
**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**



SISTEMAS Y PROCEDIMIENTOS

MANUAL DEL PROGRAMADOR

DOCENTE:

Ing. Diego Armando Herrera Flores

ESTUDIANTES:

Alejandra Claire Aguilar Mata AM19089
Junior Mauricio Villalta Flores VM19012
Jorge Eduardo Romero Garcia RG19041
Cristian Alexander Guardado Escobar GE19008
Jorge Luis Granados Umanzor GU19002

CICLO I 2021

Junio – 2021, San Miguel, El Salvador, Centroamérica

Indice

Introducción.....	3
objetivo.....	3
Requerimientos técnicos del sistema.....	4
Diagrama DCU.....	5
Normalización 1FN 2FN 3FN.....	6
DIAGRAMA ENTIDAD RELACION (ER).....	7
Descripción de las tablas de la base de datos.....	8
Descripción y definición de variables usadas en el sistema.....	9
asistencia.php.....	9
agregar.php.....	10
form_agregar.php.....	11
modificar.php.....	12
form_modificar.php.....	13
conexion.php.....	14
Head.php.....	15
Clock.js.....	16
config_horario.js.....	16
config_horario.php.....	17
eliminar.php.....	19
funciones.php.....	19
reporteRango.js.....	21
ver_llegadas.js.....	22
marcar.js.....	23
verAsistencia.php.....	24
ver_llegadas.php.....	25
reporteDiario.php.....	26
login.php.....	27
ins_marca_dia.php.....	28
Diccionario de datos.....	30
Prerrequisitos.....	31
Bitacora de cambios.....	32

Introducción

El propósito de este manual del programador, es dar a conocer al lector los códigos fuentes del programa realizado. Para ello tratamos de la forma más concisa de explicar cada uno de los códigos, junto con la programación utilizada en el desarrollo del software, esto con el fin de que el usuario pueda modificar a gusto alguno de los valores y parámetros de las funciones que se encuentran expuestas en la programación de sistema "SCAP" (Sistema de Control de Asistencia de Personal)

objetivo

Instruir el uso adecuado del Sistema "SCAP" , para el acceso oportuno del mismo, así como la descripción de los archivos relevantes del sistema los cuales nos orienten en la configuración y soporte del mismo.

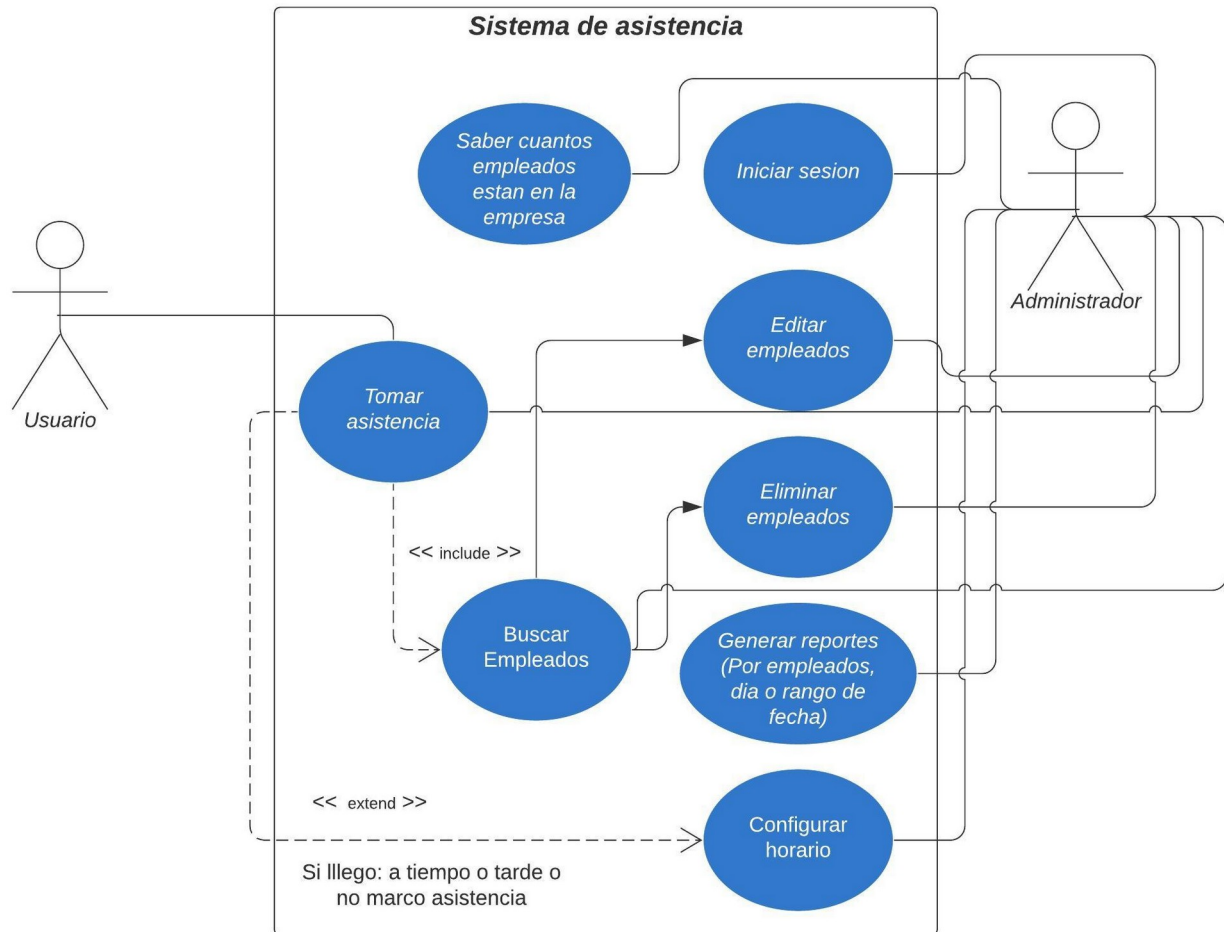
Requerimientos técnicos del sistema

El sistema puede ser instalado en cualquier sistema operativo que cumpla con los siguientes requerimientos:

- Servidor Web Apache 2.0 o Superior
- XAMP
- PHP 7.0 o Superior
- MySql 5.0 o Superior
- Navegador Web
- Windows 7 o superiores

Diagrama DCU

Diagrama de caso de uso



Normalización 1FN 2FN 3FN

Realizamos una base con todos los datos que serán utilizados.

Datos necesarios	
codigo	Código de empleado para iniciar sesión o marcar asistencia
nombres	Nombre completo del usuario.
dui	Numero único de identidad
direccion	Dirección del usuario.
telefono	Teléfono del usuario
puesto	El puesto será ingresado como texto ya que de momento no tendrá mayor utilidad.
estado	Si trabaja o no en la empresa (1-0)
tipo	Si es administrador o empleado (1-0)
contrasenia	Contraseña si es administrador
disponible	Si se encuentra dentro de la empresa.
fecha	Dia que marco registro.
hora_e	Hora de entrada. (automático)
Hora_s	Hora de salida. (automático)

Creamos una tabla con cada grupo de datos relacionados y los identificamos con una clave primaria.

empleados	
Codigo (PK)	Clave primaria
nombres	Datos del empleado
dui	Datos del empleado
direccion	Datos del empleado
telefono	Datos del empleado
puesto	Datos del empleado
estado	Datos del empleado
tipo	Datos del empleado
contrasenia	Datos del empleado
disponible	Datos del empleado

Relacionamos estos grupos mediante una clave foránea.

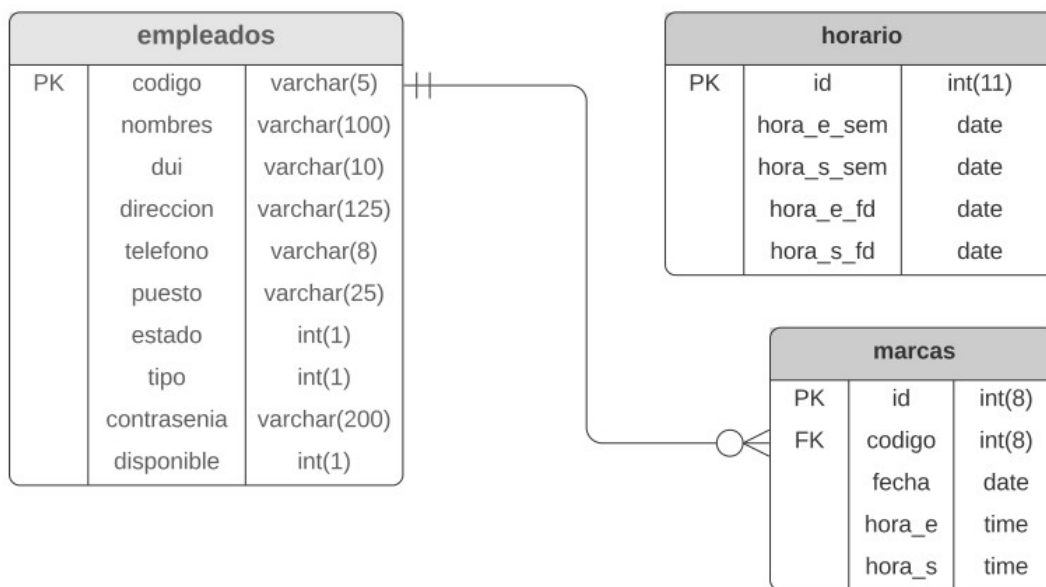
marcas	
id (PK)	Clave primaria.
codigo (FK)	Clave foránea.
fecha	Fecha. (automática)
hora_e	Hora de entrada marcada.
Hora_s	Hora de salida marcada.

Dentro de la base de datos existe una tabla que se utiliza para establecer la hora de entrada y salida de la empresa para poder comparar con la hora marcada en la tabla marcas.

horarios	
Id (PK)	Clave primaria
hora_e_sem	Hora de entrada de lunes a viernes.
hora_s_sem	Hora de salida de lunes a viernes.
hora_e_fd	Hora de entrada los fines de semana.
hora_s_fd	Hora de salida los fines de semana.

DIAGRAMA ENTIDAD RELACION (ER)

Diagrama entidad relacion
(ER)



Descripción de las tablas de la base de datos

Vamos a ir detallando el nombre de cada tabla y a continuación una descripción del contenido.

empleados

la tabla “empleados” contiene todos los empleados de la empresa con datos como:

- **codigo:** el código sera de 4 dígitos
- **nombres:** aquí se almacena el nombre completo del empleado
- **DUI:** Aquí se almacena el dui del empleado; aceptara 10 dígitos
- **dirección:** Aquí se almacena la dirección del empleado
- **teléfono:** aquí se almacenara el numero del empleado
- **puesto:** aquí se almacena el puesto que tiene el empleado, ejemplo, secretaria
- **estado:** aquí almacena si el empleado esta contratado o si ya fue despedido, 0 es si esta despedido y 1 cuando esta contratado
- **tipo:** aquí se almacena si el empleado tiene acceso de administrador, si tiene acceso de administrador es 1 y si es un empleado sin acceso de administrador 0
- **contrasenia:** si el empleado tiene acceso de administrador se le pedirá una contraseña y aquí es donde se almacena encriptada
- **disponible:** aquí se almacena si el empleado esta dentro de la empresa, si esta dentro de la empresa sera 1, si esta fuera de la empresa sera 0.

horario

la tabla “horario” contiene los horarios de lunes a viernes también del sábado y domingo

- **id:** el id del horario y único también
- **hora_e_sem:** hora de entrada de los días de lunes a viernes
- **hora_s_sem:** hora de salida de los días de lunes a viernes
- **hora_e_fd:** hora de entrada del fin de semana
- **hora_s_fd:** hora de salida del fin de semana

marcas

la tabla “marcas: contiene la marcas de asistencia de los respectivos empleados

- **id:** id de la marca
 - **codigo:** aquí almacena el código del empleado
 - **fecha:** aquí almacena la fecha en el que hizo marca de asistencia
 - **hora_e:** aquí almacena la hora de entrada en que marco el empleado
 - **hora_s:** aquí almacena la hora de salida en que marco el empleado
-

Descripción y definición de variables usadas en el sistema

asistencia.php

```
<?php
require_once("head.php");
require_once("leftmenu.php");

$activo = 1;
$consulta = "SELECT * FROM empleados WHERE activo = '$activo'";
$resultados = $con -> query($consulta);
$activos = $resultados -> fetch_all();

function iterarActivos($activos){
    $limite = count($activos);

    for ($i=0; $i < $limite; $i++) {
        imprimirActivo($i+1,$activos[$i][0],$activos[$i][1],$activos[$i][5]);
    }
}

function imprimirActivo($numero,$codigo,$nombre,$cargo){

    echo <<<_ACTIVO
        <tr>
            <td>$numero</td>
            <td>$codigo</td>
            <td>$nombre</td>
            <td>$cargo</td>
        <tr>
    _ACTIVO;
}
?>
```

En este documento php tenemos la interfaz y la lógica para ver qué empleados están trabajando en este momento. Hace una consulta a la base de datos de los empleados, donde su valor activo es igual a 1. Los itera con la función `iterarActivos()` y mientras los itera los imprime con la función `imprimirActivos()`.

- **activo:** guarda el parámetro a buscar para hacer la consulta en los empleados
 - **consulta:** guarda la sentencia completa de la consulta que se realizará.
 - **resultados:** Es un array que almacena los resultados de la consulta realizada a la base de datos
 - **con:** Es un objeto que nos permite realizar la consulta y que su explicación está en `conexion.php`, por eso en este documento
 - **activos:** Esta guarda lo que retorna resultados con el método `fetch_all()` que es, en este caso, un array numérico con los usuarios que están activos.
 - **limite:** Sirve para saber cuantas veces iterara en el ciclo for los empleados activos
-

agregar.php

```
<?php
include("conexion.php");

if(!empty($_POST)){
    $codigo = mysqli_real_escape_string($con,$_POST["codigo"]);
    $nombres = mysqli_real_escape_string($con,$_POST["nombres"]);
    $dui = mysqli_real_escape_string($con,$_POST["dui"]);
    $direccion = mysqli_real_escape_string($con,$_POST["direccion"]);
    $telefono = mysqli_real_escape_string($con,$_POST["telefono"]);
    $puesto = mysqli_real_escape_string($con,$_POST["puesto"]);
    $tipo = mysqli_real_escape_string($con,$_POST["tipo"]);
    $encriptar = mysqli_real_escape_string($con,$_POST["pass"]);
    $pass = sha1(strval($encriptar));
    //mysqli_close($con);
    if($codigo != $_SESSION['id_empleado']){
        $consulta = "SELECT * FROM empleados WHERE codigo = '$codigo'";
        $resultado = $con -> query($consulta);
        $filas = $resultado -> num_rows;

        if($filas > 0){
            $fila = $resultado -> fetch_assoc();
            if($fila["estado"] != 0){
                echo "<script type='text/javascript'>alert('Código ya existe');</script>";
            }else{
                echo "<script type='text/javascript'>alert('Intente con otro código');</script>";
            }
        }else{
            $consulta = "INSERT INTO empleados(codigo, nombres, dui, direccion, telefono, puesto, estado,tipo,contrasenia,activo) VALUES('$codi";
            $mensaje = 'Empleado agregado con exito!';
            $con -> query($consulta);
            echo "<script type='text/javascript'>alert('$mensaje');</script>";
        }
    }else{
        $mensaje = 'No puedes volver a registrarte';
        echo "<script type='text/javascript'>alert('$mensaje');</script>";
    }
}
```

En este documento incluimos conexion.php para poder acceder al objeto **con**.

El fin de este es poner guardar un empleado en la base de datos con la información que viene en la superglobal **\$_POST**. Si el código que viene es diferente al usuario que ha iniciado la sesión entonces se ejecuta la lógica de agregar, si no, se devuelve un mensaje de error. Para agregar el usuario se hace una consulta donde intenta traer los empleados que tienen el código que desea registrarse, si no se encuentra el código se hace una consulta para hacer un INSERT en la base de datos con la información del nuevo empleado.

- **codigo, nombres, dui, direccion, telefono, puesto, tipo:** Todas estas variables guardan respectivamente los datos de un empleado que su nombre describe. **encriptar:** Guarda la contraseña sin caracteres especiales que devuelve `mysqli_real_escape_string`.
- **pass:** Guarda la contraseña encriptada por sha1
- **consulta:** guarda la sentencia completa de la consulta que se realizará.
- **resultados:** Es un array que almacena los resultados de la consulta realizada a la base de datos
- **filas:** En primer instancia almacena el número de filas que retorna resultados a través del método `num_rows`. Luego más adelante guarda los resultados de la consulta.

- **mensaje:** Es un string con feedback para el usuario.

form_agregar.php

```
<div class="container mt-5">
  <form class="row d-flex flex-column" action="" method="post">
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <input type="text" name="codigo" class="form-control" placeholder="Código de empleado" required>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <input type="text" name="nombres" class="form-control" placeholder="Nombre completo" required>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <div class="row">
        <div class="col-6">
          <input type="text" name="dui" class="form-control" placeholder="DUI" required>
        </div>
        <div class="col-6">
          <input type="text" name="telefono" class="form-control" placeholder="Teléfono" required>
        </div>
      </div>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <textarea name="direccion" class="form-control" placeholder="Dirección" required></textarea>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <input type="text" name="puesto" class="form-control" placeholder="Puesto" required>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <label class="control-label">Tipo de usuario:</label>
      <select name="tipo" class="form-select form-select-sm" aria-label=".form-select-sm example">
        <option value="0">Empleado</option>
        <option value="1">Administrador</option>
      </select>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <input type="password" id="contra" name="pass" class="form-control" placeholder="Contraseña" required>
    </div>
    <div class="container mb-3 col-12 col-sm-12 col-md-12 col-lg-10 col-xl-8 col-xxl-6">
      <input type="submit" name="add" class="btn btn-sm btn-primary" value="Guardar datos">
      <a href="admin_ventana.php" class="btn btn-sm btn-danger">Cancelar</a>
    </div>
  </form>
</div>
</div>
```

Este documento es un formulario con todos los campos necesarios para que con el método POST puedan mandarse los datos al documento agregar.php, descrito en este documento.

Para este formulario hay una pequeña validación con un script de JavaScript.

```
<script>
  $(document).ready(function () {
    $("#contra").hide();
    $("#contra").removeAttr("required");
    $(".form-select").change(function () {
      if ($(this).val() == "1") {
        $("#contra").show();
        $("#contra").prop("required", true);
      } else {
        $("#contra").hide();
        $("#contra").removeAttr("required");
      }
    });
  });
</script>
```

Esta validación consta que si en select tiene el valor de 1, el campo de la contraseña se muestra, ya que es un administrador posee contraseña y si no, se oculta, Esto sucede cada vez que cambia el valor del select.

modificar.php

```
modificar.php
1  <?php
2  include("conexion.php");
3
4  if(!empty($_POST)){
5      $codigo = mysqli_real_escape_string($con,$_POST["codigo"]);
6      $nombres = mysqli_real_escape_string($con,$_POST["nombres"]);
7      $dui = mysqli_real_escape_string($con,$_POST["dui"]);
8      $direccion = mysqli_real_escape_string($con,$_POST["direccion"]);
9      $telefono = mysqli_real_escape_string($con,$_POST["telefono"]);
10     $puesto = mysqli_real_escape_string($con,$_POST["puesto"]);
11     $tipo = mysqli_real_escape_string($con,$_POST["tipo"]);
12     $encriptar = mysqli_real_escape_string($con,$_POST["pass"]);
13     $pass = sha1($encriptar);
14     //mysqli_close($con);
15     if($codigo != $_SESSION['id_empleado']){
16         $consulta = "SELECT * FROM empleados WHERE codigo = '$codigo'";
17         $resultado = $con -> query($consulta);
18         $filas = $resultado -> num_rows;
19
20         if($filas > 0){
21             $fila = $resultado -> fetch_assoc();
22             if($fila["estado"] != 0){
23                 $consulta = "UPDATE empleados SET nombres='$nombres', dui='$dui', direccion='$direccion', telefono='$telefono',
24                 puesto='$puesto',tipo='$tipo',contrasenia='$pass' WHERE codigo='$codigo'";
25                 $con -> query($consulta);
26                 echo "<script type='text/javascript'>alert('Modificado con exito!');</script>";
27             }else{
28                 echo "<script type='text/javascript'>alert('Código no encontrado.');
```

El documento modificar.php es muy parecido a agregar.php las variables y la logica es muy parecida, solo que el fin de este es modificar un empleado ya existente en la base de datos con la información que viene en la superglobal \$_POST. Si el código que viene es diferente al usuario que ha iniciado la sesión entonces se ejecuta la lógica de agregar, si no, se devuelve un mensaje de error. Para agregar el usuario se hace una consulta donde intenta traer los empleados que tienen el código que desea registrarse, si no se encuentra el código se hace una consulta para hacer un UPDATE en la base de datos con la información modificada del empleado.

- **codigo, nombres, dui, direccion, telefono, puesto, tipo:** Todas estas variables guardan respectivamente los datos de un empleado que su nombre describe. **encriptar:** Guarda la contraseña sin caracteres especiales que devuelve mysqli_real_escape_string.
- **pass:** Guarda la contraseña encriptada por sha1
- **consulta:** guarda la sentencia completa de la consulta que se realizará.

- **resultados:** Es un array que almacena los resultados de la consulta realizada a la base de datos
- **filas:** En primer instancia almacena el número de filas que retorna resultados a través del método `nums_row`. Luego más adelante guarda los resultados de la consulta.
- **mensaje:** Es un string con feedback para el usuario.

form_modificar.php

```
<?php
include("head.php");
include("leftmenu.php");
include("modificar.php");

if( isset($_GET["nik"])){
    $id = $_GET["nik"];
}

$query = "SELECT * FROM empleados WHERE codigo =$id";

$resultados = $con -> query($query);
$res = $resultados->fetch_assoc();

function imprimirTipo($tipo){
    if($tipo==1){
        echo '<option value="0" id="emp" >Empleado</option>
        <option value="1" selected="selected" id="admin" >Administrador</option>';
    }
    else{
        echo '<option value="0" selected="selected" id="emp" >Empleado</option>
        <option value="1" id="admin">Administrador</option>';
    }
}
```

Este es un formulario que se encarga de recoger la información para modificar.php. Pero contiene cierta lógica, esta realiza una consulta para traer los datos del empleado que se va a modificar si, para que en el formulario el usuario verifique la información que ya se tenía.

- **id:** guarda el código del empleado a modificar que trae la superglobal `GET`, este sirve para hacer la consulta a la base de datos y traer los datos que ya se tienen registrados de este empleado.
- **query:** Es una cadena de caracteres que tiene la consulta completa que se hará a la base de datos

-
- **resultados:** Es un array que almacena los resultados que devuelve la consulta realizada a la base de datos
 - **con:** Es un objeto que nos permite realizar la consulta y que su explicación está en conexion.php, por eso en este documento
 - **res:** Almacena el array resultados de forma asociativa.
 - **imprimirTipo():** Esta función imprime un select en el formulario con el tipo de usuario que es el empleado en el sistema.

En la parte del formulario se imprimen los datos del empleado que se almacenan en la variable res.

conexion.php

```
<?php
/*Datos de conexion a la base de datos*/
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = ""; // Nombre de tu base
//Ejecutar la conexion
$con = new mysqli($db_host, $db_user, $db_pass, $db_name);

?>
```

este documento .php sirve para conectar el sistema a la base de datos

- **db_host:** variable que almacena el host
 - **db_user:** variable que almacena tu usuario de tu localhost
 - **db_pass:** variable que almacena la contraseña en dado caso que el localhost no tenga contraseña solo se deja en blanco
 - **db_name:** variable que almacena el nombre de la base de datos
 - **con:** variable donde se hace la conexión a nuestra base de datos, con el método de mysqli solo se le pasa de parámetros las anteriores variables.
-

Head.php

```
<!--CABEZA DE LA INTERFAZ DEL ADMINISTRADOR CON SU NAVBAR-->
<?php
include("conexion.php");
session_start();
if(!isset($_SESSION['id_empleado'])) {
    header("Location: index.php");
}
$iduser = $_SESSION['id_empleado'];
$consulta = "SELECT * FROM empleados WHERE codigo = '$iduser'";
$resultado = $con -> query($consulta);
$fila = $resultado -> fetch_assoc();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR50om" >
    <script src="https://kit.fontawesome.com/c8a0597b0e.js" crossorigin="anonymous"></script>
    <title>Admin</title>
</head>
<body style="background-color: #e4efe7;">

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark d-flex justify-content-between">
        <div class="d-flex bd-highlight mx-2">
            <a class="btn btn-light d-inline p-2" data-bs-toggle="collapse" href="#multiCollapseExample1" role="button"
                aria-expanded="false" aria-controls="multiCollapseExample1">Menu</a>
        </div>

        <div class="dropdown">
            <button class="btn btn-light dropdown-toggle mx-2" type="button" id="dropdownMenuButton1" data-bs-toggle="dropdown"
                aria-expanded="false">
                <i class="fas fa-user"></i> Bienvenid@ <?php echo utf8_decode($fila['nombres']);?> </button>
            <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">
                <li><a class="dropdown-item" href="salir.php"><i class="fas fa-sign-out-alt"></i> Cerrar sesion</a></li>
            </ul>
        </div>
    </nav>

    <div class="d-flex align-items-center">
```

En este documento .php esta el header del html, para solo mandarlo a llamar en los demas documentos

- **iduser:** en esta variable se guarda el id del usuario que ingreso en el login
- **consulta:** en esta variable se hace la consulta de traer todos los datos del empleado que ingreso, aquí le pasamos el id de ese usuario que se almaceno en la variable anterior
- **resultado:** aquí preparamos la consulta que hicimos anteriormente
- **fila:** aquí obtiene una fila de resultados como una matriz asociativa.

Clock.js

```
var updateTime = function() {
    let currentDate = new Date(),
        hours = currentDate.getHours(),
        minutes = currentDate.getMinutes(),
        seconds = currentDate.getSeconds(),
        weekDay = currentDate.getDay(),
        day = currentDate.getDate(),
        month = currentDate.getMonth(),
        year = currentDate.getFullYear();

    const weekDays = [
        'Domingo',
        'Lunes',
        'Martes',
        'Miércoles',
        'Jueves',
        'Viernes',
        'Sabado'
    ];

    document.getElementById('weekDay').textContent = weekDays[weekDay];
    document.getElementById('day').textContent = day;
    const months = [
        'Enero',
        'Febrero',
        'Marzo',
        'Abril',
        'Mayo',
        'Junio',
        'Julio',
        'Agosto',
        'Septiembre',
        'Octubre',
        'Noviembre',
        'Diciembre'
    ];

    document.getElementById('month').textContent = months[month];
    document.getElementById('year').textContent = year;
    document.getElementById('hours').textContent = hours;
    if (minutes < 10) {
        minutes = "0" + minutes
    }
}
```

En este .php existe una funcion llamada updateTime el cual consiste en arrays donde se consigue la fecha y hora actual

- **currentDate:** aquí se obtienen la fecha actual y la hora
- **weekDays:** aquí se guardan los días de la semana
- **months:** aquí se guardan los meses

config_horario.js

```
$( "#horarios" ).submit(function( event ) {
    event.preventDefault();
    var stringDatos = $("#horarios").serialize();
    $.ajax({
        type: 'POST',
        url: 'config_horario.php',
        data: stringDatos,
        dataType: 'json',
        success: function(datos) {
            alert(datos.mensaje);
        }
    });
});
```

en este .js asignamos los nuevos datos del horario si en dado caso lo modifica en usuario.

- **StringDatos:** En este caso, jQuery serializa los controles exitosos dentro del formulario. Solo *horarios* se examinan los elementos para las entradas que contienen; en todos los demás casos, los elementos de entrada que se serializarán deben ser parte del conjunto pasado al .serialize() método. Seleccionar tanto el formulario como sus hijos en un conjunto provocará duplicados en la cadena serializada.
- **.ajax:** con esto actualizamos los datos de los horarios

config_horario.php

```
function inicio(){
    include_once "conexion.php";
    include("head.php");
    include("leftmenu.php");
    $sql0="SELECT * FROM horarios";
    $res0 = mysqli_query($con, $sql0);
    if(mysqli_num_rows($res0) >0){
        $row0 = mysqli_fetch_assoc($res0);
    }
}

<?php
<div class="tab-content flex-grow-1 ms-3" id="v-tabs-tabContent">
<div class="container-fluid my-5">
<h3>Configurar Horarios laborales</h3>
<form id="horarios">
<input type="hidden" id="process" name="process" value="guardar" />
<table>
<tr>
<td>Día </td>
<td>Hora entrada </td>
<td>Minutos entrada </td>
<td>Hora Salida </td>
<td>Minutos Salida </td>
</tr>
<tr>
<td colspan="5">
<?php
list($h_ent,$m_ent,$s_ent)=explode(':', $row0['hora_e_sen']);
list($h_sal,$m_sal,$s_sal)=explode(':', $row0['hora_s_sen']);
list($h_ents,$m_ents,$s_ents)=explode(':', $row0['hora_e_fd']);
list($h_sals,$m_sals,$s_sals)=explode(':', $row0['hora_s_fd']);
echo "<tr>
<td class='dia'>Lunes a Viernes</td>
<td><input type='text' id='hora_ent1' name='hora_ent1' class='datas form-control input-md integer_positive' min='0' max='23' value='",$h_ent,">
<td><input type='text' id='min_ent1' name='min_ent1' class='datas form-control input-md integer_positive' min='0' max='60' value='",$m_ent,">
<td><input type='text' id='hora_sal1' name='hora_sal1' class='datas form-control input-md integer_positive' min='0' max='23' value='",$h_sal,">
<td><input type='text' id='min_sal1' name='min_sal1' class='datas form-control input-md integer_positive' min='0' max='60' value='",$m_sal,">
</tr>
<tr>
<td class='dia'>Sabado</td>
<td><input type='text' id='hora_ent2' name='hora_ent2' class='datas form-control input-md integer_positive' min='0' max='23' value='",$h_ent,">
<td><input type='text' id='min_ent2' name='min_ent2' class='datas form-control input-md integer_positive' min='0' max='60' value='",$m_ent,">
</tr>
</table>
</div>
</div>
```

En este .php hay dos funciones, primero esta inicio() en esta funcion se muestra la tabla de los horarios donde se puede modificar

- **sql0:** aquí se hace la consulta donde traemos los datos de horario
- **res0:** aquí preparamos la consulta que hicimos anteriormente
- **row0:** aquí obtiene una fila de res0 como una matriz asociativa.
- **h_ent:** aquí guardamos la hora de entrada de lunes a viernes
- **m_ent:** aquí guardamos los minutos de entrada de lunes a viernes
- **s_ent:** aquí guardamos los segundos de entrada de lunes a viernes
- **h_sal:** aquí guardamos la hora de salida de lunes a viernes
- **m_sal:** aquí guardamos los minutos de salida de lunes a viernes
- **s_sal:** aquí guardamos los segundos de salida de lunes a viernes
- **h_ents:** aquí guardamos la hora de entrada del fin de semana
- **m_ents:** aquí guardamos los minutos de entrada del fin de semana
- **s_ents:** aquí guardamos los segundos de entrada del fin de semana
- **h_sals:** aquí guardamos la hora de salida del fin de semana
- **m_sals:** aquí guardamos los minutos de salida del fin de semana
- **s_sals:** aquí guardamos los segundos de salida del fin de semana

```
function guardar(){
    include "conexion.php";
    //variable para variar si hay espacios en blanco
    $validar=true;

    $hora_ent1=$_POST['hora_ent1'];
    if ($hora_ent1=='') {
        $validar=false;
    }
    $min_ent1=$_POST['min_ent1'];
    if ($min_ent1=='') {
        $validar=false;
    }
    $hora_sal1=$_POST['hora_sal1'];
    if ($hora_sal1=='') {
        $validar=false;
    }
    $min_sal1=$_POST['min_sal1'];
    if ($min_sal1=='') {
        $validar=false;
    }
    $hora_ent2=$_POST['hora_ent2'];
    if ($hora_ent2=='') {
        $validar=false;
    }
    $min_ent2=$_POST['min_ent2'];
    if ($min_ent2=='') {
        $validar=false;
    }
    $hora_sal2=$_POST['hora_sal2'];
    if ($hora_sal2=='') {
        $validar=false;
    }
    $min_sal2=$_POST['min_sal2'];
    if ($min_sal2=='') {
        $validar=false;
    }
    $sesem=$hora_ent1."-".$min_ent1."-00";
```

la ultima funcion es guardar, aquí se valida si en dado caso el usuario deja espacios en blanco si no deja espacios en blanco se modifican los datos

eliminar.php

```
function despedir(){
    include("conexion.php");

    $nik = mysqli_real_escape_string($con,(strip_tags($_GET["nik"],ENT_QUOTES)));
    $cek = mysqli_query($con, "SELECT * FROM empleados WHERE codigo='$nik'");
    if(mysqli_num_rows($cek) == 0){
        header('Location: reporteEmp.php?res=2');
        return <<< MSJ
        <div class="alert alert-warning alert-dismissible fade show mt-3" role="alert" id="alert">
            No se encontraron datos.
            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
        </div>
        _MSJ;
    }else{
        //este es para la opcion que aparece eliminar, pero como no haremos eso aqui, lo podes quitar
        $delete = mysqli_query($con, "UPDATE `empleados` SET `estado` = '0' WHERE `codigo` = '$nik'");
        if($delete){
            header('Location: reporteEmp.php?res=1');
        }else{
            header('Location: reporteEmp.php?res=3');
        }
    }
}

if(isset($_GET['aksi']) == 'delete'){
    despedir();
}
```

En eliminar.php es donde se hace la consulta de eliminar el usuario, aquí solo hay una funcion llamada despedir

- **nik:** aquí se obtiene el codigo del usuario a eliminar
- **cek:** aquí se hace la consulta delete
- **delete:** si en dado caso el nik es diferente a 0 , con esta variable se actualiza en la base de datos el estado a 0 que significa que esta despedido

funciones.php

```
function fecha_dmy($fecha){
    $dia = substr($fecha,8,2);
    $mes = substr($fecha,5,2);
    $a = substr($fecha,0,4);
    $fecha = "$dia-$mes-$a";
    return $fecha;
}

function fecha_ymd($fecha){
    $dia = substr($fecha,0,2);
    $mes = substr($fecha,3,2);
    $a = substr($fecha,6,4);
    $fecha = "$a-$mes-$dia";
    return $fecha;
}
```

En este .php hay algunas funciones primero explicaremos las funciones fecha_dmy y fecha_ymd estas dos funciones convierte la fecha obtenida a los dos formatos

- **dia:** aquí guarda el dia en una parte del string definida con los parametros definidos
- **mes:** aquí guarda el mes en una parte del string definida con los parametros definidos
- **a:** aquí guarda el año en una parte del string definida con los parametros definidos
- **fecha:** aquí ya se le da el formato que queremos para las dos funciones y lo retornamos

```
//funcion para observaciones segun hora de entrada y salida
function obtener_observa($he1,$he2,$tipo){
    //funcion que obtiene los objetos Datetime y retorna la observacion
    //si hubo llegads tardes o salidas temprano
    $tiempoe_diff = $he2->diff($he1);
    $invert_e    = $tiempoe_diff->invert;

    if (isset($invert_e)){
        //aquí le damos formato para imprimirlo
        $dias_tardel=$tiempoe_diff->format('%d');
        $horas_tardel=$tiempoe_diff->format('%h');
        $minutos_tardel=$tiempoe_diff->format('%i');

        if ($invert_e==1 && $tipo=="E") {
            //si es 1 y es de tipo E llego tarde
            return "<strong> Entró tarde: ".$horas_tardel." Horas, ". $minutos_tardel." Min.</strong> ";
        }
        if ($invert_e==0 && $tipo=="E") {
            //si es 0 y es de tipo E llego temprano
            return " Entró temprano ".$horas_tardel." Horas, ". $minutos_tardel." Min. ";
        }
        if ($invert_e==1 && $tipo=="S") {
            //si es 1 y es tipo S es porque salio tarde
            return "<strong> Salió tarde: ".$horas_tardel." Horas, ". $minutos_tardel." Min.</strong> ";
        }
        if ($invert_e==0 && $tipo=="S") {
            //si es 0 y es tipo S es porque salio temprano
            return "<strong> Salió temprano: ".$horas_tardel." Horas, ". $minutos_tardel." Min.</strong> ";
        }
    }
}
```

la ultima funcion que esta en este .php es obtener_observa donde se obtiene la observacion por rango de fechas

- **tiempoe_diff:** aquí se compara los dos datos que se les pasaron que es la hora en que ingreso ya sea entrada y salida y la hora que esta guardada en la base de datos
 - **invert_e:** aquí se convierte el tiempoe_diff a un string
 - **dias_tarde1:** aquí se guarda el día que llego tarde
 - **horas_tarde1:** aquí se guarda cuantas horas tarde llego el usuario
 - **minutos_tarde1:** aquí se guarda cuantos minutos tarde llego el usuario
-

reporteRango.js

```
$(document).ready(function() {  
    $('.date').datepicker({  
        format: 'dd-mm-yyyy',  
        language: 'es',  
    })  
});  
  
//seleccionar tipo orden en modal, nos ayuda a buscar y los mandamos a la siguiente  
//pagina y los metemos donde esta el id mostrar datoss que es en la tabla  
$(document).on('click', '#buscar', function(e) {  
    var fechai=$('#fechai').val();  
    var fechaf=$('#fechaf').val();  
  
    //utilizamos la libreria moment.js para manipular fechas en javascript  
    var f1=moment(fechai, "DD-MM-YYYY");  
    var f2=moment(fechaf, "DD-MM-YYYY");  
  
    //si la fecha inicio es menor o igual que la fecha fin  
    if(f1.isBefore(f2)||f1.isSame(f2)){  
        $.ajax({  
            type: 'POST',  
            url:'verAsistencia.php',  
            data: {fechai:fechai,fechaf:fechaf},  
            dataType:'html',  
            success: function(datax) {  
                $("#mostrar_datoss").html(datax);  
            }  
        });  
    } // true  
    else{  
        alert("fecha inicio debe ser menor o igual que fecha fin")  
        $("#mostrar_datoss").html('<tr><td colspan="8">No hay datos.</td></tr>');  
    }  
});
```

En este .js nos ayuda para buscar los datos para despues mostrarlos en una tabla

- **fechai:** aquí guardamos la fecha de inicio que ingrese el usuario
 - **fechaf:** aquí guardamos la fecha de final que ingrese el usuario
 - **f1:** ahora esa fechai la convertimos en formato dd-mm-yyyy
 - **f2:** ahora esa fechaf la convertimos en formato dd-mm-yyyy
 - **.ajax:** luego solo lo mandamos esos datos por ajax a mostrar_datoss que es una tabla
-

ver_llegadas.js

```
$(document).ready(function() {
    $('#date').datepicker({
        format: 'dd-mm-yyyy',
        language: 'es',
    })
});

//seleccionar tipo orden en modal, nos ayuda a buscar y los mandamos a la siguiente
//pagina y los metemos donde esta el id mostrar_datoss que es en la tabla
$(document).on('click', '#buscar', function(e) {
    var codigo=$('#codigo').val();
    var fechai=$('#fechai').val();
    var fechaf=$('#fechaf').val();

    //utilizamos la libreria moment.js para manipular fechas en javascript
    var f1=moment(fechai, "DD-MM-YYYY");
    var f2=moment(fechaf, "DD-MM-YYYY");

    //si la fecha inicio es menor o igual que la fecha fin
    if(f1.isBefore(f2)||f1.isSame(f2)){
        $.ajax({
            type: 'POST',
            url:'mostrar_llegadas.php',
            data: {codigo:codigo,fechai:fechai,fechaf:fechaf},
            dataType:'html',
            success: function(datax) {
                $("#mostrar_datoss").html(datax);
            }
        });
    } // true
    else{
        alert("fecha inicio debe ser menor o igual que fecha fin")
    }
});
});
```

En este .js nos ayuda para buscar los datos para despues mostrarlos en una tabla

- **fechai:** aquí guardamos la fecha de inicio que ingrese el usuario
- **fechaf:** aquí guardamos la fecha de final que ingrese el usuario
- **codigo:** aquí guardamos el codigo del usuario
- **f1:** ahora esa fechai la convertimos en formato dd-mm-yyyy
- **f2:** ahora esa fechaf la convertimos en formato dd-mm-yyyy
- **.ajax:** luego solo lo mandamos esos datos por ajax a mostrar_datoss que es una tabla

marcar.js

```
$(document).ready(function () {
    $( "#codigo" ).focus();
    $("input[name=entrada]").change(function () {
        $( "#salida" ).prop( "checked", false );
    });
    $("input[name=salida]").change(function () {
        $( "#entrada" ).prop( "checked", false );
    });
});

$('#codigo').keypress(function(event){
    var keycode = (event.keyCode ? event.keyCode : event.which);
    if(keycode == 13){
        event.preventDefault();
        const tiempo=1000;
        setTimeout(function() {
            consultar();},tiempo);
    }
});

function consultar(){
    var marca='E';
    var codigo=$('#codigo').val();
    if( $("#entrada").is(':checked')) {
        marca=$("#entrada").val()
    }
    if( $("#salida").is(':checked')) {
        marca=$("#salida").val()
    }
    if (codigo!="") {
        $.ajax({
            type: 'POST',
            url: 'ins_marca_dia.php',
            data: {codigo:codigo,marca:marca},
            dataType: 'json',
            success: function(data) {
                $( "#codigo" ).focus();
                $( "#codigo" ).val("");
                $( ".mensajes" ).text(data.mensaje);
            }
        });
    }
}
```

Este .js nos ayuda a la hora de marcar asistencia, la primera funcion es para que, dependiendo de la hora marcara automaticamente entrada o salida la segunda funcion nos trae el codigo y hacemos la consulta con ella

la funcion consulta nos ayuda para ver si esta chekeada la entrada o la salida para luego mandarlo a ins_marca_dia.php y limpiamos los campos por ultimo

verAsistencia.php

```
function diasTrab($codigo,$fechai,$fechaf){
    include("conexion.php");
    //include("funciones.php");
    $diasTrab=0;
    $sql = mysqli_query($con, "SELECT * FROM marcas
    WHERE codigo='$codigo'
    AND fecha BETWEEN '$fechai' AND '$fechaf'");
    while($row = mysqli_fetch_assoc($sql)){
        $codigo=$row['codigo'];
        $diasTrab++;
    }
    return $diasTrab;
}
```

La funcion diastrab nos ayuda a traernos los datos de los dias desde la fecha inicial hasta la fecha final que ingreso el usuario

- **diasTrab:** aquí guardara los dias trabajados
- **sql:** aqui hacemos la consulta

```
function cuenta_salidas_entradas($codigo,$fechai,$fechaf){
    include("conexion.php");
    $sql0="SELECT id, hora_e_sem, hora_s_sem, hora_e_fd, hora_s_fd FROM horario";
    $res0 = mysqli_query($con, $sql0);
    $row0 = mysqli_fetch_assoc($res0);

    $sql1="SELECT m.codigo,m.fecha,m.hora_e,m.hora_s FROM marcas AS m
    WHERE m.codigo='$codigo'
    AND m.fecha BETWEEN '$fechai' AND '$fechaf'";
    $res= mysqli_query($con, $sql1);

    //funcion que obtiene los objetos Datetime y retorna la observacion
    //si hubo llegads tardes o salidas temprano
    $llegatardes=0;
    $saltemprano=0;
    while($row = mysqli_fetch_assoc($res)){
        if(date("w",strtotime($row['fecha']))>0 && date("w",strtotime($row['fecha']))<6){
            //Entradas tarde al trabajo
            $hes1 = new DateTime($row['fecha']." ".$row['hora_e_sem']); //aqui agarra la config del horario y la fecha
            $hes2 = new DateTime($row['fecha']." ".$row['hora_e']); //aqui agarra la hora de entrada y la fecha

            $tiempoe_dif = $hes2->diff($hes1);
            $invert_e = $tiempoe_dif->invert;
            if (isset($invert_e)){
                if ($invert_e==1 ) {
                    $llegatardes++;
                }
            }
            //salidas temprano del trabajo
            $hss1 = new DateTime($row['fecha']." ".$row['hora_s_sem']); //aqui agarra la config del horario y la fecha
            $hss2 = new DateTime($row['fecha']." ".$row['hora_s']); //aqui agarra la hora de entrada y la fecha
            $tiempos_dif = $hss2->diff($hss1);
            $invert_s = $tiempos_dif->invert;
            if (isset($invert_s)){
                if ($invert_s==0 ) {
                    $saltemprano++;
                }
            }
        }
    }
}
```

En la funcion cuenta_salidas_entradas se buscan cuantas entradas tardes y cuantas salidas tempranas hizo el usuario

- **sql0:** aquí se hace la consulta de horario
- **res0:** aquí el resultado de esa consulta
- **row0:** aquí la fila del resultado
- **sql1:** aquí hacemos otra consulta que me traiga el código de marca, la fecha, la hora de entrada, la hora de salida de la tabla marcas donde el código sea igual al código del empleado seleccionado
- **res:** aquí el resultado de la consulta
- **llegadatardes:** aquí se almacenarán las llegadas tardes
- **saltemprano:** aquí se almacenarán las salidas temprano
- **hes1:** aquí se guarda la configuración del horario y la fecha de entrada
- **hes2:** aquí se guarda la hora de entrada y la fecha de entrada del empleado
- **tiempoe_dif:** aquí se guarda la diferencia entre hes1 y hes2
- **invert_e:** aquí se guarda tiempoe_dif ya invertido
- **hss1:** aquí se guarda la configuración del horario y la fecha de salida
- **hss2:** aquí se guarda la hora de salida y la fecha de salida del empleado
- **tiempos_dif:** aquí se guarda la diferencia entre hss1 y hss2
- **invert_s:** aquí se guarda tiempos_dif ya invertido
- **hefd1:** aquí se guarda la configuración del horario y la fecha de entrada del fin de semana
- **hefd2:** aquí se guarda la hora de entrada y la fecha de entrada del empleado del fin de semana
- **tiempoe_dif2:** aquí se guarda la diferencia entre hefd1 y hefd2
- **invert2:** aquí se guarda tiempoe_dif2 ya invertido
- **hsfd1:** aquí se guarda la configuración del horario y la fecha de entrada del fin de semana
- **hsfd2:** aquí se guarda la hora de salida y la fecha de entrada del empleado del fin de semana
- **tiempof_dif:** aquí se guarda la diferencia entre hsfd1 y hsfd2
- **invert_sf:** aquí se guarda tiempof_dif ya invertido

ver_llegadas.php

```
<?php
include("head.php");
include("leftmenu.php");
include("conexion.php");
include("funciones.php");
$codigo= $_GET['codigo'];
$fecha1=date('d-m-Y');
$fecha2=date('d-m-Y');
$sql = mysqli_query($con, "SELECT * FROM empleados WHERE codigo='$codigo'");
$nombre='';
if(mysqli_num_rows($sql) >0){
    $row = mysqli_fetch_assoc($sql);//aquí hacemos el pedido
    $nombre=$row['nombres'];
}
?>
```

En este .php solo se mostraron los datos de las llegadas por empleado

- **codigo:** aquí se trae el codigo del empleado
- **fechai:** aquí se guarda la fecha de inicia en que se buscara el reporte en formato d-m-y
- **fechaf:** aquí se guarda la fecha final en que se buscara el reporte, en formato d-m-y
- **sql:** aquí se hace la consulta donde se traera todos los datos del empleado
- **nombre:** aquí se guardara el nombre del empleado
- **row:** aquí nos traemos ya las filas

reporteDiario.php

```

if(mysqli_num_rows($sql) == 0){
    echo '<tr><td colspan="8">No hay datos.</td></tr>';
}else{
    $no = 1;
    while($row = mysqli_fetch_assoc($sql)){
        // lo siguiente es para el menu de acciones, de aqui se muestra en una tabla
        // la hora de entrada, salida, si está en la empresa o no, etc.
        $codigo = $row['codigo'];
        $query = mysqli_query($con, "SELECT * FROM empleados WHERE codigo='$codigo'");
        while($nom = mysqli_fetch_assoc($query)){
            if($nom['estado'] == 1){
                $activo = $nom['activo'] == 0 ? 'Sí' : 'No';
                echo '<tr>
                <td>.$no.</td>
                <td>.$nom['nombres'].</td>
                <td>.$row['hora_e'].</td>
                <td>.$activo.</td>
                <td>.$row['hora_s'].</td>
                <td>';
                //obtener observaciones del dia
                if(date("w")>0 && date("w")<6){//esto es para el dia de la semana
                    if($row['hora_e']){
                        $hes1 = new DateTime($row['fecha']." ".$row['hora_e_sem']);//aqui agarra la config del horario y la fecha
                        $hes2 = new DateTime($row['fecha']." ".$row['hora_e']);//aqui agarra la hora de entrada y la fecha
                        $obs_es=obtener_observa($hes1,$hes2,"E");//las observaciones (si llego tarde y asi)
                        echo $obs_es;//y se imprimen
                    }
                    else{
                        echo "<span class='text-danger'>No marco Entrada</span>";
                    }
                }

                if($row['hora_s']){
                    $hss1 = new DateTime($row['fecha']." ".$row['hora_s_sem']);//config horario
                    $hss2 = new DateTime($row['fecha']." ".$row['hora_s']);//lo mismo que arriba pero con la salida
                    $obs_ss=obtener_observa($hss1,$hss2,"S");//aqui son observacion de la salida tipo, se fue antes o no marco
                    echo $obs_ss;
                }
            }
        }
    }
}

```

En este .php se muestra el reporte diario de todos los empleados

- **no:** el numero de empleado
- **codigo:** aquí se guarda el codigo del empleado
- **nom:** aquí se guarda el nombre del empleado
- **row:** aquí se guarda la hora de entrada y salida del empleado
- **activo:** aquí se guarda si esta activo el empleado o no
- **hes1:** aquí se guarda la configuracion del horario y la fecha de entrada
- **hes2:** aquí se guarda la hora de entrada y la fecha de entrada del empleado
- **obs_es:** aquí se obtiene las observaciones de entrada

- **hss1:** aquí se guarda la configuracion del horario y la fecha de salida
- **hss2:** aquí se guarda la hora de salida y la fecha de salida del empleado
- **obs_ss:** aquí se obtiene las observaciones de salida
- **hefd1:** aquí se guarda la configuracion del horario y la fecha de entrada del fin de semana
- **hefd2:** aquí se guarda la hora de entrada y la fecha de entrada del empleado del fin de semana
- **obs_efd:** aquí se obtiene las observaciones de entrada del fin de semana
- **hsfd1:** aquí se guarda la configuracion del horario y la fecha de entrada del fin de semana
- **hsfd2:** aquí se guarda la hora de salida y la fecha de entrada del empleado del fin de semana
- **obs_sfd:** aquí se obtiene las observaciones de salida del fin de semana

login.php

```

?php
session_start();
if(isset($_SESSION['id_empleado'])){
    header("Location: index.php");
}

// Buscar usuario en base de datos
if(!empty($_POST)) {
    $usuario = mysqli_real_escape_string($con, $_POST['name']);
    $encriptar = mysqli_real_escape_string($con, $_POST['password']);
    $contrasenia = sha1($encriptar);
    $consulta = "SELECT * FROM empleados WHERE codigo = '$usuario' AND contrasenia = '$contrasenia' AND tipo = '1' AND estado = '1'";
    $resultado = $con -> query($consulta);
    $filas = $resultado -> num_rows;
    if($filas > 0){
        $fila = $resultado -> fetch_assoc();
        $_SESSION['id_empleado'] = $fila["codigo"];
        // echo $_SESSION["idEmpleado"];
        header("Location: admin_ventana.php");
    }else{
        $mensaje = 'usuario o contrasenia incorrecto';
        echo "<script type='text/javascript'>alert('$mensaje');</script>";
        // header("Location: index.php");
    }
}
?

```

En este .php se hace el login

- **usuario:** aquí se guarda el nombre del usuario
- **encriptar:** aquí se guarda el password para luego encriptarla
- **contrasenia:** aquí se encripta la contraseña
- **consulta:** aquí se hace la consulta si es la misma contraseña y usuario
- **resultado:** aquí se guarda el resultado de la consulta
- **filas:** aquí se guarda las filas del resultado
- **mensaje:** si es incorrecto el usuario o la contraseña aquí se guarda un mensaje

ins_marca_dia.php

```
//consultar si existe empleado
if(mysqli_num_rows($sql1)> 0){

    $row = mysqli_fetch_assoc($sql1);
    $datos['codigo']=$row['codigo'];
    $datos['fecha']=$row['fecha'];
    $datos['hora_e']=$row['hora_e'];
    $datos['hora_s']=$row['hora_s'];
    if($marca=='E'){
        $update = mysqli_query($con, "UPDATE marcas SET hora_e='$hora' WHERE codigo='$codigo' AND fecha='$fecha'") or die(mysqli_error());
        $update1 = mysqli_query($con, "UPDATE empleados SET disponible='1' WHERE codigo='$codigo'") or die(mysqli_error());
        $tipo_marca=" MARCA ENTRADA A LAS ";
    }
    if($marca=='S'){
        $update = mysqli_query($con, "UPDATE marcas SET hora_s='$hora' WHERE codigo='$codigo' AND fecha='$fecha'") or die(mysqli_error());
        $update1 = mysqli_query($con, "UPDATE empleados SET disponible='0' WHERE codigo='$codigo'") or die(mysqli_error());
        $tipo_marca=" MARCA SALIDA A LAS ";
    }
    if($update){
        $datos['operacion']='UPD';
        $datos['mensaje']=$nombre." : ".$tipo_marca." ".$hora;
    }
}
}else {
```

En este .php se hace las marcas de asistencia

- **row:** aquí se guarda los datos en un array
- **datos:** aquí se guardan los datos
- **marca:** aquí se guarda lo que marco el empleado puede ser E de entrada o S se salida
- **update:** aquí se guarda una consulta de actualizar en la tabla marcas
- **update1:** aquí se hace la consulta de actualizar “disponible” a 1 si en dado caso es Entrada
- **tipo_marca:** aquí se guarda un texto de marco entrada

```
//si no hay registro de marcs en la fecha actual, inserta
if($marca=='E'){
    $q_ins="INSERT INTO marcas(id,codigo, fecha, hora_e) ";
    $tipo_marca=" MARCA ENTRADA A LAS ";
}
else {
    $q_ins="INSERT INTO marcas(id,codigo, fecha, hora_s) ";
    $tipo_marca=" MARCA SALIDA A LAS ";
}
$q_ins.=" VALUES(DEFAULT,'$codigo','$fecha', '$hora')";
$insert = mysqli_query($con, $q_ins) or die(mysqli_error());

if($insert){
    $datos['operacion']='INS';
    $datos['mensaje']=$nombre." : ".$tipo_marca." ".$hora;
}else {
    $datos['operacion']='NO_EJECUTADA';
    $datos['mensaje']=$nombre." NO REGISTRO MARCA";
}
}
}else{
    $datos['operacion']='NO EXISTE';
    $datos['mensaje']=' CODIGO DE EMPLEADO: '.$codigo.', NO EXISTE ';
}
echo json_encode($datos);
?>
```

-
- **q_ins:** aquí se hace la consulta de insertar si en dado caso es primera ves a marcar
 - **insert:** aquí se hace la consulta anterior
-

Diccionario de datos

Tabla: empleados.		Fecha:	
Descripción.	En esta tabla se almacenarán todos los datos del empleado que reemplazarán el uso de registro en papel por un registro digital y si es administrador se ingresará una contraseña.		
Campo	Tamaño.	Tipo de dato.	Descripción
codigo	5	varchar	El código será utilizado para tomar la asistencia del empleado o se podrá ingresar sesión en el caso de ser administrador.
nombres	100	varchar	Nombre completo del empleado.
dui	10	varchar	Numero de identidad.
direccion	125	varchar	Dirección completa del empleado.
telefono	8	varchar	Numero de contacto.
puesto	25	varchar	El puesto será solo texto por lo que no tendrá mayor utilidad en el sistema.
estado	1	int	Si el empleado trabaja en el sistema se almacenará 1 y si ya no trabaja en la empresa será 0. No se elimina su expediente, pero si es 0 no podrá usar el sistema.
tipo	1	int	Si es tipo 1 podrá iniciar sesión para usar opciones de administración y si es 0 solo podrá tomar asistencia.
contraseña	200	varchar	Solo si el usuario es de tipo 1 podrá ingresar su contraseña.
disponible	1	int	Si el empleado esta dentro de la empresa será 1 y si ya se retiró será 0 para saber cuántos están en la empresa.

Tabla: horario.		Fecha:	
Descripción.	En esta tabla se almacenará cada registro diario de los empleados.		
Campo	Tamaño.	Tipo de dato.	Descripción
id (PK)	11	int	Cada registro tendrá su identificador por lo que será único.
codigo (FK)		date	El código del empleado nos dice toda la información de quien fue la persona que marco un determinado registro.

fecha		date	Almacena la fecha que se marcó asistencia.
hora_e		date	Almacena la hora en la que el usuario marco la entrada en la fecha que se obtuvo de manera automática.
Hora_s		date	Almacena la hora en la que el usuario marco la salida diaria en la fecha que se obtuvo de manera automática.

Tabla: marcas.		Fecha:	
Descripción.	Esta tabla no tiene una relación directa con las otras mas bien se utiliza para marcar las horas de referencia de la empresa de esta manera podrían cambiar su hora de entrada y salida que serán utilizadas para saber si el empleado llevo a tiempo o tarde a la hora de marcar asistencia.		
Campo	Tamaño.	Tipo de dato.	Descripción
id (PK)	8	int	El identificador nos sirve para actualizar los datos de la tabla.
hora_e_sem	8	int	Hora de entrada los días de semana.
hora_s_sem		date	Hora de salida los días de semana.
hora_e_fd		time	Hora de entrada el fin de semana.
hora_s_fd		time	Hora de salida el fin de semana.

Prerrequisitos

- Para generar la interfaz: **Bootstrap 5**
- Para manipulacion de elementos en el DOM: **Jquery**
- Para generar tablas con paginado: **dataTable**

Bitacora de cambios

Responsable	Implementación Principal	Fecha de Subida	Fecha de modificación	Descripción del cambio
Alejandra Claire Aguilar Mata.	Interfaz de administrador	mayo 22, 2021	mayo 22, 2021	Borro la interfaz de administrador, para implementarla de manera modular
Alejandra Claire Aguilar Mata.	Menu de ventana administrador	Mayo 22, 2021		
Jorge Eduardo Romero Garcia.	Interfaz de asistencia lista, marcar interfaz y lógica lista	Mayo 22,2021		
Junior Mauricio Villalta Flores.	Interfaz de agregar, modificar y eliminar usuario	Mayo 23, 2021	Mayo 23, 2021	Cambios a interfaz de usuario
Jorge Eduardo Romero Garcia.	Interfaz de asistencia lista, marcar interfaz y lógica lista	Mayo 22, 2021	Mayo 23, 2021	Agregó archivo faltante para guardar las marcas en la base de datos
Cristian Alexander Guardado Escobar.	Integración login con módulo de asistencia	Mayo 24, 2021	Mayo 27. 2021	Implementación de seguridad del login
Jorge Eduardo Romero Garcia.	Interfaz de asistencia lista, marcar interfaz y lógica lista	Mayo 22, 2021	Mayo 28, 2021	funcionalidad para ver empleados activos hecha
Cristian Alexander Guardado Escobar.	Integración login con módulo de asistencia	Mayo 24, 2021	Mayo 31, 2021	Mejoras en mensajes de error en login, diseño de cerrar sesión y eliminación de empleados
Cristian Alexander Guardado	Integración login con módulo de asistencia	Mayo 24, 2021	1 Junio, 2021	Corrección de bug de eliminación y cambios en el include por require_once

Escobar				
Junior Mauricio Villalta Flores.	lógica de agregar, modificar y eliminar usuario	Mayo 23, 2021	Junio 1, 2021	Corrección a la interfaz de registro y funcionalidad
Jorge Luis Granados Umanzor.	Implementación de interfaz y lógica de reportes diarios y reporte por fecha	Junio 2, 2021	Junio 3, 2021	Interfaz de reportes y funcionalidad de lista de empleados
Alejandra Claire Aguilar Mata	ver llegadas de empleados.	Junio 2, 2021		
	configuración de horario.	Junio 2, 2021		
	Reporte por empleados	Junio 2, 2021		
Jorge Eduardo Romero Garcia.	Interfaces		4 Junio, 2021	Homogeneización de interfaces, y arreglo con la base de datos
Jorge Eduardo Romero Garcia.	Lógica de agregar, modificar y eliminar usuario	Mayo 23, 2021	Junio 5, 2021	Cambios en modificar empleados
Cristian Alexander Guardado Escobar	Integración login con módulo de asistencia	Mayo 24, 2021	Junio 6, 2021	Modificación en el login, agregar y modificar contraseña encriptada
Junior Mauricio Villalta Flores. Jorge Luis Granados Umanzor	Lógica de agregar, modificar y eliminar usuario		Junio 6, 2021	Corrección de warning en lista de empleados.
			Junio 6, 2021	Modificación de los alert de eliminar empleados
	lógica de reportes diarios y por fechas	Junio 2, 2021	Junio 6, 2021	Funcionalidad reporte diario
		Junio 2, 2021	Junio 6, 2021	Modificación a reportes diarios
Alejandra Claire Aguilar Mata	Reporte por empleado	Junio 2, 2021	junio 6, 2021	Arreglo de reporte por empleado y modificaciones

Junior Mauricio Villalta Flores.	Reporte por empleado	Junio 2, 2021	Junio 6, 2021	Cambio de nombres de archivos de registro de empleados
Alejandra Claire Aguilar Mata	Reportes en general	Junio 2, 2021	Junio 7, 2021	Modificaciones en los reportes
		Junio 2, 2021	Junio 7, 2021	Arreglo de warnings
	Reporte por fecha	Junio 2, 2021	Junio 7, 2021	Reporte por fechas listo.
	Reporte por empleados	Junio 2, 2021	Junio 7, 2021	Búsqueda por empleado y plugin datable
Jorge Eduardo Romero Garcia.	lógica de agregar, modificar y eliminar usuario	Mayo 23, 2021	Junio 7, 2021	Nueva forma de borrar y corrigiendo interfaz de modificar empleados.
Cristian Alexander Guardado Escobar.	Integración login con módulo de asistencia	Mayo 24, 2021	Junio 8, 2021	Correccion de bug y modificacion de inicio de administrador
Cristian Alexander Guardado Escobar.	Interfaz de asistencia lista, marcar interfaz y lógica lista	Mayo 22, 2021	Junio 11, 2021	Corrección de bug de estado activo diario
Cristian Alexander Guardado Escobar.			Junio 11, 2021	Corrección de nombre de variables en el programa en general, para evitar errores
Alejandra Claire Aguilar Mata			Junio 11, 2021	Cambio a la base de datos
Jorge Eduardo Romero Garcia.			Junio 19, 2021	Bug de activos por fin solucionados
Alejandra Claire Aguilar Mata			Junio 19, 2021	Comentario finales en todo el sistema