

Fullstack - TFT

Lista de Regalos

Óscar Pérez Riaño
Héctor Rodríguez Iglesias
Raquel Rodríguez Hernández



ÍNDICE

1	Objetivos funcionales y definición de la aplicación	2
1.1	Estructura de pantallas y navegabilidad	2
2	Preparación del entorno	4
2.1	Repositorio	4
2.2	Instalación	4
2.3	Diseño de BBDD	8
3	Desarrollo con PHP SLIM	10
3.1	Versiones y entorno de trabajo	10
3.1.1	Framework Slim y versiones utilizadas	10
3.1.2	Estructura del proyecto desarrollado y entorno de trabajo	11
	El IDE utilizado para el desarrollo de la aplicación ha sido Visual Studio Code que es un editor ligero, potente, multiplataforma y open source. Además, es un cliente para Git y como se ha utilizado github como repositorio esto ha facilitado el desarrollo de la aplicación.	13
4	Funcionamiento de la aplicación	15
4.1	Pantalla de Login	15
4.2	Pantalla de Registro	16
4.3	Pantalla de gestión de listas del usuario	19
4.4	Ver detalle de la lista	22
4.5	Pantalla de alta de producto	27
5	Referencias y Software	29

1 Objetivos funcionales y definición de la aplicación

El objetivo de esta aplicación es que un usuario pueda crear listas de regalos para cualquier tipo de evento y poder compartir esas listas con familiares y/o amigos para que ellos puedan elegir qué regalo comprar. Además de esta manera se llevará un control de los regalos que han sido comprados porque se podrán ir marcando como tal.

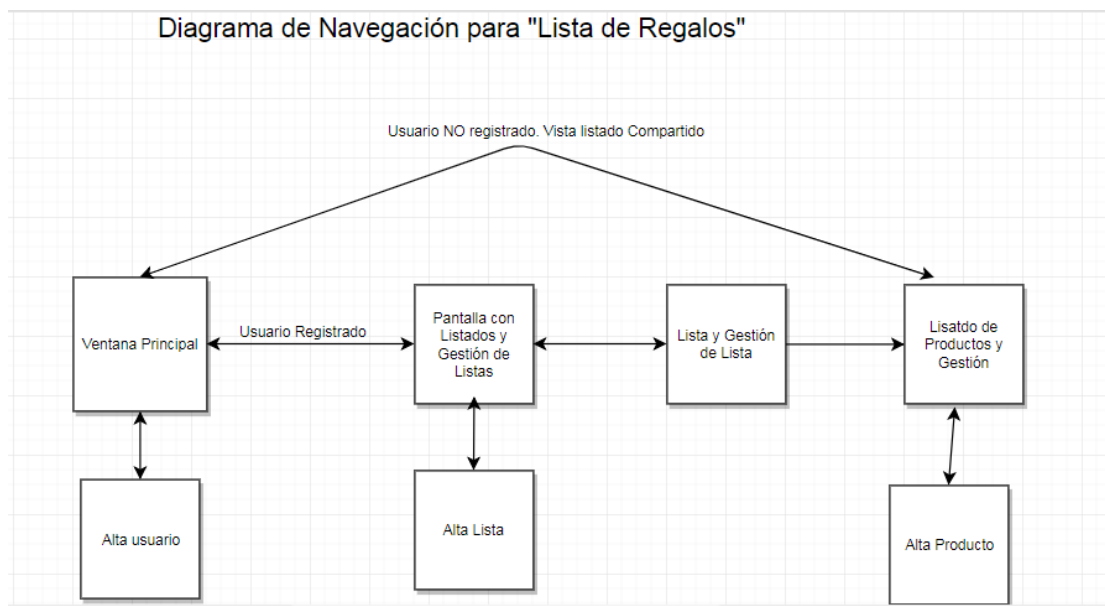
Un usuario tendrá que estar registrado en la aplicación para crear/gestionar sus listas pero no es necesario que lo estén las personas con las que la compartirá.

Como requisitos se han definido los siguientes:

- Un usuario registrado podrá tener varias listas
- Las listas podrán tener tantos productos como se deseen.
- Los productos estarán identificados por nombre, enlace e imagen descriptiva
- Para cada lista, un usuario registrado podrá:
 - añadir productos con enlaces, imágenes y texto
 - marcar los artículos como comprados o pendientes
- Un usuario registrado podrá compartir su lista con cualquier persona a la que se le envíe y este será notificado mediante correo electrónico.
- Un usuario que no esté registrado podrá acceder a la lista a través del enlace que se le comparta.

1.1 Estructura de pantallas y navegabilidad

- 1) Página de alta de usuario.
- 2) Página de login.
- 3) Página con formulario de creación de lista.
- 4) Página con formulario de creación de artículo que se añadirá a la lista.
- 5) Página con el detalle de la lista que muestra el listado de artículos.
- 6) Página con un listado, no editable, sólo para visualizar, que será el que el usuario pueda 'compartir' con otros.
- 7) Página con el detalle de un artículo.



2. Preparación del entorno

2.1 Repositorio

Todo el código de la aplicación se encuentra disponible de forma pública para su descarga en el siguiente repositorio:

<https://github.com/hectorri/fullstack-tft>

2.2 Instalación

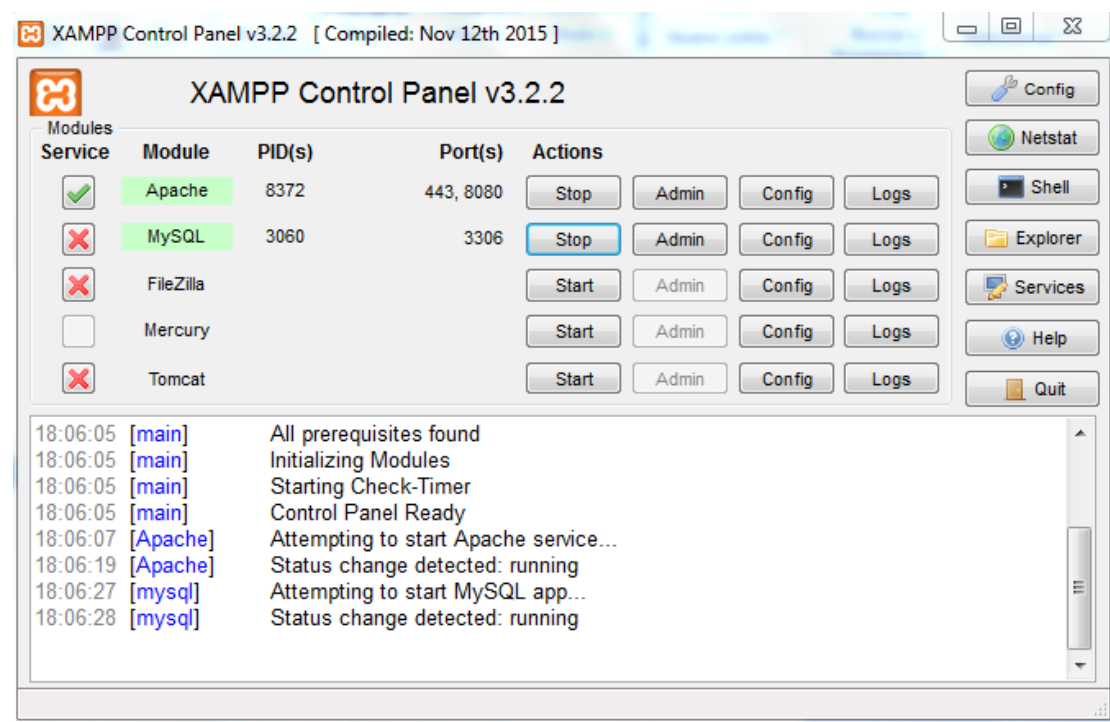
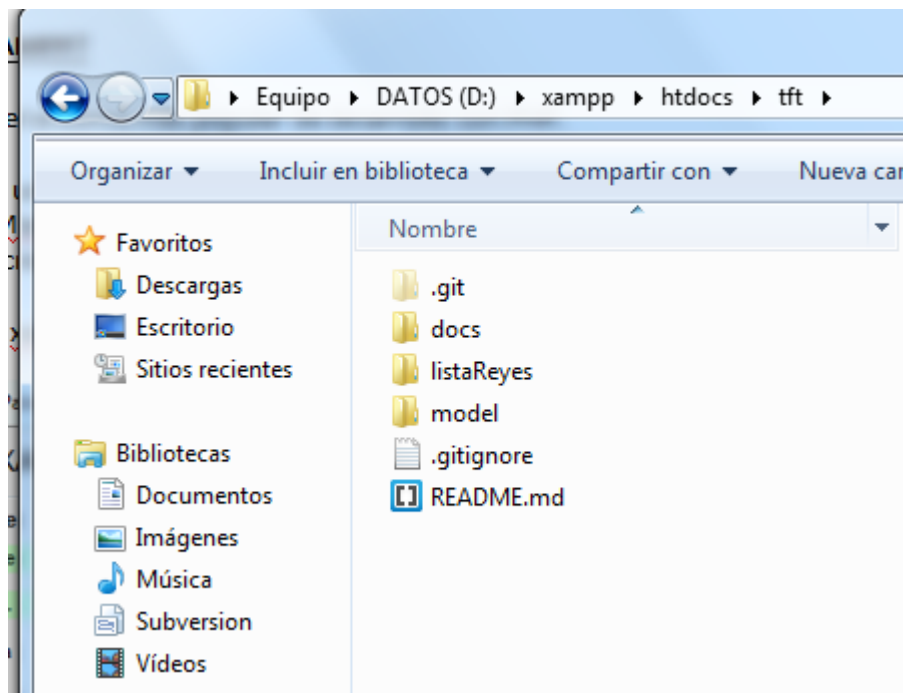
Para poder desplegar nuestra aplicación es necesario arrancar un servidor Apache y el servicio MySQL. Nosotros hemos utilizado Xampp para gestionar todos los servicios desde su panel de control.

¿Qué es XAMPP?

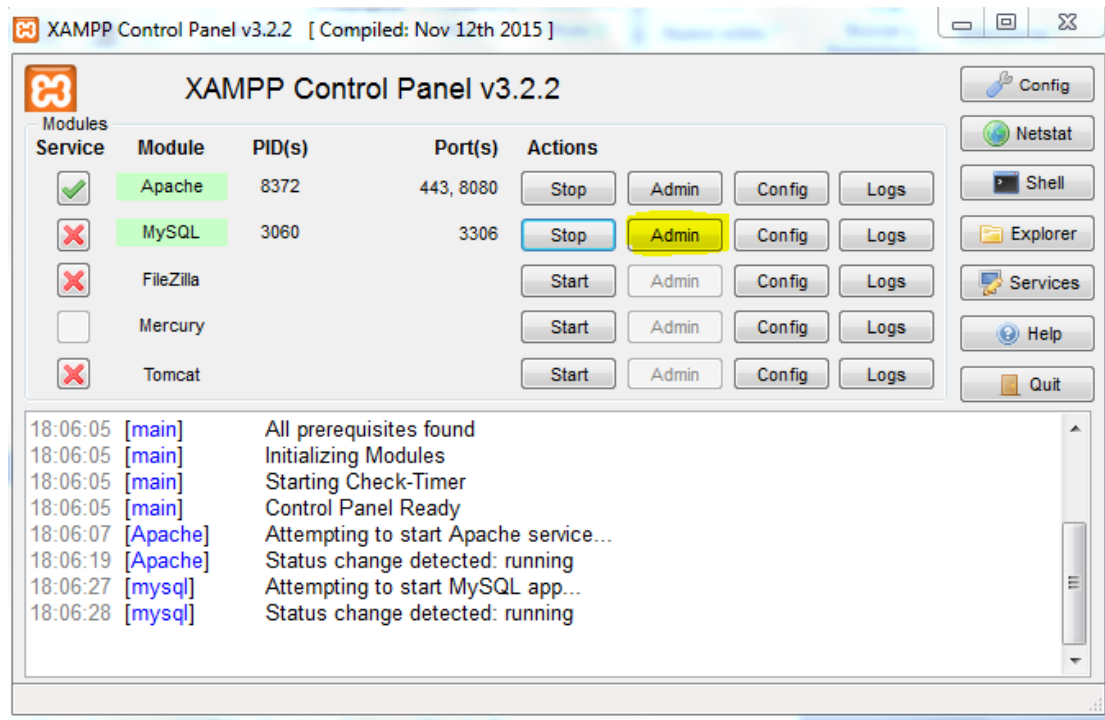
XAMPP es el entorno más popular de desarrollo con PHP.

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

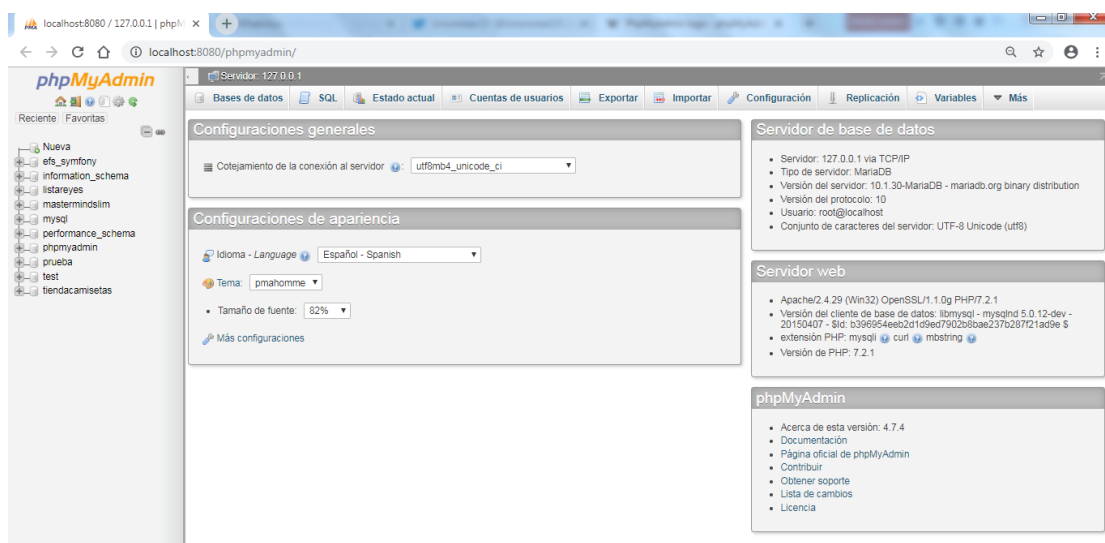
El proyecto *listaReyes* se encuentra dentro de la carpeta *htdocs* de **xampp**. De esta manera cuando arranquemos Apache nuestra aplicación estará accesible.



Una vez arrancado los servicios se debe abrir el administrador de base datos. Para ello se pulsa sobre el botón Admin (marcado en amarillo en la imagen).



Con ello abriremos *phpMyAdmin* que es una herramienta desarrollada en PHP con la finalidad de administrar nuestras bases de datos MySQL. Actualmente 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...






Estas son las versiones de servidor web y base de datos con la que la aplicación ha sido desarrollada y probada.

Servidor de base de datos

- Servidor: 127.0.0.1 via TCP/IP
- Tipo de servidor: MariaDB
- Versión del servidor: 10.1.30-MariaDB - mariadb.org binary distribution
- Versión del protocolo: 10
- Usuario: root@localhost
- Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)

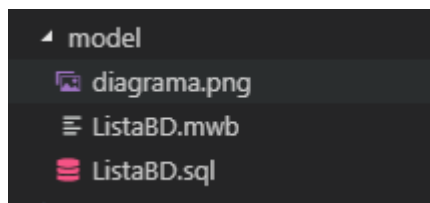
Servidor web

- Apache/2.4.29 (Win32) OpenSSL/1.1.0g PHP/7.2.1
- Versión del cliente de base de datos: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: b396954eeb2d1d9ed7902b8bae237b287f21ad9e \$
- extensión PHP: mysqli  curl  mbstring 
- Versión de PHP: 7.2.1

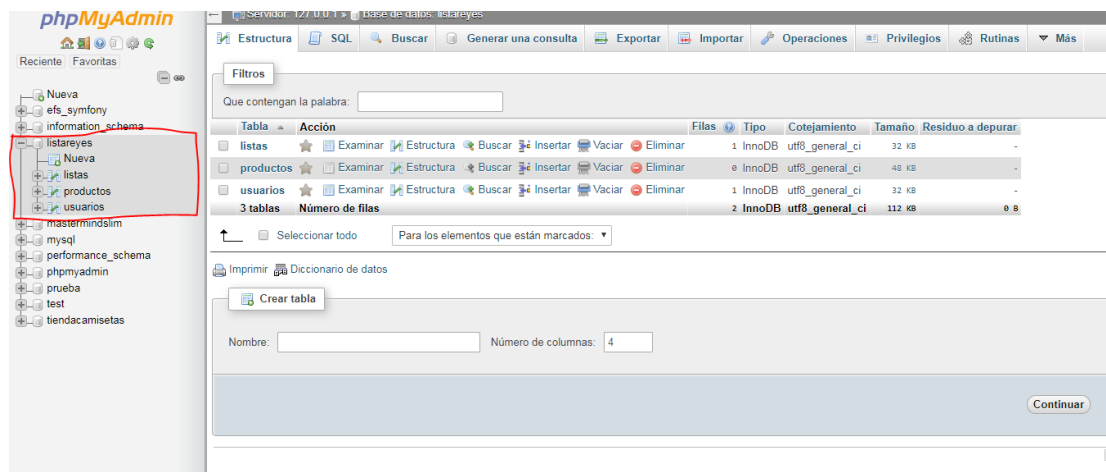
La url de acceso web al panel phpMyAdmin, en un entorno 'por defecto' sería la siguiente.

<http://localhost:8080/phpmyadmin/>

Hace falta disponer de la base de datos de la aplicación y para ello es necesario ejecutar el script de creación ListaBD.sql que se encuentra dentro del proyecto en la carpeta tft\model.



Una vez ejecutado el script la base de datos creada sería la siguiente.



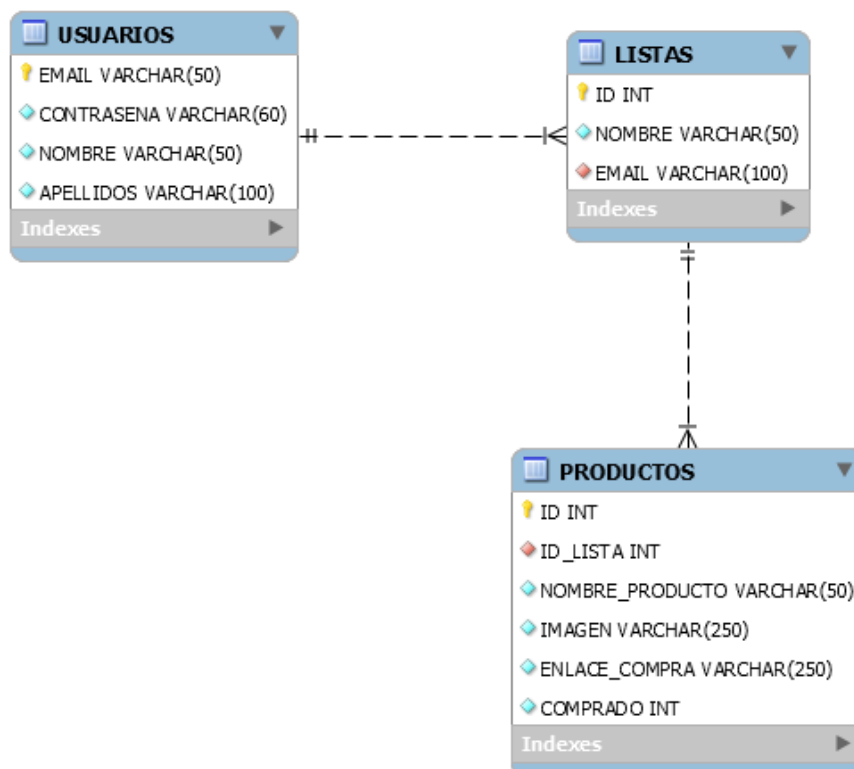
2.3 Diseño de BBDD

El diseño de la BBDD es bastante simple puesto que no es necesario mucha información para controlar y gestionar los datos de la aplicación.

Tendremos 3 tablas y con las relaciones que se muestran en el diagrama podremos mantener la información para la aplicación.

- USUARIOS
 - Email
 - Nombre
 - Apellidos
 - Contraseña
- LISTAS
 - Nombre Lista

- Email Usuario
- PRODUCTOS
 - Nombre Producto
 - Imagen
 - Enlace
 - Check Comprado/Pendiente



3. Desarrollo con PHP SLIM

Para desarrollar la aplicación hemos utilizado el lenguaje PHP y uno de sus framework denominado SLIM.

3.1 Versiones y entorno de trabajo

3.1.1 Framework Slim y versiones utilizadas

SLIM es un microframework que nos permite crear aplicaciones y Apis de forma muy rápida y sencilla bajo el lenguaje PHP y es bastante práctico y poderoso a la hora de implementar. El framework ayuda en la creación de las aplicaciones y su comunicación con el cliente, es decir:

- Validación de tipos de petición (GET, POST, INPUT, DELETE ..)
- Middleware: para interceptar el antes y después de cada petición.
- Mapeo de rutas bastante flexible

Este framework deja total libertad respecto de la gestión de la persistencia de datos además de permitir añadir todos los requerimientos necesarios para desarrollar una aplicación. En nuestro caso vamos a destacar el uso de algunos de ellos.

- Versión de PHP --> 7.2.1
- Versión del Framework SLIM 3.1 con los siguientes módulos:
 - **"illuminate/database"**: ~5.1: Para la integración con la base de datos
 - **"respect/validation"**: "^1.1": Utilizado para las validaciones de datos
 - **"slim/twig-view"**: "^2.4": Utilizado en la generación de plantillas para las vistas

Todas estas dependencias se definen en un fichero json en el proyecto (*composer.json*) que nos permite, a través del gestor de paquetes composer (<https://getcomposer.org/>) manejar de una forma sencilla todas las dependencias del proyecto.

La aplicación entregada no dispone de las librerías descargadas, a través del comando *composer update* en la ruta de la aplicación este descarga e instala todas las dependencias del proyecto (carpeta vendor).

3.1.2 Estructura del proyecto desarrollado y entorno de trabajo

El código se ha estructurado de la siguiente manera.

Existe una carpeta, **src**, donde se encuentra el código principal php y que contendrá a su vez dos carpetas:

- **Controladores.** En esta carpeta se incluyen los controladores de las diferentes entidades que se gestionan en la aplicación. En ellos se definen por ejemplo la lógica que se realiza sobre los datos introducidos en los formularios y las redirecciones tras estas acciones. Es también en el controlador donde se establece la relación entre las plantillas de los formularios y la estructura de navegación definida.
 - **ControladorLista.php:** Contiene los métodos con las acciones sobre la listas
 - ◆ Nueva lista: `public function nueva($request, $response, $args)`
 - ◆ Eliminar Lista: `public function eliminar($request, $response, $args)`
 - ◆ Mis Listas: `public function misListas($request, $response, $args)`
 - ◆ Compartir: `public function compartir($request, $response, $args)`
 - **ControladorProducto.php:** Contiene los métodos con las acciones sobre los productos.
 - ◆ Nuevos Producto: `public function nuevo($request, $response, $args)`
 - ◆ Guardar Imagen: La imagen se guardará en la carpeta `\tft\listaReyes\public\imagenes` y en bbdd se almacenará únicamente la ruta y el nombre del fichero.

```
public static function guardarImagen(UploadedFile $imagen
```
 - ◆ Eliminar: `public function eliminar($request, $response, $args)`
 - ◆ Listar Productos: `public function listarProductos($request, $response, $args)`
 - ◆ Cambiar Estado: `public function cambiarEstado($request, $response, $args)`
 - ◆ Editar: `public function editar($request, $response, $args)`
 - **ControladorUsuario.php:** Aquí estarán los métodos para todas las acciones sobre los usuarios.
 - ◆ Registro: `public function registro($request, $response, $args)`
 - ◆ Login: `public function login($request, $response, $args)`
 - ◆ Logout: `public function logout($request, $response, $args)`

Una parte importante y que es gestionada por el framework es la existencia de un fichero denominado *routes.php* donde se definen las rutas que son usadas en el controlador y que con ellas se establece la lógica de navegación.

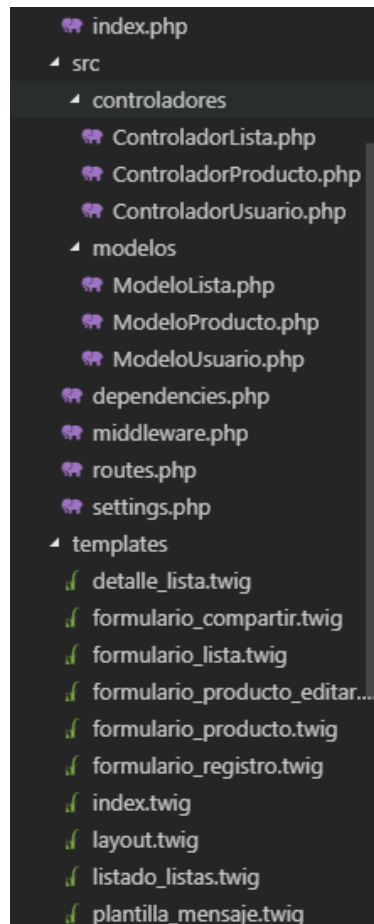
Algunas partes del fichero en las que se puede ver las rutas para el registro de usuario o para acceder a listado de listas. Aquí se puede el uso de los GET y los POST.

```
// Ruta para cargar el formulario para crear usuario
```

```
$app->get('/registro', function($request, $response, $args){  
    return $this->view->render($response, 'formulario_registro.twig');  
})->setName('usuario.registro');  
  
// ruta para crear un nuevo usuario  
$app->post('/registro', 'ControladorUsuario:registro');
```

```
// ruta para cargar el formulario para crear una lista  
$app->get('/nuevaLista', function($request, $response, $args){  
    return $this->view->render($response, 'formulario_lista.twig');  
})->setName('lista.nueva');  
  
// ruta para crear una nueva lista  
$app->post('/nuevaLista', 'ControladorLista:nueva');
```

- **Modelos:** Aquí estará el código que relaciona las tablas de la bbdd con los elementos de la aplicación
 - ModeloLista.php
 - ModeloProducto.php
 - ModeloUsuario.ph

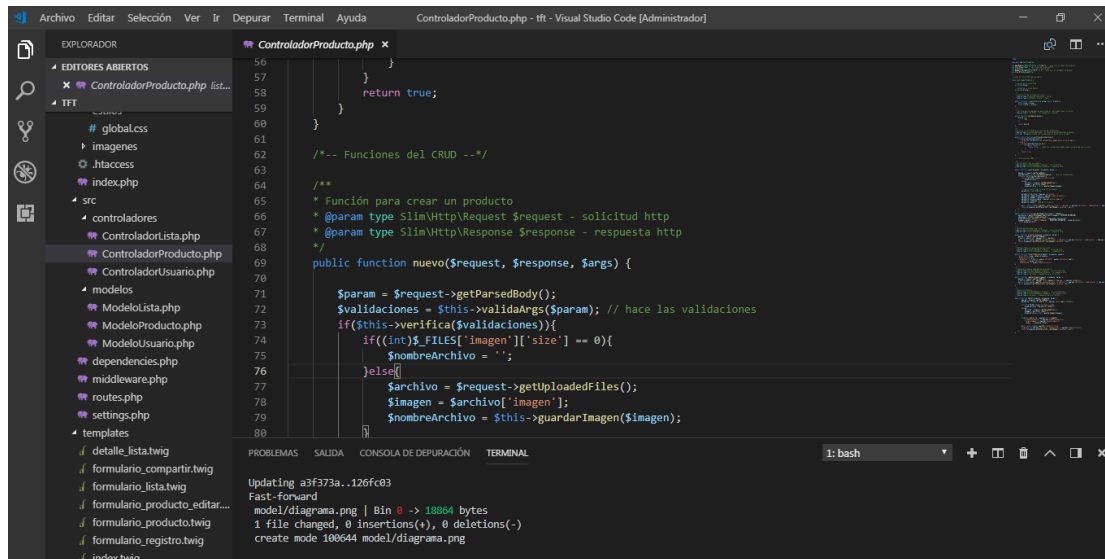


Además otra carpeta importante es **templates** donde se han definido las plantillas de los formularios de la aplicación. Estas plantillas están escritas en *Twig* que es el motor de plantillas elegido para el desarrollo de la aplicación. Cada una de ellas tiene definido el diseño de la página/formulario que se mostrará en la aplicación. El nombre es lo suficientemente intuitivo como saber qué contiene cada una de ellas. A continuación, se enumeran todas las plantillas utilizadas:

- *detalle_lista.twig* Detalle de una lista con el listado de artículos
- *formulario_compartir.twig* Formulario para compartir una lista
- *formulario_lista.twig* Formulario para la creación de una lista
- *formulario_producto_editar.twig* Formulario de edición de producto
- *formulario_producto.twig* Formulario de creación de producto
- *formulario_registro.twig* Formulario de registro de usuario
- *index.twig* Página de inicio
- *layout.twig* Plantilla común heredada por el resto
- *listado_listas.twig* Plantilla utilizada para los listados
- *plantilla_mensaje.twig* Plantilla genérica creada para mostrar mensajes

El IDE utilizado para el desarrollo de la aplicación ha sido **Visual Studio Code** que es un editor ligero, potente, multiplataforma y open source. Además, es un cliente para Git y como se ha utilizado github como repositorio esto ha facilitado el desarrollo de la aplicación.

Visual Studio Code se puede extender a través de complementos, disponible a través de un repositorio central lo que hace que trabajar con cualquier tipo de lenguaje sea muy cómodo.



4. Funcionamiento de la aplicación

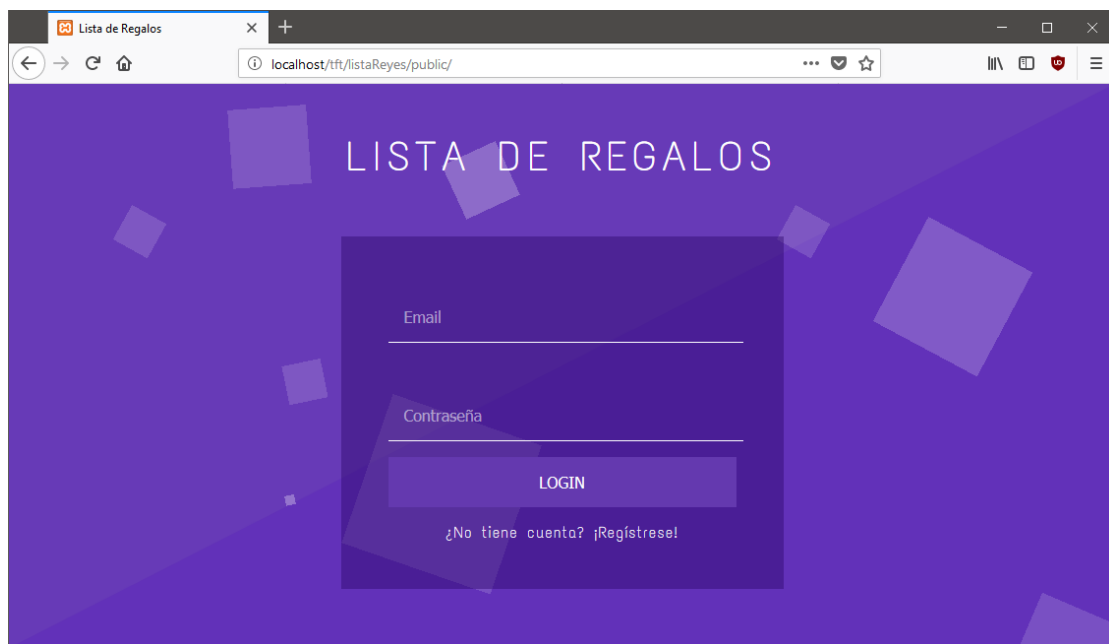
Para arrancar la aplicación se debe haber levantado el servidor Apache y el MySQL como se ha indicado en el apartado anterior. Una vez hecho esto se podrá acceder a la aplicación en la siguiente ruta.

`http://localhost:8080/tft/listaReyes/public`

A continuación, se irán mostrando cada una de las pantallas definidas y se irá explicando su funcionalidad.

4.1 Pantalla de Login

Esta es la pantalla de inicio de la aplicación “Lista de Reyes”. En ella se permite el registro o el inicio de sesión.



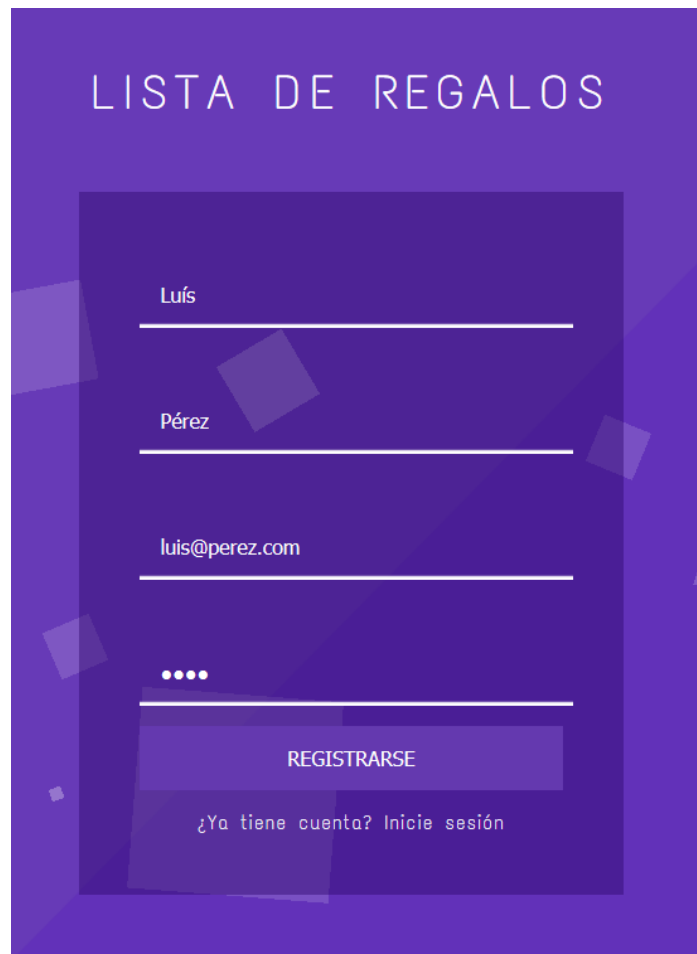
4.2 Pantalla de Registro

En esta pantalla se mostrará el formulario de registro de un nuevo usuario. Sólo será necesario un nombre, apellidos, email (que será usado como identificador) y la contraseña.

The image shows a registration form titled "LISTA DE REGALOS" on a purple background. The form is centered and contains the following elements:


- Nombre**: A text input field.
- Apellidos**: A text input field.
- Email**: A text input field.
- Contraseña**: A text input field.
- REGISTRARSE**: A button to submit the registration form.
- ¿Ya tiene cuenta? Inicie sesión**: A link to the login page.

Cuando se introducen los datos solicitados y se procede el registro se mostrará la pantalla de alta correcta y la opción de iniciar la sesión desde la misma ventana.



The image shows a registration form on a purple background with the title "LISTA DE REGALOS". The form is contained within a dark purple box and includes the following elements:

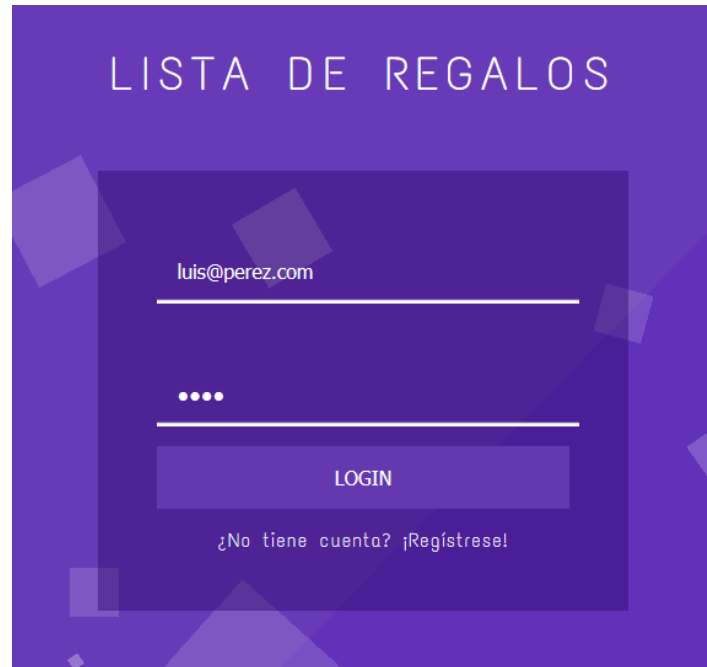
- A text input field containing "Luís".
- A text input field containing "Pérez".
- A text input field containing "luis@perez.com".
- A password input field represented by four dots.
- A button labeled "REGISTRARSE".
- A link labeled "¿Ya tiene cuenta? Inicie sesión".



The image shows the success message and login button on the same purple background with the title "LISTA DE REGALOS". The message is contained within a dark purple box and includes the following elements:

- A message: "Registro realizado correctamente".
- A button labeled "Iniciar sesión".

El siguiente paso sería iniciar la sesión y para ello se debe hacer login en la pantalla que se ha mostrado anteriormente. Cuando se han introducido los datos estaremos dentro de la aplicación.



4.3 Pantalla de gestión de listas del usuario

Una vez identificados veremos una pantalla en la que se presentarán las siguientes opciones al usuario:

- Mis Listas: Permite consultar las listas que del usuario
- Nueva Lista: Permite al usuario crear una nueva lista
- Salir: Se cerrará la sesión del usuario identificado



4.3.1 Creación de nueva Lista

En este formulario de creación de lista basta con introducir el nombre de la lista que se quiere dar de alta. Una vez creada la lista estará disponible para ir añadiendo los productos.



LISTA DE REGALOS

Lista de navidad

CREAR LISTA

VOLVER

The image shows a web form for creating a new gift list. It has a purple background. At the top, the text 'LISTA DE REGALOS' is displayed in white. Below this, there is a dark purple rectangular area containing a text input field with the placeholder text 'Lista de navidad'. Under the input field are two buttons: 'CREAR LISTA' and 'VOLVER', both in white text on a dark purple background.

4.3.2 Ver Listado de Listas



Si lo que se selecciona es la opción de Mis Listas se mostrará en pantalla un listado de las listas que ha creado el usuario.

Sobre esas listas podrá:

- Ver el contenido
- Eliminarla
- Compartirla

Si se elige “Ver el contenido” se mostrará el detalle de la lista con todos los productos añadidos. Si lo que se quiere es compartirla se enviará un mail al usuario indicado en la pantalla de envío y dicho usuario recibirá un correo con el enlace de la lista recibirá un enlace con la lista.

4.4 Ver detalle de la lista



Se mostrará el listado de todos los productos de la lista y para cada elemento de la lista se pueden realizar diferentes acciones.

- Marcar como comprado
- Eliminar
- Editar
- Abrir Enlace

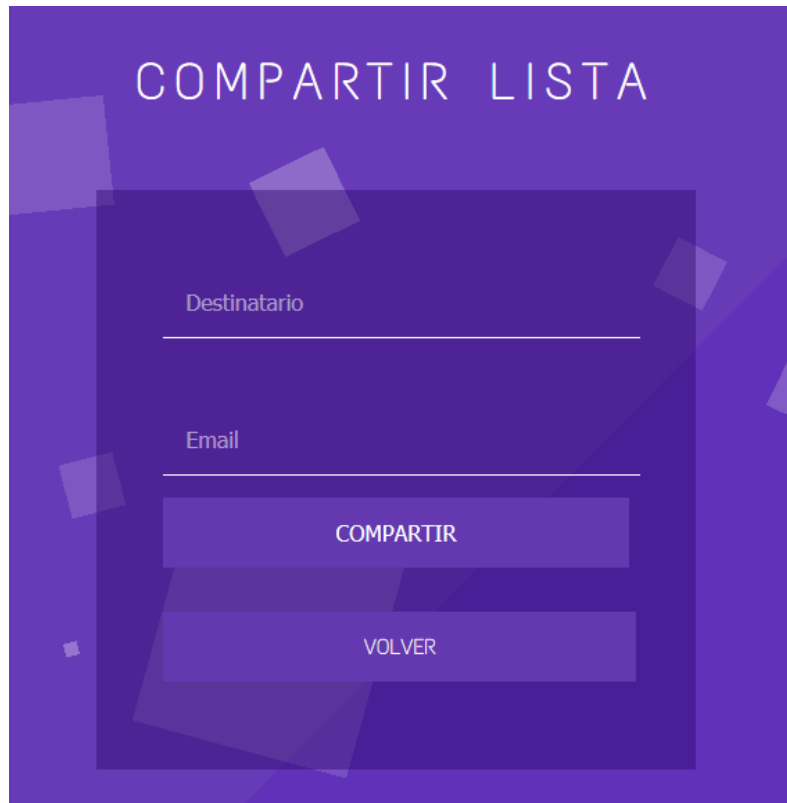
Un producto podrá eliminarse lo que hará que el producto deje de estar presente en la lista y editarse para cambiar cualquier información. Además, si se pulsa sobre el botón enlace se abrirá en otra pestaña el enlace asociado al producto.

Un producto se puede marcar como comprado con el Botón Comprado. Esto hace que el regalo deje de estar “Pendiente” y se indicaría en la parte de la derecha. Se puede observar cómo está marcado como “comprado”



4.4.1 Compartir Lista con Usuario

Con el botón “Compartir Lista” se mostrará un formulario donde se debe introducir el correo electrónico de la persona a la que enviar la lista. Este formulario enviará la url de la lista al usuario indicado y dicho usuario tendrá acceso a la lista.



COMPARTIR LISTA

Destinatario

Email

COMPARTIR

VOLVER

Este es el código para compartir la lista y el contenido del email que se generaría y se enviaría al destinatario,

```
public function compartir($request, $response, $args){
    $param = $request->getParsedBody();
    $para    = $param['email'];
    $titulo  = $_SESSION['nombre'].' ha compartido una lista contigo';
    $mensaje = 'Puede visitar la lista compartida en el siguiente enlace: ';
    $cabeceras = 'From: listaRegalos@listaRegalos.com' . "\r\n" .
        'Reply-To: '.$_SESSION['email'] . "\r\n" .
        'X-Mailer: PHP/' . phpversion();

    ini_set("SMTP","smtp.gmail.com");
    //mail($para, $titulo, $mensaje, $cabeceras);
}
```

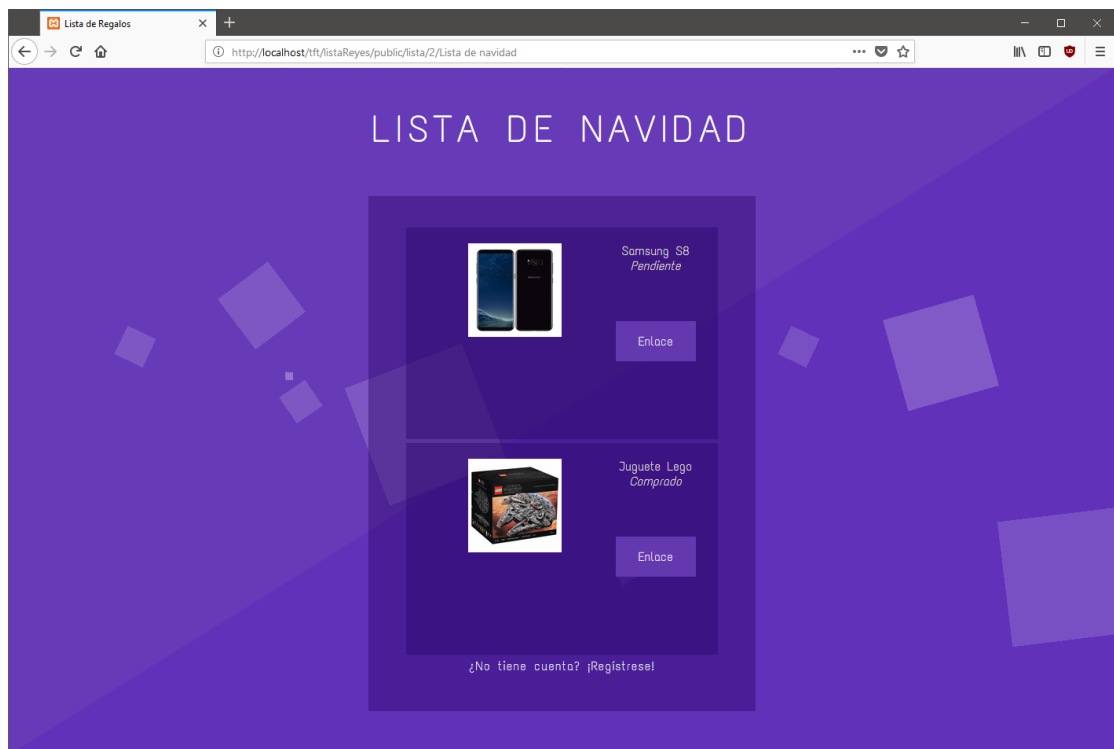
Y así es como se verá en pantalla ya que para enviar por correo hay que instalar un servidor smtp o configurar el acceso a un servidor de correo con nombre, contraseña y como no tenemos un hosting para desplegar la aplicación quedaría preparado.



4.4.2 Acceso a lista compartida

El usuario verá un listado, no editable, sólo para visualizar, que será el que el usuario ha compartido con él.

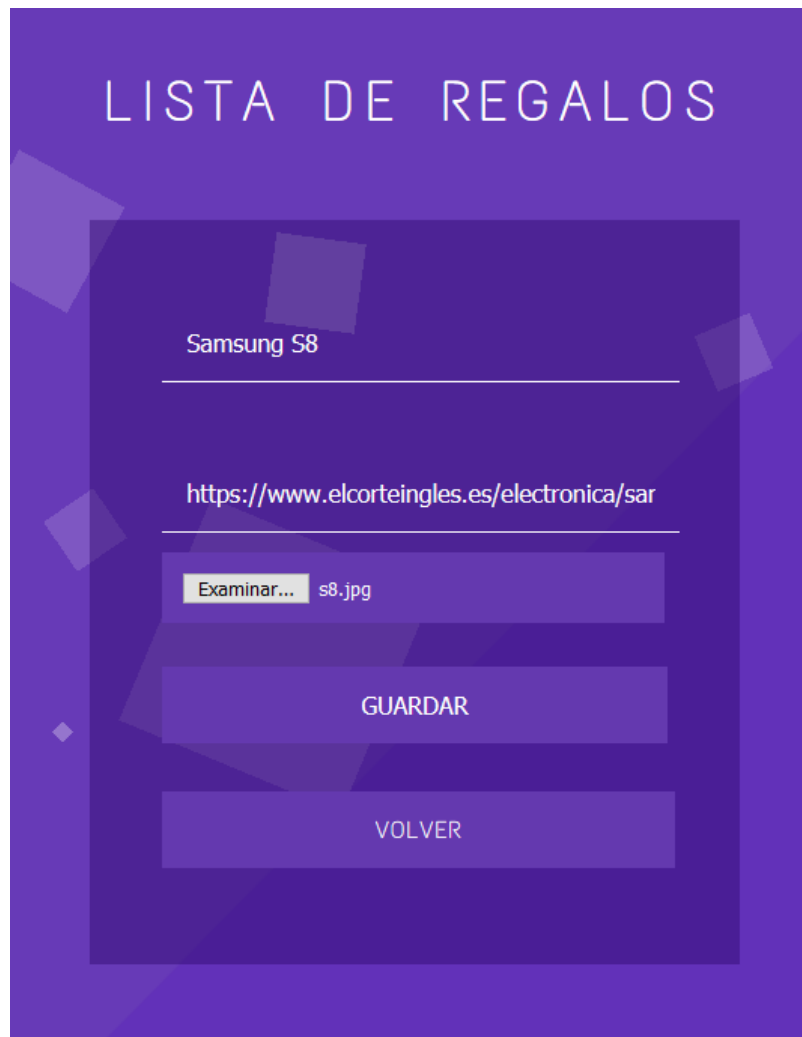
Cuando el usuario accede a la url compartida se mostrará una pantalla como la siguiente, en la que además de poder ver los productos también invitará al usuario no registrado a darse de alta en la aplicación.



4.5 Pantalla de alta de producto

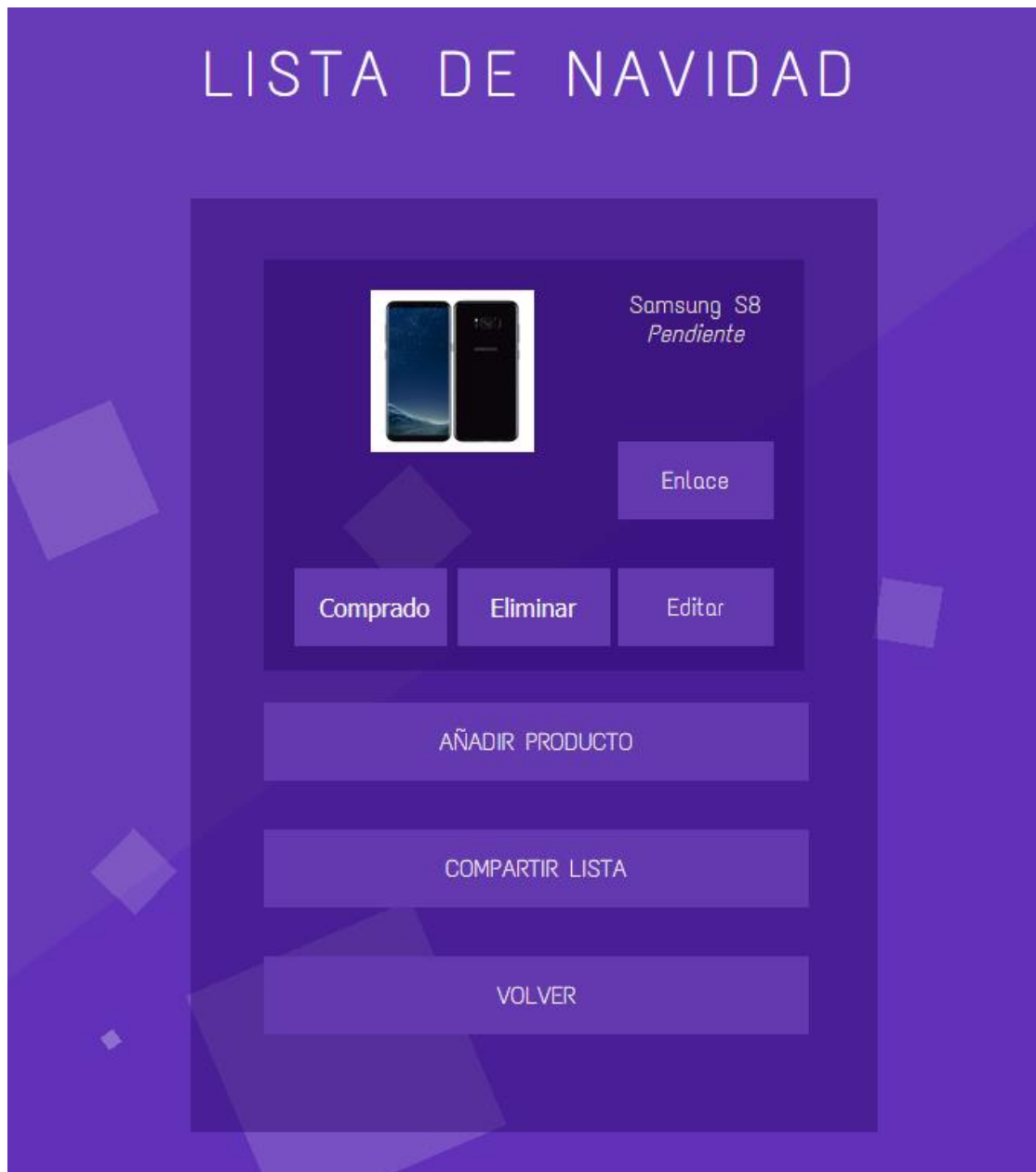
Para cada lista se pueden añadir productos y el formulario para hacerlo se muestra a continuación.

Se deberá introducir el nombre del producto, un enlace donde poder encontrar el producto, que será la información útil que se aporte al posible comprador y además una imagen para que la descripción del producto quede más completa.



The image shows a mobile application interface for adding a product to a gift list. The background is a solid purple color. At the top, the text 'LISTA DE REGALOS' is displayed in a light purple, sans-serif font. Below this, there is a dark purple rectangular area containing the form. The form has three input fields, each with a horizontal line underneath: the first contains 'Samsung S8', the second contains the URL 'https://www.elcorteingles.es/electronica/sar', and the third contains a file name 's8.jpg' preceded by a small yellow button labeled 'Examinar...'. Below the input fields are two large, light purple buttons with white text: 'GUARDAR' and 'VOLVER'.

El nuevo producto "Samsung S8" quedará añadido a la lista.



5. Referencias y Software

- ✓ XAMPP (Apache + MariaDB + PHP + Perl) <https://www.apachefriends.org/es/index.html>
- ✓ Visual Studio Code <https://code.visualstudio.com/>
- ✓ Git y Visual Studio Code
<http://lemoncode.net/lemoncode-blog/2017/12/12/git-y-visual-studio-code>
- ✓ Composer <https://getcomposer.org/>
- ✓ Github <https://github.com/>
- ✓ Slim Framework <https://www.slimframework.com/>
- ✓ Php <http://www.php.net/>
- ✓ <https://www.draw.io/>