



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

Diplomatura en programación web full stack con React JS

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 3: NodeJS Introducción

Unidad 4: Express: conexión con la base de datos



Presentación:

Ahora que ya sabemos qué es una base de datos y sus componentes. Ahora que ya podemos crear una base de datos en MySQL es momento que conectarla con Express para poder, desde nuestra aplicación, guardar, consultar, modificar y borrar la información de nuestra aplicación.



Objetivos:

Que los participantes:

- Sean capaces de conectar Express con la base de datos MySQL.
- Aprendan a realizar consultas SQL desde Express a la base MySQL.



Bloques temáticos:

1. Consultas SQL básicas
2. Estructura de conexión a una base de datos SQL en Express
3. Conexión a la base de datos MySQL desde Express
4. Consultas a la base de datos desde Express
5. Ejemplos
6. Trabajo Práctico



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.

** El MEC es el modelo de E-learning colaborativo de nuestro Centro.*

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



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. Consultas SQL básicas

DML (Data Manipulation Language)

Son las sentencias utilizadas para administrar los datos dentro de las tablas

<i>Sentencia</i>	<i>Descripción</i>
SELECT	Permite recuperar registros
INSERT	Permite agregar nuevos registros
UPDATE	Permite actualizar registros
DELETE	Permite borrar registros

Select - Recuperar registros

```
SELECT <listado de columnas separadas por coma o *>  
FROM <nombre de la tabla>  
WHERE <listado de condiciones>
```

Ejemplo

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



<i>id</i>	<i>nombre</i>	<i>turno</i>
1	Professional Webmaster	Noche
2	Experto universitario en PHP	Tarde
3	Professional Webmaster	Tarde

```
SELECT id, nombre, turno
FROM curso
```

```
SELECT *
FROM curso
WHERE id=3
```

```
SELECT count(*)
FROM curso
```

```
SELECT nombre, turno
FROM curso
WHERE nombre like '%Professional%' AND id>1
```

INSERT - Agregar un registro

```
INSERT INTO <nombre de la tabla>
(<listado de columnas, separadas por coma>)
VALUES (<listado de valores, separados por coma>)
```

Ejemplo

```
INSERT INTO curso (nombre, turno) VALUES ('Webmaster 1', 'Noche');
```



UPDATE - Actualiza registros

```
UPDATE <nombre de la tabla>  
SET <nombre de campo>=<nuevo valor>, <nombre de campo>=<nuevo valor>, ...  
WHERE <listado de condiciones>
```

Ejemplo

```
UPDATE curso  
SET nombre='Webmaster introductorio'  
WHERE nombre='Webmaster 1';
```

DELETE - Borra registros

```
DELETE FROM <nombr de la tabla>  
WHERE <listado de condiciones>
```

Ejemplo

```
DELETE FROM curso  
WHERE nombre like '%introductorio%';
```

Operadores

Son palabras reservadas o un caracter utilizada en SQL que permite modificar el resultado

Se pueden utilizar para modificar el select o el where



OPERADORES

Operador	Descripción	Ejemplo
+	Suma	$10 + 5 \Rightarrow 15$
-	Resta	$10 - 5 \Rightarrow 5$
*	Multiplicación	$10 * 5 \Rightarrow 50$
/	División	$10 / 5 \Rightarrow 2$
%	Resto de la división	$10 \% 2 \Rightarrow 0$

COMPARACIÓN

Operador	Descripción	Ejemplo
=	Verifica que los valores sean iguales	$5 = 3 \Rightarrow \text{False}$
!= o <>	Verifica que los valores NO sean iguales	$5 != 3 \Rightarrow \text{True}$
<	Verifica que el valor de la izquierda sea menor al de la derecha	$5 < 3 \Rightarrow \text{False}$
>	Verifica que el valor de la izquierda sea mayor al de la derecha	$5 > 3 \Rightarrow \text{True}$
<=	Verifica que el valor de la izquierda sea menor o igual al de la derecha	$5 <= 3 \Rightarrow \text{False}$
>=	Verifica que el valor de la izquierda sea mayor o igual al de la derecha	$5 >= 3 \Rightarrow \text{True}$

Retornan Verdadero o Falso

Se utilizan en el where o if

LÓGICOS

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



<i>Operador</i>	<i>Descripción</i>	<i>Ejemplo</i>
AND	Verifica que se cumplan las dos condiciones	2 < 3 AND 3 < 5 => True
BETWEEN	Verifica que un valor esté entre otros dos	3 BETWEEN 2 AND 5 => True
IN	Verifica que un valor esté dentro de una lista de valores	2 IN (1,2,3,4,5) => True
LIKE	Verifica que un valor esté contenido en otro (string)	'Jorge Luis Borges' like '%Luis%' => True
NOT o !	Niega el resultado de la siguiente operación	3 NOT BETWEEN 2 AND 5 => False

Retornan Verdadero o Falso

Se utilizan en el where o if

FUNCIONES AGREGADAS

Permiten realizar operaciones sobre los resultados

<i>Función</i>	<i>Descripción</i>	<i>Ejemplo</i>
MAX	Retorna el valor máximo de los valores	MAX (20,100,180) => 180)
MIN	Retorna el valor mínimo de los valores	MIN (20,100,180) => 20
SUM	Retorna la suma de los valores	SUM (20,100,180) => 300
COUNT	Retorna la cantidad de valores	COUNT (20,100,180) => 3
AVG	Retorna el promedio entre los valores	AVG (20,100,180) => 100

Se utilizan en el select



2. Estructura de conexión a una base de datos SQL en Express

Para conectar una base de datos SQL en Express se requiere:

1. Incluir la conexión a Express
2. Incluir la biblioteca "mysql"
3. Generar la estructura de conexión
4. Conectar

La estructura de conexión es:

```
//Para trabajar con base de datos
var mysql = require('mysql');
var conexion = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'prueba'
});
conexion.connect();
```

Se debe indicar el nombre de la base de datos (como se creo en PhpMyAdmin) y los datos de host, usuario y contraseña que también son definidos previamente en el PhpMyAdmin.



3. Conexión a la base de datos MySQL desde Express

```
1 var express = require('express');  
2 var app = express();
```

```
//Para trabajar con base de datos  
var mysql = require('mysql');  
var conexion = mysql.createConnection({  
  host: 'localhost',  
  user: 'root',  
  password: '',  
  database: 'prueba'  
});  
conexion.connect();
```



4. Consultas a la base de datos desde Express

```
35 app.post('/login', function(req, res) {  
36  
37   if(!req.body.usuario || !req.body.clave) {  
38     res.render('login', {error:'No envio los datos'});  
39     //res.send('No envio los datos');  
40     return;  
41   }  
42  
43   conexion.query('SELECT * FROM usuario WHERE usuario = ? AND password = ?', [req.body.usuario, req.body.clave],  
44     function(error, registros, campos){  
45       if(error) {  
46         res.send('Error al hacer el query');  
47         return;  
48       }  
49  
50       if (registros.length == 1) {  
51         req.session.usuario_id = registros[0].id; // creamos la variable de session  
52         res.send('Ya se encuentra logueado');  
53       }  
54       else {  
55         res.send('Error en el logueo');  
56       }  
57     });  
58  
59  
60   });  
61
```


5. Ejemplo

Desde el foro dispondrán el enlace a un video con un ejemplo.



6. Trabajo Práctico

Desarrollar un servidor que reciba un formulario y almacene la información (previamente validada), en una base de datos MySQL. Generar un método para recibir una petición GET a la que se le devolverá un listado con todos los datos de la tabla.

Pueden elegir libremente la temática del formulario y por ende, de la/s tabla/s de la base de datos.



Bibliografía utilizada y sugerida

Express (n. d.) Recuperado de: <https://expressjs.com/es/>

MDN - JavaScript (n.d.) Recuperado de:

<https://developer.mozilla.org/es/docs/Web/JavaScript>

NodeJs (n. d.) Recuperado de: <https://nodejs.org/es/>

NodeJS Documentacion (n. d.) Recuperado de: <https://nodejs.org/es/docs/>

NPM (n. d.) Recuperado de: <https://www.npmjs.com/package/page>

Wikipedia - NodeJS (n. d.) Recuperado de: <https://es.wikipedia.org/wiki/Node.js>



Lo que vimos:

Conexión desde Express a una base de datos MySQL.



Lo que viene:

NodeJS Intermedio.

