**Usando MySQL en React-Native con PHP**

Primero hacemos un archivo PHP en tu servidor donde tengas PHPMyAdmin, en mi caso lo hice en mi servidor que tengo, dejo el código:

config.php

<?php

//Uso un header para tener permiso a acceder a los archivos

header("Access-Control-Allow-Origin: \*");

//Realizo función de conexión a la base de datos

**function** conecta\_db(){

$servidor = "localhost";

$usuario = "root";

$clave = "";

$db = "platziVideo";

$conn = mysqli\_connect($servidor, $usuario, $clave, $db);

if(mysqli\_connect\_errno()){

echo mysqli\_connect\_error();

exit(0);

}

return $conn;

}

//Obtengo la opción que elegí cuando hago el llamado en React-Native, puedes usar un "isset" si fuera el caso

$opcion = $\_GET["opcion"];

//Uso la función de conexión

$conn = conecta\_db();

//Realizo un **switch**, ya que como en mi caso necesitare varias opciones de búsqueda; esto, depende de la variable opción.

**switch**($opcion){

case 1:

$sql = "SELECT \* from movie";

break;

}

//Realizo la consulta

$rs = mysqli\_query($conn, $sql);

//Creo un **array**, el nombre va a tu gusto.

$array = **array**();

//Pregunto si la consulta es correcta.

**if**($rs){

//Hago ciclo "mientras" para obtener los datos ""mientras" la consulta exista, importante que sea por assoc, ya que solo te devuelve una forma de parametro

while($fila = mysqli\_fetch\_assoc($rs)){

//Guardo los resultados mapeados en el array creado

$array[] = array\_map('utf8\_encode', $fila);

}

//Codifico el array anterior en una variable para que lo retorne

$res = json\_encode($array, JSON\_NUMERIC\_CHECK);

}else{

//Si la consulta no se hizo me retorna la variable como null

$res = null;

echo mysqli\_error($conn);

}

//Cierro la consulta

mysqli\_close($conn);

//Hago que el archivo PHP (si lo necesito abrir para hacer pruebas) me lo devuelva como json

header('Content-Type: application/json');

//Imprimo la variable con el array codificado y con el null, según sea el caso.

echo $res;

?>

Espero que hasta aquí vayamos bien, ahora voy a mostrar el código que use para obtener la búsqueda de PHP y es la siguiente:

//Se hace la constante para guardar la ruta de donde esta el archivo PHP con la busqueda

**const** BASE\_API = 'http://tuservidordepreferencia/carpetasilanecesitas/';

class Api {

//Hacemos la función asincrona

**async** getSuggestion(){

//Hacemos una constante donde haremos toda la consulta de la consulta (redundante no?), fijate que despues del nombre del archivo va "?opcion=1" esto es importante ya que con esto podemos elegir que acción hará el archivo de consulta en PHP.

**const** query = **await** fetch(`${BASE\_API}nombredelarchivo.php?opcion=1`, {

//Le indico que la consulta es de forma POST

method: 'POST',

//Le indico que tipo de consulta va a obtener o su contenido

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

//Le indico que el cuerpo es un JSON y quiero que lo convierta así.

body: JSON.stringify()

}).then((response) => response.json()) //Si es así que realice un tipo mapeo

.then((data) => { //Si obtiene los datos que lo guarde o haga un callback en "data"

//Hago retornar "data" para que se guarde en la constante "query"

**return** data

}).catch((error) => {

//Si no obtiene nada que me lo arroje en consola

console.error(error);

});

// Aqui ya solo retorno la constante "query" para que DidMount lo recupere y obtenga los datos y lo arroje en los estados.

**return** query;

}

**export** **default** **new** Api;

Bueno de esta forma fue como realice mi pequeña prueba a usar MySQL en React-Native con PHP, esto fue por el proyecto que quiero hacer y los datos están en SQL.