**Archivo que genera el PDF HTML2PDF**

<?php   
require \_\_DIR\_\_."/vendor/autoload.php";   
use Spipu\Html2Pdf\Html2Pdf;   
  
//if(isset($\_POST["ok"])){   
       
    ob\_start();   
    require\_once("pdf\_view.php");   
    $html=ob\_get\_clean();   
    $html2pdf = new Html2Pdf('P','A4','es','true','UTF-8');   
    $html2pdf->writeHTML($html);   
    //$html2pdf->writeHTML("<h1>GRUPO sis32</h1>");   
    //$html2pdf->writeHTML("primet pdf");   
    $html2pdf->output();   
/\*}else{   
    echo "   
        <form method=post>   
        <input type=submit name=ok value='ver'>   
        </form>   
    ";   
}   
\*/   
  
?>

**Pagina a cargar en el archivo PDF HTML2PDF**

<!DOCTYPE HTML>   
<html lang='es'>   
<head>   
<meta charset="UTF-8" />   
<title>Generador PDF</title>   
<style type="text/css">   
    #cabecera img{   
        float: right;   
    }   
    #cabecera h1{   
        display: block;   
        float: right;   
    }   
</style>   
</head>   
<body>   
<div id=cabecera>   
  
<img src="html2pdf.png" />   
  
<h1>Ejemplo uso libreria HTML2PDF</h1>   
  
</div>   
<h2><?php echo "Ricardo";?></h2>   
</body>   
</html> 

**LA OTRA CLASE**

**plantilla.php es para generar el encabezado y pie de pagina**

<?php   
    require 'fpdf/fpdf.php';   
       
    class PDF extends FPDF   
    {   
        function Header()   
        {   
            $this->Image('fpdf.jpg', 5, 5, 30 );   
            $this->SetFont('Arial','B',15);   
            $this->Cell(30);   
            $this->Cell(120,10, 'Reporte De Estados',0,0,'C');   
            $this->Ln(20);   
        }   
           
        function Footer()   
        {   
            $this->SetY(-15);   
            $this->SetFont('Arial','I', 8);   
            $this->Cell(0,10, 'Pagina '.$this->PageNo().'/{nb}',0,0,'C' );   
        }           
    }   
?>

**Archivo que genera el PDF con FPDF**

<?php   
    include 'plantilla.php';   
    //require 'conexion.php';   
    require\_once("Persona.php");   
    $person=new Persona();   
    /\*$query = "SELECT e.estado, m.id\_municipio, m.municipio FROM t\_municipio AS m INNER JOIN t\_estado AS e ON m.id\_estado=e.id\_estado";   
    $resultado = $mysqli->query($query);   
    \*/   
       
    $datos=$person->reporteFPDF();   
       
       
       
    $pdf = new PDF();   
    $pdf->AliasNbPages();   
    $pdf->AddPage();   
       
    $pdf->SetFillColor(232,232,232);   
    $pdf->SetFont('Arial','B',12);   
    $pdf->Cell(50,6,'Nombres',1,0,'C',1);   
    $pdf->Cell(50,6,'Apellidos',1,0,'C',1);   
    $pdf->Cell(20,6,'Edad',1,0,'C',1);   
    $pdf->Cell(50,6,'Direccion',1,1,'C',1);   
    $pdf->SetFont('Arial','',10);   
       
    while($row = $datos->fetch\_assoc())   
    {   
        $pdf->Cell(50,6,utf8\_decode($row['nombres']),1,0,'C');   
        $pdf->Cell(50,6,$row['apellidos'],1,0,'C');   
        $pdf->Cell(20,6,utf8\_decode($row['edad']),1,0,'C');   
        $pdf->Cell(50,6,utf8\_decode($row['direccion']),1,1,'C');   
    }   
    $pdf->Output();   
?>

**CLASE CONEXION.PHP**

<?php   
  
include 'credenciales.php';    
class Conexion{    
    protected $conn;     
    const SALT = 'EstoEsUnSalt';//llave para encriptar   
       
    public function Conexion(){    
     $this->conn=new mysqli(SERVIDOR,USUARIO,CONTRA,BASEDATOS);    
    }    
       
     public function desconectar(){    
          $this->conn->close();    
     }    
       
     public function consultar($sql){    
          $this->Conexion();    
          $res = $this->conn->query($sql);              
        $this->desconectar();    
          return $res;    
     }    
       
     public function ejecutar($sql){    
          $this->Conexion();    
          $this->conn->query($sql);              
        $this->desconectar();    
          return true;    
     }    
       
    function login($usuario, $password) {   
        // el uso de declaraciones preparadas evita las   
        // inyecciones sql.    
        $this->Conexion();    
       
        $sqlUsuario="   
                SELECT    
                    idusuario,usuario, contra,nivel    
                FROM    
                    usuarios    
                WHERE    
                    usuario = ? LIMIT 1";       
        if ($stmt = $this->conn->prepare($sqlUsuario)) {   
               
            $stmt->bind\_param('s', $usuario);   
            $stmt->execute();// ejecutamos la consulta.   
            $stmt->store\_result();   
            // obtenomos las variables del resultado   
            //bind\_result recomendado cuando solo es un registro   
            $stmt->bind\_result($idusuario, $usuario, $contra, $nivel);   
            $stmt->fetch();   
  
            // clave encriptada.       
            $password = $this->encriptar($password);   
  
            if ($stmt->num\_rows == 1) {   
                // si el usuario existe verificamos si la cuenta   
                // esta bloqueada por fuerza bruta   
                if ($this->checkbrute($idusuario) == true) {   
                    // bloquear la cuenta    
                    // enviar notificacion del bloqueo .   
                    echo "cuenta cerrada temporalmente intente dentro de 2 horas";   
                       
                    return false;   
                } else {   
                    // verificamos que las claves sean iguales.   
                       
                    if ($contra == $password) {   
                        // la clave fue correcta   
                        // obtenomos el navegador del cliente   
                        $user\_browser = $\_SERVER['HTTP\_USER\_AGENT'];   
  
                        // XSS protection as we might print this value   
                        $idusuario = preg\_replace("/[^0-9]+/", "", $idusuario);   
                        $\_SESSION['user\_id'] = $idusuario;   
  
                        // XSS protection as we might print this value   
                        $usuario = preg\_replace("/[^a-zA-Z0-9\_\-]+/", "", $usuario);   
  
                        $\_SESSION['usuario']["nivel"] = $nivel;   
                        $\_SESSION['usuario']["nombre"] = $usuario;   
                        header("Location:mantoPersona.php");   
  
                        // ingreso al sistema    
                        return true;   
                    } else {   
                        // la clave fallo   
                        // almacenamos este falso ingreso en la base de datos    
                        $now = time();   
                        if ($this->conn->query("INSERT INTO login\_attempts(user\_id, time)    
                                        VALUES ('$idusuario', '$now')")) {   
                                               
                            echo "<script>alert('datos erroneos');</script>";   
                            header("Location: loguin.php?error");   
                            exit();   
                        }   
                        return false;   
                    }   
                }   
            } else {   
                // El usuario no existe.    
                return false;   
            }   
        } else {   
            // error con la base de datos   
            header("Location: ../error.php?err=Database error: cannot prepare statement");   
            exit();   
        }   
    }   
  
    function checkbrute($user\_id) {   
        // obtenemos la hora del sistema    
        $now = time();   
        // Consultamos todos los falsos ingresos en las ultimas    
        // 2 horas.    
        $valid\_attempts = $now - (2 \* 60 \* 60);   
        $this->Conexion();   
        $sqlAtentados="SELECT time FROM login\_attempts WHERE user\_id = ? AND time > '$valid\_attempts'";   
        if ($stmt = $this->conn->prepare($sqlAtentados)){   
            $stmt->bind\_param('i', $user\_id);   
            // ejecutamos la consulta preparada.    
            $stmt->execute();   
            $stmt->store\_result();   
            // si hay mas de 5 intentos devolvemos true    
            if ($stmt->num\_rows > 5){   
                return true;   
            } else {   
                return false;   
            }   
        } else {   
            // Error con la base de datos   
            header("Location: ../error.php?err=Database error: cannot prepare statement");   
            exit();   
        }   
           
    }   
  
    function SQLsegura($strVar){   
        $banned = array("select", "drop","|","'", ";", "--", "insert","delete", "xp\_");        
        $vowels = $banned;   
        $no = str\_replace($vowels, "", $strVar);   
        $final = str\_replace( "'","",$no);   
        return $final;   
    }//End Function   
  
    public static function encriptar($password) {   
        return hash('sha512', self::SALT . $password);   
    }   
       
       
    public static function verificar($password, $hash) {   
        return ($hash == self::hash($password));   
    }   
       
    public function vacio(){          
        $sql="select \* from usuarios";   
        $this->Conexion();    
          $res = $this->conn->query($sql);    
        //$resultado= $conexion->query($sql);          
        $cuantos = mysqli\_num\_rows($res);    
        $this->desconectar();    
          return $cuantos;   
              
    }   
       
       
       
       
}    
?>   

**PAGINA acceso.php Modificada**

<?php    
session\_start();    
  
require\_once( 'conexion.php');   
 if(isset($\_REQUEST["btnIngresar"])){         
    $usuario=$\_REQUEST["txtUsuario"];    
     $contra=$\_REQUEST["txtContra"];    
    $c = new Conexion();    
    $c->login($usuario, $contra);   
    if($c->login($usuario, $contra)){   
           
    }else{   
        header("location: mantousuario.php");   
    }   
  
}    
if(isset($\_REQUEST["cerrar"])){         
    session\_destroy();         
    header("Location:loguin.php");    
}    
  
show\_source("Acceso.php");   
show\_source("conexion.php");   
?>