

Laboratorio de Casas Inteligentes

Almacenamiento de datos en bases de datos.

Práctica No. 06

Objetivo: Instalar el manejador de bases de datos *MySQL* y almacenar datos en una base de datos provenientes de los sensores y actuadores utilizando la herramienta BlackBoard.

Duración: Dos semanas

Desarrollo:

1. Instale el servidor de base de datos *MySQL* ejecutando el siguiente comando en una terminal:

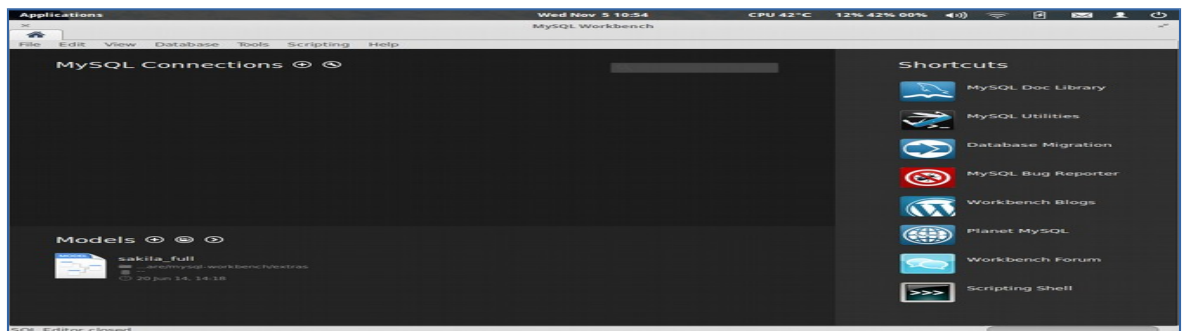
```
sudo apt-get install mysql-server
```

Se recomienda instalar la herramienta (cliente) *MySQL workbench* para facilitar la manipulación de las bases de datos en un ambiente gráfico. Puede descargar el paquete desde la siguiente liga: <http://dev.mysql.com/downloads/workbench/>

Una vez descargado, instálelo utilizando el siguiente comando:

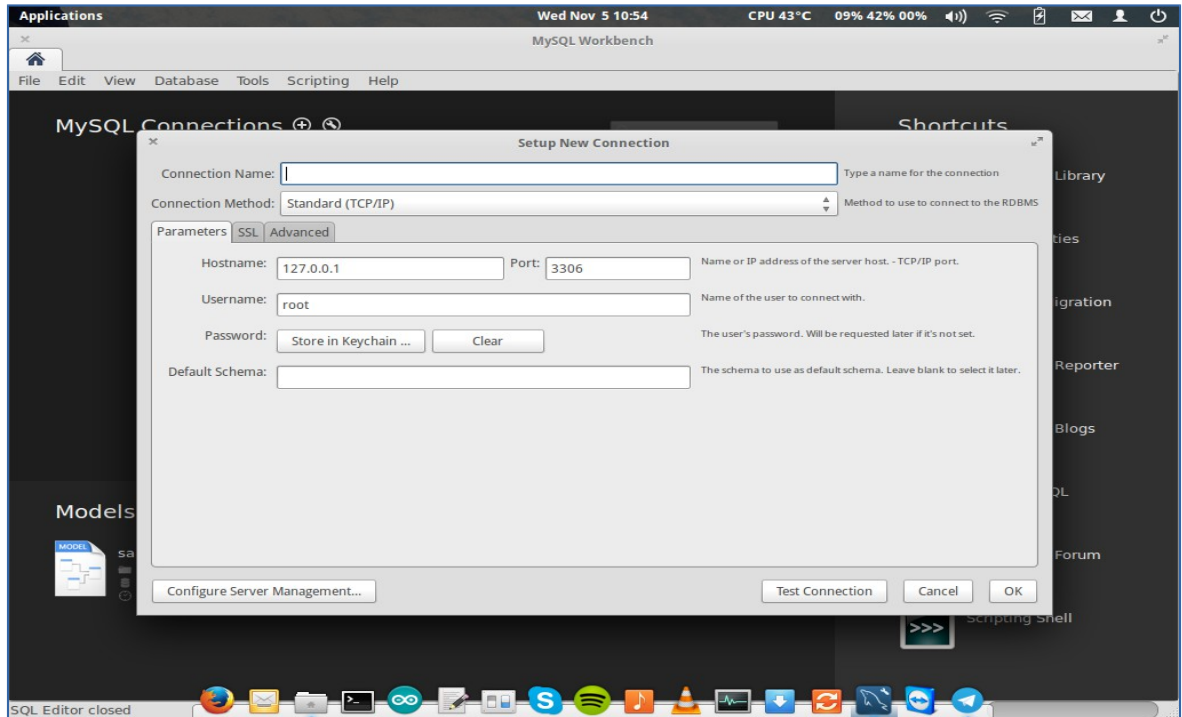
```
sudo dpkg -i nombre_del_archivo_descargado
```

2. Ejecute el cliente *MySQL workbench* y cree una nueva conexión con el servidor local (*localhost*). Para ello haga click en el símbolo “+” que aparece a la derecha del título “MySQL Connections”

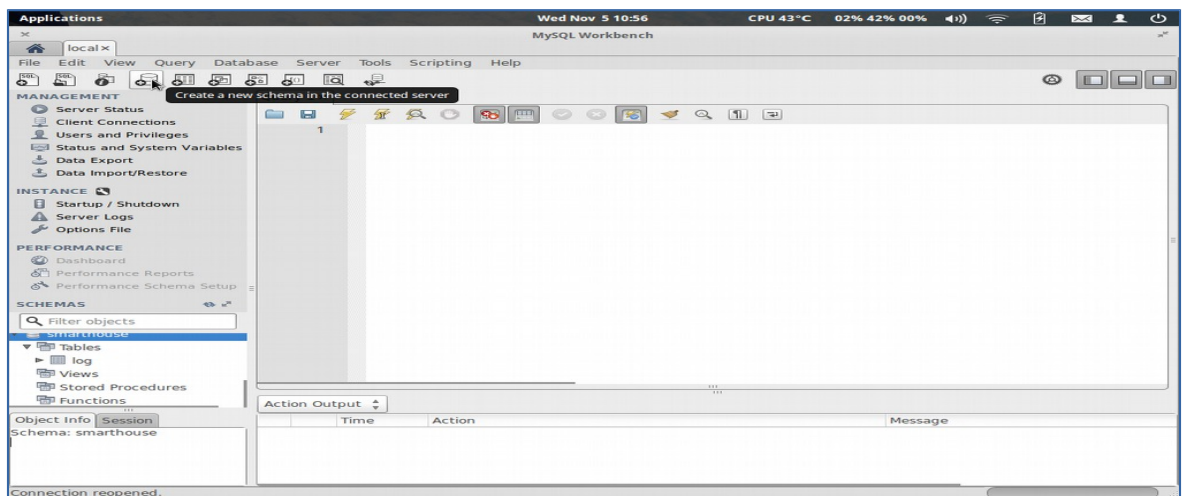


Escriba un nombre para la conexión y de click en OK dejando los demás campos con el valor por default. El programa solicitará la

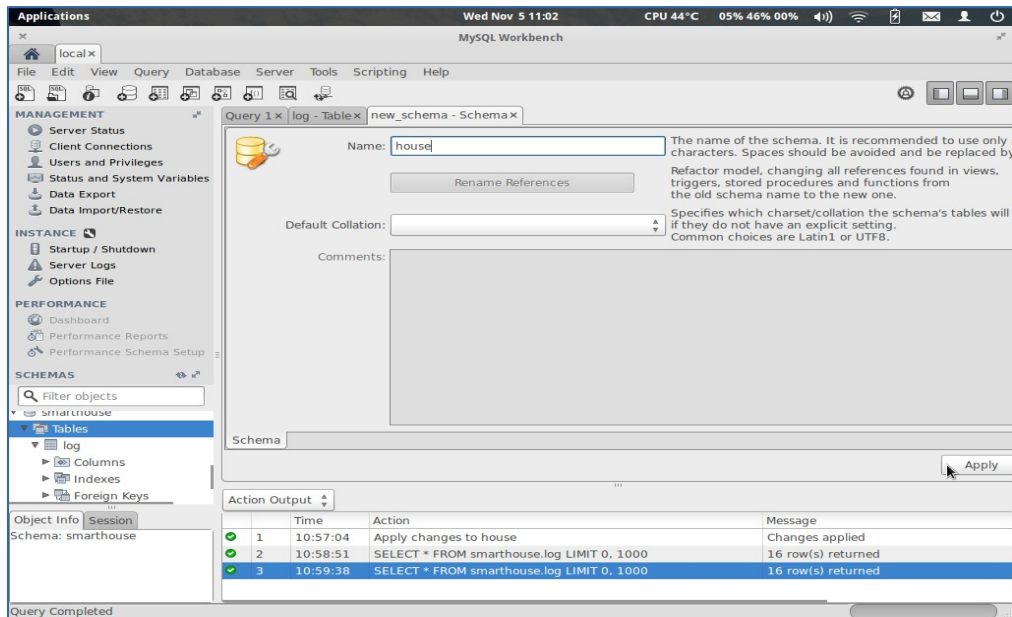
contraseña para el usuario *root* que definió al momento de la instalación del servidor. Sino recuerda este password teclee root.



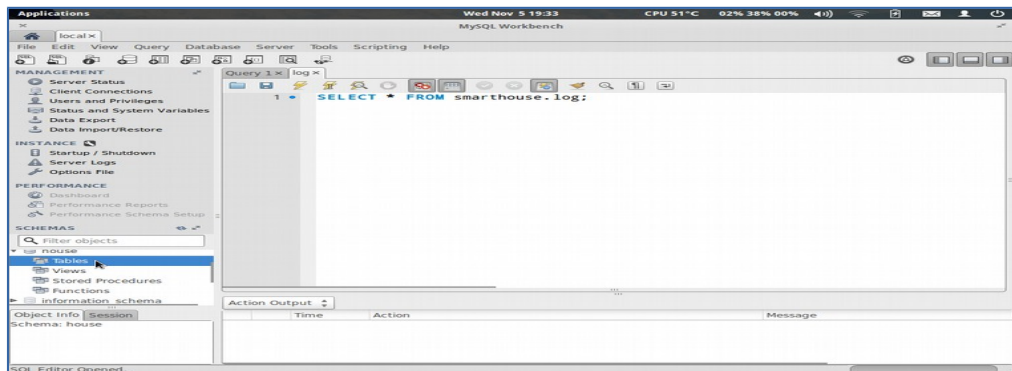
- i. Cree una nueva base de datos dando click en el botón señalado “Create a new schema in the connected server”



Escriba un nombre para la nueva base de datos y de click en apply.



- ii. Defina una tabla nueva llamada *log*. Para ello de click con el botón derecho del ratón sobre el apartado “Tables” de la base de datos y posteriormente en “Create new table”.

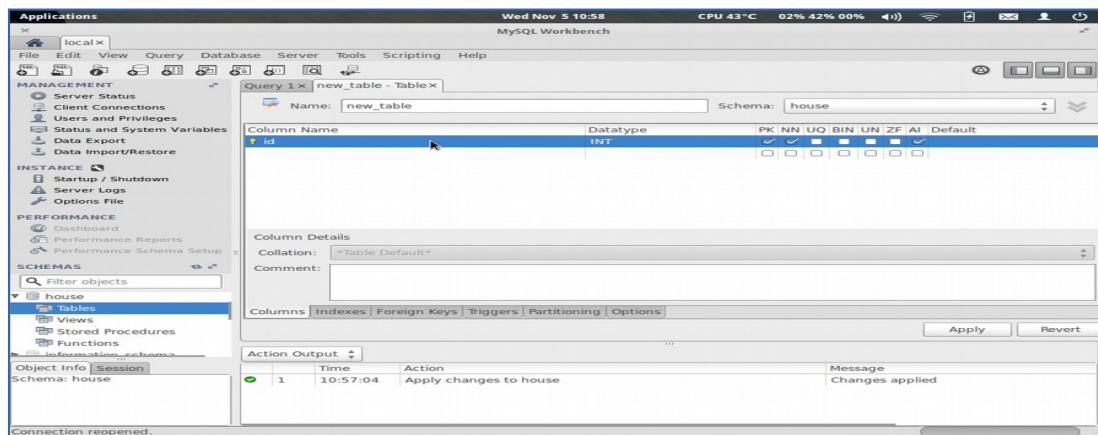


Añada las siguientes columnas, atendiendo al tipo de dato y restricciones:

- i. *Id*: Esta columna será el identificador único de los registros; para ello deberá ser marcada como llave primaria (PK), deberá ser definida como un tipo de dato entero (int) y auto incrementable (AI).
- ii. *Device*: Esta columna identificará el dispositivo de origen de los datos. Su tipo de dato será varchar(4).
- iii. *Source*: Esta columna almacenará la naturaleza del sensor o actuador del cual se están enviando los datos. Su tipo de dato será varchar(20).

- iv. *ValueCommand*: Aquí se almacenará el valor del sensor en cuestión en el instante de la medición, o la orden enviada al actuador. Su tipo de dato será varchar(45).
- v. *Timestamp*: Almacenará la fecha y hora en la cual se envió el valor del sensor. Su tipo de dato será datetime.
- vi. *Type*: Servirá para discriminar si los datos almacenados pertenecen a un sensor o un actuador. Su tipo de dato será bit; adquirirá el valor de 0 al tratarse de un sensor y 1 en caso de ser un actuador.

NOTA: Para añadir una nueva columna haga doble click debajo de la última columna definida y rellene los campos.



Ejemplo de datos almacenados en la tabla *log*:

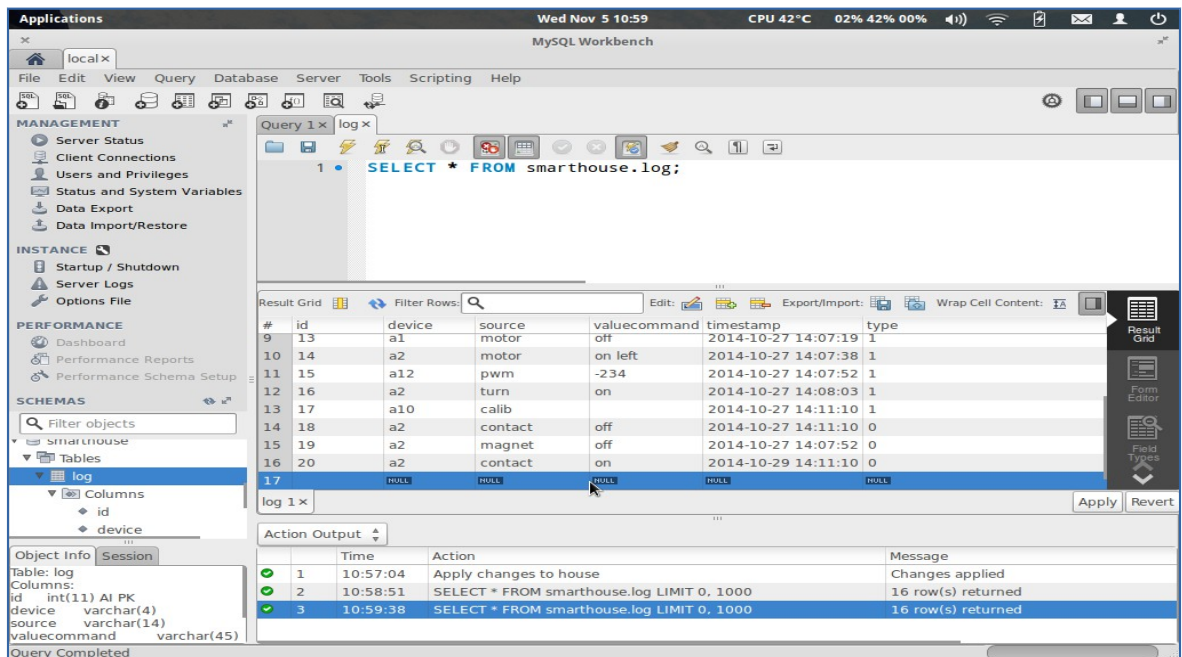
Id	Device	Source	ValueCommand	Timestamp	Type
79	A1	photora	128	1980-12-08 22:50:45	0
80	A2	acelerometer_x	68	2014-02-54 18:49:01	0
81	A8	turn	on	2011-12-21 23:59:59	1
82	A1	motor	on left	2015-10-21 04:04:33	1

3. Instale el conector para manipular base de datos mediante Python MySQL-connector. Una forma de hacer esto es utilizando el comando :

```
sudo apt-get install python-mysqldb
```

4. En el apéndice viene parte del código de un programa en Python que inserta datos simulados a la base de datos, insert_data.py, la versión completa viene en la pagina de las practicas en el archivo practica_6_extras.tar.gz . Pruebe este código y compruebe que los datos son insertados en la base de datos.

Puede utilizar el cliente *MySQL Workbench* para insertar datos manualmente, o bien revisar el contenido de las tablas. Para ello puede dar click derecho en el nombre de la tabla y seleccionar la opción “Select Rows - Limit 1000” lo que mostrará los primeros 1000 registros de la tabla.



Para agregar datos de forma manual haga click debajo del último registro de la tabla e ingrese los datos. No olvide dar click en Aplicar para guardar los cambios.

Modifique el código insert_data.py para que se encargue de leer los mensajes de los sensores y actuadores provenientes de la tarjeta Arduino y los almacene en la tabla de bases de datos del punto 3; dicho

programa deberá conectarse al BlackBoard (como se vio en la práctica 5) y suscribirse a las variables compartidas que almacenan los mensajes del Arduino (sensor) y los escritos por los usuarios utilizando el teclado (actuador).

Referencias:

<http://dev.mysql.com/downloads/> Descargas de paquetes.

<http://dev.mysql.com/doc/connector-python/en/connector-python-reference.html> Sobre cómo usar mySQL con Python.

Anexo: Programa básico para inserción de datos en una base de datos MySQL con Python.

```
*****
#*   insert_data.py   *
#*   *               *
*****

# Importar conector python-mysql
import mysql.connector
import datetime
import time
from random import randrange, uniform

# Objeto con los datos de la conexión al servidor
connection= mysql.connector.connect(user='root',password='root',host='localhost',database='house')

# Cursor de consulta
cursor = connection.cursor()

# It gets the time stamp
Timestamp=datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

#-----
# Inserting new data directly
query = "INSERT INTO `house`.`log` (`Device`, `Source`, `ValueCommand`, `Timestamp`, `Type`)
VALUES ('A1', 'photora', '2.1', '2015-11-02 13:50:48', 0);"

# Execute the query
cursor.execute(query)
# Update the data
connection.commit()
print query
```

```

# Insert new data from variables
#-----
# Insert new data from variable accel_x
ValueCommand= str(uniform(0, 100))
Type='0'
Source='accel_x'
Device='A1'
query = "INSERT INTO `house`.`log` (`Device`, `Source`, `ValueCommand`, `Timestamp`,
`Type`)"
query += "VALUES
("+Device+", "+Source+", "+ValueCommand+", "+Timestamp+", "+Type+");"
cursor.execute(query)
connection.commit()
print query

# ...

#-----
# Read all the data from the database
query = "SELECT id, Device, Source, ValueCommand, Timestamp, Type FROM `house`.`log`;"
cursor.execute(query)
print query
# Print the data
for c in cursor:
    print c[0],c[1],c[2],c[3],c[4],c[5]

connection.close()

```