

## A03 – Ultima actividad con Angular

Las dos actividades anteriores fueron fáciles de realizar pues eran simples y guiadas paso a paso, pues el propósito de ellas era conocer la herramienta y aprender lo básico.

Esta actividad no será tan fácil, y por eso te solicito que comiences a realizarla desde ya.

Pensando en esta dificultad adicional, y que son feriado el jueves y viernes 2, el plazo para realizar esta actividad será:

Avance	hasta las 22:00 de lunes 5 de abril	(30%)
Final	hasta las 22:00 del domingo 11 de abril	(70%)

### Descripción: Aplicación Angular que muestre datos obtenidos desde una tabla MySQL

Para esta actividad puedes usar la misma aplicación ya desarrollada o una nueva, diseñada por ti.

#### Primero crea una Base de Datos MySQL, que posea una tabla con al menos cuatro campos

- Si no tienes XAMPP descárgalo desde <https://www.apachefriends.org/es/download.html>
- Instala e inicia XAMPP. Durante la instalación quita la marca de chequeo a las opciones:
  - FileZilla FTP Server
  - Mercury Mail Server
  - Perl
  - Webalizer
  - Fake Sendmail
- Ingresa a la carpeta **C:\xampp**, allí ejecuta el **programa xampp-control** con permisos de administrador.
  - Desde ese panel inicia **Apache** y **MySQL** (si se levanta el firewall autoriza acceso)
  - Si marca error al iniciar Apache por estar ocupado el **puerto 80** averigua que programa lo está usando y abórtalo si es posible. Lo mismo si marca error al iniciar MySQL
- Estando **Apache** y **MySQL** correctamente iniciados dale **Clic** al botón **Admin** de **Apache**, debería aparecer la página principal de XAMPP (si no es así no podrás seguir).
- En esa página dale **Clic** a la opción **phpMyAdmin**, debería aparecer un panel de control desde el cual podrás crear una nueva Base de Datos
- Crea una base de datos con una tabla de **al menos cuatro campos**
  - **Clic** en **Nueva** (al lado), en la ranura escribe un **nombre** para tu Base de Datos, **Clic** en **Crear**
  - En ranura **Nombre** escribe el nombre que deseas para la tabla que vas a crear
  - En la ranura **Número de Columnas** escribe un valor igual o **superior a 4**, **Clic** en **Continuar**
  - Define los campos de tu tabla. Para cada campo define su **Nombre**, **Tipo** y **Longitud** (si lo requiere)
  - Uno de esos campos debe ser **CLAVE PRIMARIA** (para poder buscar un registro)
  - **Clic** en **Guardar**
- **Clic** en **Insertar** (arriba) y entonces inventa datos para, al menos, **diez** registros de dato
- **Clic** en **Exportar** (arriba), **Clic** en **Continuar**, guarda en un lugar seguro (**deberás entregar esta exportación junto con tu proyecto, para que yo pueda revisarlo**)

Luego crea un script PHP que lea los datos de esa tabla cuando los solicite la aplicación Angular

Comparto un script que funciona, pero no es obligación usar este. En este script se hace una consulta a una tabla **alumnos**, que tiene un campo clave de nombre **codigo**, en una base de datos de nombre **ej**. Este script, si encuentra ese registro, devuelve el contenido de un campo **nombre**.

Este es solo un ejemplo (en un script que devuelva datos a una aplicación Angular quizá sería mejor que entregue esos datos en formato JSON)

`<?php`

```
if(isset($_GET['cod'])) {  
    $codigo = $_GET['cod'];  
    $nombre = "No encontrado";  
    $mysqli = new mysqli("localhost", "root", "", "ej");  
    $sql = "SELECT nombre FROM alumnos WHERE codigo='".$codigo."'";  
    $result = $mysqli->query($sql);  
    $numfi = $result->num_rows;  
    if($numfi>0) {  
        $registro = $result->fetch_assoc();  
        $nombre = $registro["nombre"];  
    }  
    echo $nombre;  
    $mysqli->close();  
}
```

Después de...  
WHERE codigo=

hay un APOSTROFE, a  
continuación COMILLA

Después de . \$codigo.

hay UNA COMILLA,  
luego APOSTROFE y al  
final COMILLA

`?>`

- Crea en C:\xampp\htdocs una carpeta de nombre Ejemplo
- Escribe este script y guárdalo en C:\xampp\htdocs\Ejemplo con nombre buscar.php

### Explicaciones

Con `if(isset($_POST['cod']))` se logra que el código contenido entre las llaves del `if` se ejecute solo si ha sido recibido un parámetro de nombre **cod** mediante la petición **POST**.

Con `$codigo = $_POST['cod'];` se asigna a la variable **\$codigo** el valor que contiene el parámetro **cod**

Con `$mysqli = new mysqli("localhost", "root", "", "ej");` se crea una conexión a la base de datos. Los parámetros que necesita esta instrucción son la **dirección del servidor** (en este caso es **localhost**), el **nombre del usuario con permiso** para acceder a la base de datos (durante desarrollo es **root**), la contraseña de este usuario (sin contraseña en este caso) y el nombre de la base de datos (**ej** en este caso)

Con `$sql = "SELECT nombre FROM alumnos WHERE codigo='".$codigo."'";` se seleccionan desde la tabla **alumnos** de esa base de datos todos los registros que en su campo **codigo** tengan el mismo valor que contiene la variable **\$codigo** (como el código es de tipo VARCHAR fue encerrado entre apóstrofes)

Con `$result = $mysqli->query($sql);` se obtienen todos los registros que cumplan esa condición (sabemos que en este ejercicio será como máximo uno solo, pues el campo **codigo** es clave primaria)

Con `$numfi = $result->num_rows;` se obtiene la cantidad de registros que contiene **\$result** (sabemos que en este ejercicio será como máximo uno solo)

Con `$registro = $result->fetch_assoc();` se obtiene el registro contenido en **\$result**

## Comprueba el funcionamiento de este script de ejemplo

- Crea una base de datos de nombre **ej**
- Crea en ella una tabla de nombre **alumnos**, con **dos** campos
- Define en ella los campos **codigo** de tipo **VARCHAR** largo **2**, que debe ser **CLAVE PRIMARIA**, y el campo **nombre** de tipo **VARCHAR** de largo **20**.
- Digita varios registros de dato
- **Momentáneamente** modifica el archivo consultar.php cambiando **POST** por **GET** en dos de sus filas
- Ahora, en la ranura de direcciones de tu navegador, escribe **localhost/Ejemplo/buscar.php?cod="01"**  
Si tienes un registro con ese código mostrará el **nombre**, de lo contrario el texto **No encontrado**
- Si funciona edita **consultar.php** y repone **POST** en las dos líneas en que aparece **GET**

Explicación: Se usó GET para poder probar una consulta digitada directamente en el navegador, pero se usará POST para hacer consultas desde la aplicación Angular

## Ahora la parte "difícil" de esta actividad

Averigua en Internet como hacer desde Angular una llamada a un script php para buscar un cierto registro dentro de una Base de Datos, mostrando toda la información que contiene, si lo encuentra, o una advertencia en caso contrario.

Es decir, esta última aplicación es casi igual a las dos anteriores, pero ahora los datos no estarán definidos de modo fijo en el código de la aplicación, sino que **serán obtenidos desde tu Base de Datos para ser mostrados en la página**

Es decir que, la parte "difícil" de esta actividad, es solo esa... **como obtener los datos entregados por el script php**, pues mostralos en pantalla ya lo hiciste en las dos actividades anteriores

- No descuides el diseño visual de tu aplicación
- No olvides que debes entregar un avance antes de las 22:00 del lunes 5

## Importante:

Al ejecutar el proyecto Angular el servidor que se inicia no es el Apache que hay en **C:\xampp**. Por lo tanto, el **script PHP** no deberás colocarlo en la carpeta **C:\xampp\htdocs** sino que **en una creada en la misma carpeta de tu proyecto Angular** (esto forma parte de lo que deberás averiguar y experimentar)

## ¿Qué debes entregar?

Un archivo comprimido conteniendo:

- La **carpeta src** del proyecto Angular
- La **exportación SQL** de tu **Base de Datos**
- Si estimas necesario explicar algo entonces agrega un archivo TXT