



MANUAL TÉCNICO

**DESARROLLO DEL MANUAL TÉCNICO DEL SISTEMA WEB PARA LA GESTIÓN
DE CLIENTES, PRODUCTOS Y SERVICIOS DEL GIMNASIO “HOOLIGAN’S
GYM”**

ÍNDICE

CARÁTULA	1
1. INTRODUCCION	4
2. OBJETIVO	4
3. REQUERIMIENTOS FUNCIONAMIENTOS.....	4
4. INSTALACIÓN SUBLIME TEXTS	5
5. INSTALACIÓN XAMPP	8
6. CODIFICACIÓN	15

ÍNDICE DE FIGURA

Figura 1 Primer paso de Instalación Sublime Text.....	5
Figura 2 Segundo paso de Instalación de Sublime Text.....	5
Figura 3 Tercer paso de Instalación de Sublime Text	6
Figura 4 Cuarto paso de Instalación de Sublime Text.....	6
Figura 5 Quinto paso de Instalación de Sublime Text.....	7
Figura 6 Sexto paso de Instalación de Sublime Text.....	7
Figura 7 Primer paso de Instalación XAMPP	8
Figura 8 Segundo paso de Instalación XAMPP	8
Figura 9 Tercer paso de Instalación XAMPP	9
Figura 10 Cuarto paso de Instalación XAMPP	9
Figura 11 Quinto paso de Instalación XAMPP	10
Figura 12 Sexto paso de Instalación XAMPP	10
Figura 13 Séptimo paso de Instalación XAMPP	11
Figura 14 Octavo paso de Instalación XAMPP	11
Figura 15 Noveno paso de Instalación XAMPP.....	12
Figura 16 Decimo paso de Instalación XAMPP.....	12
Figura 17 Creación de una Base de Datos.....	13
Figura 18 Nombre de la Base de Datos	13
Figura 19 Creación de Campos	14
Figura 20 Tabla de base de datos.....	14

1. INTRODUCCION

La finalidad de todo manual técnico es la de proporcionar al lector las pautas de configuración y la lógica con la que se ha desarrollado una aplicación, la cual se sabe que es propia de cada programador; por lo que se considera necesario ser documentada.

Aclarando que este manual no pretende ser un curso de aprendizaje de cada una de las herramientas empleadas para el desarrollo del sitio, sino documentar su aplicación en el desarrollo del sitio. Para un mayor detalle acerca de cada una de las herramientas utilizadas, y su forma de operación.

2. OBJETIVO

Proporcionar una guía para el lector, del desarrollo de la interfaz y de la instalación del sistema web.

3. REQUERIMIENTOS FUNCIONAMIENTOS

Los siguientes requerimientos de este sistema son para el correcto funcionamiento del sistema web.

➤ Navegadores

Un buen navegador de internet actualizado permite el acceso óptimo al sistema web. En este caso se recomienda el uso de Google Chrome.

➤ SUBLIME TEX

Se recomienda Sublime Text es un editor de texto para escribir código en casi cualquier formato de archivo

➤ XAMPP

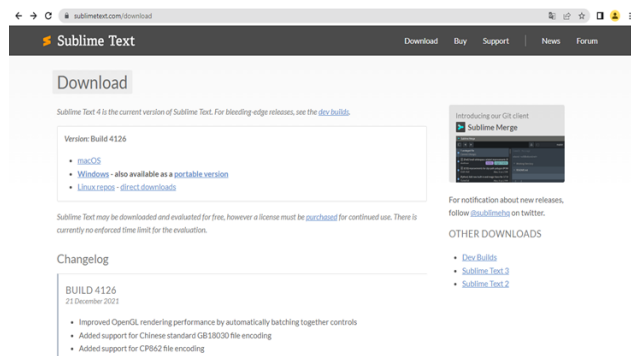
Se recomienda el uso de la versión 3.3.0 de XAMPP para no tener dificultades con los puertos de Apache y MySQL.

4. INSTALACIÓN SUBLIME TEXTS

Primero vamos a descargar el instalador de Sublime text para ello nos dirigimos a www.sublimetext.com y vamos a la sección de DOWNLOAD, allí podremos descargar el instalador de acuerdo con el sistema operativo de nuestro computador. En mi caso voy a descargar el instalador para Windows de 64 bits, y esperamos a que se realice la descarga.

Figura 1

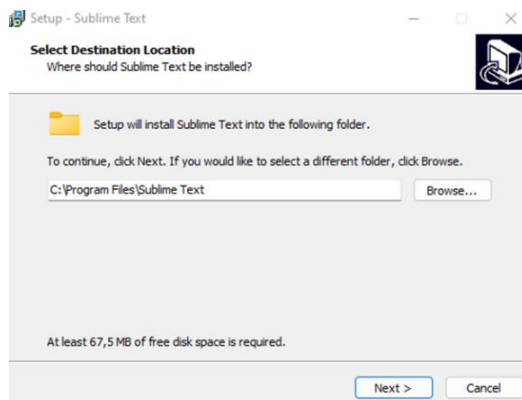
Primer paso de Instalación Sublime Text



Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Figura 2

Segundo paso de Instalación de Sublime Text

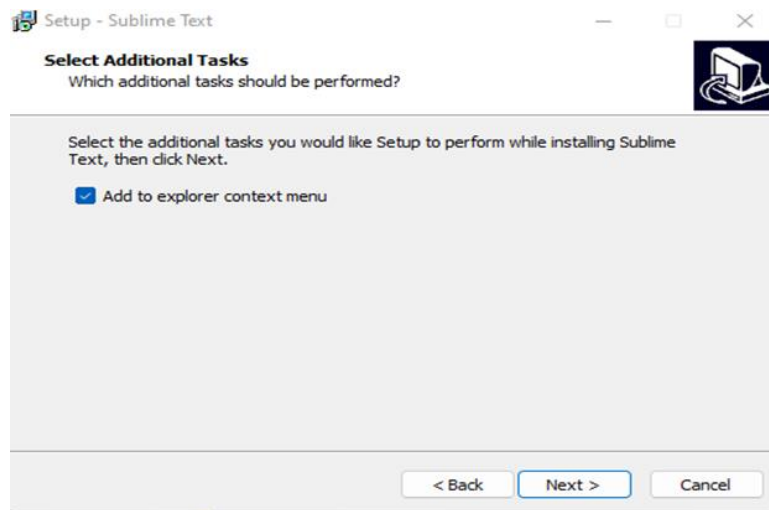


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Si desea que Sublime Text 3 aparezca en el menú del botón derecho, marque la casilla de verificación y haga clic en el botón Siguiente.

Figura 3

Tercer paso de Instalación de Sublime Text

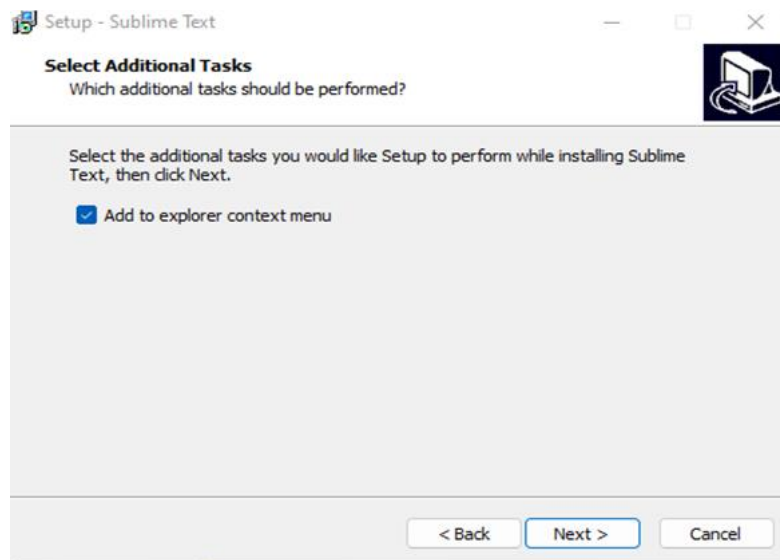


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Presione el botón de instalación.

Figura 4

Cuarto paso de Instalación de Sublime Text

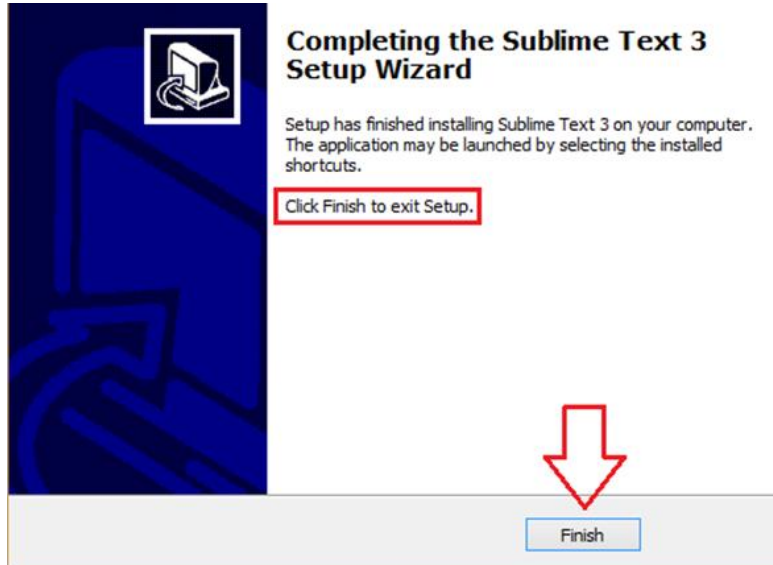


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Termina con el proceso de instalación.

Figura 5

Quinto paso de Instalación de Sublime Text

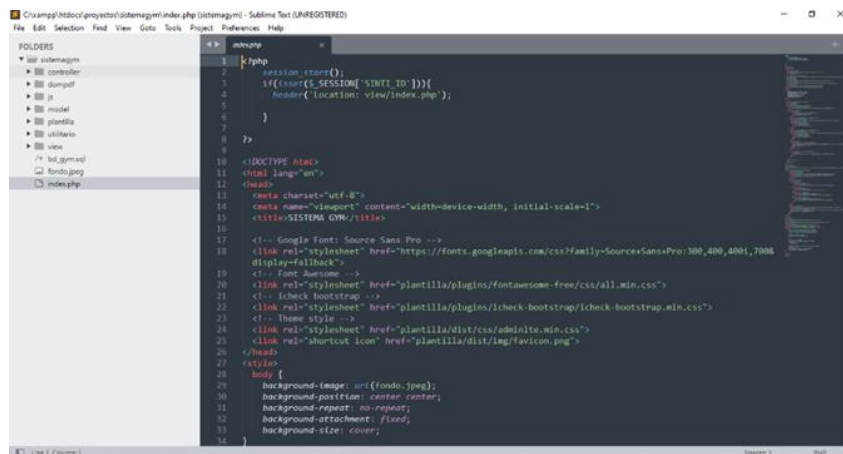


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Una vez finalizado la instalación abrimos nuestro sublime text y desarrollamos nuestro sistema web

Figura 6

Sexto paso de Instalación de Sublime Text



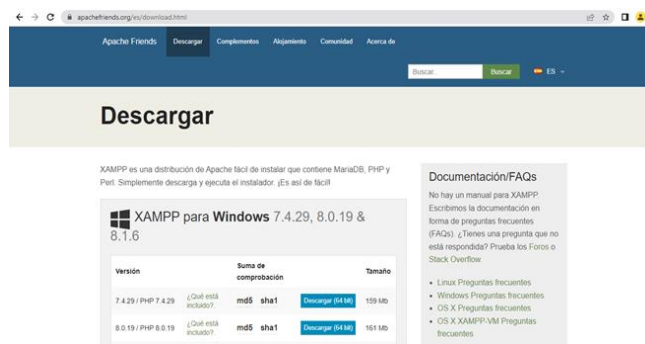
Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

5. INSTALACIÓN XAMPP

Para descargar has clic en este enlace www.apachefriends.org/es/download.html desde donde podrá descargar XAMPP, para diferentes plataformas como Windows, Linux, Mac OS X. Como estamos hablando de cómo instalar XAMPP en Windows, vamos a elegir la opción de Windows como se muestra a continuación.

Figura 7

Primer paso de Instalación XAMPP

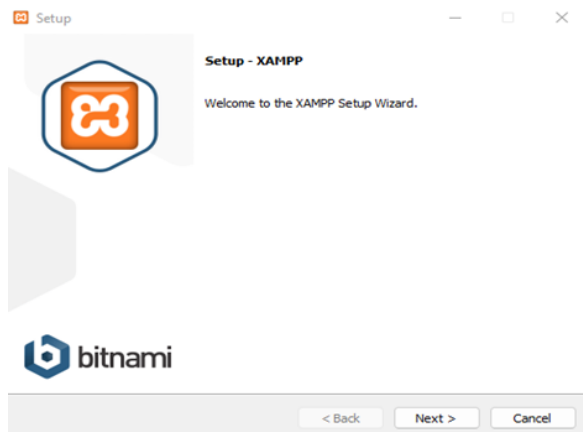


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Una vez descargado, se abrirá el asistente de instalación de XAMPP. Ahora haga clic en el botón Siguiente para continuar

Figura 8

Segundo paso de Instalación XAMPP

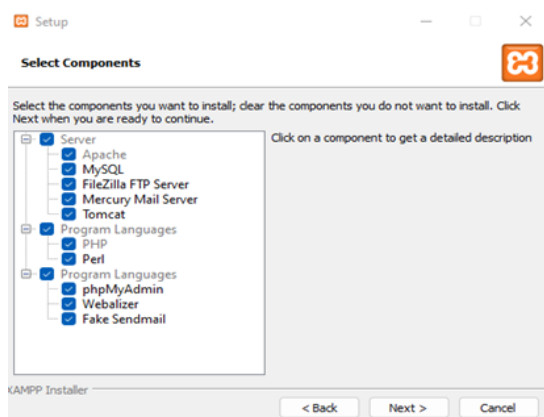


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

A continuación, tendrá que marcar los componentes que quiera instalar y puede desmarcar o dejar como está lo que le interese. Verá que hay algunas opciones de color azul, estas son las opciones que se requieren para ejecutar el software y se instalarán automáticamente. Ahora haga clic en el botón Siguiente para continuar.

Figura 9

Tercer paso de Instalación XAMPP

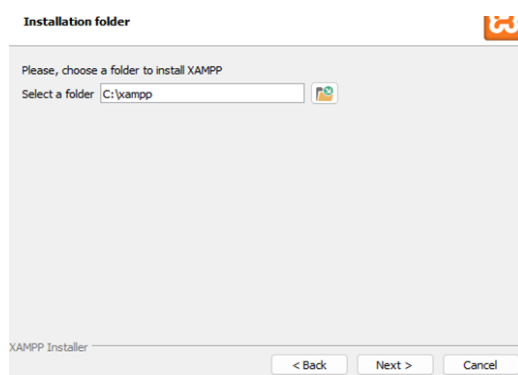


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Ahora elegirá la carpeta donde puede instalar XAMPP. Elija la ubicación predeterminada o seleccione cualquier ubicación de su elección y pulse Siguiente para seguir adelante.

Figura 10

Cuarto paso de Instalación XAMPP

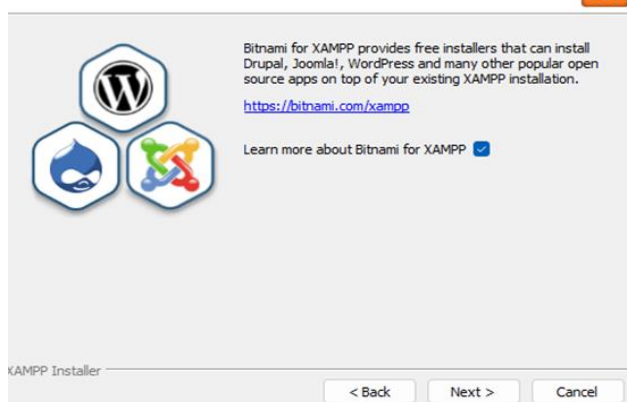


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Ahora verá una ventana que le mostrará información sobre Bitnami. Basta con hacer clic en el botón Siguiente para seguir adelante. Sin embargo, si quiere aprender más sobre Bitnami, entonces marque la casilla que dice Aprender más sobre Bitnami para XAMPP.

Figura 11

Quinto paso de Instalación XAMPP

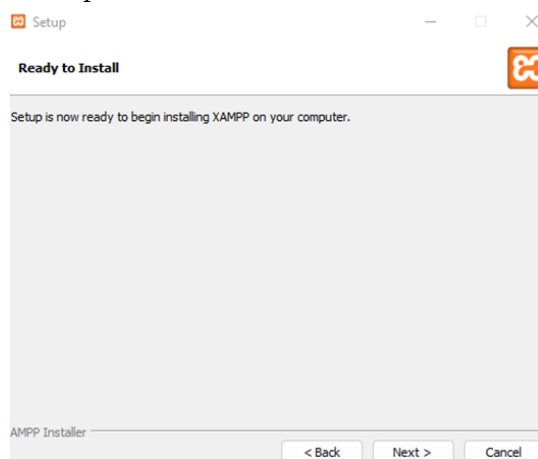


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Como se muestra a continuación. Sólo tiene que pulsar el botón Siguiente para continuar.

Figura 12

Sexto paso de Instalación XAMPP

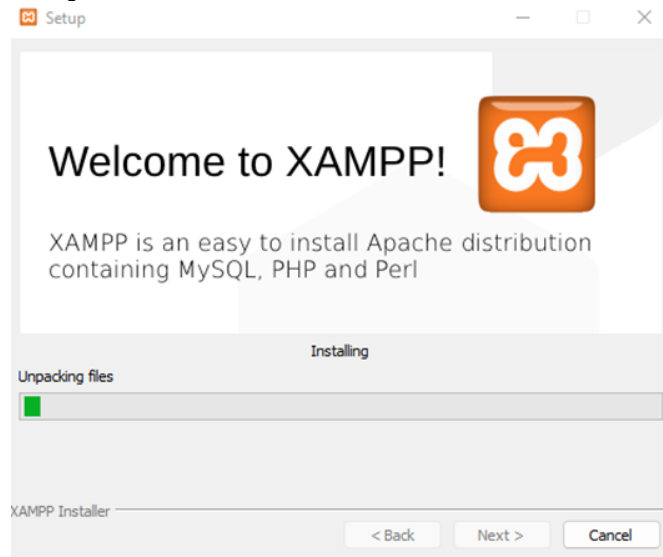


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Esperar hasta que se complete la instalación.

Figura 13

Séptimo paso de Instalación XAMPP



Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Una vez finalizada la instalación, marque la casilla, haga clic en el botón Finalizar y compruebe si XAMPP funciona bien.

Figura 14

Octavo paso de Instalación XAMPP



Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

En cuanto haga clic en el botón Finalizar en el último paso del proceso de instalación de XAMPP, se le pedirá que seleccione su idioma predefinido entre el inglés y el alemán. La elección del idioma depende de usted. A continuación, haga clic en el botón Guardar para confirmar el idioma seleccionado.

Figura 15

Noveno paso de Instalación XAMPP



Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Si todo el proceso de instalación de XAMPP fue correcto, entonces el panel de control se abrirá sin problemas. Ahora haga clic en el botón ‘Start’ correspondiente a Apache y MySQL

Figura 16

Decimo paso de Instalación XAMPP

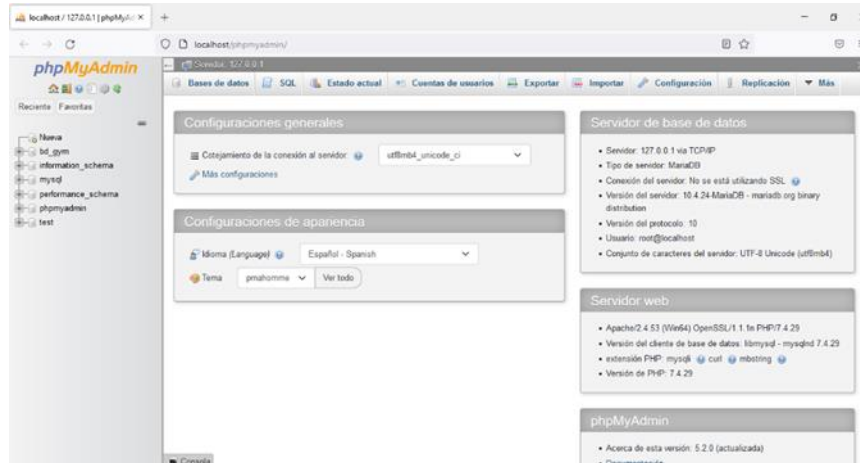


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Nos aparecerá esta ventana donde le daremos clic en “Nueva” para crear la carpeta de la base de datos y podremos exportar la base de datos

Figura 17

Creación de una Base de Datos

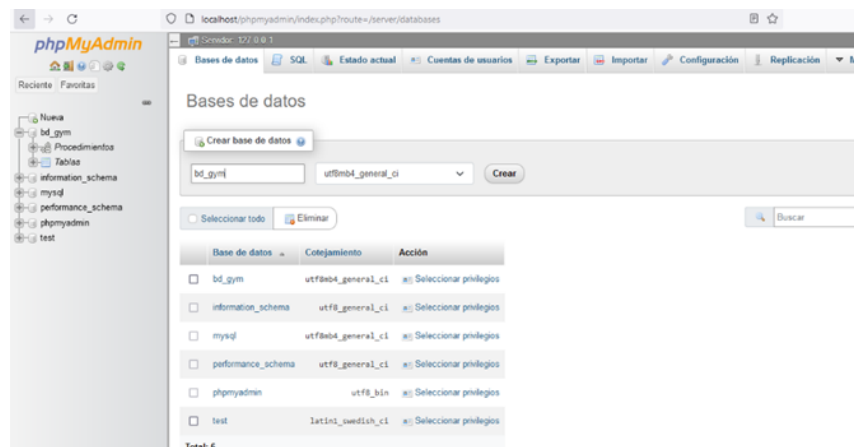


Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Cuando estemos creando la carpeta le pondremos el nombre que quieran en este caso cree con el nombre bd_gym y le daremos en “CREAR” y listo ya está creado.

Figura 18

Nombre de la Base de Datos



Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Una vez creado la base de datos creamos las tablas y campos que va utilizar el sistema.

Figura 19

Creación de Campos

Nombre de la tabla: clientes

Agregar 1 columna(s) Continuar

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
Nombres	INT		Ninguno			☐	--
Apellido	INT		Ninguno			☐	--
Direccion	INT		Ninguno			☐	--
Correo	INT		Ninguno			☐	--

Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

Así quedaría las tablas creadas de nuestra base de datos

Figura 20

Tabla de base de datos

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
categoria	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8_spanish_ci	16.0 KB	-
cliente	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8_spanish_ci	16.0 KB	-
detalle_factura	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	utf8_spanish_ci	48.0 KB	-
empleado	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish_ci	32.0 KB	-
factura	Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8_spanish_ci	48.0 KB	-
membresia	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_spanish_ci	64.0 KB	-
plan	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_spanish_ci	16.0 KB	-
producto	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8_spanish_ci	32.0 KB	-
rol	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_spanish_ci	16.0 KB	-
usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_spanish_ci	32.0 KB	-
10 tablas	Número de filas	71	InnoDB	utf8mb4_general_ci	320.0 KB	0 B

Nota: Elaborado por, (Elder Cerda y Jonathan Shiguango)

6. CODIFICACIÓN

CODIGOS DE MODULO CLIENTES

```
<?php
require_once 'model_conexion.php';

class Modelo_Cliente extends conexionBD{

    public function Registrar_Cliente($ced,$nom,$ape,$tel,$dir,$cor,$sed,$fec,$gene){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_CLIENTE(?,?,?,?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$ced);
        $query -> bindParam(2,$nom);
        $query -> bindParam(3,$ape);
        $query -> bindParam(4,$tel);
        $query -> bindParam(5,$dir);
        $query -> bindParam(6,$cor);
        $query -> bindParam(7,$sed);
        $query -> bindParam(8,$fec);
        $query -> bindParam(9,$gene);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }

    public function Modificar_Cliente($id,$ced,$nom,$ape,$tel,$dir,$cor,$ses,$sed,$fec,$gene){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_CLIENTE(?,?,?,?,?,?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$ced);
        $query -> bindParam(2,$nom);
        $query -> bindParam(3,$ape);
        $query -> bindParam(4,$tel);
        $query -> bindParam(5,$dir);
        $query -> bindParam(6,$cor);
        $query -> bindParam(7,$id);
        $query -> bindParam(8,$ses);
        $query -> bindParam(9,$sed);
        $query -> bindParam(10,$fec);
        $query -> bindParam(11,$gene);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
    }
}
```

```

    }
    conexionBD::cerrar_conexion();
}

public function Listar_Cliente(){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_CLIENTE()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}
}
?>

```

CODIGOS DE MODULO EMPLEADOS

```

<?php
require_once 'model_conexion.php';

class Modelo_Empleado extends conexionBD{

    public function Registrar_Empleado($ced,$nom,$ape,$tel,$dir,$cor,$naci,$estado,$car,$sue){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_EMPLEADO(?,?,?,?,?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$ced);
        $query -> bindParam(2,$nom);
        $query -> bindParam(3,$ape);
        $query -> bindParam(4,$tel);
        $query -> bindParam(5,$dir);
        $query -> bindParam(6,$cor);
        $query -> bindParam(7,$naci);
        $query -> bindParam(8,$estado);
        $query -> bindParam(9,$car);
        $query -> bindParam(10,$sue);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }
}

```



```

public function Modificar_Empleado($id,$ced,$nom,$ape,$tel,$dir,$cor,$es,$naci,$estado,$car,$sue){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_MODIFICAR_EMPLEADO(?,?,?,?,?,?,?,?,?,?,?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$ced);
    $query -> bindParam(2,$nom);
    $query -> bindParam(3,$ape);
    $query -> bindParam(4,$tel);
    $query -> bindParam(5,$dir);
    $query -> bindParam(6,$cor);
    $query -> bindParam(7,$id);
    $query -> bindParam(8,$es);
    $query -> bindParam(9,$naci);
    $query -> bindParam(10,$estado);
    $query -> bindParam(11,$car);
    $query -> bindParam(12,$sue);
    $query->execute();
    if($row = $query->fetchColumn()){
        return $row;
    }
    conexionBD::cerrar_conexion();
}

```

```

public function Listar_Empleado(){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_EMPLEADO";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

```

```

public function eliminar_empleado($id){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_ELIMINAR_EMPLEADO(?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$id);
    $query->execute();
    $result = $query->execute();
    if($result){
        return 1;
    }else{

```

```

        return 0;
    }
    conexionBD::cerrar_conexion();
}
}
?>

```

CODIGOS DE MODULO PRODUCTOS

```

<?php
require_once 'model_conexion.php';

class Modelo_Producto extends conexionBD{

    public function Registrar_Producto($pro,$pre,$cat,$sto,$cod){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_PRODUCTO(?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$pro);
        $query -> bindParam(2,$pre);
        $query -> bindParam(3,$cat);
        $query -> bindParam(4,$sto);
        $query -> bindParam(5,$cod);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }

    public function Modificar_Producto($pro,$pre,$cat,$id,$estatus,$stock,$cod){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_PRODUCTO(?,?,?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$pro);
        $query -> bindParam(2,$pre);
        $query -> bindParam(3,$cat);
        $query -> bindParam(4,$id);
        $query -> bindParam(5,$estatus);
        $query -> bindParam(6,$stock);
        $query -> bindParam(7,$cod);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }
}

```

```

public function Listar_Producto(){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_PRODUCTO()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

public function listar_select_categoria(){
    $c = conexionBD::conexionPDO();

    $sql = "CALL SP_SELECT_CATEGORIA()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll();
    foreach($resultado as $resp){
        $arreglo[] = $resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}
?>

```

CODIGOS DE MODULO DE CATEGORIA DE PRODUCTOS

```

<?php
require_once 'model_conexion.php';
class Modelo_Categoria extends conexionBD{

    public function Registrar_Categoria($cat){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_CATEGORIA(?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$cat);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }
}

```

```

public function Modificar_Categoria($id,$cat,$es){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_MODIFICAR_CATEGORIA(?,?,?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$id);
    $query -> bindParam(2,$cat);
    $query -> bindParam(3,$es);
    $query->execute();
    if($row = $query->fetchColumn()){
        return $row;
    }
    conexionBD::cerrar_conexion();
}

public function Listar_Categoria(){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_CATEGORIA()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

}
?>

```

CODIGOS DE MODULO VENTAS

```

<?php
require_once 'model_conexion.php';
class Modelo_Venta extends conexionBD{
    public function
    Registrar_Venta($idcliente,$tip,$idusu,$importe,$idpro,$precio,$cantidad,$subtotal,$monigv){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_VENTA(?,?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$idcliente);
        $query -> bindParam(2,$tip);
        $query -> bindParam(3,$idusu);
        $query -> bindParam(4,$importe);
        $query -> bindParam(5,$subtotal);
        $query -> bindParam(6,$monigv);
    }
}

```

```

$query->execute();
if($row = $query->fetchColumn()){
    if ($row!="") {
        $array_idpro = explode(",",$idpro);
        $array_precio = explode(",",$precio);
        $array_cantidad = explode(",",$cantidad);
        $nregistradas=0;
        for ($i=0; $i < count($array_idpro); $i++) {
            //die();
            $c2 = conexionBD::conexionPDO();
            $sql2 = "CALL SP_REGISTRAR_VENTA_DETALLE(?,?,?,?)";
            $query2 = $c2->prepare($sql2);
            $query2->bindParam(1,$row);
            $query2->bindParam(2,$array_idpro[$i]);
            $query2->bindParam(3,$array_precio[$i]);
            $query2->bindParam(4,$array_cantidad[$i]);
            $resultado2 = $query2->execute();
            if($resultado2){
                $nregistradas++;
            }
        }
        if ($nregistradas>0) {
            return 1;
        }else{
            return 0;
        }
    }
}
conexionBD::cerrar_conexion();
}

```

```

public function Listar_Venta($inicio,$fin){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_VENTA(?,?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$inicio);
    $query -> bindParam(2,$fin);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

```

```

public function listar_select_producto($id){

```

```

$c = conexionBD::conexionPDO();

$sql = "CALL SP_SELECT_PRODUCTO(?)";
$arreglo = array();
$query = $c->prepare($sql);
$query -> bindParam(1,$id);
$query->execute();
$resultado = $query->fetchAll();
foreach($resultado as $resp){
    $arreglo[] = $resp;
}
return $arreglo;
conexionBD::cerrar_conexion();
}
}
?>

```

CODIGOS DE MODULO MEMBRESIA

```

<?php
require_once 'model_conexion.php';
class Modelo_Membresia extends conexionBD{
    public function Registrar_Membresia($cli,$pla,$ini,$fin,$id,$pre){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_MEMBRESIA(?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$cli);
        $query -> bindParam(2,$pla);
        $query -> bindParam(3,$ini);
        $query -> bindParam(4,$fin);
        $query -> bindParam(5,$id);
        $query -> bindParam(6,$pre);
        $result = $query->execute();
        if($result){
            return 1;
        }else{
            return 0;
        }
        conexionBD::cerrar_conexion();
    }
    public function Modificar_Membresia($idmem,$pla,$ini,$fin,$id,$pre){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_MEMBRESIA(?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$idmem);
    }
}

```

```

$query -> bindParam(2,$pla);
$query -> bindParam(3,$ini);
$query -> bindParam(4,$fin);
$query -> bindParam(5,$id);
$query -> bindParam(6,$pre);

$query->execute();
$result = $query->execute();
if($result){
    return 1;
}else{
    return 0;
}
conexionBD::cerrar_conexion();
}

public function Eliminar_Membresia($id){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_ELIMINAR_MEMBRESIA(?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$id);
    $query->execute();
    $result = $query->execute();
    if($result){
        return 1;
    }else{
        return 0;
    }
    conexionBD::cerrar_conexion();
}

public function Finalizar_Membresia($id){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_FINALIZAR_MEMBRESIA(?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$id);
    $query->execute();
    $result = $query->execute();
    if($result){
        return 1;
    }else{
        return 0;
    }
    conexionBD::cerrar_conexion();
}

public function Listar_Membresia(){

```

```

$c = conexionBD::conexionPDO();
$sql = "CALL SP_LISTAR_MEMBRESIA()";
$arreglo = array();
$query = $c->prepare($sql);
$query->execute();
$resultado = $query->fetchAll(PDO::FETCH_ASSOC);
foreach($resultado as $resp){
    $arreglo["data"][]=$resp;
}
return $arreglo;
conexionBD::cerrar_conexion();
}

public function Listar_Membresia_Vencida(){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_LISTAR_MEMBRESIA_VENCIDA()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    foreach($resultado as $resp){
        $arreglo["data"][]=$resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

public function listar_select_cliente(){
    $c = conexionBD::conexionPDO();

    $sql = "CALL SP_SELECT_CLIENTE()";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query->execute();
    $resultado = $query->fetchAll();
    foreach($resultado as $resp){
        $arreglo[] = $resp;
    }
    return $arreglo;
    conexionBD::cerrar_conexion();
}

public function listar_select_plan(){
    $c = conexionBD::conexionPDO();

    $sql = "CALL SP_SELECT_SERVICIOS()";
    $arreglo = array();

```



```

$query = $c->prepare($sql);
$query->execute();
$resultado = $query->fetchAll();
foreach($resultado as $resp){
    $arreglo[] = $resp;
}
return $arreglo;
conexionBD::cerrar_conexion();
}
}
?>

```

CODIGOS DE MODULO DE SERVICIOS

```

<?php
require_once 'model_conexion.php';
class Modelo_Plan extends conexionBD{
    public function Registrar_Plan($pla,$des,$pre){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_SERVICIOS(?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$pla);
        $query -> bindParam(2,$des);
        $query -> bindParam(3,$pre);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }
    public function Modificar_Plan($id,$pla,$des,$ses,$pr){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_SERVICIO(?,?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$id);
        $query -> bindParam(2,$pla);
        $query -> bindParam(3,$des);
        $query -> bindParam(4,$ses);
        $query -> bindParam(5,$pr);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }
    public function Listar_Plan(){

```

```

        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_LISTAR_SERVICIOS()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
        foreach($resultado as $resp){
            $arreglo["data"][]=$resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }
}
?>

```

CODIGOS DE MODULO DE USUARIO

```

<?php
require_once 'model_conexion.php';
class Modelo_Usuario extends conexionBD{
    public function Verificar_Usuario($usu,$con){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_VERIFICAR_USUARIO(?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->bindParam(1,$usu);
        $query->execute();
        $resultado = $query->fetchAll();
        foreach($resultado as $resp){
            if(password_verify($con,$resp['usuario_password'])){
                $arreglo[]=$resp;
            }
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }

    public function Registrar_Usuario($rol,$usu,$con,$ced){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_USUARIO(?,?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$usu);
        $query -> bindParam(2,$con);
        $query -> bindParam(3,$rol);
        $query -> bindParam(4,$ced);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
    }
}

```

```

    }
    conexionBD::cerrar_conexion();
}

public function Restablecer_Contra($ced,$usu,$con){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_RESTABLECER_CONTRA(?,?,?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$ced);
    $query -> bindParam(2,$usu);
    $query -> bindParam(3,$con);
    $query->execute();
    if($row = $query->fetchColumn()){
        return $row;
    }
    conexionBD::cerrar_conexion();
}

public function Modificar_Usuario_Contra($id,$con){
    $c = conexionBD::conexionPDO();
    $sql = "CALL SP_MODIFICAR_USUARIO_CONTRA(?,?)";
    $arreglo = array();
    $query = $c->prepare($sql);
    $query -> bindParam(1,$id);
    $query -> bindParam(2,$con);
    $resul = $query->execute();
    if($resul){
        return 1;
    }else{
        return 0;
    }
    conexionBD::cerrar_conexion();
}

public function Modificar_Usuario($id,$cedula,$rol){
    $c = conexionBD::conexionPDO();

    $sql = "CALL SP_MODIFICAR_USUARIO(?,?,?)";
    $query = $c->prepare($sql);
    $query->bindParam(1,$id);
    $query->bindParam(2,$cedula);
    $query->bindParam(3,$rol);
    $resultado = $query->execute();
    if($resultado){
        return 1;
    }else{
        return 0;
    }
}

```

```

        conexionBD::cerrar_conexion();

    }

    public function Modificar_Usuario_Estatus($id,$estatus){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_USUARIO_ESTATUS(?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$id);
        $query -> bindParam(2,$estatus);
        $resul = $query->execute();
        if($resul){
            return 1;
        }else{
            return 0;
        }
        conexionBD::cerrar_conexion();
    }

    public function Widget_Membresia(){
        $c = conexionBD::conexionPDO();

        $sql = "CALL SP_LISTAR_WIDGET_MEMBRESIA()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll();
        foreach($resultado as $resp){
            $arreglo[] = $resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }

    public function Finalizar_Membresia(){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_ACTUALIZAR_TODAS_MEMBRESIA_VENCIDA()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $resul = $query->execute();
        if($resul){
            return 1;
        }else{
            return 0;
        }
        conexionBD::cerrar_conexion();
    }

```

```

    }

    public function Listar_Usuario(){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_LISTAR_USUARIO()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
        foreach($resultado as $resp){
            $arreglo["data"][]=$resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }

    public function listar_select_rol(){
        $c = conexionBD::conexionPDO();

        $sql = "CALL SP_SELECT_ROL()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll();
        foreach($resultado as $resp){
            $arreglo[] = $resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }

    public function cargar_notificacion(){
        $c = conexionBD::conexionPDO();

        $sql = "CALL SP_LISTAR_NOTIFICACION()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll();
        foreach($resultado as $resp){
            $arreglo[] = $resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }
}
?>

```

CODIGOS DE MODULO DE ROLES

```
<?php
require_once 'model_conexion.php';

class Modelo_Rol extends conexionBD{

    public function Registrar_Rol($rol){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_REGISTRAR_ROL(?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$rol);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }

    public function Modificar_Rol($id,$rol,$ses){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_MODIFICAR_ROL(?,?,?)";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query -> bindParam(1,$id);
        $query -> bindParam(2,$rol);
        $query -> bindParam(3,$ses);
        $query->execute();
        if($row = $query->fetchColumn()){
            return $row;
        }
        conexionBD::cerrar_conexion();
    }

    public function Listar_Rol(){
        $c = conexionBD::conexionPDO();
        $sql = "CALL SP_LISTAR_ROL()";
        $arreglo = array();
        $query = $c->prepare($sql);
        $query->execute();
        $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
        foreach($resultado as $resp){
            $arreglo["data"][]=$resp;
        }
        return $arreglo;
        conexionBD::cerrar_conexion();
    }
}
?>
```