

**INSTITUTO DE FORMACIÓN TÉCNICA
SUPERIOR N° 4**

FUNCIONALIDAD: NEWSLETTER

**Tecnicatura Superior en
Desarrollo de Software**

Desarrollo de Sistemas Web

**Profesora:
Maria del Carmen Acri**

**Grupo 2:
Sergio Ottati
Daniel Ons
Fabián Jerez**

Índice

1. Consigna	pág 1
2. Estructura de Archivos PHP/JS.....	pág 1
3. Base de Datos.....	pág 2
4. Base de datos que nos pasaron.....	pág 2
5. Pasos para enviar el newsletter.....	pág 3
6. Baja automática.....	pág 3
7. Requisitos.....	pág 4
8. Panel de gestión (usuarios.php).....	pág 4
9. Consideraciones.....	pág 4
10. Archivos, funcionalidad e integración.....	pág 5
11. Páginas de prueba.....	pág 5
12. Cargamos la base de datos con valores de prueba.....	pág 6
13. Creación de páginas web para un marco de referencia.....	pág 6
14. Botón de alta al newsletter.....	pág 7
15. Buscador por nombre o email.....	pág 7
16. Botón Ejecutar baja automática.....	pág 8

1- Consigna

#2 Base de estudiantes para envíos de mensajes

Usar (PHP CSS eventos JS)

1. Tres tipos de usuarios, estudiantes, profesores, administrativos
2. Estudiantes, solo puede realizar alta.
3. Profesores, sólo pueden listar estudiantes.
4. Administrativos, pueden enviar mail, y realizar una baja lógica de los estudiantes.
5. Luego de x tiempo (4 años), se produce una baja lógica.

2- Estructura de Archivos PHP/JS

ifts4/

```

|— panel.php
|— login.php
|— logout.php
|— newsletter/
|   |— cron_baja.php
|   |— dar_alta_newsletter.php
|   |— dar_baja_newsletter.php      # baja lógica (administrativo)
|   |— enviar_newsletter.php       # envío de mails (administrativo)
|   |— listado.php                 # listado según rol (profesor)
|   |— newsletter_unsuscribe.php
|   |— suscripcion.php             # formulario para estudiante
|   |— test_admin.php
|   |— test_profesor.php
|   |— test_suscripcion.php
|   |— usuarios.php
|   |— includes/
|       |— auth.php                #roles de usuario (sesión)
|       |— config.php
|       |— db.php                  #conexión con la BD
|       |— utils.php               # funciones comunes
|       |— PHPMailer/
|           |— PHPMailer.php
|           |— SMTP.php
|           |— Exception.php

```

2- Base de Datos

Usaremos:

1. id_usuario int(11) NOT NULL
2. nombre varchar(100) NOT NULL
3. apellido varchar(100) NOT NULL
4. email varchar(150) NOT NULL
5. activo tinyint(1) DEFAULT 1 #estado activo/inactivo
6. fecha_suscripcion datetime NOT NULL DEFAULT current_timestamp()
7. unsubscribe_token VARCHAR(64) DEFAULT NULL

3- Base de datos que nos pasaron

```
CREATE TABLE usuarios (
  id_usuario int(11) NOT NULL,
  nombre varchar(100) NOT NULL,
  apellido varchar(100) NOT NULL,
  email varchar(150) NOT NULL,
  password varchar(255) NOT NULL,
  rol enum('estudiante','profesor','administrativo') NOT NULL DEFAULT 'estudiante',
  activo tinyint(1) DEFAULT 1,
  estado_aprobacion enum('pendiente','aprobado','rechazado') DEFAULT 'aprobado',
  reset_token varchar(64) DEFAULT NULL,
  reset_expira bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

4- Agregar campos a la tabla usuarios.sql

1. Newsletter

```
ALTER TABLE usuarios ADD newsletter TINYINT(1) DEFAULT 0;
```

newsletter = 1 → está suscripto

newsletter = 0 → no está suscripto

2. Fecha de suscripción

```
ALTER TABLE usuarios ADD fecha_suscripcion DATETIME DEFAULT NULL
```

3. Token único

Funcionamiento: Cuando se presiona el botón en el mail (desuscribirse) este me lleva a una página que procesa la baja lógica (newsletter=0)

```
ALTER TABLE usuarios ADD unsubscribe_token VARCHAR(64) DEFAULT  
NULL;
```

5- Pasos para enviar el newsletter

`enviar_newsletter.php`

ADMIN entra → se verifica si alguien cumplió 4 años → se desuscribe automáticamente



Envía mensaje desde el formulario



PHPMailer configura SMTP y arma el mail



Se agregan todos los destinatarios activos y suscriptos



Se envía el mail (y se informa si tuvo éxito o no)

6- Baja automática

Se ejecuta en:

- `enviar_newsletter.php` → antes de enviar el correo.
- `listado.php` (o cualquier panel de administración) → cuando se entra al sistema.

Esto asegura que cada vez que se usa el sistema de forma real, se actualizan automáticamente los suscriptores vencidos (más de 4 años).

Aclaración:

`cron_baja.php` solo sirve para testeo, es para controlar en la actualidad si funciona correctamente modificando uno de los datos de suscripción de la base de datos de forma manual y probando el php para que actualice y nos devuelve la siguiente información:

Los usuarios que fueron dados de baja (nombre, apellido, fecha de suscripción) eso en forma de lista y como es un **array** nos muestra a través del **count** la cantidad de usuarios que fueron dados de baja del newsletter.

7- Requisitos

Al momento de loguearse debe guardar las siguientes variables globales

```
$_SESSION['id_usuario'] = $usuario['id_usuario'];
$_SESSION['rol'] = $usuario['rol'];
$_SESSION['nombre'] = $usuario['nombre'];
```

Esto asegura que todas las funciones de **auth.php** puedan funcionar.

8- Panel de gestión (**usuarios.php**)

Tiene diferentes **funcionalidades** según el **rol** del usuario que accede.

Rol	Funcionalidad
Administrativo	<ol style="list-style-type: none"> 1. Ver todos los usuarios (activos/inactivos) 2. dar bajas lógicas 3. ver suscripciones
Profesor	<ol style="list-style-type: none"> 1. Ver listado de estudiantes suscriptos
Estudiante	<ol style="list-style-type: none"> 1. nada, aunque debería poder darse de baja (no es lo recomendable dado que se brinda información relevante del IFTS 4)

9- Consideraciones

Consideramos que el nombre de la base de datos es "**ifts4**". De lo que estamos seguros es que la tabla se llama **usuarios.sql**

10- Archivos, funcionalidad e integración

suscripcion.php

1. Verifica que el usuario esté logueado y que sea estudiante.
2. Consulta si ya estás suscripto.
3. Si no lo está y envía el formulario, lo suscribe y guarda la fecha actual.
4. Si ya está suscripto, muestra un mensaje y no permite volver a suscribirse.

¿Cómo integrarlo al sistema (página web del IFTS 4)?

Podés poner un botón o enlace en el panel del estudiante que diga:

`Suscribirme al newsletter`

11- Páginas de prueba

Objetivos: Validar que el newsletter funciona correctamente.

1. Suscripción
2. Envío por administrador
3. Visualización de estudiantes
4. Baja automática (4 años)
5. Desuscripción vía e-mail.

Archivo	Propósito
<code>test_suscripcion.php</code>	Simular acceso del estudiante al botón de suscripción
<code>test_admin.php</code>	Simular acceso administrativo: ver usuarios + enviar newsletter
<code>test_profesor.php</code>	Simular vista del profesor para listar estudiantes suscriptos
<code>test_login.php</code>	Forzar sesión de prueba para estudiante / admin / profesor (sin login.php)

¿Cómo Testear?

1. Entrar a [test_suscripcion.php](#) como estudiante → presionar suscribirse.
2. Ir a [test_admin.php](#) y enviar el newsletter → confirmar que llega el correo al estudiante.
3. Hacer clic en "desuscribirse" desde el correo → confirmar que se da de baja revisando la base de datos.
4. Volvés a [test_admin.php](#) y corrés [cron_baja.php](#) si querés probar la lógica automática.

12- Cargamos la base de datos con valores de prueba

Se crearon los siguientes valores para poblar la tabla **usuarios** con:

- 10 estudiantes
- 2 profesores
- 1 administrativo

“Pocos valores bien elegidos” Luis Perez

Todos los usuarios tienen una **contraseña** simple (1234). En la página real el login utiliza el **hash** de la **contraseña**.

13- Creación de páginas web para un marco de referencia

Estas páginas serán reemplazadas luego por las páginas originales del IFTS

4. Las utilizaremos para testear la funcionalidad del newsletter, una vez que funcione correctamente se procede a comentar el código para que no influyan en el correcto funcionamiento de la página oficial.

Creamos las siguientes las siguientes páginas:

1. [login.php](#)
2. [logout.php](#)
3. [panel.php](#) (punto de entrada al sistema de pruebas)
4. [registro.php](#)

Objetivo

- A. Generar usuarios de Prueba a través de [registro.php](#), es útil para poder enviar emails reales y probar que el [PHPMailer](#) funcione correctamente.
- B. Permitir el logueo como estudiante, profesor o administrativo desde un [login.php](#).
- C. Guardar el **id_usuario**, **rol**, y **nombre** en la **sesión**.
- D. Salir con [logout.php](#).
- E. Estas páginas sólo se usarán mientras desarrollamos y probamos la funcionalidad del newsletter, luego se comentan y se utilizan los botones reales de la página del IFTTS 4 que enlazan a las páginas del newsletter, es por eso que están fuera de la carpeta newsletter, pensando siempre en la modularidad del proyecto.

Flujo de prueba sugerido

1. Elegir un usuario existente en la tabla [usuarios](#) o crear uno propio usando [registro.php](#)
2. Entrar por [login.php](#)
3. Según el rol, se redirige a [panel.php](#) con los accesos correctos.
4. Desde ahí podemos navegar a todas las funcionalidades.
5. Salir con [logout.php](#).

14- Botón de alta al newsletter

Objetivo:

En la sección Administrador podrá dar de alta a un usuario en particular. Es muy útil en caso en caso de que se diera de baja manual por error y volver a darlo de alta, sobre todo en tareas de mantenimiento y control de los alumnos suscriptos al newsletter.

La misma función se encuentra en [dar_alta_newsletter.php](#)

15- Buscador por nombre o email

El administrador podrá:

1. Elegir el criterio de búsqueda ([nombre completo](#) o [email](#))
2. Ver resultados correctos según ese filtro
3. Limpiar fácilmente la búsqueda

16- Botón Ejecutar baja automática

Objetivo:

Permite al administrador ejecutar la baja lógica de los alumnos que tengan más de 4 años desde la fecha de suscripción al newsletter.