con notación de punto asociarle el método execute(), con el cual le pasamos en lenguaje sql la instrucción para que cree la base de datos.

Luego de ejecutarlo podemos chequear desde la interfaz gráfica como se ha agregado al menú de la izquierda.

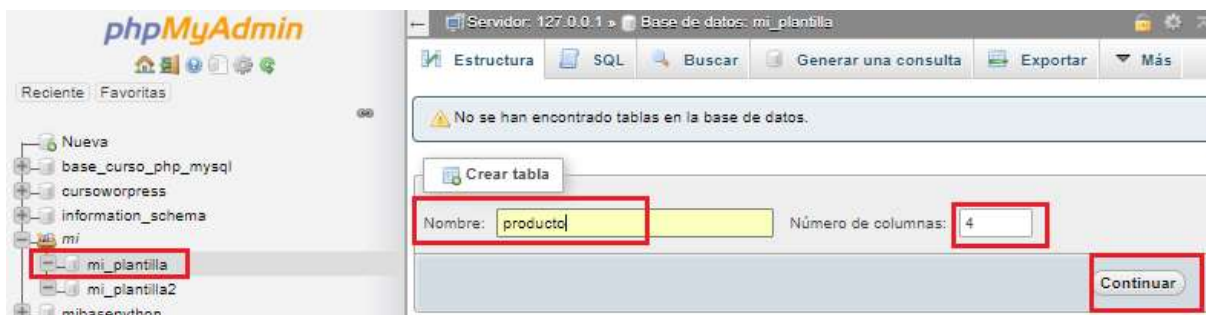


Paso 4 – Creamos una tabla

Las bases de datos relacionales poseen tablas para guardar los datos, y trabajar con ellas es similar a trabajar con hojas de Excel. Nuevamente podemos crear las tablas de forma gráfica o mediante código.

Opción 1 – Desde phpmyadmin

Para crear una tabla, nos paramos en la base que queremos adicionarla y le damos un nombre seleccionando además la cantidad de columnas que va a poseer, luego presionamos en “Continuar”.





En este caso he agregado una tabla de cuatro columnas:

- **id:** Campo de enteros (int) autoincremental y clave primaria (campo clave)
- **titulo:** Campo de tipo varchar (strings) de 128 caracteres.
- **ruta:** Campo de tipo varchar (strings) de 128 caracteres.
- **descripcion:** Campo de texto, ocupa más memoria.

Nota: no utilizar acentos ni caracteres en español para los campos.

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo índice
id	INT		Ninguno			PRIMARY
titulo	VARCHAR	128	Ninguno			
ruta	VARCHAR	128	Ninguno			
descripcion	TEXT		Ninguno			

Opción 2 – Desde el código

Esta opción requiere un poco más de conocimiento del lenguaje de la base de datos, sin embargo salvo que hemos indicado como parámetro de la conexión la base de datos dentro de la cual se va a agregar la tabla, la estructura es la misma que ya hemos utilizado al crear la base.

creartabla.py

```
import mysql.connector
```

```
mibase = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    passwd="",  
    database="mi_plantilla2"  
)  
micursor = mibase.cursor()
```

```
micursor.execute("CREATE TABLE producto( id int(11) NOT NULL PRIMARY KEY  
AUTO_INCREMENT, titulo VARCHAR(128) COLLATE utf8_spanish2_ci NOT NULL, ruta  
varchar(128) COLLATE utf8_spanish2_ci NOT NULL, descripcion text COLLATE  
utf8_spanish2_ci NOT NULL )")
```

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

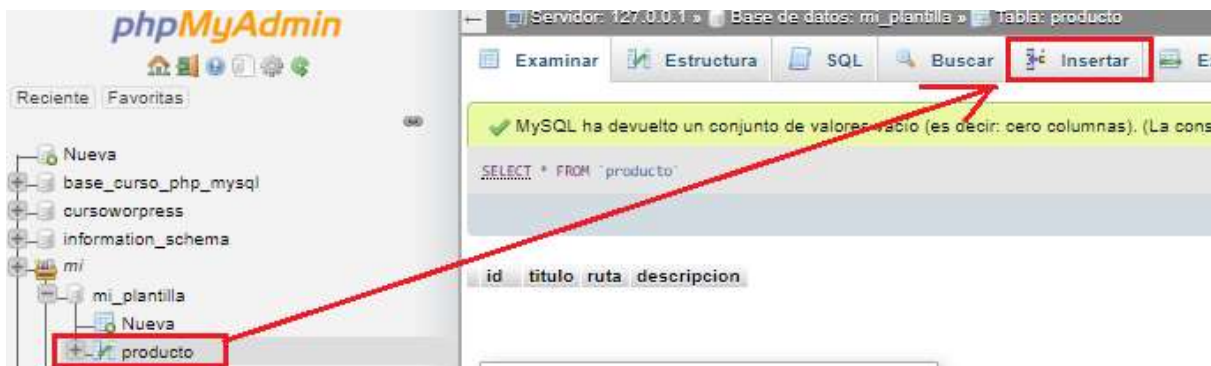
www.sceu.frba.utn.edu.ar/e-learning



Paso 5 - Insertamos registros

Opción 1 – Desde phpmyadmin

Nuevamente la opción 1 es la más fácil de implementar para insertar registros, solamente tendríamos que pararnos en la tabla en el menú de la izquierda, y luego desde el menú superior ir a “Insert” para poder agregar los nuevos registros. Tener en cuenta que el campo “id” no debe ser completado, ya que al ser autoincremental el valor es agregado automáticamente.





Columna	Tipo	Función	Nulo	Valor
id	int(11)	<input type="text"/>	<input type="text"/>	
titulo	varchar(128)	<input type="text"/>	<input type="text"/>	Titulo 1
ruta	varchar(128)	<input type="text"/>	<input type="text"/>	imagen1.png
descripcion	text	<input type="text"/>	<input type="text"/>	Descripción de imagen 1

+ Opciones

				id	titulo	ruta	descripcion
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Título 1	imagen1.png	Descripción de primer producto.
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Título 2	imagen2.png	Descripción de segundo producto.

Opción 2 – Desde el código

Desde el código podemos insertar registros de la siguiente forma, teniendo en cuenta que:

- El campo id como lo definimos auto incremental no debe ser agregado
- Para que se registre el cambio debemos realizar un commit.

insert.py

```
import mysql.connector
```

```
mibase = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    passwd="",  
    database="mi_plantilla2"  
)
```

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



```
micursor = mibase.cursor()

sql = "INSERT INTO producto (titulo, ruta, descripcion) VALUES (%s, %s, %s)"
datos = ("Tema 3", "ruta3", "descripción 3")

micursor.execute(sql, datos)

mibase.commit()

print(micursor.rowcount, "Cantidad de registros agregados.")
```

Si en lugar de utilizar “execute()”, utilizamos “executemany()”, podemos pasar una lista de tuplas.

```
datos = [("Tema 4", "ruta4", "descripción 4"),
("Tema 5", "ruta5", "descripción 5"),
("Tema 6", "ruta6", "descripción 6"),
]

micursor.executemany(sql, datos)
```

Paso 6 – Borrar registro

Opción 1 – Desde phpmyadmin

Borrar datos gráficamente es muy simple, solo tenemos que presionar en el ícono que se encuentra en la fila del registro que queremos borrar.





Opción 2 – Desde el código

Para borrar el registro desde el código debemos ejecutar “DELETE FROM tabla WHERE registro”

insert.py

```
import mysql.connector

mibase = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="mi_plantilla2"
)
micursor = mibase.cursor()

sql = "DELETE FROM producto WHERE id = %s"
dato = ('2',)

micursor.execute(sql, dato)

mibase.commit()

print(micursor.rowcount, "Registro borrado")
```

Paso 7 – Seleccionar registros

En el caso de seleccionar registros en ambos casos debemos introducir el código sql de los datos a seleccionar.

Opción 1 – Desde phpmyadmin

Desde phpmyadmin, podemos ir a la solapa “SQL” en donde contamos con algunas herramientas para armar la consulta específica que queremos realizar.



Opción 2 – Desde el código

Podemos recuperar los datos mediante un “SELECT”, si queremos traer todos los campos simplemente ponemos un asterisco, en caso contrario seleccionamos los campos separados por coma. Para darle formato a los datos recuperados utilizamos “fetchall()”

resultado.py

```
import mysql.connector

mibase = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="mi_plantilla2"
)
micursor = mibase.cursor()

sql = "SELECT * FROM producto"

micursor.execute(sql)
resultado = micursor.fetchall()

for x in resultado:
    print(x)
```

Nos retorna:

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning

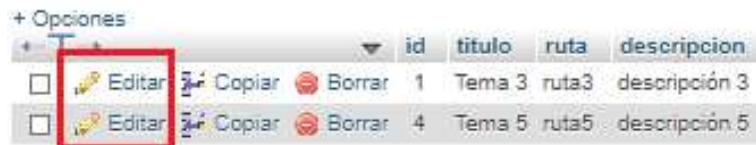


(1, 'Tema 3', 'ruta3', 'descripción 3')
(4, 'Tema 5', 'ruta5', 'descripción 5')

Paso 8 – Actualizar registros

Opción 1 – Desde phpmyadmin

Actualizar registros desde la interfaz gráfica nuevamente es muy simple, solo tenemos que seleccionar el ícono del lápiz que se encuentra al lado de cada registro para editarlos.



Opción 2 – Desde el código

Para editarlo desde el código debemos ejecutar un “UPDATE” e indicar el dato que queremos agregar, en lugar de un dato ya existente.

actualizar.py

```
import mysql.connector

mibase = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="mi_plantilla2"
)
micursor = mibase.cursor()

sql = "UPDATE producto SET titulo = 'Titulo 8' WHERE titulo = 'Tema 3'"

micursor.execute(sql)
mibase.commit()

print(micursor.rowcount, "Cantidad de registros afectados.")
```



Bibliografía utilizada y sugerida

Libros

Programming Python 5th Edition – Mark Lutz – O'Reilly 2013

Programming Python 4th Edition – Mark Lutz – O'Reilly 2011

Manual online

<https://docs.python.org/3.7/tutorial/>

<https://docs.python.org/3.7/library/index.html>



Lo que vimos

En esta unidad trabajamos con archivos y con la base de datos del tipo relacional MySQL.



Lo que viene:

En la siguiente unidad comenzaremos a trabajar expresiones regulares.