

## Tecnólogo en Análisis y Desarrollo de Sistemas de Información

### ID 2205627

**Competencia:** Implantar la solución que cumpla con los requisitos para su operación.

**Resultado de Aprendizaje:** Elaborar informes técnicos relacionados con la solución informática implantada, de acuerdo con las propuestas de alternativas aplicadas, teniendo en cuenta las técnicas de comunicación y según normas y protocolos establecidos.

**Actividad de Aprendizaje 2.2:** Practicar la implementación de notificaciones con variables bandera.

*La Formación Profesional Integral es el proceso mediante el cual la persona adquiere y desarrolla de manera permanente **conocimientos, destrezas y aptitudes e identifica, genera y asume valores y actitudes** para su realización humana y su participación en el trabajo productivo y en la toma de decisiones sociales. **Acuerdo 12 de 1985 – Unidad Técnico-pedagógica.***

Lo invito a que de manera dinámica, atenta y propositiva realice el presente instructivo con el ánimo de generar un proceso de aprendizaje (apropiación de conocimientos) entorno a la temática de reportes usando la clase FPDF.

Ejemplo:

**Actividad de Aprendizaje 2.2: Practicar la implementación de notificaciones con variables bandera.**

#### Descripción de la Actividad

Las notificaciones son funcionalidades que tienen como finalidad el aviso de cambios o novedades en el sistema y que requieren de una rápida acción a atención por parte de un rol o usuario en específico, es así como podemos ver en varios aplicativos que usamos a diario esta funcionalidad, como por ejemplo Facebook, WhatsApp o Messenger



A continuación, revisaremos un ejercicio y donde implementamos las notificaciones de manera funcional o visual, a pesar de ser muy sencillo, es lo necesario para iniciar a desarrollar e implementar



esta funcionalidad en nuestros aplicativo.

En el ejercicio observaremos una bandeja para enviar mensajes y una para visualizarlos, cada vez que se envíe un mensaje nuevo, aparecerá una notificación. ¡¡¡Manos a la obra!!!

## Ejercicio Demostrativo

### 1. Base de datos

Para este ejercicio, tendremos una base de datos muy sencilla que contara de una sola tabla con la siguiente estructura.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	id_men	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar  Eliminar  Primaria  Más
<input type="checkbox"/> 2	asu_men	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar  Eliminar  Primaria  Más
<input type="checkbox"/> 3	con_men	varchar(150)	latin1_swedish_ci		No	Ninguna			Cambiar  Eliminar  Primaria  Más
<input type="checkbox"/> 4	est_men	varchar(15)	latin1_swedish_ci		No	Ninguna			Cambiar  Eliminar  Primaria  Más

**id\_men** Este atributo es la llave primaria auto incrementable, por lo tanto, es de tipo entero.

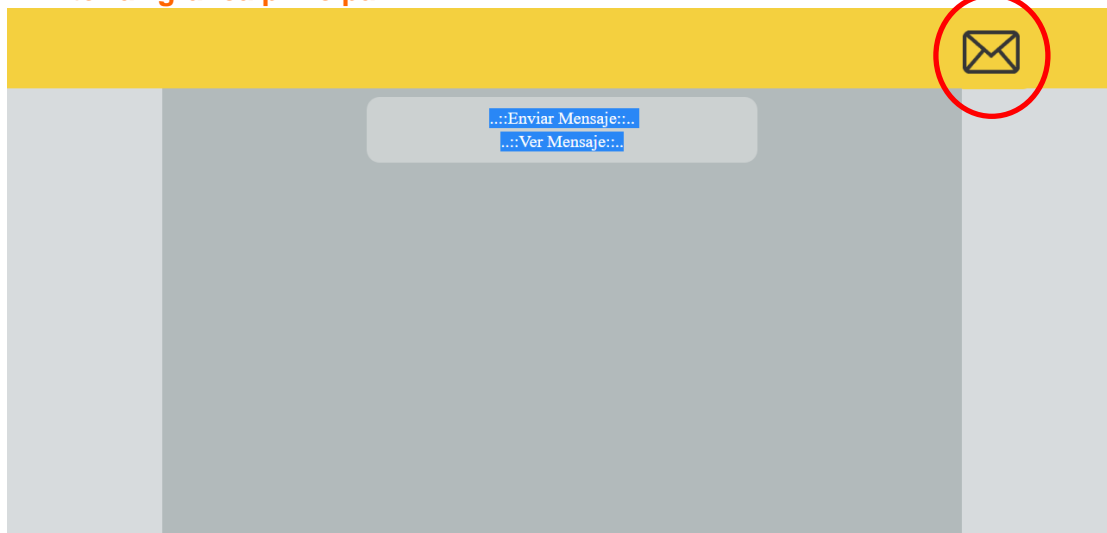
**asu\_men** Este atributo sirve para guardar el asunto o “titulo” de mensaje que sea enviado, este es de tipo varchar y tiene una longitud de 50 caracteres.

**con\_men** Este atributo sirve para guardar el contenido del mensaje, este es de tipo varchar y tiene una longitud de 150 caracteres.

**est\_men** Sirve para guardar el estado del mensaje, este atributo es clave y es el responsable de controlar la notificación y la visualización de la misma.

Estos cuatro atributos son suficientes para el desarrollo de la aplicación, la idea es que el ejercicio sea lo más sencillo posible pero totalmente explicativo a la temática.

### 2. Interfaz gráfica principal



Para el ejercicio se ha creado una interfaz gráfica muy básica que consta de una HTML con un encabezado <header> y una parte central <section>.

En el encabezado (color amarillo) es donde se va a hacer visible la notificación una vez se realice el envío



de un mensaje.

En la parte central (color gris) aparecer dios opciones de navegación “Enviar mensaje” que nos llevara a la bandeja de escritura (salida) y “Ver mensaje” nos llevara a la bandeja de entrada para ver los mensajes que han llegado.

### 3. Enviar mensaje

Una vez el usuario de clic en “Enviar mensaje”, cargara la siguiente interfaz.

Esta es la interfaz para redactar y enviar un mensaje, consta de dos cajas de texto, “Asunto”, es donde se debe escribir el título del mensaje y “Mensaje”, es donde se escribe el contenido del mensaje, además un botón “Enviar”.

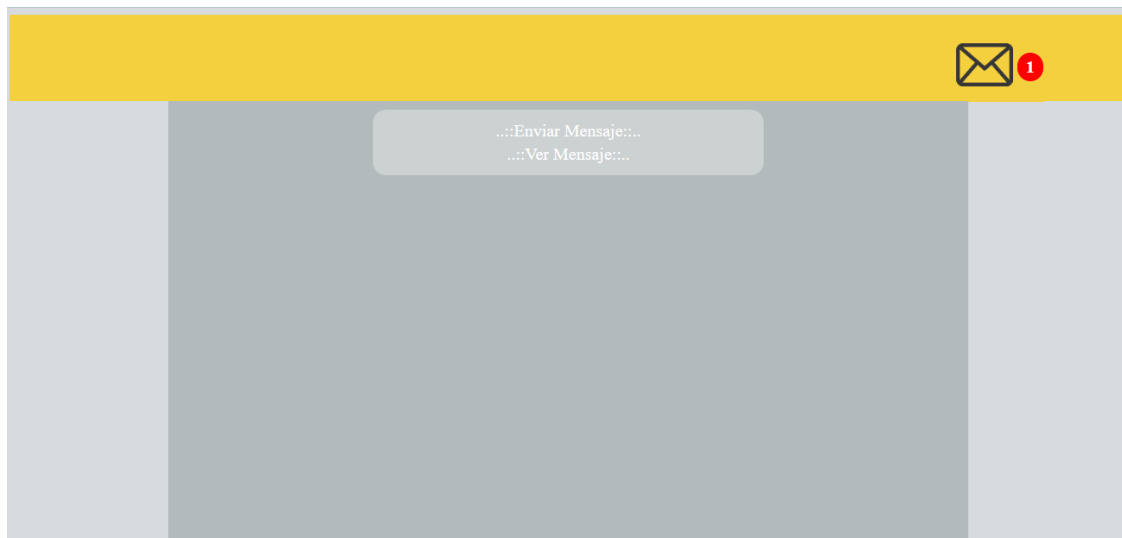
SI el usuario digita los campos y da clic en el botón “Enviar”, el mensaje se guardará en la tabla de la base de datos que se creó en el paso 1.

### 4. Ver Mensaje

ASUNTO	CONTENIDO	ESTADO
Hola	Mensaje de prueba	Ver
Envio1	Esto es un mensaje de prueba	Ver
Envio2	Esto es una nueva prueba	Ver
Prueba3	Esto es una tercera prueba	Ver

Al dar clic en la opción “Ver Mensaje” de la página inicial, se cargará una interfaz gráfica con el contenido de todos los mensajes enviados hasta el momento (consulta general a la tabla).

## 5. Notificaciones



Si se ha enviado un mensaje, al regresar a la interfaz principal ya debe aparecer la notificación, esta será una señal que sobresalga a la vista y genere una especie de reacción del usuario.

Para este caso es una imagen en color rojo con un número adentro.

El numero muestra la cantidad de mensajes pendientes por visualizar.

## Como Implementar Las Notificaciones

**A.** Lo primero es, al momento de enviar un mensaje, realizar un insert a la tabla de la BD con la siguiente



INSTRUCTIVO PARA LA EJECUCIÓN DE LA FPI

información:

Asunto: Titulo del mensaje (caja de texto).

Mensaje: Contenido del mensaje (área de texto).

Estado: Este estado se debe enviar internamente en el SELECT y debe ser un texto por defecto con la palabra “pendiente” como se observa en la imagen.

```
34 <?php
35 if (isset($_POST['btne'])) {
36     $asu=$_POST['txta'];
37     $men=$_POST['txtm'];
38     $est="Pendiente";
39
40     try{
41         $sql="INSERT INTO mensaje(id_men, asu_men, con_men, est_men) VALUES ('', :a,
42             :m, :e)";
43         $resultado=$base->prepare($sql);
44         $resultado->execute(array(":a"=>$asu, ":m"=>$men, ":e"=>$est));
45         <script language="javascript">window.alert("Mensaje enviado exitosamente!!!")</script>
46     }<?php
47     }catch(Exception $e){
48         die('Error: '.$e->getMessage());
49     }
50 }
```

En la base de datos se guardarán los mensajes de la siguiente forma.

+ Opciones					id_men	asu_men	con_men	est_men
<input type="checkbox"/>	Editar	Copiar	Borrar		1	Hola	Mensaje de prueba	Pendiente
<input type="checkbox"/>	Editar	Copiar	Borrar		2	Envio1	Esto es un mensaje de prueba	Pendiente
<input type="checkbox"/>	Editar	Copiar	Borrar		3	Envio2	Esto es una nueva prueba	Pendiente
<input type="checkbox"/>	Editar	Copiar	Borrar		4	Prueba3	Esto es una tercera prueba	Pendiente
↑ <input type="checkbox"/> Seleccionar todo Para los elementos que están marcados: Editar Copiar Borr								

**B.** La lógica de funcionamiento es la siguiente: SI una persona envía un mensaje nuevo, este se guarda en la BD con un estado “pendiente”, este estado es la **bandera** que me permitirá visualizar las notificaciones. Para eso se debe de programar una estructura IF en el lugar donde quiero que se vean las notificaciones.

**SI**, el mensaje tiene como estado pendiente, cuéntelo (contador), se sumará 1 a la variable contador cada vez que encuentre un mensaje con el estado “pendiente”.



INSTRUCTIVO PARA LA EJECUCIÓN DE LA FPI

```
C:\xampp\htdocs\notificacion\uno.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

uno.php
7 <body>
8 <?php
9 require('conec.php');
10 $contador=0;
11 try{
12     $sql="SELECT asu_men, con_men, est_men FROM mensaje";
13     $resultado=$base->prepare($sql);
14     $resultado->execute(array());
15     while($consulta=$resultado->fetch(PDO::FETCH_ASSOC)){
16         if ($consulta['est_men']=='Pendiente') {
17             $contador=$contador + 1;
18         }
19     }
20 catch(Exception $e){
21     die('Error: '.$e->getMessage());
22 }
23 }
24 <header>
25 <section class="cabecera">
26 <table class="tuno">
27 <tr>
28 <td></td>
```

Finalmente, para visualizar las notificaciones (en este caso se deberán ver en la parte superior del header), se debe crear una estructura IF en donde se preguntará por la variable contador, SI contador es mayor a 0 haga visible el contenido de la variable contador, así nos mostrar cuantos mensajes están pendientes por visualizar.

```
C:\xampp\htdocs\notificacion\uno.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

uno.php
24 <header>
25 <section class="cabecera">
26 <table class="tuno">
27 <tr>
28 <td></td>
29 <td>
30 <?php
31 if ($contador > 0) {
32 <?
33 <section class="notificacion">
34 <a href="ver2.php">
35 <?php
36 echo $contador;
37 <?
38 </a>
39 </section>
40 <?php
41 }
42 <?
43 </td>
44 </tr>
45 </table>
```

Por último, el efecto de botón rojo y número blanco se realiza con CSS.

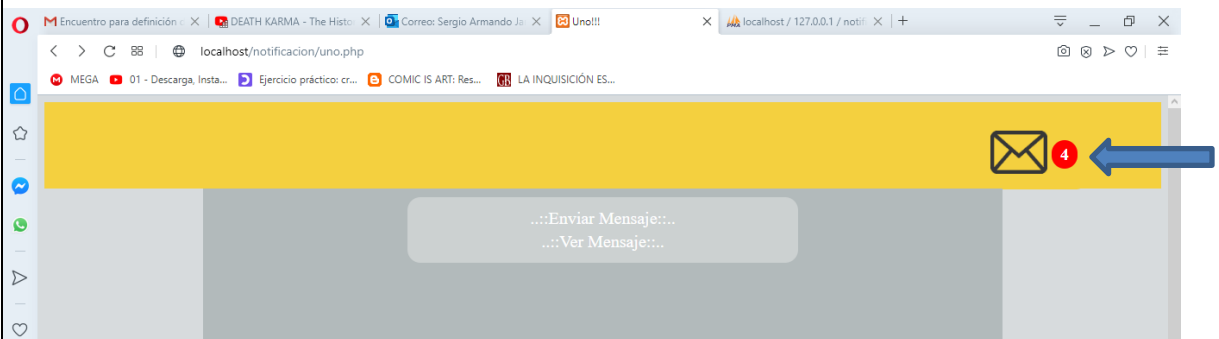




```
C:\xampp\htdocs\notificacion\estilo.css - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

uno.php x estilo.css x

52 .cabecera{
53     margin-left: 1000px;
54     padding-top: 20px;
55 }
56 .notificacion{
57     background-color: red;
58     width: 20px;
59     text-align: center;
60     border-radius: 20px;
61     padding: 5px;
62     color: white;
63     font-weight: bold;
64 }
65 .uno{
66     background-color: #F4D03F;
67     width: 30%;
68 }
```

Con la correcta implementación de los códigos, se logrará el siguiente resultado.



	<p align="center"><b>CENTRO DE GESTIÓN Y DESARROLLO SOSTENIBLE SURCOLOMBIANO</b></p> <p align="center"><b>INSTRUCTIVO PARA LA EJECUCIÓN DE LA FPI</b></p>	
--	---	--

**QUE DEBE TENER EL ENTREGABLE (EVIDENCIA).**

*Después de estudiar este material, usted deberá replicar este ejercicio y entregarlo con todas las funcionalidades que se trabajaron en este documento completamente operativas.*

**¡La práctica hace al maestro!**

<b>Listado de recursos, materiales, equipos, herramientas</b>	<ul style="list-style-type: none"> <li>- Computador, Tablet o smartphone (celular).</li> <li>- Conexión a internet o datos de celular.</li> <li>- XAMPP correctamente instalado.</li> <li>- Gestor de código fuente (el que usted prefiera).</li> <li>- Actitud positiva y propositiva.</li> </ul>							
<b>Tipo de Evidencia</b>	Desempeño		Conocimiento		Producto	<b>X</b>	No produce entrega de evidencia	
<b>Producto y Forma de Entrega</b>	<p><b>Entregable:</b> Archivos fuentes de generación de reportes.</p> <p><b>Forma de entrega:</b> Principalmente enlace correspondiente en TERRITORIUM, pero en caso de no ser posible, también puede usar WhatsApp, email misena, Gmail, u Outlook.</p>							
<b>Fecha de Inicio:</b>	30/11/2020							
<b>Fecha de Fin:</b>	30/11/2020							
<b>Intensidad /duración</b>	4 horas.							