

MANUAL DEL PROGRAMADOR- GESTIÓN CONSTRUCCIÓN



COMPUTO

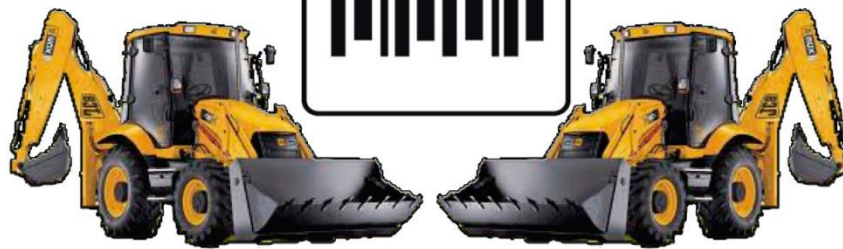
Informar para explicar el código desarrollado

Oscar René Rodríguez
Zimmermann 3ro

Informática e Inglés Técnico



Gestión Construcción.



Explicación del código.

Planteamiento del proyecto.

- Para que la plataforma “Gestión de Construcción” funcione correctamente se tiene que utilizar un usuario y contraseña para poder acceder a ella, que se les dará a cada persona que pueda controlar o utilizar este software que será único y personal, por su seguridad en la base de datos esta encriptado

el campo de “Contraseña” (se escribe sin “Ñ” para no tener mayor problema al momento de programar).

A la hora de Utilizar este programa, el solicitante de proyecto o edificación tiene que estar presente al personal controlador de este software para especificar lo que necesita guardarse como proyecto y así dar los detalles al equipo que realizara dicho proyecto.

- El Proyecto está hecho para las personas que trabajan en la rama de la Ingeniería Civil, en este proyecto puede guardar datos de proyectos que se necesiten edificar, del personal trabajador que se encuentra laborando en la empresa, el control de los usuarios que administrarían o llenarían los datos, lleva un control de ingresos, también lleva una parte de Reportes que le permite imprimir ya sea el personal laborando en la empresa, los proyectos y los gastos económicos realizados semanalmente.
- Base de datos inscriptica, la cual fue importada a PhpMyAdmin.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `ProyectosCiviles` DEFAULT CHARACTER SET utf8 ;
USE `ProyectosCiviles` ;

-----
-- Table `ProyectosCiviles`.`Proveedores`
-----

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Proveedores` (
  `idProveedor` INT NOT NULL AUTO_INCREMENT ,
  `NombreProveedor` VARCHAR(250) NOT NULL ,
  `NIT` VARCHAR(14) NOT NULL ,
```

```

`RazonSocial` VARCHAR(250) NULL DEFAULT NULL ,
`Direccion` VARCHAR(250) NOT NULL ,
`Telefonos` VARCHAR(80) NOT NULL ,
PRIMARY KEY (`idProveedor`) )
ENGINE = InnoDB;

-----

-- Table `ProyectosCiviles`.`Entradas`
-----

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Entradas` (
  `idEntradas` INT NOT NULL AUTO_INCREMENT ,
  `idProveedor` INT NOT NULL ,
  `FechaCompra` DATE NOT NULL ,
  `CantidadProducto` INT NOT NULL ,
  `ValorProducto` FLOAT NOT NULL ,
  `PorcentajeIVA` INT NOT NULL ,
  `CalculoIVA` FLOAT NOT NULL ,
  `StockMinimo` INT NOT NULL ,
  `StockMaximo` INT NOT NULL ,
  `Anulada` BIT NULL DEFAULT NULL ,
  PRIMARY KEY (`idEntradas`) ,
  INDEX `fk_idproveedor_idx` (`idProveedor` ASC) ,
  CONSTRAINT `fk_idproveedor`
    FOREIGN KEY (`idProveedor`)
      REFERENCES `ProyectosCiviles`.`Proveedores` (`idProveedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

```

-- Table `ProyectosCiviles`.`TipoProyecto`

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`TipoProyecto` (
 `idTipoProyecto` INT NOT NULL AUTO_INCREMENT ,
 `Nombre` VARCHAR(45) NOT NULL ,
 `Descripcion` TEXT NOT NULL ,
 `Imagen` VARCHAR(100) NULL ,
 PRIMARY KEY (`idTipoProyecto`))
ENGINE = InnoDB;

-- Table `ProyectosCiviles`.`Proyectos`

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Proyectos` (
 `idProyectos` INT NOT NULL AUTO_INCREMENT ,
 `idTipoProyecto` INT NOT NULL ,
 `NombreProyecto` VARCHAR(250) NOT NULL ,
 `FechaInicio` DATETIME NOT NULL ,
 `FechaFin` DATETIME NULL DEFAULT NULL ,
 `CostoProyectado` FLOAT NOT NULL ,
 PRIMARY KEY (`idProyectos`),
 INDEX `fk_tipo_proyecto_idx` (`idTipoProyecto` ASC) ,
 CONSTRAINT `fk_tipo_proyecto`
 FOREIGN KEY (`idTipoProyecto`)
 REFERENCES `ProyectosCiviles`.`TipoProyecto` (`idTipoProyecto`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `ProyectosCiviles`.`Experiencia`

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Experiencia` (
 `idExperiencia` INT NOT NULL AUTO_INCREMENT ,
 `Descripcion` VARCHAR(20) NOT NULL ,
 PRIMARY KEY (`idExperiencia`))
ENGINE = InnoDB;

-- Table `ProyectosCiviles`.`TipoEmpleado`

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`TipoEmpleado` (
 `idTipoEmpleado` INT NOT NULL AUTO_INCREMENT ,
 `idExperiencia` INT NOT NULL ,
 `Descripcion` VARCHAR(150) NOT NULL ,
 PRIMARY KEY (`idTipoEmpleado`) ,
 UNIQUE INDEX `idExperiencia_UNIQUE` (`idExperiencia` ASC) ,
 CONSTRAINT `fk_experiencia`
 FOREIGN KEY (`idExperiencia`)
 REFERENCES `ProyectosCiviles`.`Experiencia` (`idExperiencia`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `ProyectosCiviles`.`Personal`

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Personal` (
 `idPersonal` INT NOT NULL AUTO_INCREMENT ,
 `idTipoEmpleado` INT NOT NULL ,

```

`Nombres` VARCHAR(30) NOT NULL ,
`Apellidos` VARCHAR(30) NOT NULL ,
`FechaIngreso` DATE NOT NULL ,
`FechaNacimiento` DATE NOT NULL ,
`Habilidad1` VARCHAR(50) NULL DEFAULT NULL ,
`Habilidad2` VARCHAR(50) NULL DEFAULT NULL ,
`Habilidad3` VARCHAR(50) NULL DEFAULT NULL ,
`DUI` VARCHAR(10) NOT NULL ,
`NIT` CHAR(15) NOT NULL ,
`Pasaporte` CHAR(15) NULL DEFAULT NULL ,
`Nacionalidad` VARCHAR(45) NOT NULL ,
`AFP` CHAR(12) NOT NULL ,
`NUP` CHAR(12) NULL ,
`ISSS` CHAR(9) NOT NULL ,
`SalarioMensual` DECIMAL(4,2) NOT NULL ,
`Personalcol` VARCHAR(45) NULL ,
PRIMARY KEY (`idPersonal`) ,
INDEX `fk_tipo_empleado_idx` (`idTipoEmpleado` ASC) ,
CONSTRAINT `fk_tipo_empleado`
    FOREIGN KEY (`idTipoEmpleado`)
    REFERENCES `ProyectosCiviles`.`TipoEmpleado` (`idTipoEmpleado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`Salidas`
-----

```

```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Salidas` (
    `idSalidas` INT NOT NULL AUTO_INCREMENT ,
    `idPersonal` INT NOT NULL ,

```

```

`idProyecto` INT NOT NULL ,
`CantidadUtilizada` INT NOT NULL ,
`FechaSalida` DATETIME NOT NULL ,
PRIMARY KEY (`idSalidas`),
INDEX `fk_idproyectos_idx` (`idProyecto` ASC),
INDEX `fk_personal_idx` (`idPersonal` ASC),
CONSTRAINT `fk_idproyectos`
  FOREIGN KEY (`idProyecto`)
    REFERENCES `ProyectosCiviles`.`Proyectos` (`idProyectos`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_personal`
  FOREIGN KEY (`idPersonal`)
    REFERENCES `ProyectosCiviles`.`Personal` (`idPersonal`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`Unidad`
-----

```

```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Unidad` (
  `idUnidad` INT NOT NULL AUTO_INCREMENT ,
  `Descripcion` VARCHAR(80) NOT NULL ,
  PRIMARY KEY (`idUnidad`))
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`Lineas`
-----

```



```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Lineas` (
  `idLineas` INT NOT NULL AUTO_INCREMENT ,
  `idUnidad` INT NOT NULL ,
  `Descripcion` VARCHAR(150) NOT NULL ,
  PRIMARY KEY (`idLineas`) ,
  INDEX `fk_unidad_idx` (`idUnidad` ASC) ,
  CONSTRAINT `fk_unidad`
    FOREIGN KEY (`idUnidad` )
      REFERENCES `ProyectosCiviles`.`Unidad` (`idUnidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`UnidadesMedida`
-----

```

```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`UnidadesMedida` (
  `idUnidadesMedida` INT NOT NULL AUTO_INCREMENT ,
  `Descripcion` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idUnidadesMedida`))
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`Especifico`
-----

```

```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Especifico` (
  `idEspecifico` INT NOT NULL AUTO_INCREMENT ,
  `idSalidas` INT NOT NULL ,
  `idEntradas` INT NOT NULL ,
  `idLinea` INT NOT NULL ,

```

```

`idUnidadMedida` INT NOT NULL ,
`Descripcion` VARCHAR(250) NOT NULL ,
PRIMARY KEY (`idEspecifico`) ,
INDEX `fk_Especifico_Entradas1_idx` (`idEntradas` ASC) ,
INDEX `fk_Especifico_Salidas1_idx` (`idSalidas` ASC) ,
INDEX `fk_linea_idx` (`idLinea` ASC) ,
INDEX `fk_unidad_medida_idx` (`idUnidadMedida` ASC) ,
CONSTRAINT `fk_Especifico_Salidas1`
  FOREIGN KEY (`idSalidas` )
  REFERENCES `ProyectosCiviles`.`Salidas` (`idSalidas` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Especifico_Entradas1`
  FOREIGN KEY (`idEntradas` )
  REFERENCES `ProyectosCiviles`.`Entradas` (`idEntradas` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_linea`
  FOREIGN KEY (`idLinea` )
  REFERENCES `ProyectosCiviles`.`Lineas` (`idLineas` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_unidad_medida`
  FOREIGN KEY (`idUnidadMedida` )
  REFERENCES `ProyectosCiviles`.`UnidadesMedida` (`idUnidadesMedida` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`NivelConfianza`

```

```

-----
CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`NivelConfianza` (
  `idNivelConfianza` INT NOT NULL AUTO_INCREMENT ,
  `Descripcion` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idNivelConfianza`) )
ENGINE = InnoDB;

```

```

-----
-- Table `ProyectosCiviles`.`Usuarios`
-----

```

```

CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Usuarios` (
  `idUsuarios` INT NOT NULL AUTO_INCREMENT ,
  `idNivelConfianza` INT NOT NULL ,
  `idPersonal` INT NOT NULL ,
  `Usuario` VARCHAR(45) NOT NULL ,
  `Contrasena` VARCHAR(45) NOT NULL ,
  `Activo` TINYINT(1) NOT NULL DEFAULT 1 ,
  PRIMARY KEY (`idUsuarios`) ,
  INDEX `fk_nivel_confianza_idx` (`idNivelConfianza` ASC) ,
  UNIQUE INDEX `Usuario_UNIQUE` (`Usuario` ASC) ,
  INDEX `fk_id_personal_idx` (`idPersonal` ASC) ,
  CONSTRAINT `fk_nivel_confianza`
    FOREIGN KEY (`idNivelConfianza`)
      REFERENCES `ProyectosCiviles`.`NivelConfianza` (`idNivelConfianza`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_id_personal`
    FOREIGN KEY (`idPersonal`)
      REFERENCES `ProyectosCiviles`.`Personal` (`idPersonal`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;
```

```
-----  
-- Table `ProyectosCiviles`.`Proyectos_has_Personal`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ProyectosCiviles`.`Proyectos_has_Personal` (  
  `Proyectos_idProyectos` INT NOT NULL AUTO_INCREMENT ,  
  `idPersonal` INT NOT NULL ,  
  `idProyecto` INT NOT NULL ,  
  PRIMARY KEY (`Proyectos_idProyectos`),  
  INDEX `fk_Proyectos_has_Personal_Personal1` (`idPersonal` ASC) ,  
  INDEX `fk_Proyectos_has_Personal_Proyectos1_idx` (`idProyecto` ASC) ,  
  CONSTRAINT `fk_Proyectos_has_Personal_Proyectos1`  
    FOREIGN KEY (`idProyecto` )  
      REFERENCES `ProyectosCiviles`.`Proyectos` (`idProyectos` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Proyectos_has_Personal_Personal1`  
    FOREIGN KEY (`idPersonal` )  
      REFERENCES `ProyectosCiviles`.`Personal` (`idPersonal` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
USE `ProyectosCiviles` ;
```

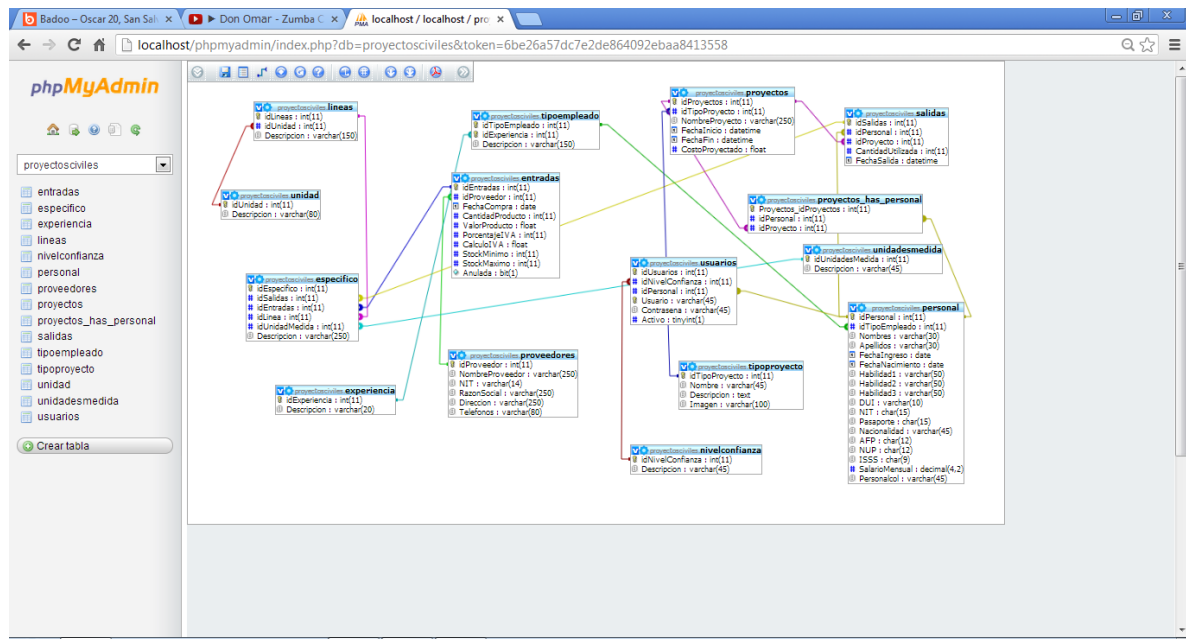
```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Estas son ejemplos de las tablas y los campos que aparecen en la base de datos, en las imágenes se muestran la tabla de Proveedores y entradas con sus respectivos campos.

Al momento de Importar la base de datos a PhpMyAdmin, aparecerá con sus relaciones y todos los datos que se muestran en la imagen siguiente:



- El programa está hecho en lenguaje PHP. A continuación las capturas de pantalla de algunas páginas de registro.

Programa utilizado para poder programar “Sublime Text”, en la imagen que se muestra es un ejemplo del código utilizado para el proyecto “Gestión de Construcción”, cada código va dependiendo a su función para tener un mayor orden, cada Case 1, Case 2, Case 3, Case 4 se identifican así dependiendo a su función ya sea, Guardar, Eliminar, Modificar y Buscar.

Case 1. Ejemplo de Registros de Personal. //buscar personal

```
switch ($accion) {
```

```

case 1: //buscar personal

$texto = isset($_POST['txt_texto_personal']) ? $_POST['txt_texto_personal'] :
";

$opc = isset($_POST['sel_buscar_personal']) ?
$_POST['sel_buscar_personal'] : "";

$array_data = array();
if ($texto != "") {
    $sql = "SELECT P.*,TE.Descripcion TipoEmp FROM Personal P ";
    $sql.="INNER JOIN TipoEmpleado TE ";
    $sql.="ON P.idTipoEmpleado=TE.idTipoEmpleado ";
    $sql.="WHERE " . $opc . " like '%" . $texto . "%' ORDER BY
P.Nombres,P.Apellidos ASC";

    $query = mysql_query($sql, $connection);
    $i = 0;
    while ($row = mysql_fetch_array($query)) {
        $array_data[$i] = array(
            "id" => $row["idPersonal"],
            "id_tipo_empleado" => $row["idTipoEmpleado"],
            "label_tipo_empleado" => $row["TipoEmp"],
            "nombres" => $row["Nombres"],
            "apellidos" => $row["Apellidos"],
            "fecha_ingreso" => $row["FechaIngreso"],
            "fecha_nacimiento" => $row["FechaNacimiento"],
            "habilidad1" => $row["Habilidad1"],
            "habilidad2" => $row["Habilidad2"],
            "habilidad3" => $row["Habilidad3"],
            "dui" => $row["DUI"],
            "nit" => $row["NIT"],
            "pasaporte" => $row["Pasaporte"],
            "nacionalidad" => $row["Nacionalidad"],

```

```

        "afp" => $row["AFP"],
        "nup" => $row["NUP"],
        "iss" => $row["ISS"],
        "salario" => $row["SalarioMensual"],
        "personal" => $row["Personalcol"]
    );
    $i++;
}

$jsonData["mensaje"] = "RECUPERANDO DATOS ";
$jsonData["bandera"] = 1;
$jsonData["total_rows"] = $i;
$jsonData["rows"] = $array_data;
} else {
    $jsonData["mensaje"] = "FALTAN CAMPOS POR LLENAR";
    $jsonData["bandera"] = 0;
}
break;

```

Este es el código que genera la búsqueda, en la cual indica que hace una selección al campo de la base de datos que necesita buscarse haciendo un envío de búsqueda con un “mysql_query”, se hace con una función while en la cual se utiliza una variable “mysql_fetch_array” que funciona para recuperar una fila de resultados como un array asociativo, un array numérico o como ambos y con una variable “jsonData” que permite mostrar un texto el cual sería el mensaje que dice "RECUPERANDO DATOS" mientras es 1 y si es 0 dirá "FALTAN CAMPOS POR LLENAR" la cual termina con un “Sleep” que permite retrasar la ejecución de un proceso y un “Break” se utiliza para terminar la ejecución de la estructura actual en este caso sería la función While y acepta un argumento numérico opcional al cual indica de cuantas estructuras anidadas encerradas se debe salir.

Case 2 Ejemplo de Registros de Personal. //adicionando un nuevo registro}

case 2: //adicionando un nuevo registro

//validaciones para el archivo

```
$sql = "insert into Personal
(idTipoEmpleado,Nombres,Apellidos,FechaIngreso,FechaNacimiento,
$mysql.=
"Habilidad1,Habilidad2,Habilidad3,DUI,NIT,Pasaporte,Nacionalidad,AFP,NUP,
ISSS,SalarioMensual,Personalcol) ";
$sql.=
"values(" . $sel_tipo_empleado_personal . "," .
$txt_nombre_personal . "," . $txt_apellidos_personal . "," .
$txt_fecha_ingreso_personal . "," .
$sql.= "" . $txt_fecha_nacimiento_personal . "," .
$texta_habilidad1_personal . "," . $texta_habilidad2_personal . "," .
$texta_habilidad3_personal . "," .
$sql.= "" . $txt_uit_personal . "," . $txt_nit_personal . "," .
$txt_pasaporte_personal . "," . $txt_nacionalidad_personal . "," .
$sql.= $txt_afp_personal . "," . $txt_nup_personal . "," .
$txt_iss personal . "," . $txt_salario_personal . "," .
$txt_personal_personal . ")";
mysql_query("SET NAMES 'utf8'");
$respuesta = mysql_query($sql, $connection);
if ($respuesta) {
    $mensaje = "PERSONAL AGREGADO";
    $bandera = 1;
} else {
    $mensaje = "SE PRODUJO UN ERROR AL MOMENTO DE REGISTRAR
EL PERSONAL. INTENTE DE NUEVO " . $sql;
    $bandera = 0;
}
```



```
$jsonData["mensaje"] = $mensaje;  
$jsonData["bandera"] = $bandera;  
  
break;
```

Este código es el que permite agregar un nuevo registro en la base de datos y comienza con un \$Sql se hace el insert dentro de las líneas y los campos que necesitan llenarse, mostrando un mensaje que dice “LINEA AGREGADA” si el valor es “\$Bandera = 1”, si la opción fuera “\$Bandera = 0” mostraría un mensaje de "SE PRODUJO UN ERROR AL MOMENTO DE REGISTRAR LA NUEVA LINEA. INTENTE DE NUEVO" y siempre termina con un “Break” que en este caso acepta un argumento numérico opcional al cual indica de cuantas estructuras anidadas encerradas se debe salir.

Case 3. Ejemplo de Registros de Personal. //modificando la línea

```
case 3: //modificando al personal  
  
    $sql      =      "UPDATE      Personal      SET      idTipoEmpleado="      .  
$sel_tipo_empleado_personal . ",Nombres=" . $txt_nombre_personal . " ,";  
    $sql.="Apellidos="      . $txt_apellidos_personal      . " ,FechaIngreso="      .  
$txt_fecha_ingreso_personal . " ,";  
    $sql.="FechaNacimiento="      . $txt_fecha_nacimiento_personal . " ,DUI="      .  
$txt_oui_personal . " ,";  
    $sql.="NIT="      . $txt_nit_personal      . " ,Nacionalidad="      .  
$txt_nacionalidad_personal . " ,";  
    $sql.="AFP="      . $txt_afp_personal      . " ,ISSS="      . $txt_iss_s_personal      .  
" ,SalarioMensual="      . $txt_salario_personal . "" ;  
  
    if ($text_a_habilidad1_personal != "") {
```

```
$sql.="Habilidad1=" . $texta_habilidad1_personal . """;  
}  
  
if ($texta_habilidad2_personal != "") {  
    $sql.="Habilidad2=" . $texta_habilidad2_personal . """;  
}  
  
if ($texta_habilidad3_personal != "") {  
    $sql.="Habilidad3=" . $texta_habilidad3_personal . """;  
}  
  
if ($txt_pasaporte_personal != "") {  
    $sql.="Pasaporte=" . $txt_pasaporte_personal . """;  
}  
  
if ($txt_nup_personal != "") {  
    $sql.="NUP=" . $txt_nup_personal . """;  
}  
  
if ($txt_personal_personal != "") {  
    $sql.="Personalcol=" . $txt_personal_personal . """;  
}  
  
$sql.=" WHERE idPersonal=" . $cod_personal;  
  
$respuesta = mysql_query($sql, $connection);  
if ($respuesta) {  
    $mensaje = "PERSONAL MODIFICADO";  
    $bandera = 1;  
} else {  
    $mensaje = "SE PRODUJO UN ERROR AL MOMENTO DE ACTUALIZAR
```

```
EL PERSONAL. INTENTE DE NUEVO. ";
```

```
    $bandera = 0;
```

```
}
```

```
$jsonData["mensaje"] = $mensaje;
```

```
$jsonData["bandera"] = $bandera;
```

```
break;
```

Este es el código que permite modificar un dato ya existente en la base de datos y comienza siempre indicando donde se va a actualizar utilizando una variable “UPDATE” y un “SET” que guarda datos en el servidor por la actualización y al momento de modificarse muestra un mensaje que dice "LINEA MODIFICADA" si el valor es “\$Bandera = 1”, si la opción fuera “\$Bandera = 0” mostraría un mensaje de "SE PRODUJO UN ERROR AL MOMENTO DE ACTUALIZAR LA LINEA. INTENTE DE NUEVO." Terminando el código con un “Break” se utiliza para terminar la ejecución de la estructura actual en este caso sería la función IF y acepta un argumento numérico opcional al cual indica de cuantas estructuras anidadas encerradas se debe salir.

Case 4 Ejemplo de Registros de Personal. //eliminando línea

```
case 4:
```

```
    $sql = "SELECT COUNT(*) total from Proyectos where idTipoProyecto=" .  
$cod_personal;
```

```
    $query = mysql_query($sql, $connection);
```

```
    $row = mysql_fetch_array($query);
```

```
    if ($row['total'] == 0) {
```

```
        //eliminando al personal
```

```
        $sql = "DELETE FROM TipoProyecto where idTipoProyecto=" .  
$cod_personal;
```

```

    $respuesta = mysql_query($sql, $connection);
    if ($respuesta) {
        $mensaje = "TIPO DE PROYECTO ELIMINADO";
        $bandera = 1;
    } else {
        $mensaje = "SE PRODUJO UN ERROR AL MOMENTO DE ELIMINAR
EL TIPO DE PROYECTO. INTENTE DE NUEVO";
        $bandera = 0;
    }
} else {
    $mensaje = "NO SE PUEDE ELIMINAR EL TIPO DE PROYECTO
PORQUE EXISTEN REGISTROS QUE DEPENDE DE EL";
    $bandera = 0;
}
$jsonData["mensaje"] = $mensaje;
$jsonData["bandera"] = $bandera;

    break;
}
sleep(1);
echo json_encode($jsonData);
?>

```

Este código permite eliminar una línea de la base de datos, siguiendo la misma función que en los casos 1, 2 y 3, solamente que los mensajes serían cambiados si es “\$Bandera=1” aparecería "LINEA ELIMINADA" y si es “\$Bandera = 0” aparecerá como "SE PRODUJO UN ERROR AL MOMENTO DE ELIMINAR LA LINEA. INTENTE DE NUEVO " termina con un “Break” que en este caso acepta un argumento numérico opcional al cual indica de cuantas estructuras anidadas encerradas se debe salir.

Al final del código aparece un echo json_encode(\$jsonData); la cual se utiliza porque Devuelve la representación JSON de un valor.

- Beneficios:

Las personas que son beneficiadas con el software “Gestión de Construcción” serán muchas, entre ellas los ingenieros, los contratistas e incluso la misma empresa en donde se implemente dicho software.

En caso de cualquier empresa de construcción pueden llevar de una forma más segura y ordenada sus datos, sin temor a poder perder algún detalle guardado en la base de datos y así llevando un registro de cada dato guardado que se realizó por la persona solicitante de dicho proyecto.

Un beneficio importante es que la plataforma puede dominarse ya sea por LAN y por WEB.

```
<?php

require("../data.php");

$accion = isset($_GET['accion']) ? $_GET['accion'] : '';
$cod_personal = isset($_POST['hidden_cod_personal']) ? $_POST['hidden_cod_personal'] : '';

$sel_tipo_empleado_personal = isset($_POST['sel_tipo_empleado_personal']) ?
$_POST['sel_tipo_empleado_personal'] : '';
$txt_nombre_personal = isset($_POST['txt_nombre_personal']) ? $_POST['txt_nombre_personal']
: '';
$txt_apellidos_personal = isset($_POST['txt_apellidos_personal']) ?
$_POST['txt_apellidos_personal'] : '';
$txt_fecha_ingreso_personal = isset($_POST['txt_fecha_ingreso_personal']) ?
$_POST['txt_fecha_ingreso_personal'] : '';
$txt_fecha_nacimiento_personal = isset($_POST['txt_fecha_nacimiento_personal']) ?
$_POST['txt_fecha_nacimiento_personal'] : '';
$txt_oui_personal = isset($_POST['txt_oui_personal']) ? $_POST['txt_oui_personal'] : '';
$txt_nit_personal = isset($_POST['txt_nit_personal']) ? $_POST['txt_nit_personal'] : '';
$txt_pasaporte_personal = isset($_POST['txt_pasaporte_personal']) ?
$_POST['txt_pasaporte_personal'] : '';
$txt_afp_personal = isset($_POST['txt_afp_personal']) ? $_POST['txt_afp_personal'] : '';
$txt_nup_personal = isset($_POST['txt_nup_personal']) ? $_POST['txt_nup_personal'] : '';
$txt_iss_personal = isset($_POST['txt_iss_personal']) ? $_POST['txt_iss_personal'] : '';
$txt_nacionalidad_personal = isset($_POST['txt_nacionalidad_personal']) ?
$_POST['txt_nacionalidad_personal'] : '';
$texta_habilidad1_personal = isset($_POST['texta_habilidad1_personal']) ?
$_POST['texta_habilidad1_personal'] : '';
```

```
$texta_habilidad2_personal = isset($_POST['texta_habilidad2_personal']) ?  
$_POST['texta_habilidad2_personal'] : '';  
$texta_habilidad3_personal = isset($_POST['texta_habilidad3_personal']) ?  
$_POST['texta_habilidad3_personal'] : '';  
$txt_salario_personal = isset($_POST['txt_salario_personal']) ? $_POST['txt_salario_personal'] : '';  
$txt_personal_personal = isset($_POST['txt_personal_personal']) ?  
$_POST['txt_personal_personal'] : '';
```

Este código comienza abriendo un lenguaje PHP, utilizando un Required que es idéntico a include excepto que en caso de fallo producirá un error fatal de nivel E_COMPILE_ERROR. En otras palabras, éste detiene el script mientras que include sólo emitirá una advertencia (E_WARNING) lo cual permite continuar el script y en medio de los corchetes que hace una llamada a una página llamada data.php que es donde se hace la conexión a la base de datos.

```
<?php  
$connection = mysql_connect("localhost","root","");  
$db = mysql_select_db("proyectosciviles",$connection);  
  
?>
```

Y termina con las conexiones a los campos que existen en la base de datos, en este caso en la tabla de personal, utilizando las variables \$_GET y un isset que determina si una variable está definida y no es NULL y después va seguido de una variable \$_POST Asumiendo que el usuario envió por el método POST nombre= Leonel el resultado del ejemplo sería algo similar a: Leonel en la tabla personal y todos los datos que se han agregado.

```
<?php  
  
require("data.php");  
  
$txtusuario = $_POST['textuser'];  
$txtpass = $_POST['textpass'];  
$q="SELECT * FROM usuarios WHERE usuario = '$txtusuario' and Contraseña =  
md5('$txtpass')";  
echo $q;  
$query = mysql_query($q, $connection);  
$usuario = "";  
$pass = "";  
  
while ($row = mysql_fetch_array($query)) {  
    $usuario = $row['Usuario'];  
    $pass = $row['Contraseña'];  
}
```

```

}

if ($txtusuario == $usuario and md5($txtpass == $pass)) {
    session_start();
    $_SESSION["autenticado"] = "si";
    $_SESSION["usuario"] = $usuario;
    header("location:index.php");
} else {
    ?>
    <script language="Javascript" type="text/javascript">
        alert("Combinación incorrecta de Usuario / Contraseña");
    </script>

    <?php
}
?>

```

Este código comienza abriendo un lenguaje PHP, utilizando un Required que es idéntico a include excepto que en caso de fallo producirá un error fatal de nivel E_COMPILE_ERROR. En otras palabras, éste detiene el script mientras que include sólo emitirá una advertencia (E_WARNING) lo cual permite continuar el script y en medio de los corchetes que hace una llamada a una página llamada data.php que es donde se hace la conexión a la base de datos.

Se indican las variables para poderlas utilizar al momento de hacer la función, en este caso se utiliza un "If" si el usuario y la contraseña están existentes en la base de dato, buscara la página index.php y si no está el <script> que es un mensaje de alerta, si la contraseña y el usuario no se ha encontrado en la base de datos, mostrara un mensaje de alerta que dirá ("Combinación incorrecta de Usuario / Contraseña"), después se cierra el lenguaje php y termina la función de logear.

```

<?php
include("data.php");

$Nombres = $_POST['Nombres'];
$Apellidos = $_POST['Apellidos'];
$FechaIngreso = $_POST['FechaIngreso'];
$FechaNacimiento = $_POST['FechaNacimiento'];
$Habilidad1 = $_POST['Habilidad1'];
$Habilidad2 = $_POST['Habilidad2'];
$Habilidad3 = $_POST['Habilidad3'];
$DUI = $_POST['DUI'];
$NIT = $_POST['NIT'];

```

```

$Pasaporte = $_POST['Pasaporte'];
$Nacionalidad = $_POST['Nacionalidad'];
$AFP= $_POST['AFP'];
$NUP = $_POST['NUP'];
$ISSS = $_POST['ISSS'];
$SalarioMensual = $_POST['SalarioMensual'];

$query = mysql_query("INSERT INTO `proyectosciviles`.`personal` (`Nombres`,
`Apellidos`, `FechaIngreso`, `FechaNacimiento`, `Habilidad1`, `Habilidad2`,
`Habilidad3`, `DUI`, `NIT`, `Pasaporte`, `Nacionalidad`, `AFP`, `NUP`, `ISSS`,
`SalarioMensual`) VALUES ('$Nombres', '$Apellidos', '$FechaIngreso',
'$FechaNacimiento', '$Habilidad1', '$Habilidad2', '$Habilidad3', '$DUI', '$NIT',
'$Pasaporte', '$Nacionalidad', '$AFP', '$NUP', '$ISSS', '$SalarioMensual');");

header('location: RegistrarUsuarios.php')

?>

```

Primero se comienza abriendo el lenguaje php, después se conecta la base de datos, que es el data.php con un include incluye y evalúa el archivo especificado. Después de eso se indican las variables de \$Nombres = \$_POST['Nombres'] utilizando el método POST con todos los datos que seria, nombre, apellido, fecha ingreso, fecha fin, habilidad1, habilidad2. Habilidad3, DUI, NIT, Pasaporte, nacionalidad, AFP, NUP, ISSS, salario mensual. Después se hace una conexión con un \$query que envía datos que envía una consulta MySQL con la función Insert into proyectos civiles que es la base de datos existente en el servidor.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Gestión de Construcción</title>
    <link rel="stylesheet" href="css/jquery-ui.css" />
    <script src="js/jquery-1.9.1.js"></script>
    <script src="js/jquery-ui.js"></script>
    <script src="js/registroLinea.js"></script>
    <?php
    include("header.php");
    require("data.php");
    ?>

    <link href="SpryAssets/SpryValidationTextField.css" rel="stylesheet"
type="text/css">

```



```

</head>

<body class="home">
  <!-- WRAPPER -->
  <div id="wrapper">
    <?php include("baner.php") ?>
    <!-- MAIN -->
    <div id="main">

      <!-- content --><!-- ENDS content -->

      <!-- Twitter -->
      <div id="twitter">
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p>&nbsp;</p>
      </div>
      <!-- ENDS Twitter -->

    </div>
    <!-- ENDS MAIN -->

    <!-- FOOTER -->
    <div id="footer">
      <div id="content_form">
        <h2>Registros de Linea</h2>
        <div id="id_div_form_buscar">
          <form id="id_form_buscar_linea" class="form-style">
            <table width="100%" >
              <tr>
                <td width="18%">Buscar linea por:</td>
                <td width="10%">
                  <select id="id_sel_buscar_linea"
name="sel_buscar_linea">
                    <option value="">Seleccione...</option>
                    <option value="l.Descripcion">Nombre de
Linea</option>
                    <option value="u.Descripcion">Nombre de
Unidad</option>
                  </select>
                </td>
                <td width="50%">
                  <input type="text" id="id_txt_texto_linea"
name="txt_texto_linea">
                </td>
                <td width="5%" align="right">

```

```

        <input type="button" id="id_btn_buscar_linea"
name="btn_buscar_linea" value="Buscar">
    </td>
    <td width="17%"></td>
</tr>
</table>
</form>
<div id="id_result_busqueda_linea" align="center" style="overflow-
y:auto; "></div>
</div>
<div id="id_div_form_registrar">

    <form id="id_registrar_linea" class="form-style">
        <table width="100%">
            <tr>
                <td>Unidad (*):</td>
                <td>
                    <select id="id_sel_unidad_linea"
name="sel_unidad_linea">
                        <option value="">Selecione..</option>
                        <?php
                            $query = mysql_query("SELECT * FROM Unidad
ORDER BY Descripcion ASC", $connection);

                            while ($row = mysql_fetch_array($query)) {
                                ?>
                                <option value="<?php echo $row['idUnidad'];
?>"><?php echo $row['Descripcion']; ?></option>
                                <?php
                                    }
                                ?>
                            </select>
                        </td>
                        <td></td>
                    </tr>
                    <tr>
                        <td width="25%">Descripcion (*):</td>
                        <td width="45%">
                            <input type="hidden" id="id_hidden_cod_linea"
name="hidden_cod_linea">
                            <textarea id="id_texta_descripcion_linea"
name="texta_descripcion_linea"></textarea>
                        </td>
                        <td width="30%" aling="left"></td>
                    </tr>
                    <tr>
                        <td></td>

```

```

                <td align="right">
                    <input type="button" id="id_btn_add_linea"
name="btn_add_linea" value="Agregar">
                    <input type="button" id="id_btn_update_linea"
name="btn_update_linea" value="Modificar" style="display: none">
                    <input type="button" id="id_btn_delete_linea"
name="btn_delete_linea" value="Eliminar" style="display: none">
                    <input type="button" id="id_btn_clean_linea"
name="btn_clean_linea" value="Limpiar">
                    <input type="button" id="id_btn_cancel_linea"
name="btn_cancel_linea" value="Cancelar" style="display: none">
                </td>
            </td></td>
        </tr>
        <tr>
            <td colspan="3"><span style="color: #FFD800">Son
campos requeridos (*)</span></td>
        </tr>
    </table>
</form>
</div>

</div>
<div id="dialog"></div>

<!-- footer-cols -->                                <!-- ENDS footer-cols -->
<div id="bottom">
    <?php include("bottom.php") ?>
</div>

<!-- ENDS Bottom -->
</div>
<!-- ENDS FOOTER -->

</div>
<!-- ENDS WRAPPER -->

</html>

```

Este código esta combinado con código de estructura HTML y código de lenguaje programable que es PHP, más que todo aquí solo se utilizan los Include para llamar algunas partes, como la del menú, o la del header, entre otras para tener un control más ordenado de los datos y no digitar el código a cada momento. Por ejemplo:

```
<?php
include("header.php");
require("data.php");
?>
```

En este se manda a llamar el header.php el código de header.php es básico pero se utiliza mucho en el proyecto.

```
<?php
require("data.php");
?>
<!-- CSS -->
    <link rel="stylesheet" href="css/style.css" type="text/css"
media="screen" />
    <link rel="stylesheet" href="css/social-icons.css" type="text/css"
media="screen" />
    <!--[if IE 8]>
        <link rel="stylesheet" type="text/css" media="screen"
href="/css/ie8-hacks.css" />
    <![endif]-->
    <!-- ENDS CSS -->

    <!-- GOOGLE FONTS -->
    <!--<link href='http://fonts.googleapis.com/css?family=Ubuntu'
rel='stylesheet' type='text/css'>-->

    <!-- JS -->
    <script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
    <script type="text/javascript" src="js/jquery-ui-
1.8.13.custom.min.js"></script>
    <script type="text/javascript" src="js/easing.js"></script>
    <script type="text/javascript" src="js/jquery.scrollTo-1.4.2-
min.js"></script>
    <script type="text/javascript" src="js/quicksand.js"></script>
    <script type="text/javascript" src="js/jquery.cycle.all.js"></script>
    <script type="text/javascript" src="js/custom.js"></script>
    <!--[if IE]>
        <script
src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->

    <!--[if IE 6]>
        <script type="text/javascript"
src="js/DD_belatedPNG.js"></script>
    <script>
```

```

/* EXAMPLE */
//DD_belatedPNG.fix('*');
</script>
<![endif]-->
<!-- ENDS JS -->

<!-- Nivo slider -->
<link rel="stylesheet" href="css/nivo-slider.css" type="text/css"
media="screen" />
<script src="js/nivo-slider/jquery.nivo.slider.js"
type="text/javascript"></script>
<!-- ENDS Nivo slider -->

<!-- tabs -->
<link rel="stylesheet" href="css/tabs.css" type="text/css"
media="screen" />
<script type="text/javascript" src="js/tabs.js"></script>
<!-- ENDS tabs -->

<!-- prettyPhoto -->
<script type="text/javascript"
src="js/prettyPhoto/js/jquery.prettyPhoto.js"></script>
<link rel="stylesheet" href="js/prettyPhoto/css/prettyPhoto.css"
type="text/css" media="screen" />
<!-- ENDS prettyPhoto -->

<!-- superfish -->
<link rel="stylesheet" media="screen" href="css/superfish.css" />
<!--<link rel="stylesheet" media="screen" href="css/superfish-
left.css" />-->
<script type="text/javascript" src="js/superfish-
1.4.8/js/hoverIntent.js"></script>
<script type="text/javascript" src="js/superfish-
1.4.8/js/superfish.js"></script>
<script type="text/javascript" src="js/superfish-
1.4.8/js/supersubs.js"></script>
<!-- ENDS superfish -->

<!-- poshytip -->
<link rel="stylesheet" href="js/poshytip-1.0/src/tip-twitter/tip-
twitter.css" type="text/css" />
<link rel="stylesheet" href="js/poshytip-1.0/src/tip-yellowsimple/tip-
yellowsimple.css" type="text/css" />
<script type="text/javascript" src="js/poshytip-
1.0/src/jquery.poshytip.min.js"></script>
<!-- ENDS poshytip -->

```

```

        <!-- Tweet -->
        <!--<link rel="stylesheet" href="css/jquery.tweet.css" media="all"
type="text/css"/>-->
        <script src="js/tweet/jquery.tweet.js" type="text/javascript"></script>
        <!-- ENDS Tweet -->

        <!-- Fancybox -->
        <link rel="stylesheet" href="js/jquery.fancybox-
1.3.4/fancybox/jquery.fancybox-1.3.4.css" type="text/css" media="screen" />
        <script type="text/javascript" src="js/jquery.fancybox-
1.3.4/fancybox/jquery.fancybox-1.3.4.pack.js"></script>
        <!-- ENDS Fancybox -->

        <!-- SKIN -->
        <link rel="stylesheet" href="skins/plastic/style.css" type="text/css"
media="screen" />

```

En el header.php es puro código CSS que se utiliza para diseño de las diferentes páginas que se utilizan en el proyecto “Gestión Construcción”.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Inicio de Sesión :: Gestión de Construcción</title>
    <?php include("header.php") ?>
    <script src="SpryAssets/SpryValidationTextField.js"
type="text/javascript"></script>
    <link href="SpryAssets/SpryValidationTextField.css" rel="stylesheet"
type="text/css">
  </head>

  <body class="home">
    <!-- WRAPPER -->
    <div id="wrapper">

      <?php include("baner.php") ?>

      <!-- MAIN -->
      <div id="main">

        <!-- content --><!-- ENDS content -->

```

```

<!-- Twitter -->
<div id="twitter"></div>
<!-- ENDS Twitter -->

</div>
<!-- ENDS MAIN -->

<!-- FOOTER -->
<div id="footer">

    <!-- footer-cols -->
    <ul id="footer-cols">
        <li class="col"></li>
        <li class="col">
            <h6>&nbsp;</h6>
            <ul>
                <li></li>
                <li></li>
                <li>
                    <form name="form3" method="post" action="logear.php">
                    </form>
                </li>
            </ul>
        </li>
    </ul>
    </li>
    <?php
    if (isset($_SESSION["autenticado"])) {
        echo "<h2>Usted se ha autenticado como: " .
$_SESSION['usuario'] . "</h2>";
    } else {
        ?>
        <li class="col">
            <h6>LOGIN:</h6>
        </li>

        <li class="col">
            <form name="form4" method="post" action="logear.php">
                <p>
                    <label for="textfield3"></label>
                    Usuario:</p>
                <p>
                    <label for="textfield3"></label>
                    <span id="sprytextfield1">
                        <input name="textuser" type="text" class="highlight"
id="textfield3">
                        <span class="textfieldRequiredMsg">Digite su
Usuario.</span></span></p>
            </form>
        </li>
    </ul>
    </div>

```

```

        <p>&nbsp;</p>
        <p>Contraseña:</p>
        <p>
            <label for="textfield4"></label>
            <span id="sprytextfield2">
                <input name="textpass" type="password"
class="highlight" id="textfield4">
                <span class="textfieldRequiredMsg">Digite su
contraseña.</span></span></p>
        <p>&nbsp;</p>
        <p>
            <input type="submit" name="button" id="button"
value="Ingresar">
        </p>
    </form>
</li>
<?php } ?>

</ul>
<!-- ENDS footer-cols -->

<!-- Bottom -->
<div id="bottom">
    <?php include("bottom.php") ?>
</div>
<!-- ENDS Bottom -->
</div>
<!-- ENDS FOOTER -->

</div>
<!-- ENDS WRAPPER -->

<script type="text/javascript">
    var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1");
    var sprytextfield2 = new Spry.Widget.ValidationTextField("sprytextfield2");
</script>
</body>

</html>

```

El código más importante que tiene el index.php es este:

```

<?php
if (isset($_SESSION["autenticado"])) {
    echo "<h2>Usted se ha autenticado como: " .
$_SESSION['usuario'] . "</h2>";

```



```
} else {  
?>
```

Este código es utilizado para un autenticado de sesión, para que cuando el usuario se conecte salga su nombre de usuario, es como una bienvenida, utilizando un isset que determina si una variable está definida y no es NULL y cuando todo este correcto le mostrara el mensaje en la parte inferior de la página que dirá, “Usted se ha autenticado como: y mi usuario que sería Zimmer.

```
<?php  
session_start();  
session_destroy();  
header("Location:index.php");  
?>
```

Este es el código denominado Logout.php que es utilizado únicamente para cerrar la sesión del proyecto.

```
?php  
  
//vars  
$subject = $_POST['subject'];  
$to = explode(',', $_POST['to'] );  
  
$from = $_POST['email'];  
  
//data  
$msg = "NAME: " .$_POST['name'] . "<br>\n";  
$msg .= "EMAIL: " .$_POST['email'] . "<br>\n";  
$msg .= "WEBSITE: " .$_POST['web'] . "<br>\n";  
$msg .= "COMMENTS: " .$_POST['comments'] . "<br>\n";  
  
//Headers  
$headers = "MIME-Version: 1.0\r\n";  
$headers .= "Content-type: text/html; charset=UTF-8\r\n";  
$headers .= "From: <".$from. ">" ;  
  
//send for each mail  
foreach($to as $mail){  
    mail($mail, $subject, $msg, $headers);  
}  
  
?>
```

Este es el código que utiliza la página web para mandar un mensaje de las personas que necesitan contratar el servicio de la empresa de construcción.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Gestión de Construcción</title>
    <link rel="stylesheet" href="css/jquery-ui.css" />
    <script src="js/jquery-1.9.1.js"></script>
    <script src="js/jquery-ui.js"></script>
    <script src="js/money/jquery.price_format.1.8.min.js"></script>
    <script src="js/registroPersonal.js"></script>
    <?php
    include("header.php");
    require("data.php");
    ?>

    <link href="SpryAssets/SpryValidationTextField.css" rel="stylesheet"
type="text/css">
  </head>

  <body class="home">
    <!-- WRAPPER -->
    <div id="wrapper">
      <?php include("baner.php") ?>
      <!-- MAIN -->
      <div id="main">

        <!-- content --><!-- ENDS content -->

        <!-- Twitter -->
        <div id="twitter">
          <p>&nbsp;</p>
          <p>&nbsp;</p>
          <p>&nbsp;</p>
        </div>
        <!-- ENDS Twitter -->
      </div>
      <!-- ENDS MAIN -->

      <!-- FOOTER -->
      <div id="footer">
        <div id="contents">

          <div id="content_form">
            <h2>Registros de Personal</h2>
```

```

<div id="id_div_form_buscar">
  <form id="id_form_buscar_personal" class="form-style">
    <table width="100%" >
      <tr>
        <td width="20%">Buscar Personal por:</td>
        <td width="10%">
          <select id="id_sel_buscar_personal"
name="sel_buscar_personal">
            <option value="">Seleccione...</option>
            <option value="P.Nombres">Nombres</option>
            <option value="P.Apellidos">Apellidos</option>
            <option value="P.DUI">DUI</option>
            <option value="P.NIT">NIT</option>
          </select>
        </td>
        <td width="50%">
          <input type="text" id="id_txt_texto_personal"
name="txt_texto_personal">
        </td>
        <td width="5%" align="right">
          <input type="button" id="id_btn_buscar_personal"
name="btn_buscar_personal" value="Buscar">
        </td>
        <td width="17%"></td>
      </tr>
    </table>
  </form>
  <div id="id_result_busqueda_personal" align="center"
style="overflow-y:auto; "></div>
</div>
<div id="id_div_form_registrar">

  <form id="id_registrar_personal" class="form-style">
    <table width="100%">
      <tr>
        <td>Tipo de Empleado (*):</td>
        <td>
          <select id="id_sel_tipo_empleado_personal"
name="sel_tipo_empleado_personal">
            <option value="">Seleccione..</option>
            <?php
              $query = mysql_query("SELECT * FROM
TipoEmpleado ORDER BY Descripcion ASC", $connection);

              while ($row = mysql_fetch_array($query)) {
                ?>
                <option value="<?php echo

```

```

$row['idTipoEmpleado']; ?>"><?php echo $row['Descripcion']; ?></option>
        <?php
            }
        ?>
    </select>
</td>
<td></td>
</tr>
<tr>
    <td width="25%">Nombres (*):</td>
    <td width="45%">
        <input type="hidden" id="id_hidden_cod_personal"
name="hidden_cod_personal">
        <input type="text" id="id_txt_nombre_personal"
name="txt_nombre_personal">
    </td>
    <td width="30%" align="left"></td>
</tr>
<tr>
    <td>Apellidos (*):</td>
    <td>
        <input type="text" id="id_txt_apellidos_personal"
name="txt_apellidos_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>Fecha Ingreso (*):</td>
    <td>
        <input type="text" id="id_txt_fecha_ingreso_personal"
name="txt_fecha_ingreso_personal" readonly="true">
    </td>
    <td></td>
</tr>
<tr>
    <td>Fecha Nacimiento (*):</td>
    <td>
        <input type="text"
id="id_txt_fecha_nacimiento_personal" name="txt_fecha_nacimiento_personal"
readonly="true">
    </td>
    <td></td>
</tr>
<tr>
    <td>DUI (*):</td>
    <td>
        <input type="text" id="id_txt_dui_personal"

```

```

name="txt_dui_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>NIT (*):</td>
    <td>
        <input type="text" id="id_txt_nit_personal"
name="txt_nit_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>Pasaporte:</td>
    <td>
        <input type="text" id="id_txt_pasaporte_personal"
name="txt_pasaporte_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>AFP (*):</td>
    <td>
        <input type="text" id="id_txt_afp_personal"
name="txt_afp_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>NUP:</td>
    <td>
        <input type="text" id="id_txt_nup_personal"
name="txt_nup_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>ISSS (*):</td>
    <td>
        <input type="text" id="id_txt_iss_s_personal"
name="txt_iss_s_personal">
    </td>
    <td></td>
</tr>
<tr>
    <td>Nacionalidad (*):</td>
    <td>

```

```

                <input type="text" id="id_txt_nacionalidad_personal"
name="txt_nacionalidad_personal">
            </td>
        </td></td>
    </tr>
    <tr>
        <td>Habilidad 1:</td>
        <td>
            <textarea id="id_texta_habilidad1_personal"
name="texta_habilidad1_personal"></textarea>
        </td>
        <td></td>
    </tr>
    <tr>
        <td>Habilidad 2:</td>
        <td>
            <textarea id="id_texta_habilidad2_personal"
name="texta_habilidad2_personal"></textarea>
        </td>
        <td></td>
    </tr>
    <tr>
        <td>Habilidad 3:</td>
        <td>
            <textarea id="id_texta_habilidad3_personal"
name="texta_habilidad3_personal"></textarea>
        </td>
        <td></td>
    </tr>
    <tr>
        <td>Salario Mensual (*):</td>
        <td>
            <input type="text" id="id_txt_salario_personal"
name="txt_salario_personal">
        </td>
        <td></td>
    </tr>
    <tr>
        <td>Personal:</td>
        <td>
            <input type="text" id="id_txt_personal_personal"
name="txt_personal_personal">
        </td>
        <td></td>
    </tr>
    <tr>
        <td></td>

```

```

                <td align="right">
                    <input type="button" id="id_btn_add_personal"
name="btn_add_personal" value="Agregar">
                    <input type="button" id="id_btn_update_personal"
name="btn_update_personal" value="Modificar" style="display: none">
                    <input type="button" id="id_btn_delete_personal"
name="btn_delete_personal" value="Eliminar" style="display: none">
                    <input type="button" id="id_btn_clean_personal"
name="btn_clean_personal" value="Limpiar">
                    <input type="button" id="id_btn_cancel_personal"
name="btn_cancel_personal" value="Cancelar" style="display: none">
                </td>
            </td></td>
        </tr>
        <tr>
            <td colspan="3"><span style="color: #FFD800">Son
campos requeridos (*)</span></td>
        </tr>

```

```

    </table>

```

```

    </form>

```

```

</div>

```

```

</div>

```

```

<div id="dialog"></div>

```

```

<br>

```

```

<br>

```

```

<br>

```

```

<br>

```

```

<br>

```

```

<br>

```

```

</div>

```

```

<!-- footer-cols -->

```

```

<!-- ENDS footer-cols -->

```

```

<div id="bottom">

```

```

    <?php include("bottom.php") ?>

```

```

</div>

```

```

<!-- ENDS Bottom -->

```

```

</div>

```

```

<!-- ENDS FOOTER -->

```

```

</div>

```

```

<!-- ENDS WRAPPER -->

```

```
</html>
```

Este es la página de RegistrarPersonal.php el código más importante sería el de las librerías del calendario, para tener un calendario más dinámico. Explicando más o menos el código HTML los `
` son para dar un espacio en la hoja, los `<div>` son para dividir una tabla ya existente dentro de otra tabla, un `<tr>` es como dar un salto o un enter, siempre son utilizados para espacios

```
<?php
    $query = mysql_query("SELECT * FROM
TipoEmpleado ORDER BY Descripcion ASC", $connection);

    while ($row = mysql_fetch_array($query)) {
        ?>
        <option value="<?php echo
$row['idTipoEmpleado']; ?>"><?php echo $row['Descripcion']; ?></option>
        <?php
```

Primero siempre indicando que es un lenguaje php, utilizando una variable `$query = mysql_query` que envía una consulta MySQL a la base de datos, haciendo una función sql seleccionando la tabla de Tipo Empleado utilizando una sentencia sql que es el ORDER BY que funciona para ordenar la llegada de los datos en el campo de Descripcion haciendo la conexión con una variable `$connection`.

Después se utiliza una función “while” hagarrando el valor, ingresándolo en una variable llamada ROW

Como pueden ver el código está documentado para los casos de programación o espacios para no perderse, teniendo el código más ordenado y no perderse al momento de buscar algún código.