Fullstack - TFT

Lista

de

Regalos

Óscar Pérez Riaño

Héctor Rodríguez Iglesias

Raquel Rodríguez Hernández



INDICE

[1 Objetivos funcionales y definición de la aplicación 2](#_Toc8803)

[1.1 Estructura de pantallas y navegabilidad 2](#_Toc19474)

[2. Preparación del entorno 4](#_Toc31055)

[2.1 Instalación 4](#_Toc18830)

[2.2 Diseño de BBDD 7](#_Toc32544)

[3. Desarrollo con PHP SLIM 9](#_Toc8429)

[3.1 Versiones y entorno de trabajo 9](#_Toc5035)

[4. Funcionamiento de la aplicación 13](#_Toc5109)

[4.1 Pantalla de Login 13](#_Toc9910)

[4.2 Pantalla de Registro 13](#_Toc7174)

[4.3 Pantalla de gestión de listas del usuario 15](#_Toc14863)

[4.3.1 Creación de nueva Lista 16](#_Toc4597)

[4.3.2 Ver Listado de Listas 17](#_Toc12399)

[4.4 Ver detalle de la lista 18](#_Toc28141)

[4.4.1 Compartir Lista con Usuario 19](#_Toc2145)

[4.4.2 Acceso a lista compartida 20](#_Toc10029)

[4.5 Pantalla de alta de producto 21](#_Toc2943)

[5. Referencias y Software 24](#_Toc15066)

# 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 una descripción, enlace e imagen descriptiva

- Para cada lista, un usuario registrado podrá:

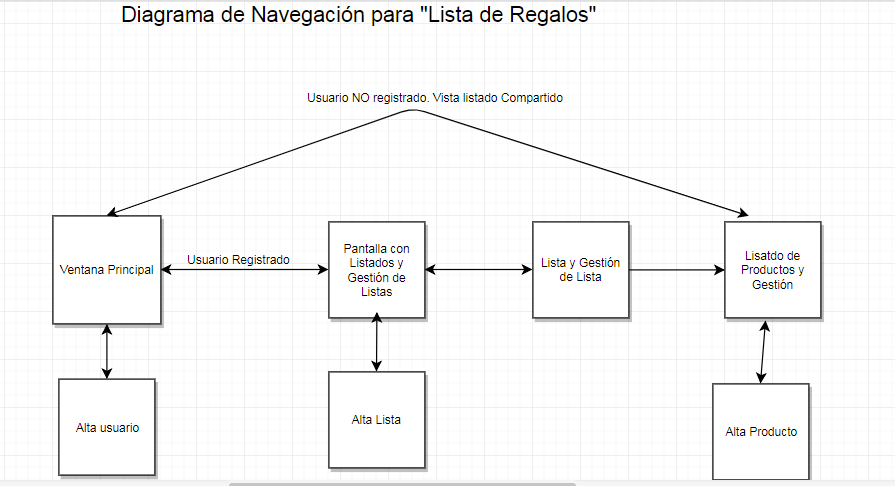
* añadir productos con enlaces, imágenes y texto
* marcar las cosas como compradas o como pendientes

- Un usuario registrado podrá compartir su lista con cualquier persona a la que se le envíe el enlace 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.
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.



# **Preparación del entorno**

## Instalación

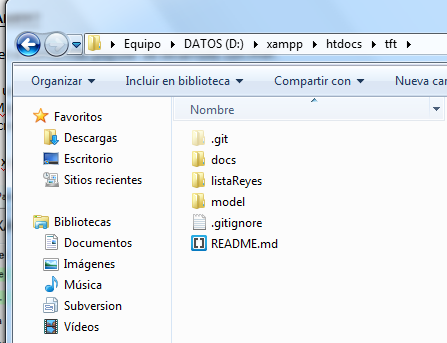
Para poder desplegar nuestra aplicación es necesario arrancar un servidor Apache y el servicio MySQL. Nosotros hemos utilizado Xampp para gestionar todos los servicios desde su panel de control.

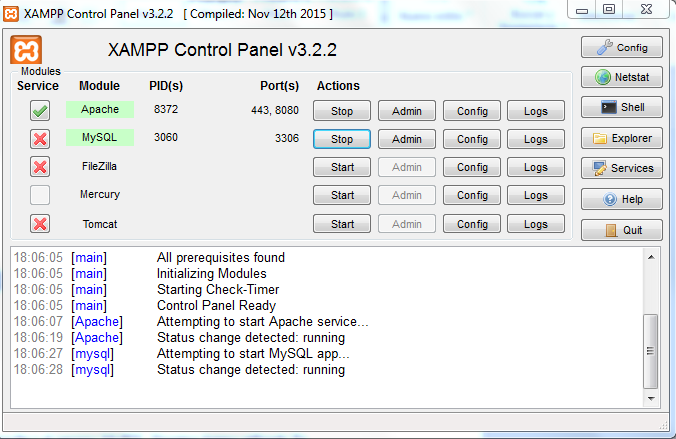
**¿Qué es XAMPP?**

XAMPP es el entorno más popular de desarrollo con PHP.

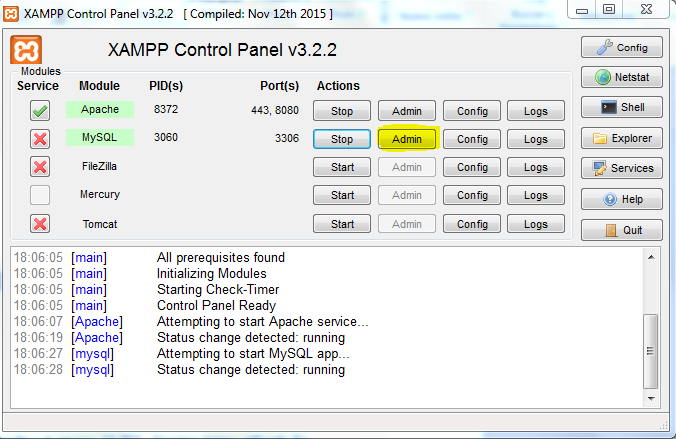
XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

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

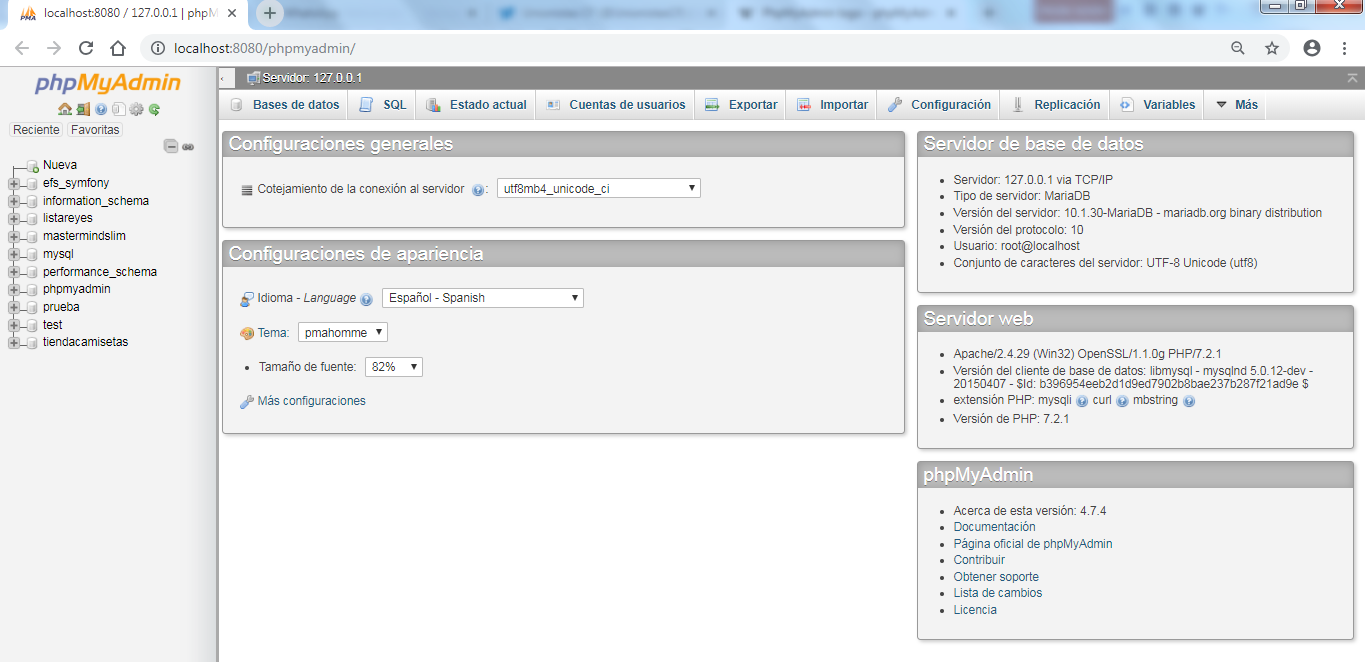




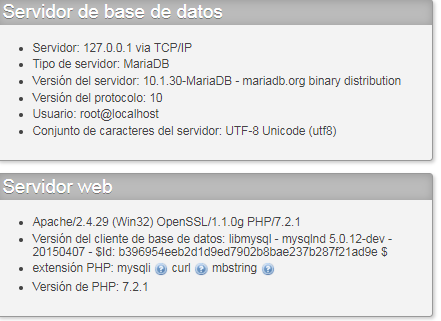
Una vez arrancado los servicios se debe abrir el administrador de base datos. Para ello se pulsa sobre el botón Admin (marcado en amarillo en la imagen).



Con ello abriremos *phpMyAdmin* que es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. 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.



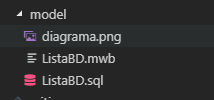
Estas son las versiones de servidor y demás información que proporciona la herramienta.



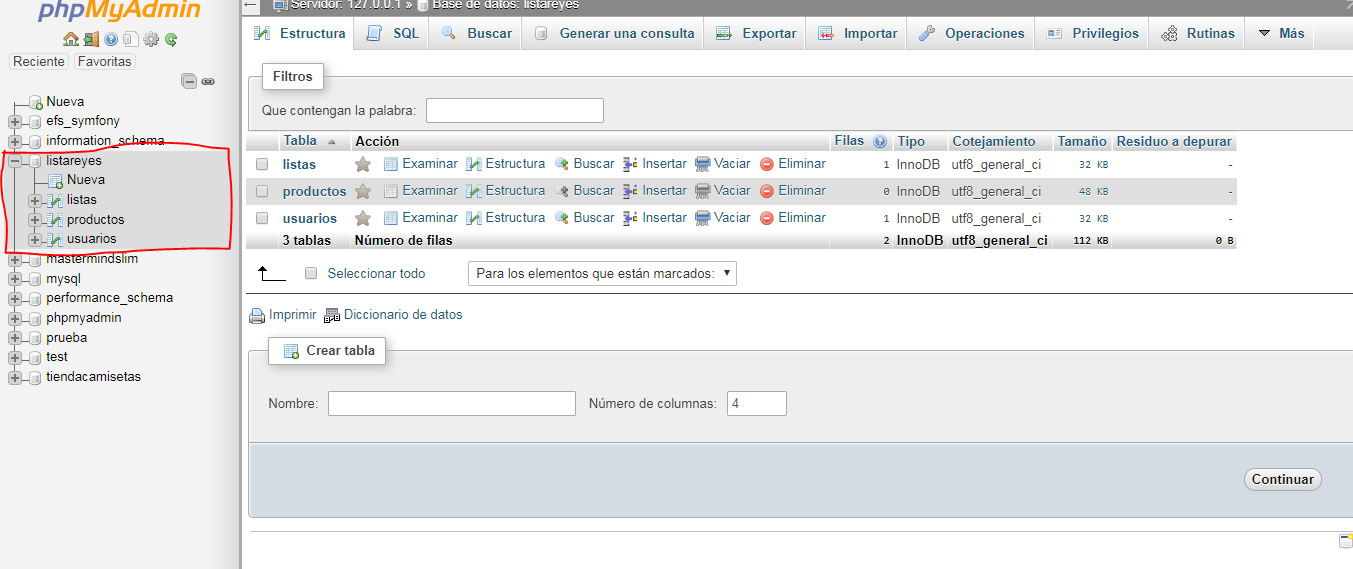
La url de acceso web sería la siguiente.

http://localhost:8080/phpmyadmin/

Hace falta tener la base de datos y para ellos 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.

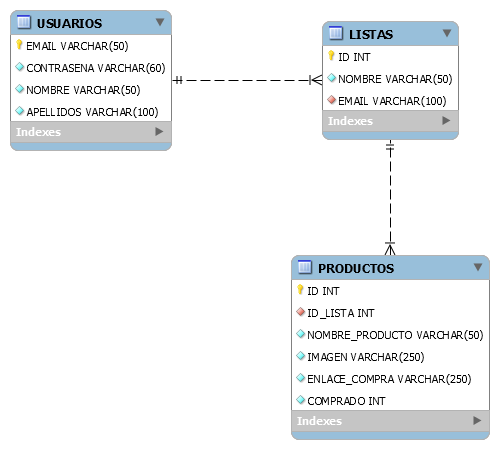


## ***Diseño de BBDD***

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



# Desarrollo con PHP SLIM

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

## ***Versiones y entorno de trabajo***

### Framework Slim y versiones utilizadas

Es un framework para crear REST Api usando PHP y es bastante práctico y poderoso a la hora de implementar. El framework ayuda en la creación del API y su comunicación con el cliente, es decir:

* Validación de tipos de petición (GET, POST, INPUT, DELETE ..)
* Mddleware: 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 --> 5.5.0
* Versión del Framework SLIM 3.1 con los siguientes requerimientos:
  + ***“illuminate/database"***: ~5.1: Para la definición de la base de datos
  + ***"respect/validation"***: "^1.1: Utilizado para las validaciones de los formularios
  + ***"slim/twig-view"***: "^2.4": Se utiliza para las plantillas

Todas estas dependencias se define en un fichero json en el proyecto (*composer.json)*

### 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 los diferentes elementos que se tienen que gestionar en la aplicación. En ellos se definen las acciones sobre los formularios y las redirecciones. Es 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

* + - Elliminar: 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){

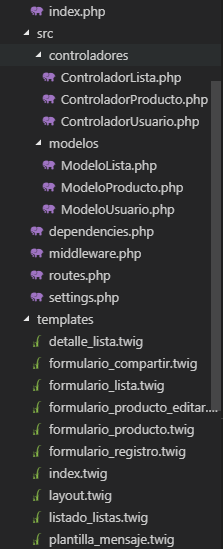
    return $this->view->render($response, 'formulario\_lista.twig');

})->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

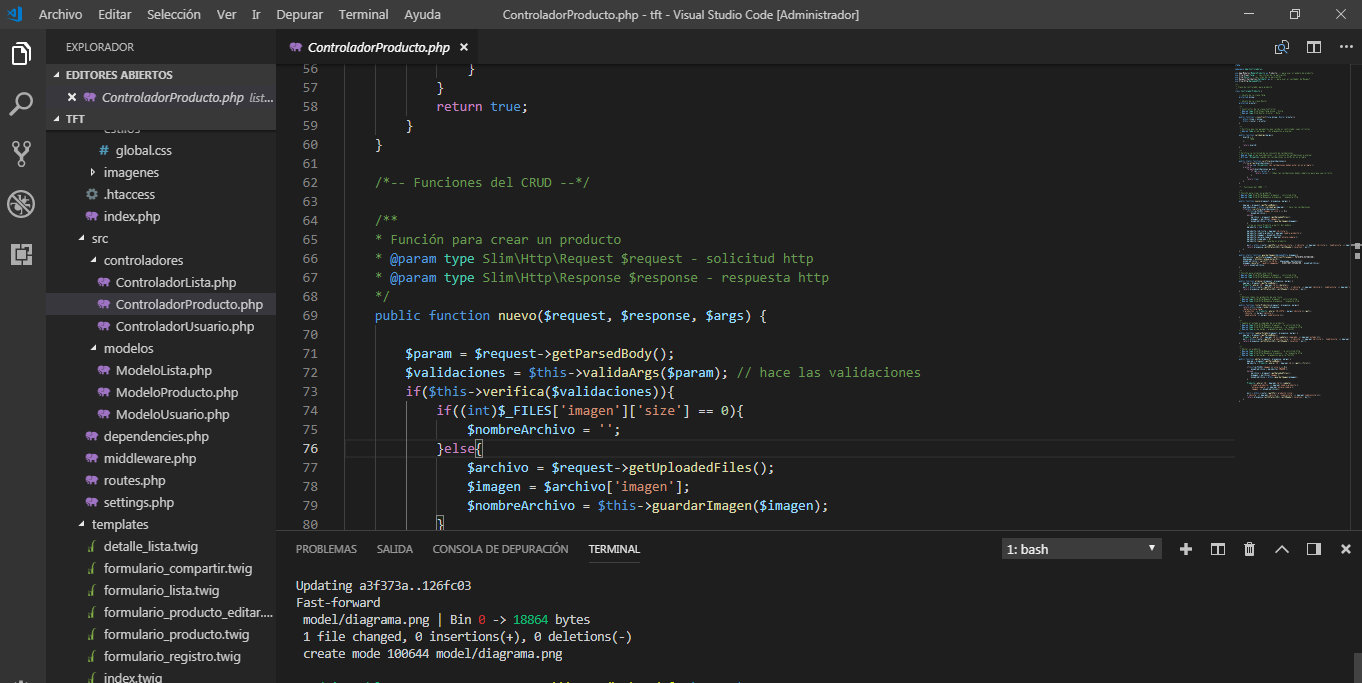


Además otra carpeta importante es **templates** donde se han definido las plantillas de los formularios de la aplicación. Estas plantillas están escritas en *Twig* que es el motor de plantillas para PHP. Cada una de ellas tiene definido el diseño de la página/formulario que se mostrará en la aplicación.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

* detalle\_lista.twig
* formulario\_compartir.twig
* formulario\_lista.twig
* formulario\_producto\_editar.twig
* formulario\_producto.twig
* formulario\_registro.twig
* index.twig
* layout.twig.
* listado\_listas.twig. Plantilla utilizada para los listados
* plantilla\_mensaje.twig Plantilla genérica creada para para mostrar mensajes

### El editor que se ha utilizado para trabajar 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.



# **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.

### 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.



### 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 usuario) y la contraseña.



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.





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.



### Pantalla de gestión de listas del usuario

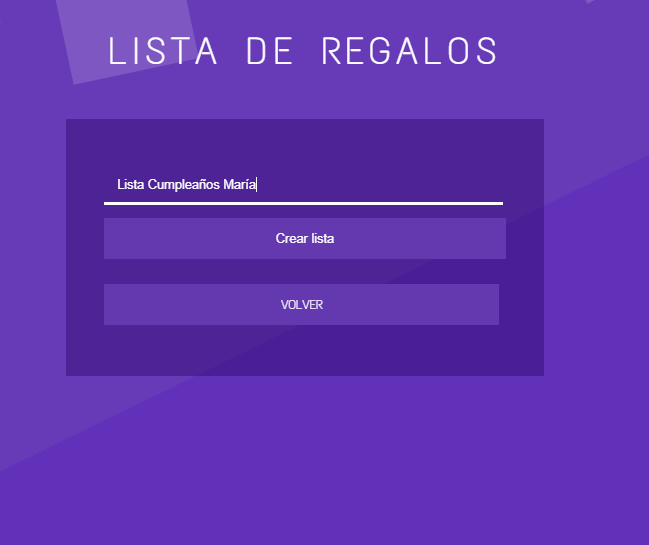
Una vez *logados* veremos una pantalla en la que se presentarán las siguientes opciones:

* Mis Listas: Dónde se consultará las listas que tiene el usuario
* Nueva Lista: Desde donde se puede dar de alta las listas
* Salir: Donde se cerrará la sesión del usuario



##### 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.



##### Ver Listado de Listas



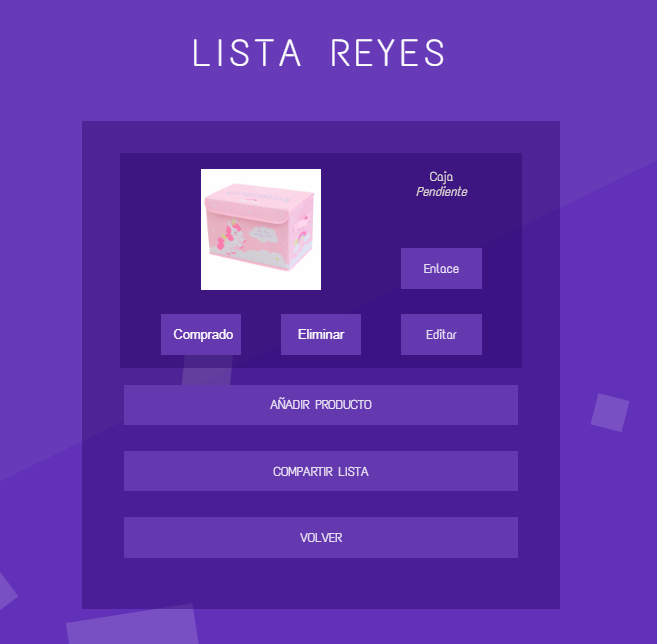
Si lo que se selecciona es la opción de Mis Listas se mostrará en pantalla un listado de las listas que ha creado el usuario.

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.

### 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

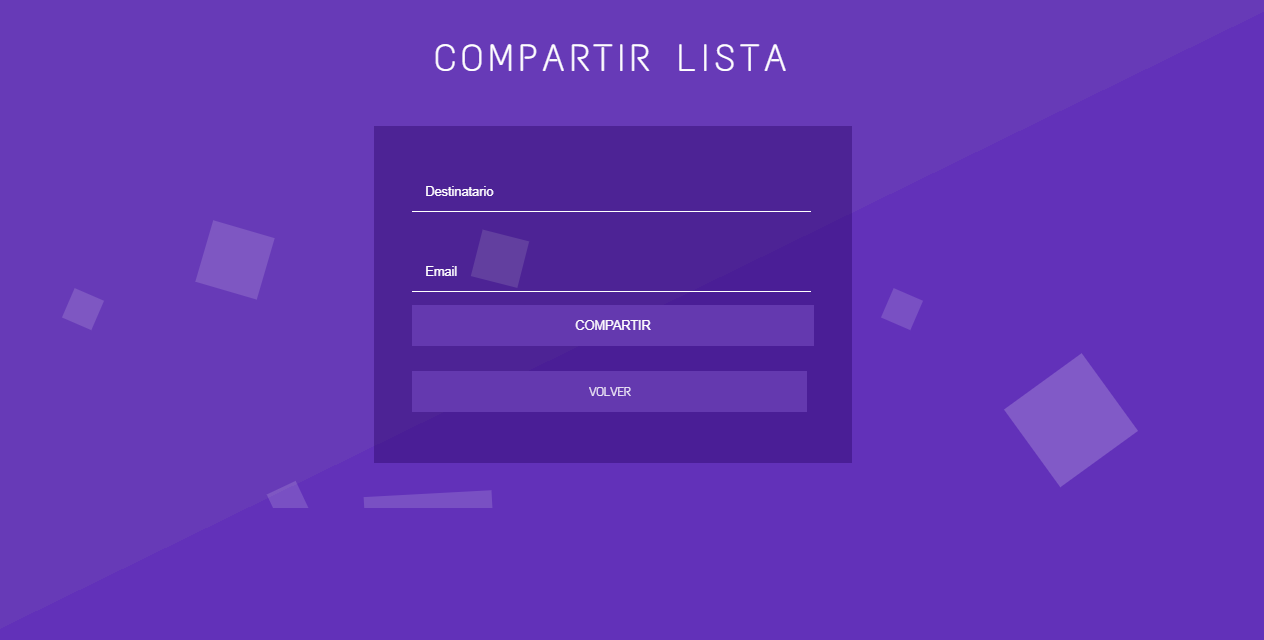
Un producto podrá eliminarse lo que hará que el producto deje de estar presente en la lista y editarse para cambiar cualquier información. Además, si se pulsa sobre el botón enlace se abrirá en otra pestaña el enlace asociado al producto.

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 como está marcado como “comprado”

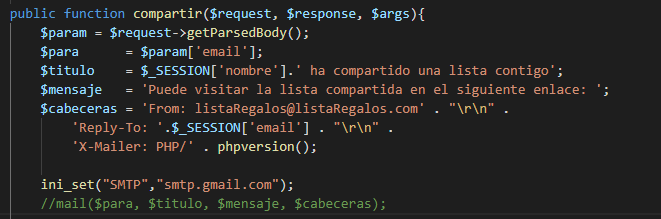


##### Compartir Lista con Usuario

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.



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



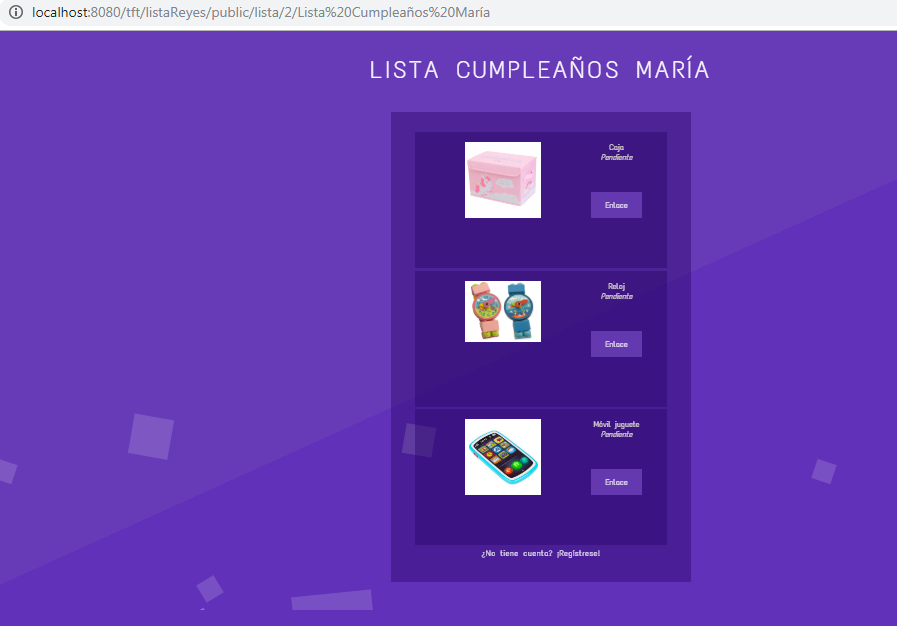
Y así es como se verá en pantalla ya que para enviar por correo hay que instalar un servidor smtp o configurar el acceso a un servidor de correo con nombre, contraseña y como no tenemos un hosting para desplegar la aplicación quedaría preparado.



##### 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.



### Pantalla de alta de producto

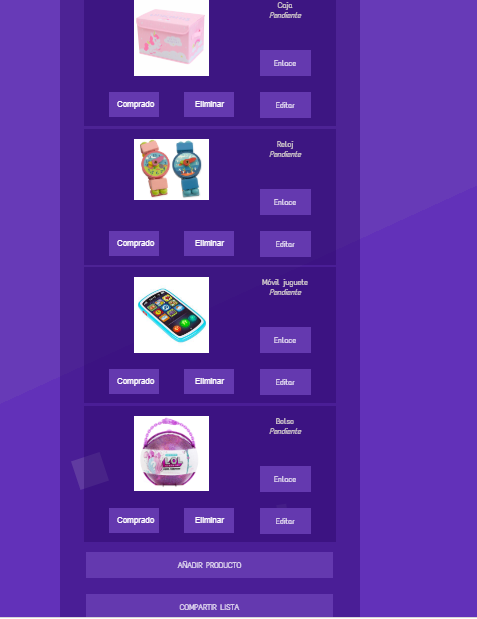
Para cada lista se pueden añadir productos y el formulario para hacerlo se muestra a continuación.



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.



El nuevo producto “bolso” quedará añadido a la lista.



# **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/