Desarrollo de Aplicaciones Web

Tema Nº9:NODE JS – DATABASE I

Indicador de logro Nº9:Construye aplicaciones CRUD con conexión a base de datos a través de Node.JS en una arquitectura orientada a eventos y basado en el motor V8.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº9:**

NODE JS – DATABASE I

**Subtema 9.1:**

Comunicación con el gestor de base de datos MySQL

Una actividad muy común de una aplicación web es el acceso a un gestor de base de datos. Veremos en este concepto como podemos comunicarnos desde nuestra aplicación Node.js con el servidor de base de datos MySQL.

Una base de datos en una página web la convierte en una herramienta de ventas sumamente potente, ya que la convierte en una web dinámica al permitirnos almacenar, datos de usuarios, productos, etc.

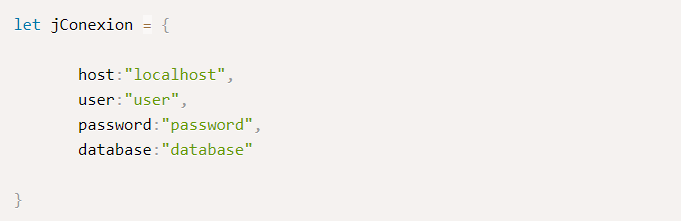
Si tú página almacena el correo de tus usuarios podrás realizar campañas de email marketing para promocionar nuevos productos o simplemente para estar en contacto con tus usuarios, así que tener una base de datos en tu página es indispensable.



**Subtema 9.2:**

¿Cómo conectar MySQL con Node JS?

Para contestar esta pregunta se escribirá algunos trozos de códigos, por ejemplo, comencemos con los parámetros que necesitamos para realizar la conexión.



Es muy fácil deducir cada uno de los parámetros:

* **host**, IP del servidor de base de datos
* **user**, usuario registrado en la base de datos
* **password**, contraseña del usuario registrado en la base de datos
* **database**, nombre de la base de datos

**NPM MYSQL**

Para poder crear un enlace entre NodeJs y MySQL debemos utilizar una paquetería que nos permita hacer uso de un conector, en nuestro caso utilizaremos "mysql" el cual instalaremos en nuestro proyecto nodejs a través de la siguiente instrucción desde la consola.



**Subtema 9.3:**

Creando el objeto de conexión entre MySQL y Node JS

Ahora que ya se ha definido cada parámetro de la base de datos y ya se tiene instalado el conector, simplemente tenemos que crear los objetos necesarios e invocar los métodos de conexión, por ejemplo:

var con = mysql.createConnection(jConexion);

//Abrimos conexión a la base de datos, "connect" regresa un error si no podemos

//conectarnos a la base de datos

con.connect(function(error){

try {

//si error es true mandamos el mensaje de error

if(error){

console.log("Error al establecer la conexión a la BD -- " + error);

}else{

//conexión exitosa, en este punto ya hemos establecido la conexión a

//base de datos

console.log("Conexión exitosa");

//Aquí debes escribir el código que necesites, un INSERT, SELECT etc.

}

} catch(x){

console.log("Contacto.agregarUsuario.connect --Error-- " + x);

}

});

**1. MATERIALES**

Para la experiencia a realizar se requiere lo siguiente:

1. EQUIPO

Se requiere tener un equipo de cómputo (PC o laptop).

1. HERRAMIENTAS

* Visual Stude Code
* Node.js
* Npm
* JavaScript
* MySQL

**2. PROCEDIMIENTO**

Node.js se puede utilizar en aplicaciones de bases de datos. Una de las bases de datos más populares es MySQL.

**Instalar el controlador MySQL**

Una vez que tenga MySQL en funcionamiento en su computadora, puede acceder a él usando Node.js.

Para acceder a una base de datos MySQL con Node.js, necesita un controlador MySQL. Este ejemplo utilizará el módulo "mysql", descargado de NPM. Para descargar e instalar el módulo "mysql", abra la Terminal de comandos y ejecute lo siguiente:



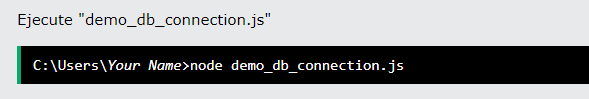
Ahora ha descargado e instalado un controlador de base de datos mysql. Node.js puede usar este módulo para manipular la base de datos MySQL:



**Crear conexión**



Guarde el código anterior en un archivo llamado "demo\_db\_connection.js" y ejecute el archivo:

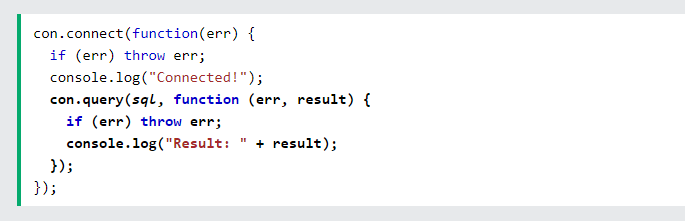


Lo que te dará este resultado:



**Consultar una base de datos**

Utilice sentencias SQL para leer (o escribir en) una base de datos MySQL. Esto también se llama "consultar" la base de datos. El objeto de conexión creado en el ejemplo anterior tiene un método para consultar la base de datos:

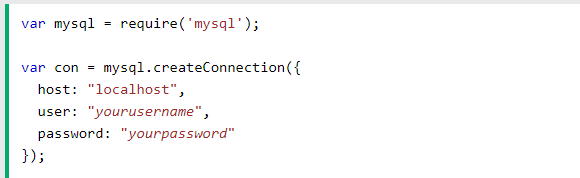


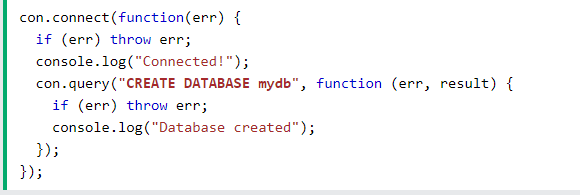
El método de consulta toma declaraciones SQL como parámetro y devuelve el resultado.

**Crear una base de datos**

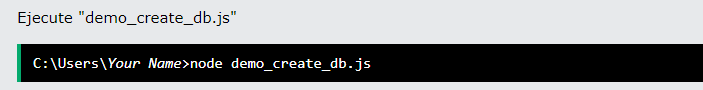
Para crear una base de datos en MySQL, use la declaración "CREATE DATABASE":

Ejemplo: Cree una base de datos llamada "mydb":





Guarde el código anterior en un archivo llamado "demo\_create\_db.js" y ejecute el archivo:



Lo que te dará este resultado:

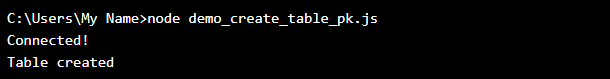


**Crear una tabla**

Para crear una tabla en MySQL, use la instrucción "CREATE TABLE". Asegúrese de definir el nombre de la base de datos cuando cree la conexión:

var mysql = require('mysql');  
  
var con = mysql.createConnection({  
  host: "localhost",  
  user: "yourusername",  
  password: "yourpassword",  
  database: "mydb"  
});  
  
con.connect(function(err) {  
  if (err) throw err;  
  console.log("Connected!");  
  var sql = "CREATE TABLE customers (**id INT AUTO\_INCREMENT PRIMARY KEY**,

name VARCHAR(255), address VARCHAR(255))";  
  con.query(sql, function (err, result) {  
    if (err) throw err;  
    console.log("Table created");  
  });  
});



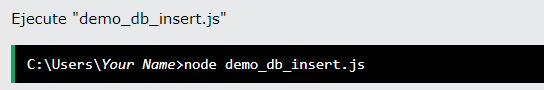
**Insertar en la tabla**

Para llenar una tabla en MySQL, use la instrucción "INSERT INTO".

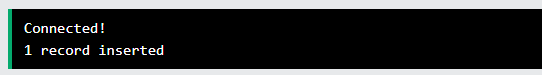
var mysql = require('mysql');  
  
var con = mysql.createConnection({  
  host: "localhost",  
  user: "yourusername",  
  password: "yourpassword",  
  database: "mydb"  
});  
  
con.connect(function(err) {  
  if (err) throw err;  
  console.log("Connected!");  
**var sql = "INSERT INTO customers (name, address)**

**VALUES ('Company Inc', 'Highway 37')";**  con.query(sql, function (err, result) {  
    if (err) throw err;  
    console.log("1 record inserted");  
  });  
});

Guarde el código anterior en un archivo llamado "demo\_db\_insert.js" y ejecute el archivo:



Lo que te dará este resultado:



**ACTIVIDAD VIRTUAL:**

1. **CUESTIONARIO TÉCNICO**

Revisa y analiza el tema desarrollado en la presente sesión, luego responde las siguientes preguntas propuestas:

* ¿Qué es y para qué sirve Node JS?
* ¿Cómo se conecta Node JS con MySQL?
* ¿Cuál es el mejor ORM de MySQL para Node js? De una breve explicación
* Realice una App aplicando todo lo aprendido de esta guía

1. **ENLACES DE REFERENCIA**

* https://ewebik.com/nodejs/con-mysql
* https://www.tutorialesprogramacionya.com/javascriptya/nodejsya/detalleconcepto.php?punto=15&codigo=15&inicio=0
* https://www.w3schools.com/nodejs/nodejs\_mysql.asp

1. **CONCLUSIONES DE LA EXPERIENCIA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_