

2019

# Cliente De Correo Web

**ADMINISTRACION DE SISTEMAS INFORMÁTICOS EN RED**

MANUEL GÓMEZ RODRÍGUEZ | IES LA MARISMA



## Índice

1.	Introducción .....	2
2.	Protocolos de correo electrónico .....	3
2.1.	POP3. ....	3
2.2.	IMAP .....	3
2.3.	SMTP .....	4
3.	Objetivos .....	5
4.	Recursos utilizados .....	6
4.1.	Lenguajes de programación. ....	6
4.2.	Sistema operativo.....	6
4.3.	Software. ....	6
4.4.	Hardware. ....	6
4.5.	Librerías de PHP. ....	6
5.	Planificación .....	9
6.	Ejecución del proyecto.....	12
7.	Configuración.....	30
8.	Conclusiones finales.....	31
8.1.	Grado de cumplimiento de los objetivos fijados.....	31
8.2.	Propuestas de modificaciones o ambiciones futuras del proyecto. ....	31
9.	Referencias.....	32



## 1. Introducción

El webmail es un sistema de acceso al correo electrónico que permite acceder a sus mensajes mediante un navegador web, sin tener que descargar los correos a su ordenador personal.

Se provee una interfaz web por la que accede al correo electrónico, el webmail permite listar, desplegar y borrar vía un navegador web los correos almacenados en el servidor remoto. Con el webmail además de leer mensajes también se pueden enviar mensajes a otras cuentas de correo.

Los correos pueden ser consultados posteriormente desde otro dispositivo que disponga de un navegador web. La privacidad de los usuarios de webmail se lleva a cabo mediante la utilización de nombres de usuario y contraseña únicos.

*Según cita la Wikipedia (Wikipedia, 2019)*

### Ventajas de webmail

- Los mensajes pueden leerse, escribirse y enviarse desde cualquier lugar con un explorador y conexión a Internet.
- Los mensajes no tienen que descargarse al ordenador.
- Las cuentas de correo pueden crearse fácilmente, lo que permite crear cuentas para uso anónimo fácilmente.

### Desventajas de webmail

- El usuario tiene que estar conectado a Internet mientras lee y escribe los mensajes.
- Los servidores de webmail comerciales normalmente ofrecen espacio limitado para el almacenamiento de los mensajes y muestran propaganda en los mensajes
- Cuando la conexión a Internet es lenta, puede ser difícil enviar los mensajes.

*Información extraída de la página web (Masadelante, 2019).*



## 2. Protocolos de correo electrónico

### 2.1. POP3.

Post Office Protocol version 3 (POP3) es un protocolo standard de correo que se usa para la recepción de correo desde un servidor remoto a un cliente de correo local.

POP3 te permite descargar los mensajes de correo en tu ordenador local y leerlos cuando estés offline. Ten en cuenta que cuando usas POP3 para conectarte a tu cuenta de correo, los mensajes se descargan localmente y se eliminan del servidor de correo.

Esto significa que, si accede a tu cuenta desde varias localizaciones o dispositivos, puede que esta no sea la mejor opción. Por otra parte, si usas POP3, tus mensajes se guardan en tu dispositivo local, lo que reduce el espacio que tus correos ocupan en tu servidor web.

Por defecto, el protocolo POP3 trabaja con dos puertos:

1. **Puerto 110** - Este es el predeterminado POP3 no cifrado.
2. **Puerto 995** -Este es el puerto que debes usar si quieres conectarte usando POP3 de forma segura (cifrada).

#### Ventajas:

- El correo es accesible desde cualquier dispositivo con acceso a Internet.
- Al trabajar directamente con el servidor de correo, cualquier cambio es visible desde todos los dispositivos que accedan a la cuenta.
- Si el dispositivo desde el que se conecta a la cuenta tiene una avería, no se pierden los correos al estar almacenados directamente en el servidor.

#### Desventajas:

- Al almacenar el correo en el servidor es más sencillo llenar la capacidad del buzón.
- Si se crean carpetas IMAP, no son accesibles desde configuraciones POP3.

### 2.2. IMAP.

Internet Message Access Protocol (IMAP) es un protocolo de correo usado para acceder al correo de un servidor web remoto desde un cliente local. IMAP y POP3 son los dos protocolos de correo de internet más utilizados para la Recuperación de Correos. Ambos protocolos son compatibles con todos los clientes de correo modernos y servidores web.



Mientras que el protocolo POP3 asume que solo se accede a tu correo desde una única aplicación, IMAP permite el acceso desde varios clientes de correo. Por eso IMAP es el más adecuado si vas a acceder desde diferentes localizaciones o si sus mensajes son gestionados por varios usuarios...

Por defecto, el protocolo IMAP trabaja con dos puertos:

1. **Puerto 143** - Este es el predeterminado para IMAP no cifrado.
2. **Puerto 993** - Este es el puerto que debes usar si quieres conectarte usando IMAP de forma segura (cifrada).

#### **Ventajas:**

- Poder conservar un histórico del correo en local sin necesidad de conexión a Internet.
- El espacio en el servidor de correo no se llena porque una vez descargado el correo se borra del servidor.

#### **Desventajas:**

- Los correos enviados solo son accesibles de forma local.
- No se puede ver el correo desde distintos dispositivos. Si el dispositivo sufre una avería, el correo puede llegar a perderse y no es accesible.
- Consume espacio local.

### 2.3. SMTP

Simple Mail Transfer Protocol (SMTP) es un protocolo estándar para el envío de correo de Internet. Se conecta entre el cliente de correo y el servidor SMTP para poder enviar correos, en caso de que el correo se envíe a una cuenta de correo de otro proveedor se produce el relevo de información de servidor SMTP a servidor SMTP y por último al cliente de correo receptor del mensaje

Por defecto, el protocolo SMTP trabaja con 3 puertos:

3. **Port 25** - Este es el predeterminado para SMTP no cifrado.
4. **Port 465 o 587** - Este es el puerto que debes usar si quieres conectarte usando SMTP de forma segura (cifrada).

*Tal y como cita (Steground, 2019) en el siguiente artículo.*



### 3. Objetivos

1. Estudio de los protocolos POP e IMAP.
2. Estudio de funciones PHP IMAP.
3. Estudio de funciones PHP MAILER.
4. Permitir a los usuarios conectarse a cuentas en servidores de correo POP3 e IMAP de su servidor de correo.
5. Leer mensajes de correo.
6. Enviar mensajes de correo.
7. Responder a mensaje de correo.
8. Reenviar mensajes de correo.
9. Borrar correo de la cuenta.
10. Formularios con validación de datos.
11. Creación de base de datos y tablas.
12. Encriptación de contraseñas en la base de datos.
13. Hosting, nombre de dominio y conexión a base de datos.
14. Creación de cuentas de usuario e inicio de sesión.
15. Cierre de sesión de cuentas.
16. Opción de multicuentas en el mismo usuario (Gmail, Outlook, Yahoo).
17. Interfaz fácil e intuitiva para el usuario.



## 4. Recursos utilizados

### 4.1. Lenguajes de programación.

- PHP y JavaScript.
- SQL.
- HTML5 y CSS.

### 4.2. Sistema operativo.

- Windows 7 Professional 64 bits.

### 4.3. Software.

- Editor de texto (Notepad ++, Visual Studio Code).
- Servidor local (XAMPP XAMPP).
- Navegador Web.

### 4.4. Hardware.

- Servidor web para alojar los archivos (Hostinger).
- Ordenador.

### 4.5. Librerías de PHP.

- PHPMAILER y PHP IMAP.

**PHPMailer** es utilizado para enviar emails vía sendmail, PHP mail(), o con SMTP. Lo más recomendable es usando smtp por dos razones:

- Con PHPMailer se pueden usar varios servidores SMTP. Esto permite repartir la carga entre varios servidores, con lo que se podrán enviar un mayor número de mensajes en un tiempo menor.
- Además, los servidores SMTP permiten mensajes con múltiples to's (destinatarios), cc's (Las direcciones que aparezcan en este campo recibirán el mensaje. Todos los destinatarios verán las direcciones que aparecen en la sección Cc), bcc's (Las direcciones que aparezcan en el campo Bcc recibirán



El mensaje. Sin embargo, ninguno de los destinatarios podrá ver las direcciones que fueron incluidas en la sección Bcc) y Reply-tos (direcciones de respuesta)

PHPMailer hace fácil esta difícil tarea de enviar correos con estas características y puedes incluso utilizar tu propio servidor smtp, aunque éste requiera autenticación (un nombre de usuario y contraseña).

Necesitamos el fichero **class.phpmailer.php** que es la clase PHPMailer propiamente dicha y también el fichero **class.smtp.php** que nos permite el envío de emails a través de un servidor smtp.

*La descarga de estos archivos de configuración de PHP Mailer ha sido utilizada gracias a GitHub (Synchro, 2019).*

```
<?php
// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// Load Composer's autoloader
require 'vendor/autoload.php';

// Instantiation and passing `true` enables exceptions
$mail = new PHPMailer(true);

try {
    //Server settings
    $mail->SMTPDebug = 2; // Enable verbose debug output
    $mail->isSMTP(); // Set mailer to use SMTP
    $mail->Host = 'smtp1.example.com;smtp2.example.com'; // Specify main and backup SMTP servers
    $mail->SMTPAuth = true; // Enable SMTP authentication
    $mail->Username = 'user@example.com'; // SMTP username
    $mail->Password = 'secret'; // SMTP password
    $mail->SMTPSecure = 'tls'; // Enable TLS encryption, `ssl` also accepted
    $mail->Port = 587; // TCP port to connect to

    //Recipients
    $mail->setFrom('from@example.com', 'Mailer');
    $mail->addAddress('joe@example.net', 'Joe User'); // Add a recipient
    $mail->addAddress('ellen@example.com'); // Name is optional
    $mail->addReplyTo('info@example.com', 'Information');
    $mail->addCC('cc@example.com');
    $mail->addBCC('bcc@example.com');
```





```
// Attachments
$mail->addAttachment('/var/tmp/file.tar.gz');      // Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

// Content
$mail->isHTML(true);                               // Set email format to HTML
$mail->Subject = 'Here is the subject';
$mail->Body    = 'This is the HTML message body <b>in bold!</b>';
$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

$mail->send();
echo 'Message has been sent';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```



## 5. Planificación

Conexión con el host SMTP, esto es imprescindible en cada archivo que vaya a interactuar con el servidor de correo, sin abrir la conexión no podemos hacer nada. Para poder iniciar la conexión se utiliza la función de IMAP PHP llamada **imap\_open**. Esta función cuenta con 3 variables obligatorias **\$hostname** es el servidor de correo, **\$username** es el correo electrónico del servidor asociado y por último **\$password** que es la contraseña asociada al correo electrónico, con uno de estas variables incorrecta no se iniciará la conexión.

Recepción de mensajes, para poder hacer esto tuve que estudiar el funcionamiento de las funciones de PHP IMAP, esta librería contiene muchas funciones a utilizar para la conexión, recepción, eliminación de mensajes...

La más importante para la recepción de mensajes es la función **imap\_fetchstructure**, esta función tiene una serie de cabeceras (from, date, message, header...) que son las que se visualizarán en un mensaje de correo mediante el uso de un bucle para así ver todos los mensajes.

Borrar mensajes de la cuenta, para esto he usado la librería PHP IMAP, con la función **imap\_delete** para marcar un mensaje para el borrado y la función **imap\_expunge** que hace la confirmación del borrado del mensaje o mensajes seleccionados, sin la función **imap\_expunge** el mensaje realmente no se eliminará.

*La siguiente información fue extraída de PHP IMAP (PHP.net/manual, 2019).*

Enviar mensajes, para esto he usado la librería PHP Mailer, es la librería más utilizada y segura para el envío de mensajes, ya que permite utilizar la conexión cifrada de POP e IMAP, además de las características más actualizadas.

**Base de datos**, al principio lo hice con XAMPP y phpMyAdmin, con esta herramienta se puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos.

La base de datos del cliente de correo consta de 2 tablas, una llamada **usuarios** y otra llamada **cuentas**.





La tabla **Usuarios** contiene las filas llamadas **id\_usuario** (**Clave Primaria**), **nombre**, **usuario**, **clave** y **teléfono**, estos datos serán insertados en la base de datos mediante un formulario.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id_usuario</b>	int(255)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nombre</b>	varchar(50)	utf8_unicode_ci		Sí	NULL		
<input type="checkbox"/> 3	<b>usuario</b>	varchar(50)	utf8_unicode_ci		Sí	NULL		
<input type="checkbox"/> 4	<b>clave</b>	varchar(255)	utf8_unicode_ci		Sí	NULL		
<input type="checkbox"/> 5	<b>telefono</b>	int(9)			Sí	NULL		

Las contraseñas se encriptan en la base de datos mediante **MD5**

usuario	clave
manumanu	827ccb0eea8a706c4c34a16891f84e7b

La tabla **Cuentas** contiene las filas **id\_cuentas** (**Clave Primaria**), **username**, **password** e **id\_usuario** (**clave foranea**) esta tabla contiene las cuentas reales de correo electrónico.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id_cuentas</b>	int(255)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>username</b>	varchar(255)	utf8_unicode_ci		Sí	NULL		
<input type="checkbox"/> 3	<b>password</b>	varchar(255)	utf8_unicode_ci		Sí	NULL		
<input type="checkbox"/> 4	<b>id_usuario</b>	int(255)			No	Ninguna		

username	password
manuhuelva8@gmail.com	64bbb61af62514adcf568bbdf8fd075

Ha sido necesaria la información de la Wikipedia para completar este proyecto (Wikipedia, 2019).

Lo siguiente fue comprar un Hosting y nombre de dominio para mi página Web, utilicé **Hostinger** para comprar el hosting y GoDaddy para elegir el nombre de dominio **correomarisma.es**.

Una vez que la compra se hizo efectiva tenía que apuntar los registros DNS de GODADDY hacia los de Hostinger para que la página estuviera visible en la web. Este proceso es fácil porque simplemente hay que copiar y pegar los DNS que te dice Hostinger y pegarlos en un nuevo registro de GODADDY. Este proceso tarda unas 12 horas, a partir de ahí ya está visible la web en Internet.

GoDaddy fue la página seleccionada para el nombre de dominio (GoDaddy, 2019).



El hosting venia con el protocolo HTTPS que había que activarlo desde el dashboard de la página de Hostinger se hace bastante fácil, además de tener asistentes virtuales que periódicamente te preguntan y ayudan.

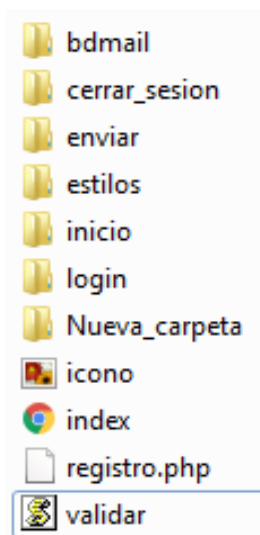
Para utilizar la base de datos que tenía en XAMPP era necesario primero crear una base de datos en la web y después importar la base de datos de XAMPP a HOSTINGER y cambiar las variables de conexión que tenía en XAMPP a las variables de conexión **usuario, contraseña, nombre base datos y host**.

*Hostinger fue la página seleccionada para tener la web en funcionamiento (Hostinger, 2019).*



## 6. Ejecución del proyecto

### 6.1. Árbol de directorios y archivos.



### 6.2. Código fuente del proyecto.

index.html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
5      <title>WebMail - Registro</title>
6      <link rel="icon" type="image/png" href="icono.png">
7      <link rel="stylesheet" href="/estilos/estilos.css">
8      <script src="validar.js"></script>
9  </head>
10 <body>
11     <form action="registro.php" method="POST" class="form-register" onsubmit="return validar();">
12         <h2 class="form-titulo">Crea tu cuenta de usuario</h2>
13         <div class="contenedor-inputs">
14             <input type="text" id="nombre" name="nombre" placeholder="Nombre" class="input-48" >
15             <input type="text" id="usuario" name="usuario" placeholder="Nombre de usuario" class="input-48" >
16             <input type="password" id="clave" name="clave" placeholder="Clave" class="input-48" >
17             <input type="password" id="reclave" name="reclave" placeholder="Confirme la Clave" class="input-48" >
18             <input type="text" id="telefono" name="telefono" placeholder="Nº Teléfono" class="input-48">
19             <input type="submit" value="Registrarse" class="btn-enviar">
20             <p class="form-link">¿Ya tiene una cuenta? <a href=" ../login/login.html">Ingresa aquí</a></p>
21         </div>
22     </form>
23 </body>
24 </html>

```



validar.js

```
JS validar.js x
1  // CREACION DE FUNCION "VALIDAR"
2  function validar(){
3
4  // DECLARACION DE VARIABLES A UTILIZAR
5      var nombre;
6      var usuario;
7      var clave;
8      var reclave;
9      var telefono;
10
11 // RECOGER VALORES DEL FORMULARIO MEDIANTE EL VALOR "ID"
12 nombre = document.getElementById("nombre").value;
13 usuario = document.getElementById("usuario").value;
14 clave = document.getElementById("clave").value;
15 reclave = document.getElementById("reclave").value;
16 telefono = document.getElementById("telefono").value;
17
18 // LOS CAMPOS NO PUEDEN ESTAR VACIOS
19 if (nombre === "" || usuario === "" || clave === "" || reclave === "" || telefono === ""){
20     alert("COMPLETA TODOS LOS CAMPOS");
21     return false;
22 }
23
24 // LOS CAMPOS NO PUEDEN TENER SOLO ESPACIOS EN BLANCO
25 else if (nombre ==0 || usuario==0 || clave ==0 || reclave ==0 || telefono ==0){
26     alert("NO SE PERMITEN ESPACIOS EN BLANCO");
27     return false;
28 }
29
30 // LONGITUD MAXIMA DEL NOMBRE
31 else if (nombre.length>20){
32     alert ("EL NOMBRE ES DEMASIADO LARGO");
33     return false;
34 }
35
36 // LONGITUD MINIMA DE LA CLAVE DE USUARIO
37 else if (clave.length<5){
38     alert ("LA CLAVE ES DEMASIADO CORTA (MINIMO 5 CARACTERES)");
39     return false;
40 }
41
42 // LA CLAVE Y EL USUARIO NO PUEDEN SER IDENTICAS
43 else if (clave === usuario){
44     alert ("LA CLAVE NO PUEDE SER IGUAL QUE EL NOMBRE DE USUARIO")
45     return false;
46 }
```



```
47
48 // LA CLAVE Y LA CONFIRMACION DE LA CLAVE TIENEN QUE SER IGUALES
49     else if (clave != reclave){
50         alert ("LAS CLAVES NO COINCIDEN")
51         return false;
52     }
53
54 // LONGITUD MAXIMA DEL TELEFONO
55     else if (telefono.length>=10){
56         alert ("EL TELEFONO TIENE DEMASIADOS NUMEROS");
57         return false;
58     }
59
60 // LONGITUD MINIMA DEL TELEFONO
61     else if (telefono.length<9){
62         alert ("EL TELEFONO NO ES UN NUMERO REAL");
63         return false;
64     }
65
66 // LA FUNCION "ISNaN" OBLIGA A SOLO PERMITIR NUMEROS
67     else if (isNaN(telefono)){
68         alert("ESCRIBA CORRECTAMENTE EL TELEFONO");
69         return false;
70     }
71 }
```



registro.php

```
registro.php ●
1  <?php
2  // SE INICIA LA SESION
3  session_start();
4
5  // "REQUIRE" AÑADE LAS LINEAS DEL ARCHIVO DE CONEXION
6  // ASI NO HAY QUE ESCRIBIR CONSTANTEMENTE LAS LINEAS DE CONEXION
7  require 'bdmail/conexion.php';
8
9  // RECIBIR DATOS DEL FORMULARIO Y ALMACENAR EN VARIABLES
10 // CON "MD5" LA STRING SE CIFRA CON CARACTERES ALEATORIOS
11 $nombre = $_POST['nombre'];
12 $usuario = $_POST['usuario'];
13 $clave=md5($_POST['clave']);
14 $telefono = $_POST['telefono'];
15
16 $_SESSION['usuario'] = $usuario;
17
18 // COMANDO SQL PARA INSERTAR DATOS
19 $insertar = "INSERT INTO usuarios(nombre, usuario, clave, telefono)
20 VALUES ('$nombre', '$usuario', '$clave', '$telefono')";
21
22 // VERIFICAR QUE EL USUARIO NO EXISTE EN LA BASE DE DATOS
23 $verificar = mysqli_query($conexion, "SELECT * FROM usuarios
24 WHERE usuario = '$usuario' and clave = '$clave'");
25
26 // SI EL NUMERO DE FILAS ES MAYOR QUE 0 SIGNIFICA QUE HAY DATOS INSERTADOS
27 // SI HAY DATOS INSERTADOS DA MENSAJE DE ERROR Y VUELVE AL REGISTRO
28 if (mysqli_num_rows($verificar) > 0){
29     echo '<script>
30         alert("El nombre de usuario ya se encuentra registrado");
31         window.history.go(-1);
32     </script>';
33     exit;
34 }
35
36 // EN CASO CONTRARIO EL USUARIO ES NUEVO Y PUEDE CONTINUAR HACIA LA PAGINA LOGIN
37 else {
38     header ("location:/login/login.html");
39     sleep (3);
40 }
41
42 //COMANDO PARA HACER QUE INSERTE LOS DATOS EN LA BASE DE DATOS
43 $resultado = mysqli_query($conexion, $insertar);
44
45 // COMANDO PARA CERRAR LA CONEXION
46 mysqli_close ($conexion);
47 ?>
```





bdmail/conexión.php

```
1  <?php
2  // CREO UNA VARIABLE PARA GUARDAR LOS VALORES NECESARIOS PARA LA CONEXION
3  // LOS VALORES SON "HOST" "USUARIO" "CONTRASEÑA" "NOMBRE_BASE_DATOS"
4  // SI LA CONEXION FALLA DEVUELVE EL ERROR ESPECIFICO
5  $conexion = mysqli_connect("localhost", "root", "", "bdmail")
6  or die ("No se pudo conectar" . mysql_error($conexion));
7  ?>
```

login/login.html

```
<> login.html •
1  <?php
2  session_start();
3  ?>
4
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
9      <title>WebMail - Login</title>
10     <link rel="icon" type="image/png" href="../icono.png">
11     <link rel="stylesheet" href="../estilos/estilos.css">
12     <script src="validar_login.js"></script>
13 </head>
14 <body>
15     <form action="login.php" method="POST" class="form-register" onsubmit="return validar_login();">
16         <h2 class="form-titulo">Inicia sesión</h2>
17         <div class="contenedor-inputs">
18             <input type="text" name="usuario" id="usuario" placeholder="#9000; Usuario">
19             <input type="password" name="clave" id="clave" placeholder="#128272; Clave">
20             <input type="submit" value="Entrar" class="btn-enviar">
21             <p class="form-link">¿No tiene ninguna cuenta? <a href="../index.html">Registrese aquí</a></p>
22         </div>
23     </form>
24 </body>
25 </html>
```



login/validar\_login.js

```
JS validar_login.js ●
1  //CREACION DE FUNCION PARA VALIDAR
2  function validar_login(){
3
4  //DECLARACION DE VARIABLES A USAR
5      var usuario;
6      var clave;
7
8  //RECOGER DATOS DE LOS ID DEL FORMULARIO
9      usuario = document.getElementById("usuario").value;
10     clave = document.getElementById("clave").value;
11
12     //SI EL CAMPO "USUARIO" ESTA VACIO MUESTRA MENSAJE DE ERROR
13     if(usuario === ""){
14         alert ("INTRODUCE UN NOMBRE DE USUARIO");
15         return false;
16     }
17
18     //SI EL CAMPO "CLAVE" ESTA VACIO MUESTRA MENSAJE DE ERROR
19     else if (clave === ""){
20         alert ("INTRODUCE LA CLAVE");
21         return false;
22     }
23 }
```



login/login.php

```
login.php •
1  <?php
2  // INICIA LA SESION
3  session_start();
4
5  // "REQUIRE" AÑADE LAS LINEAS DEL ARCHIVO DE CONEXION
6  // ASI NO HAY QUE ESCRIBIR CONSTANTEMENTE LAS LINEAS DE CONEXION
7  require '../bdmail/conexion.php';
8
9  // SE RECOGEN LAS VARIABLES DE FORMULARIO
10 $usuario = $_POST['usuario'];
11 $clave = $_POST['clave'];
12
13 $_SESSION['usuario'] = $usuario;
14
15 // VERIFICAR QUE USUARIO Y CLAVE(ENCRIPADA) ESTA EN LA BASE DE DATOS
16 $consulta="SELECT usuario FROM usuarios
17 WHERE usuario = '$usuario' and clave = MD5('$clave')";
18 $resultado = mysqli_query($conexion, $consulta);
19
20 // SI EXISTEN LOS DATOS EL RESULTADO ES = 1
21 // SI NO EXISTEN LOS DATOS EL RESULTADO ES = 0
22 $filas=mysqli_num_rows($resultado);
23
24 // SI EL RESULTADO ES MAYOR QUE 0 ENTONCES HACE LOGIN Y VA A LA SIGUIENTE PAGINA
25 if($filas>0){
26     header ("location:../inicio/index.php");
27 }
```

```
28
29 // EN CASO CONTRARIO MUESTRA MENSAJE DE ERROR Y VUELVE AL LOGIN
30 else {
31     echo '<script>
32         alert("Usuario o clave incorrectas");
33         window.history.go(-1);
34     </script>';
35     exit;
36 }
37
38 // COMANDO PARA LIBERAR MEMORIA
39 mysqli_free_result($resultado);
40
41 // COMANDO PARA CERRAR LA CONEXION
42 mysqli_close($conexion);
43
44 ?>
```



inicio/index.php

```

index.php
1  <?php
2  // SE INICIA LA SESION
3
4  session_start();
5  // CREO VARIABLE PARA GUARDAR SESION DE USUARIO
6  $varsesion = $_SESSION['usuario'];
7
8  // SI LA SESION ES NULA O ESTA VACIA DEVUELVE A PAGINA DE LOGIN
9  // PORQUE NO HAS INICIADO SESION
10 if ($varsesion == null || $varsesion == ''){
11     sleep (1);
12     header("location:../login/login.html");
13     die;
14 }
15 ?>

```

```

16 <!DOCTYPE html>
17 <html>
18 <head>
19     <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
20     <link rel="stylesheet" href="../estilos/estilos.css">
21     <title>WebMail - AñadeCorreo @</title>
22     <link rel="icon" type="image/png" href="../icono.png">
23     <script src="validar.js"></script>
24 </head>
25 <body>
26     <form action="registro_correo.php" method="POST" class="form-register" onsubmit="return validar();">
27         <h2 class="form-titulo"><?php echo "Bienvenid@: " . $_SESSION['usuario'] ;
28         ' <br>' ?></h2>
29         <div class="contenedor-inputs">
30             <br>
31             <input type="text" name="username" id="username" placeholder="Correo electronico" class="input-48">
32             <input type="password" name="password" id="password" placeholder="Contraseña Correo" class="input-48">
33             <input type="submit" value="Enviar" class="btn-enviar">
34             <p class="form-link">Inicie sesion en su correo <a href="login_correo.html">Ingresa aquí</a></p>
35         </div>
36     </form>
37 </body>
38 </html>

```



inicio/registro\_correo.php

```
registro_correo.php •
1  <?php
2  // SE INICIA LA SESION
3  session_start();
4
5  // "REQUIRE" AÑADE LAS LINEAS DEL ARCHIVO DE CONEXION
6  // ASI NO HAY QUE ESCRIBIR CONSTANTEMENTE LAS LINEAS DE CONEXION
7  require '../bdmail/conexion.php';
8
9  // RECIBIR DATOS DEL FORMULARIO Y ALMACENAR EN VARIABLES
10 $username = $_POST['username'];
11 $password = md5($_POST['password']);
12 $_SESSION['username'] = $username;
13 $_SESSION['password'] = $password;
14
15 // COMANDO SQL PARA INSERTAR DATOS
16 $insertar = "INSERT INTO cuentas(username, password)
17 VALUES ('$username', '$password')";
18
19 // VERIFICAR QUE EL CORREO ELECTRONICO NO EXISTE EN LA BASE DE DATOS
20 $verificar = mysqli_query($conexion, "SELECT * FROM cuentas
21 WHERE username = '$username' and password = '$password'");
22
23 if (mysqli_num_rows($verificar) > 0){
24     echo '<script>
25         alert("El correo electronico ya se encuentra registrado");
26         window.history.go(-1);
27     </script>';
28     exit;
29 }
30
31 // EN CASO CONTRARIO EL USUARIO ES NUEVO Y PUEDE CONTINUAR HACIA LA PAGINA LOGIN
32 else {
33     header ("location:login_correo.html");
34     sleep (3);
35 }
```



```
35
36  //COMANDO PARA HACER QUE INSERTE LOS DATOS EN LA BASE DE DATOS
37  $resultado = mysqli_query($conexion, $insertar);
38
39  //ESTO ES SOLO PARA CONFIRMAR QUE TODO VA BIEN
40  if (!$resultado){
41      echo "HA HABIDO UN ERROR AL REGISTRARSE";
42  }
43  else{
44      echo "LOS DATOS SE HAN INSERTADO CORRECTAMENTE";
45  }
46
47  //CERRAR CONEXION
48  mysqli_close ($conexion);
49  ?>
```

inicio/login\_correo.html

```
<> login_correo.html •
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
5      <title>WebMail - LoginCorreo @</title>
6      <link rel="icon" type="image/png" href="../icono.png">
7      <link rel="stylesheet" href="../estilos/estilos.css">
8      <script src="validar.js"></script>
9  </head>
10 <body>
11     <form action="validar_correo.php" method="POST" class="form-register" onsubmit="return validar();">
12         <h2 class="form-titulo">Inicia sesión con tu correo electrónico</h2>
13         <div class="contenedor-inputs">
14             <input type="text" name="username" id="username" placeholder="Correo electrónico">
15             <input type="password" name="password" id="password" placeholder="Contraseña">
16             <input type="submit" value="Entrar" class="btn-enviar">
17         </div>
18     </form>
19 </body>
20 </html>
```



inicio/validar.js

```
JS validar.js ●
1  // CREACION DE FUNCION "VALIDAR"
2  function validar(){
3
4  // DECLARACION DE VARIABLES A UTILIZAR
5      var username;
6      var password;
7
8  // RECOGER VALORES DEL FORMULARIO MEDIANTE EL VALOR "ID"
9      username = document.getElementById("username").value;
10     password = document.getElementById("password").value;
11
12     // CREACION DE EXPRESION REGULAR PARA VALIDAR CORREO
13     // TEXTO@DOMINIO.TEXTO
14     correo = /\w+@\w+\.[a-z]/;
15
16
17 // EL CAMPO USERNAME NO PUEDE ESTAR VACIO
18     if (username === ""){
19         alert ("NO HAS INTRODUCIDO EL CORREO");
20         return false;
21     }
22
23 // EL CAMPO USERNAME NO PUEDE TENER ESPACIOS EN BLANCO
24     else if (username == 0){
25         alert ("CORREO ELECTRONICO NO ES CORRECTO");
26         return false;
27     }
```

```
28
29 // SI LA EXPRESION QUE SE INTRODUCE EN CORREO NO ES VALIDA MUESTRA ERROR
30     else if (!correo.test(username)){
31         alert ("EL CORREO NO ES VALIDO");
32         return false;
33     }
34
35 // EL CAMPO PASSWORD NO PUEDE ESTAR VACIO
36     else if (password === ""){
37         alert ("NO HAS INTRODUCIDO LA CONTRASEÑA");
38         return false;
39     }
40
41 // EL CAMPO PASSWORD NO PUEDE TENER ESPACIOS EN BLANCO
42     else if (password == 0){
43         alert ("NO HAS INTRODUCIDO LA CONTRASEÑA");
44         return false;
45     }
46 }
```



enviar/index.html

```
<> index.html •
1  <html>
2  <head>
3      <link rel="stylesheet" href="../estilos/estilos.css">
4      <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
5      <script src="validar.js"></script>
6      <link rel="icon" type="image/png" href="../icono.png">
7      <title>WebMail - Envía Mensajes</title>
8  </head>
9  <body>
10     <form action="../enviar/action.php" method="POST" class="form-register" onsubmit="return validar();">
11     <h2 class="form-titulo">Envío de correo electrónico</h2>
12     <div class='contenedor-inputs'>
13         <input type='text' placeholder='Destinatario:' name="destinatario" class="input-48"/>
14         <input type='text' placeholder='Asunto:' name="asunto" class="input-48"/>
15         <textarea name="cuerpo" rows="10" cols="40" placeholder="Escribe aquí el mensaje"></textarea>
16         <input type='submit' class="btn-enviar"/>
17     </div>
18 </form>
19 </body>
20 </html>
```

enviar/action.php

```
🐞 enviar.php •
1  <?php
2  // UTILIZO ARCHIVOS DE LA CARPETA PHPMAILER
3  use PHPMailer\PHPMailer\PHPMailer;
4  use PHPMailer\PHPMailer\Exception;
5
6  // CON EL COMANDO REQUIRE EVITO COPIAR Y PEGAR
7  // EL CODIGO DE LOS ARCHIVOS REFERENCIADOS
8  require 'PHPMailer/Exception.php';
9  require 'PHPMailer/SMTP.php';
10 require 'PHPMailer/PHPMailer.php';
11
12 // INICIO SESION
13 session_start();
14
15 // RECOJO VARIABLES DE SESION
16 $username = $_SESSION['username'];
17 $password = $_SESSION['password'];
```





```
18
19 // $MAIL ES LA VARIABLE GLOBAL DE PHPMAILER
20 $mail = new PHPMailer(true);
21 try {
22     $mail->isSMTP();
23     $mail->SMTPDebug = 0;
24     $mail->Host = 'smtp.gmail.com';
25     $mail->SMTPAuth = true;
26     $mail->Username = $_SESSION['username'];
27     $mail->Password = $_SESSION['password'];
28     $mail->SMTPSecure = 'tls';
29     $mail->Port = 587;
30     $mail->setFrom($_SESSION['username']);
31     $mail->addAddress($_POST['destinatario']);
32     $mail->isHTML(true);
33     $mail->Subject = $_POST['asunto'];
34     $mail->Body = $_POST['cuerpo'];
35     $mail->send();
36     sleep (2);
37     header ("Location: ../inicio/login_correo.html");
38 }
39 catch (Exception $e) {
40     echo 'Error: ', $mail->ErrorInfo;
41 }
42 ?>
```



enviar/validar.js

```
JS validar.js x
1  // CREACION DE FUNCION "VALIDAR"
2  function validar(){
3
4  // DECLARACION DE VARIABLES A UTILIZAR
5      var destinatario;
6      var asunto;
7      var cuerpo;
8
9  // RECOGER VALORES DEL FORMULARIO MEDIANTE EL VALOR "ID"
10     destinatario = document.getElementById("destinatario").value;
11     asunto = document.getElementById("asunto").value;
12     cuerpo = document.getElementById("cuerpo").value;
13
14     // CREACION DE EXPRESION REGULAR PARA VALIDAR CORREO
15     // TEXTO@DOMINIO.TEXTO
16     correo = /\w+@\w+\.[a-z]/;
17
18     // EL CAMPO DESTINATARIO NO PUEDE ESTAR VACIO
19     if (destinatario === ""){
20         alert ("NO HAS ESCRITO EL DESTINATARIO");
21         return false;
22     }
```



```
23
24 // EL CAMPO DESTINATARIO NO PUEDE TENER ESPACIOS EN BLANCO
25     else if (destinatario == 0){
26         alert ("EL DESTINATARIO NO ES CORRECTO");
27         return false;
28     }
29
30 // SI LA EXPRESION QUE SE INTRODUCE EN CORREO NO ES VALIDA MUESTRA ERROR
31     else if (!correo.test(destinatario)){
32         alert ("EL CORREO NO ES CORRECTO");
33         return false;
34     }
35
36 // EL CAMPO ASUNTO NO PUEDE ESTAR VACIO
37     else if (asunto === ""){
38         alert ("ESCRIBE EL ASUNTO DEL MENSAJE");
39         return false;
40     }
41
42 // EL CAMPO ASUNTO NO PUEDE TENER ESPACIOS EN BLANCO
43     else if (asunto == 0){
44         alert ("ESCRIBE UN ASUNTO CORRECTO");
45         return false;
46     }
```

```
47
48 // EL CAMPO DE CUERPO NO PUEDE ESTAR VACIO
49     else if (cuerpo === ""){
50         alert ("ESCRIBE EL MENSAJE");
51         return false;
52     }
53
54 // EL CAMPO CUERPO NO PUEDE TENER ESPACIOS EN BLANCO
55     else if (cuerpo == 0){
56         alert ("ESCRIBE EL CUERPO DEL MENSAJE");
57         return false;
58     }
59 }
```



cerrar\_sesion/cerrar.php

```
cerrar.php
1  <?php
2      session_start();
3      session_unset();
4      session_destroy();
5      sleep(1);
6      header("location:../login/login.html");
7  ?>
```

estilos/estilos.css

```
# estilos.css x
1  * {box-sizing: border-box;
2  }
3
4  body {
5      margin: 0;
6      font-family: sans-serif;
7      background-color: #206892;
8      padding: 200px;
9  }
10
11  h1{
12      color: #fff;
13      text-align: center;
14  }
15
16  .form-register {
17      width: 80%;
18      max-width: 410px;
19      margin: auto;
20      background: white;
21      border-radius: 7px;
22  }
23  }
```



```
24
25  .form-titulo {
26      background: deepskyblue;
27      color: #fff;
28      padding: 16px;
29      text-align: center;
30      font-weight: 100;
31      font-size: 30px;
32      border-top-left-radius: 8px;
33      border-top-right-radius: 8px;
34  }
35
36  .contenedor-inputs{
37      padding: 10px 30px;
38      display: flex;
39      flex-wrap: wrap;
40      justify-content: space-between;
41  }
42
43  input{
44      margin-bottom: 15px;
45      padding: 15px;
46      font-size: 16px;
47      border-radius: 2px;
48      border: 1px solid darkgray;
49      display: block;
50  }
```



```
51
52  .input-48{
53  |    width: 48%;
54  |  }
55
56  .input-100{
57  |    width: 100%;
58  |  }
59
60  .btn-enviar{
61  |    background: ■ rgb(56, 127, 209);
62  |    color: ■ #fff;
63  |    padding: 10px 42px;
64  |    cursor: pointer;
65  |    display: block;
66  |    float:center;
67  |  }
```

```
68
69  .btn-enviar:active{
70  |    transform: scale(1.03);
71  |  }
72
73  .form-link{
74  |    width: 100%;
75  |    margin: 7px;
76  |    text-align: center;
77  |    font-size:17px;
78  |  }
```

Gracias a las enseñanzas mostradas en el canal J&C Proyectos Web (Proyectos, 2017)

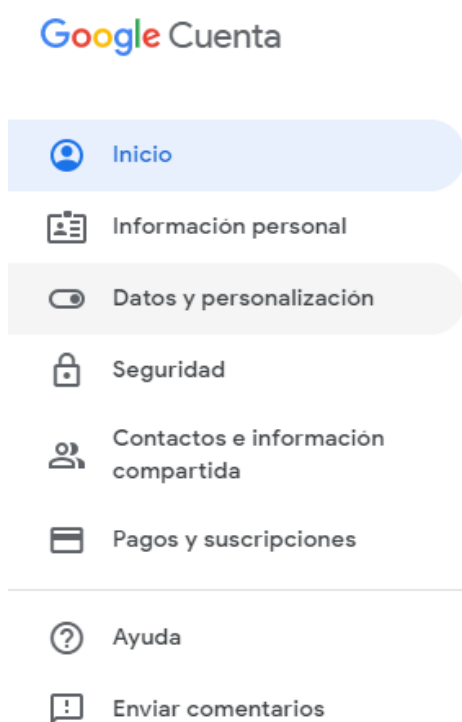
Gracias a las enseñanzas mostradas en el canal CodigoFacilito (CodigoFacilito, 2019)



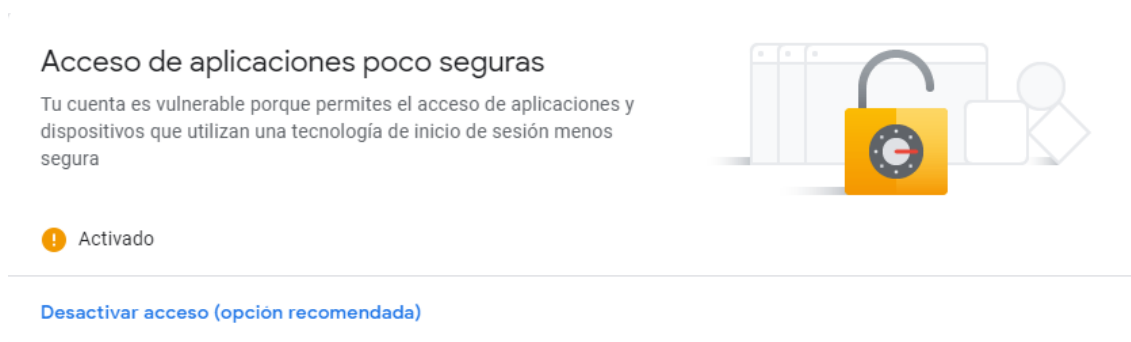
## 7. Configuración

Para poder utilizar esta aplicación en Gmail hay que activar una opción de la cuenta llamada “acceso de aplicaciones poco seguras” sin esta opción el servidor SMTP rechazará todos los intentos de conexión creyendo que es un acceso no autorizado a la cuenta Gmail.

Clic en **Seguridad**.



Clic en **activar**.



*Ha sido necesario el soporte de Google para solucionar este problema (Google, 2019).*



## 8. Conclusiones finales

### 8.1. Grado de cumplimiento de los objetivos fijados.

Para mí el proyecto ha sido un reto personal, ya que para crear un cliente de correo he tenido que utilizar todas las asignaturas que he cursado en ASIR, pero la que sobresale para poder hacerlo es programación, y es la que peor se me da, por eso lo escogí, quería aprender más sobre esta asignatura, como mejorar en ella y utilizar lo aprendido en clase para poder hacer el proyecto.

He intentado hacer todos los objetivos posibles, además de otros que a medida que iba avanzando en el proyecto he pensado que podrían ser útiles para la aplicación, aunque no estuvieran en la lista, otros objetivos se me han hecho más difícil de implementar y no he podido completarlos como yo quería.

Mi conclusión final es que en el tiempo que he tenido para hacer este proyecto (que merece bastantes horas) he hecho más de lo que yo me esperaba, aunque me hubiera gustado poder entregarlo mucho más completo y profesional.

*Mención honorífica para Silvestre Martínez Jiménez por todos sus conocimientos y motivación aportadas para este proyecto (Jimenez, 2019).*

### 8.2. Propuestas de modificaciones o ambiciones futuras del proyecto.

1. Recuperación de contraseña.
2. Envío de archivos adjuntos.
3. Recepción de mensajes multipart/alternative y multipart/mixed.
4. Mayor seguridad para base de datos.
5. Aplicación para GMAIL, YAHOO, OUTLOOK.
6. Modificación de datos y eliminación de cuenta.
7. Excepciones para errores.
8. Mejora de interfaz.





## 9. Referencias

- Abhinav. (15 de Abril de 2018). <https://stackoverflow.com/>. Obtenido de <https://stackoverflow.com/questions/52815563/how-to-read-emails-from-gmail-server-by-php-imap>
- CodigoFacilito. (2019). <https://www.youtube.com/>. Obtenido de <https://www.youtube.com/user/codigofacilito>
- GoDaddy. (Mayo de 2019). <https://es.godaddy.com/>. Obtenido de <https://es.godaddy.com/>
- Google. (2019). <https://support.google.com/>. Obtenido de <https://support.google.com/a/answer/6260879?hl=es>
- Hostinger. (Mayo de 2019). <https://www.hostinger.es/>. Obtenido de <https://www.hostinger.es/>: <https://www.hostinger.es/>
- Jimenez, S. M. (2019). *Silvestre Martinez Jimenez*. Huelva: Silvestre Martinez Jimenez.
- Masadelante. (13 de Junio de 2019). *Masadelante*. Obtenido de [www.masadelante.com](http://www.masadelante.com): <https://www.masadelante.com/faqs/que-significa-webmail>
- PHP.net/manual. (2019). <https://www.php.net/manual/>. Obtenido de <https://www.php.net/manual/es/book.imap.php>
- Proyectos, W. J. (13 de Febrero de 2017). <https://www.youtube.com/>. Obtenido de <https://www.youtube.com/watch?v=iN0xYdgRAzk>
- Steground. (2019). <https://www.siteground.es>. Obtenido de <https://www.siteground.es/tutoriales/correo/protocolos-pop3-smtp-imap/>
- Synchro. (2019). <https://github.com>. Obtenido de <https://github.com/PHPMailer/PHPMailer>
- Wikipedia. (10 de Abril de 2019). <https://es.wikipedia.org>. Obtenido de <https://es.wikipedia.org/wiki/Webmail>
- Wikipedia. (2019). <https://es.wikipedia.org/wiki/>. Obtenido de <https://es.wikipedia.org/wiki/PhpMyAdmin>