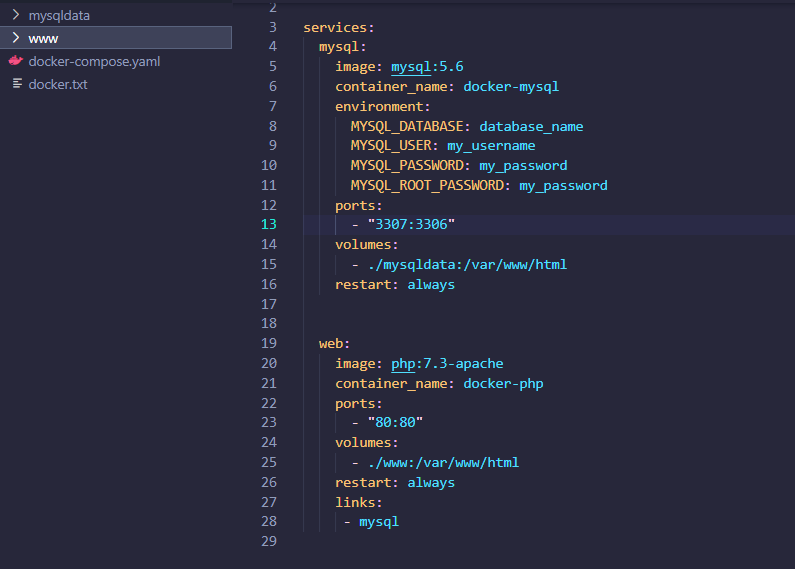
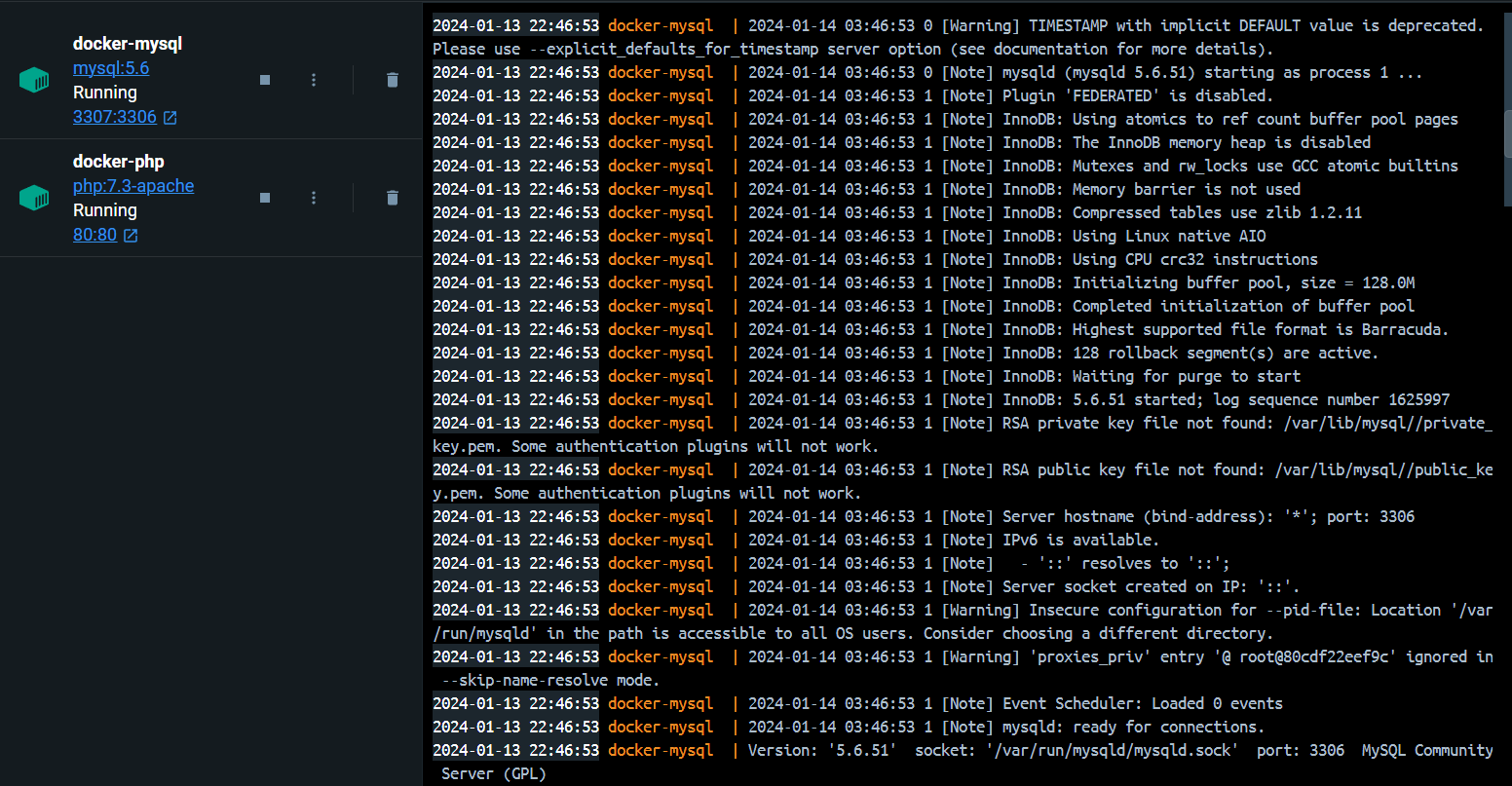
Manejo y uso de docker con comandos

Pasos.

1. Intalamos Docker
2. En visual estudio creamos las diferentes carpetas y el Docker-compose para la instalación de php y mysql en el Docker junto con el archivo .yaml
3. Verificamos la instalación por Docker y por consola de vb.

Texto

Descripción generada automáticamente

Cabe resaltar que se guarda el archivo Docker-compose creado y se ejecuta el comando up- d para la instalación del mismo, hay que verificar cada uno de los puertos en la sintaxis de código:

services:

  mysql:

    image: mysql:5.6

    container\_name: docker-mysql

    environment:

      MYSQL\_DATABASE: database\_name

      MYSQL\_USER: my\_username

      MYSQL\_PASSWORD: my\_password

      MYSQL\_ROOT\_PASSWORD: my\_password

    ports:

      - "3307:3306"

    volumes:

      - ./mysqldata:/var/www/html

    restart: always

  web:

    image: php:7.3-apache

    container\_name: docker-php

    ports:

      - "80:80"

    volumes:

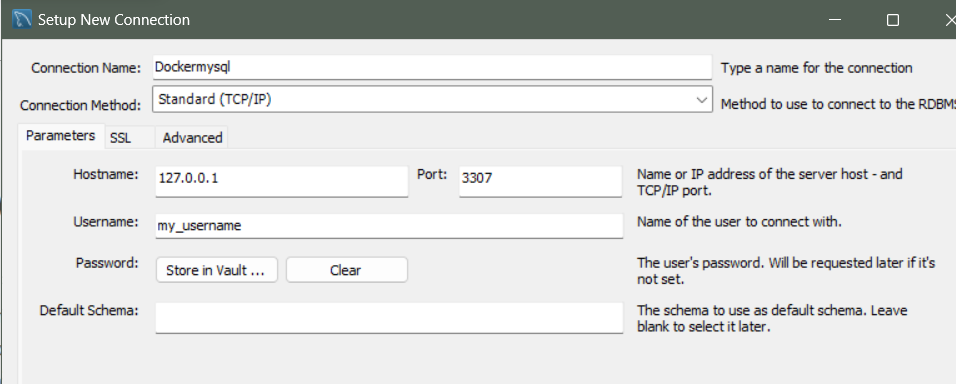
      - ./www:/var/www/html

    restart: always

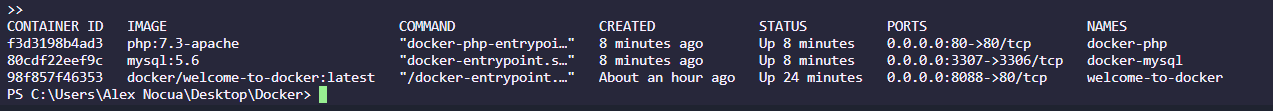
    links:

     - mysql

En este caso me genero error por el puerto 3306 que esta por defecto, esto se cambia junto con el 8080 según los puertos que estén en el Docker y así mismo se configuran en myworkbrech.



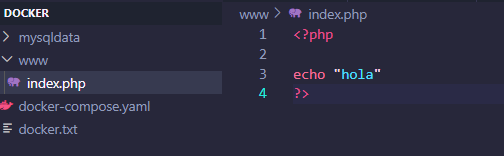
Con el comando docker ps se revisa si nuestro Docker esta funcionando.



Luego de eso creamos e ingresamos a workbrech, teniendo consigo nuestra base de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Luego de esto creamos nuestro index.php y verificamos en nuestro localhost que este funcionando. 

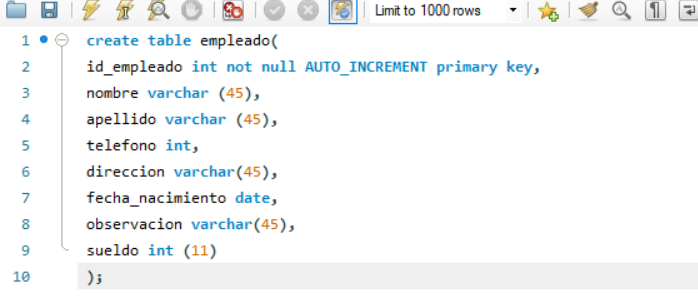
Interfaz de usuario gráfica, Aplicación

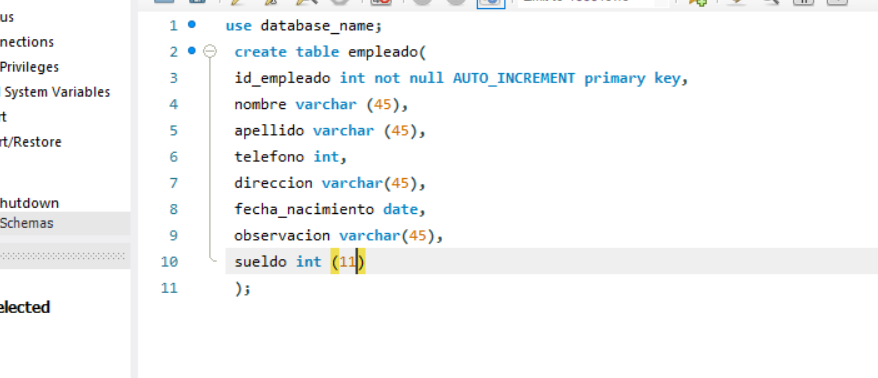
Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

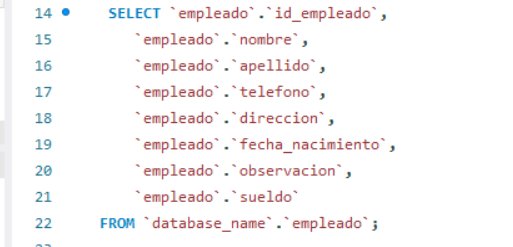
Con esto ahora creamos una tabla en myworkbrechs





Verificamos que todo este bien Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Creamos una sentend¿cia de código 

Y vemos que todo este funcionando.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Conexión con la base de datos.

Creamos el siguiente código en php para verificar la conexión con la base de datos.

<?php

*// coneccion con una base de datos*

$conexion = mysqli\_connect("mysql", "my\_username", "my\_password", "database\_name", "3307");

if (mysqli\_connect\_errno()) {

    echo "la coneccion con la base de datos ha fallado:" . mysqli\_connect\_errno();

} else {

    echo "La coneccion con la base de datos fue correcta";

}

*//consulta para configurar la cogificacion de caracteres*

mysqli\_query($conexion, "SET NAMES 'utf8'" )

?>

Hahora necesitamos instalar una utilidad “mysqli” en nuestro Docker-php

Para eso rectificamos que todo se este ejecutando bien con Docker ps y luego ejecutamos el siguiente comando.

docker exec -it docker-php bash

esto para ver si esta instalada esta utilidad en nuestro contenedor.

con esto ahora estamos en comando de Linux, ahora ejecutamos el siguiente comando

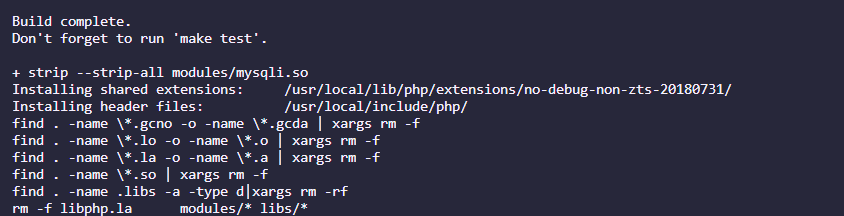
Docker-php-ext-enable mysqli

Texto

Descripción generada automáticamente

No existe, entonces para instalarlo ingresamoe el siguiente comando.

docker-php-ext-install mysqli



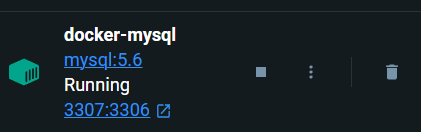
Con esto sabemos que la instalación esta completa

Ahora salimos del contenedor, recordemos que cuando ingresemos en el contenedor se manejan comandos de Linux.

Para salir de este solo con el comando exit.

Con eso tedremos nuestro script para la conexión con la base de datos.

Por lo general genera un error ya que nosotros colocamos el puerto 3307 este se cambia a 3306 y se verifica que todas las conexiones con las bases de datos estén correctas, aquí esto se cambia debido a que el nuestro docker tenemos esa puerta:



Por eso se debe colocar la que esta 3306

Ahora creamos el siguiente código con cada una de las sentencias de coneccion con la base de datos.

<?

if (isset($\_POST)) {

    require\_once 'conexion.php';

*//mysqli\_real\_escape\_string con esto lombiamos sin tener espacios : si no se cumple o vienw un parametro vacio este se llenaria como falso.*

    $nombre = isset($\_POST['nombre']) ? mysqli\_real\_escape\_string($conexion, $\_POST['nombre']) : false;

    $apellido = isset($\_POST['apellido']) ? mysqli\_real\_escape\_string($conexion, $\_POST['apellido']) : false;

    $telefono = isset($\_POST['telefono']) ? mysqli\_real\_escape\_string($conexion, $\_POST['telefono']) : false;

    $direccion = isset($\_POST['direccion']) ? mysqli\_real\_escape\_string($conexion, $\_POST['direccion']) : false;

    $fecha\_nacimiento = isset($\_POST['fecha\_nacimiento']) ? mysqli\_real\_escape\_string($conexion, $\_POST['fecha\_nacimiento']) : false;

    $observacion = isset($\_POST['observacion']) ? mysqli\_real\_escape\_string($conexion, $\_POST['observacion']) : false;

    $sueldo = isset($\_POST['sueldo']) ? mysqli\_real\_escape\_string($conexion, $\_POST['sueldo']) : false;

    $errores = array(); *//en este se agregan los errores*

    if (!empty($nombre) && !is\_numeric($nombre) && !preg\_match("/[0.9]/", $nombre)) { *// si no viene vacio y no es un numero y por medio de una exprecion regular si no viene un numero del 0 añl 9*

        $nombre\_validado = true;

    } else {

        $nombre\_validado = false;

        $errores['nombre'] = "El nombre no es valido.";

    }

    if (!empty($apellido) && !is\_numeric($apellido) && !preg\_match("/[0.9]/", $apellido)) {

        $apellido\_validado = true;

    } else {

        $apellido\_validado = false;

        $errores['apellido'] = "El apellido no es valido.";

    }

    if (!empty($telefono) && is\_numeric($telefono)) { *// se le quita el ! ya que debe de ser un numero, igual la exprecion regular.*

        $telefono\_validado = true;

    } else {

        $telefono\_validado = false;

        $errores['telefono'] = "El telefono no es valido.";

    }

    if (!empty($direccion) && !is\_numeric($direccion)) {

        $direccion\_validado = true;

    } else {

        $direccion\_validado = false;

        $errores['direccion'] = "La direccion no es valida";

    }

    if (!empty($fecha\_nacimiento)) {

        $fecha\_nacimiento\_validado = true;

    } else {

        $fecha\_nacimiento\_validado = false;

        $errores['fecha\_nacimiento'] = "La fecha de nacimiento no es valida";

    }

    if (!empty($observacion)) {

        $observacion\_validado = true;

    } else {

        $observacion\_validado = false;

        $errores['observacion'] = "La observacion no es valida";

    }

    if (!empty($sueldo) && is\_numeric($sueldo)) {

        $sueldo\_validado = true;

    } else {

        $sueldo\_validado = false;

        $errores['sueldo'] = "El sueldo no es valido";

    }

    if (count($errores) == 0) {

        $sql = "insert into empleado (nombre, apellido, telefono, direccion, fecha\_nacimiento, observacion, sueldo) values ('$nombre','$apellido',$telefono,'$direccion','$fecha\_nacimiento','$observacion','$sueldo')";

*// si el dato no es numeroco se deben colocar la comillas simples, caso contrario con telefono*

        $guardar = mysqli\_query($conexion, $sql); *//esta es una fincion que necesita la sentencia sql y la coneccion*

        echo "Guardado con exito";

    } else {

        foreach ($errores as $val) {

            echo $val;

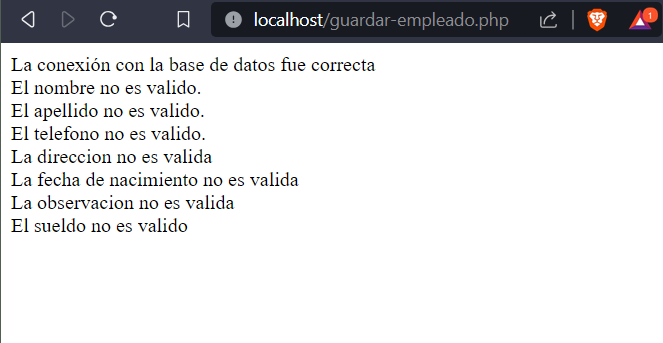
            echo "<br>";

        }

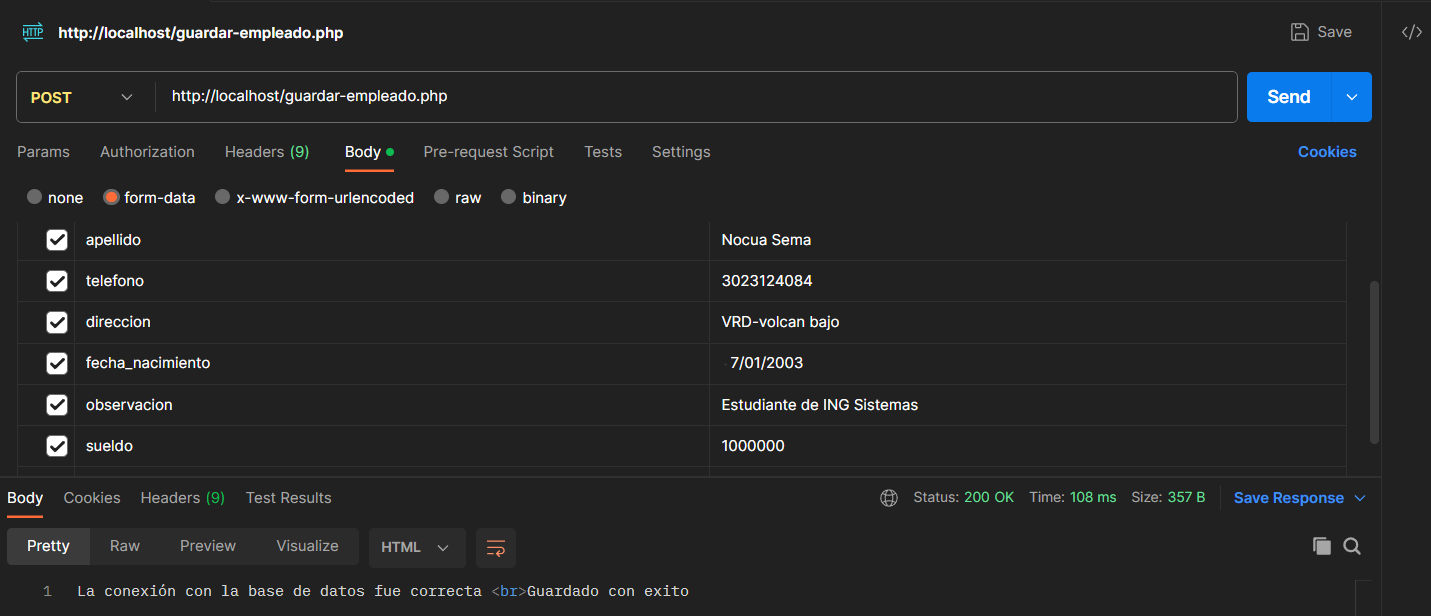
    }

}

Testeamos que todo este correcto añadiendo la dirección en nuestro local host:



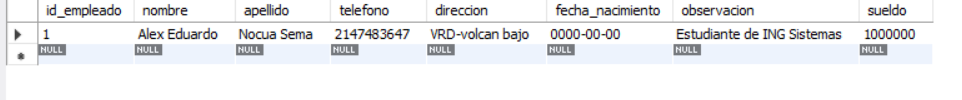
Con esto accedemos ahora a postman para ingresar nuestros datos desde allí.



Seleccionamos post y pegamos la dirección de nuestro local host

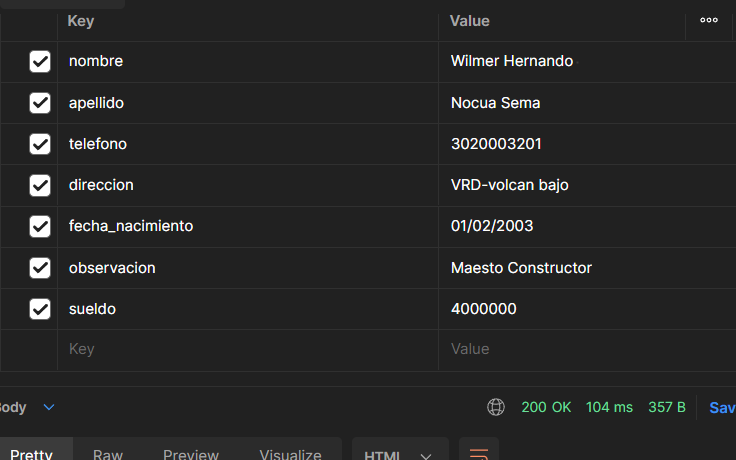
Podemos hacer unas pruebas con send ya que no hemos instancido nuestras variables, con esto para verificar si el mensaje que arroja es el correcto.

Después de esto creamos las llaves nombre, apellido… con el fin de que en el valor de las mismas llaves se ingrese la información y al momento de dar send este se guarde automáticamente en la base de datos, cabe resaltar que el nombre de estas llaves debe de ser el mismo de las variables que nombramos en nuestro scrip de php.



Prueba guardado de datos.

Otro ejemplo.





Mostrar usuarios: para esto se crea el siguiente código

<?php

require\_once 'conexion.php';

*//con esto vamos a ver los regfistros de la base de datos*

$query = mysqli\_query($conexion, "select \* from empleado"); *//con esta linea se peden colocar consultas o lineas de comandos*

*// esta es la estructura de la tabla.*

echo '<table >

    <thead>

    <tr>

    <th>nombre </th>

    <th>apellido </th>

    <th>telefono </th>

    <th>direccion </th>

    <th>fecha\_nacimiento </th>

    <th>observacion </th>

    <th>sueldo </th>

    </tr>

    </thead>

    <tbody>';

while ($empleado = mysqli\_fetch\_assoc($query)) { *//la variable $ con esta funcion vamos a recorrer todo lo que traiga la tabla empelado y lo va a guardar en arrays*

    echo '<tr>';

*//esto es un registo en filas sin estilo en css*

    echo '<td>' . $empleado['nombre'] . '</td>';

    echo '<td>' . $empleado['apellido'] . '</td>';

    echo '<td>' . $empleado['telefono'] . '</td>';

    echo '<td>' . $empleado['direccion'] . '</td>';

    echo '<td>' . $empleado['fecha\_nacimiento'] . '</td>';

    echo '<td>' . $empleado['observacion'] . '</td>';

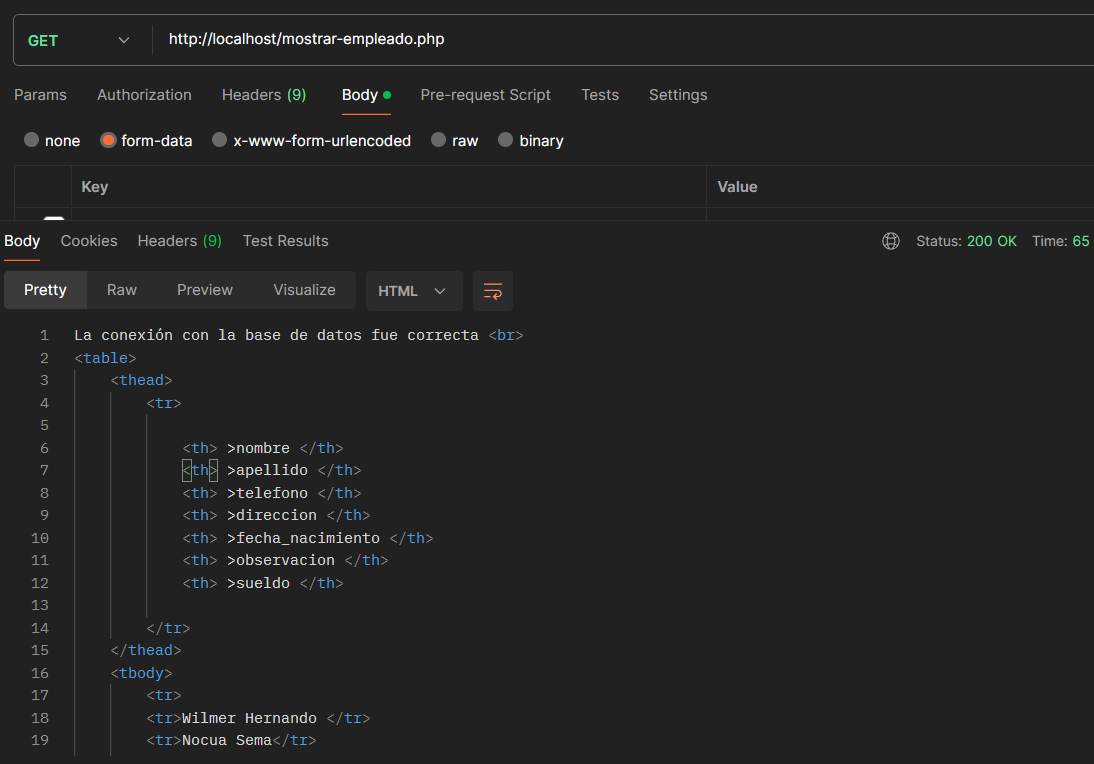
    echo '<td>' . $empleado['sueldo'] . '</td>';

    echo '</tr>';

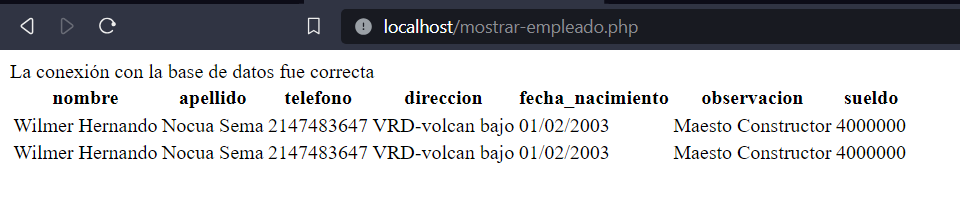
}

echo '</tbody></table>' *//cearramos la estructura*

    ?>



Escogemos gen en postgrets y nos tiene que dar esa estructura, con ello verificamos en la url de navegador.



Actualizar información

Para actualizar la información se crea el siguiente código:

<?

if (isset($\_POST)) {

    require\_once 'conexion.php';

*//mysqli\_real\_escape\_string con esto lombiamos sin tener espacios : si no se cumple o vienw un parametro vacio este se llenaria como falso.*

    $id\_empleado = isset($\_POST['id\_empleado']) ? mysqli\_real\_escape\_string($conexion, $\_POST['id\_empleado']) : false; *// en caso de existir un emplesdo lo guarde en esa variable*

    $nombre = isset($\_POST['nombre']) ? mysqli\_real\_escape\_string($conexion, $\_POST['nombre']) : false;

    $apellido = isset($\_POST['apellido']) ? mysqli\_real\_escape\_string($conexion, $\_POST['apellido']) : false;

    $telefono = isset($\_POST['telefono']) ? mysqli\_real\_escape\_string($conexion, $\_POST['telefono']) : false;

    $direccion = isset($\_POST['direccion']) ? mysqli\_real\_escape\_string($conexion, $\_POST['direccion']) : false;

    $fecha\_nacimiento = isset($\_POST['fecha\_nacimiento']) ? mysqli\_real\_escape\_string($conexion, $\_POST['fecha\_nacimiento']) : false;

    $observacion = isset($\_POST['observacion']) ? mysqli\_real\_escape\_string($conexion, $\_POST['observacion']) : false;

    $sueldo = isset($\_POST['sueldo']) ? mysqli\_real\_escape\_string($conexion, $\_POST['sueldo']) : false;

    $errores = array(); *//en este se agregan los errores*

    if (!empty($id\_empleado) && is\_numeric($id\_empleado)) { *// si no viene vacio y no es un numero*

        $id\_validado\_validado = true;

    } else {

        $id\_validado\_validado = false;

        $errores['id\_empleado'] = "El id\_empleado no es valido.";

    }

    if (!empty($nombre) && !is\_numeric($nombre) && !preg\_match("/[0.9]/", $nombre)) { *// si no viene vacio y no es un numero y por medio de una exprecion regular si no viene un numero del 0 añl 9*

        $nombre\_validado = true;

    } else {

        $nombre\_validado = false;

        $errores['nombre'] = "El nombre no es valido.";

    }

    if (!empty($apellido) && !is\_numeric($apellido) && !preg\_match("/[0.9]/", $apellido)) {

        $apellido\_validado = true;

    } else {

        $apellido\_validado = false;

        $errores['apellido'] = "El apellido no es valido.";

    }

    if (!empty($telefono) && is\_numeric($telefono)) { *// se le quita el ! ya que debe de ser un numero, igual la exprecion regular.*

        $telefono\_validado = true;

    } else {

        $telefono\_validado = false;

        $errores['telefono'] = "El telefono no es valido.";

    }

    if (!empty($direccion) && !is\_numeric($direccion)) {

        $direccion\_validado = true;

    } else {

        $direccion\_validado = false;

        $errores['direccion'] = "La direccion no es valida";

    }

    if (!empty($fecha\_nacimiento)) {

        $fecha\_nacimiento\_validado = true;

    } else {

        $fecha\_nacimiento\_validado = false;

        $errores['fecha\_nacimiento'] = "La fecha de nacimiento no es valida";

    }

    if (!empty($observacion)) {

        $observacion\_validado = true;

    } else {

        $observacion\_validado = false;

        $errores['observacion'] = "La observacion no es valida";

    }

    if (!empty($sueldo) && is\_numeric($sueldo)) {

        $sueldo\_validado = true;

    } else {

        $sueldo\_validado = false;

        $errores['sueldo'] = "El sueldo no es valido";

    }

    if (count($errores) == 0) {

*// se cqambia esto para la actualizacion de los datos.*

        $sql = "update  empleado set nombre ='$nombre',apellido ='$apellido',telefono=$telefono,direccion='$direccion',fecha\_nacimiento='$fecha\_nacimiento',observacion = '$observacion',sueldo='$sueldo' where id\_empleado = $id\_empleado";

*// si el dato no es numeroco se deben colocar la comillas simples, caso contrario con telefono*

        $guardar = mysqli\_query($conexion, $sql); *//esta es una fincion que necesita la sentencia sql y la coneccion*

        echo "Actualizado con exito";

    } else {

        foreach ($errores as $val) {

            echo $val;

            echo "<br>";

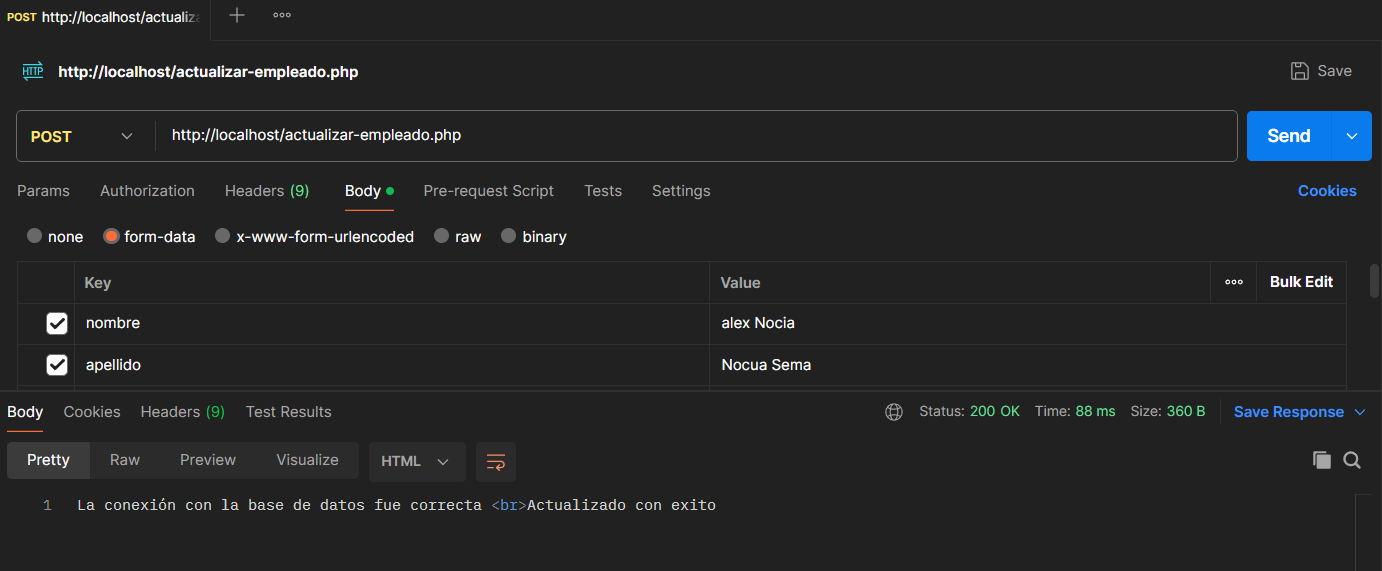
        }

    }

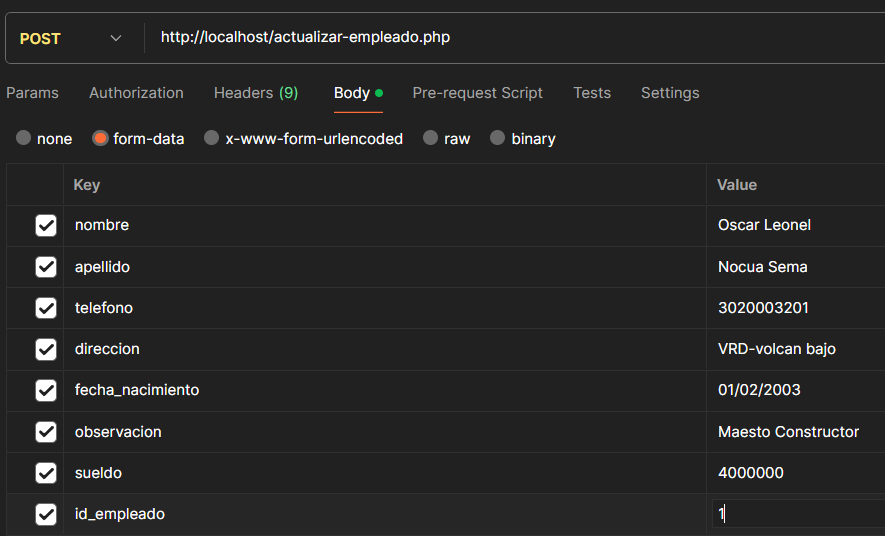
}

En este se crea la variable id\_empleado y así mismo se hacen los respectivos ajustes, es muy parecido al código de agregar empleado.

Para acceder a este en postman se hace de la siguiente manera:



En este caso el id 1 tiene esa información incorrecta, entonces selecciono el id 1 donde esta y lo cambio por oscar leonel:



Recuerde siempre colocar el id que queremos actualizar.

Actualizamos la tabla: Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamente

Eliminar

Para esto utilizamos el siguiente código:

<?

if (isset($\_POST)) {

    require\_once 'conexion.php';

*//mysqli\_real\_escape\_string con esto lombiamos sin tener espacios : si no se cumple o vienw un parametro vacio este se llenaria como falso.*

    $id\_empleado = isset($\_POST['id\_empleado']) ? mysqli\_real\_escape\_string($conexion, $\_POST['id\_empleado']) : false; *// en caso de existir un emplesdo lo guarde en esa variable*

    $errores = array(); *//en este se agregan los errores*

    if (!empty($id\_empleado) && is\_numeric($id\_empleado)) { *// si no viene vacio y no es un numero*

        $id\_validado\_validado = true;

    } else {

        $id\_validado\_validado = false;

        $errores['id\_empleado'] = "El id\_empleado no es valido.";

    }

    if (count($errores) == 0) {

*// se elimiminar una fila por el ID*

        $sql = "delete from empleado where id\_empleado = $id\_empleado"; *// cambiamos esa consulta*

*// si el dato no es numeroco se deben colocar la comillas simples, caso contrario con telefono*

        $guardar = mysqli\_query($conexion, $sql); *//esta es una fincion que necesita la sentencia sql y la coneccion*

        echo "Eliminado con exito";

    } else {

        foreach ($errores as $val) {

            echo $val;

            echo "<br>";

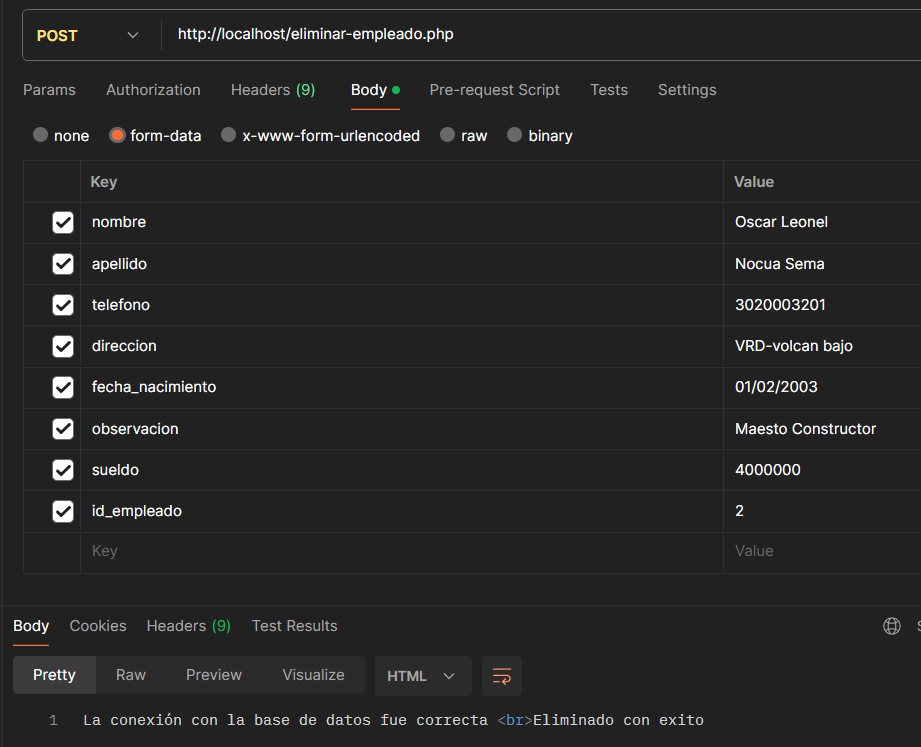
        }

    }

}

?>

Con esto solo cambiamos el comando de sql para delete, utilizamos el post en postman y verificamos.



En este caso borramos los registros del id 2 que quedaba y rectificamos en la base de datos:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente