



TÉCNICO EN DESARROLLO DE APLICACIONES WEB  
Departamento de Informática

PROYECTO  
HoFinControl

**Manual Técnico**

Autor: Alejandro González Macías  
Curso Académico: 2019/2020

## Indice

1. Introducción .....	4
2. Objetivos .....	4
3. Tecnologías involucradas. ....	5
3.1 Herramientas utilizadas en el desarrollo. ....	5
3.2. Diseño de la base de datos. ....	5
3.2.1 Tabla Usuarios . ....	6
3.2.2 Tabla Países. ....	6
3.2.3 Tabla Permisos_Usuarios. . ....	6
3.2.4 Tabla Permisos. . ....	6
3.2.5. Tabla Transacciones. ....	6
3.2.6 Tabla tipo_ingreso_gasto. ....	6
3.2.7 Tabla Cuentas. ....	7
3.3. Diagrama de casos de usos. ....	7
3.4. Modelo lógico. ....	9
3.5. Lenguajes de desarrollo. ....	9
3.5.1 PHP . ....	9
3.5.2 JavaScript . ....	10
3.6. Frameworks. ....	10
3.6.1 AngularJs . ....	10
3.6.2 Bootstrap . ....	10
3.7. Librerías . ....	11
3.7.1. JQuery . ....	11
3.7.2 ChartJs . ....	11
3.8. Otros lenguajes . ....	11
3.8.1 HTML . ....	11
3.9 Guía de estilos. . ....	11
3.9.1 Introducción. . ....	11
3.9.2 Navegación. . ....	11
3.9.3 Estructura. . ....	12
3.9.4 Elementos de ordenación. . ....	12
3.9.4.1 Área superior páginas. ....	13
3.9.4.2 Área de navegación páginas. ....	13
3.9.4.3 Área central páginas. . ....	13
3.9.5 Elementos básicos. . ....	13
3.9.5.1 Tipografía. . ....	13
3.9.5.2 Colores. . ....	13
3.9.5.3 Imágenes. . ....	14
3.9.5.4 Iconografía. ....	15
3.10 Estructura del proyecto. . ....	17
3.10.1 Carpeta php. ....	18
3.10.2 Carpeta model. ....	18
3.10.3 Carpeta js. ....	18
3.10.4 Carpeta include. ....	18
3.10.5 Carpeta img. ....	18
3.10.6 Carpeta dao. . ....	18
3.10.7 Carpeta css. . ....	18



3.10.8 Carpeta controller.....	19
3.11 Proceso de desarrollo . . . . .	19
3.11.1 Implementación. . . . .	19
3.11.2 Ciclo de vida de la aplicación. . . . .	20
4. Despliegue de la aplicación. . . . .	21
4.1 Preparativos previos. . . . .	21
4.2 Instalación de XAMP. . . . .	21
4.3 Instalación de la base de datos. . . . .	22
4.4 Paso previo instalación del software. . . . .	23
4.5 Instalación software. . . . .	23
4.6 Configuración del acceso a la base de datos HoFinControl. . . . .	24
5. Propuesta de mejora o trabajos futuros. . . . .	25
6. Bibliografía. . . . .	25

## 1. Introducción.

A lo largo de los últimos años, el avance de la tecnología ha sido tal, que un gran número de acciones han pasado a ser realizadas a través de Internet. Desde la simple comunicación entre dos personas, hasta la compra de productos por Internet, consulta de nuestras facturas o información de nuestra cuenta bancaria.

A diario salen a la luz nuevas aplicaciones web, cada vez más fáciles y cómodas, que nos hacen posible la realización de tareas cotidianas de manera más sencilla y eficaz.

Es sin duda que las ventajas superan los inconvenientes. Este tipo de tecnologías han ayudado a mejorar muchos aspectos de la vida cotidiana, por ejemplo, el acceso a la información, el almacenamiento de datos personales de forma segura o la compra de manera inmediata, sencilla y desde cualquier lugar

## 2. Objetivos

El desarrollo de una aplicación web para gestionar los gastos de casa, viene motivado por la inminente necesidad que tiene los hogares del siglo XXI de llevar un control de gastos que les permitan gestionar de manera rápida y eficiente todos los gastos e ingresos ocurridos dentro de un ámbito familiar.

De esta manera podemos decir que el objetivo principal de este proyecto es desarrollar una aplicación web para el control de gastos, donde los usuarios podrán registrarse, gestionar y consultar todos aquellos gastos anuales, así como ver estadísticas relevantes según dichas transacciones.

Otro de los principales objetivos de un producto de estas características es la forma de llegar al usuario, por lo que una interfaz amigable y la facilidad de uso se hacen indispensables en este proyecto.

La robustez es otro de los objetivos de cualquier aplicación, a través de la cual se mide, en su mayoría, la calidad del producto desarrollado.

Dentro de la aplicación web, se desarrollará una gestión de cuentas, donde el usuario gestionará las diferentes cuentas bancarias disponibles.

La aplicación web dispondrá de una gestión de ingresos y gastos para que el usuario pueda guardar y asociar esa información a las diferentes cuentas configuradas.

La aplicación dispondrá de un cuadro de mando, en el cual el usuario puede ver en tiempo real el estado de sus cuentas.

## 3. Tecnologías involucradas.

### 3.1 Herramientas utilizadas en el desarrollo.

Para el desarrollo de nuestra aplicación se han utilizado diversas herramientas en el desarrollo.

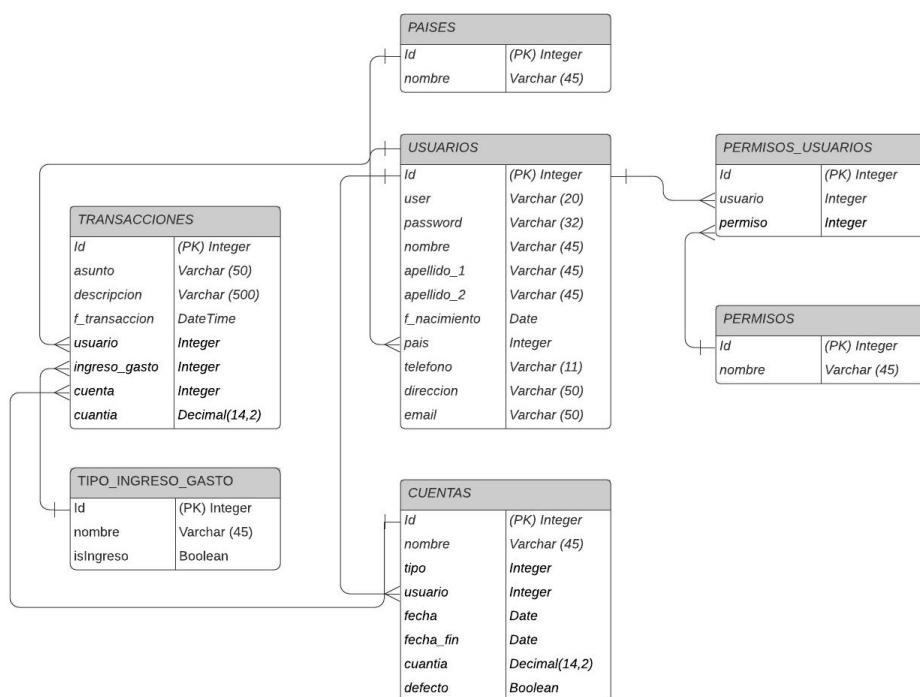
En primera instancia hemos utilizado como IDE de desarrollo NetBeans 8.2. tanto para el desarrollo front como back de la aplicación. Además hemos necesitado como motor de base de datos MySQL en su versión 5.2. Hemos utilizado el cliente MySQL Workbench, el cual nos ha facilitado mucho la tarea en el diseño de la base de datos y su gestión debido a su interfaz intuitiva.

Para que todo nuestro desarrollo funcione y se pueda ejecutar, necesitamos un servidor de aplicaciones, para ello se ha utilizado Apache en su versión 2.4.

Los navegadores utilizados para su visualización y pruebas son Chrome en su última versión, Mozilla Firefox e Internet Explorer.

### 3.2. Diseño de la base de datos.

En este apartado se tratan los aspectos de diseño relacionados con la base de datos del servidor. Como motor de base de datos se ha utilizado MySql. En un principio se muestra un diseño modelo entidad-relación, al final del apartado se muestra el detalle de tablas.



A

continuación pasaremos a detallar el contenido de cada tabla:

## 3.2.1 Tabla Usuarios

Esta tabla permitirá llevar la gestión de los usuarios de la aplicación, llevará relaciones 1:n con la tabla permisos\_usuarios, 1:n con la tabla transacciones y 1:1 con la tabla de paises. Como punto a tener en cuenta la columna password tendrá codificada su información en md5.

## 3.2.2 Tabla Paises.

La tabla paises tendrá la información relevante a los paises de un usuario. Tendrá solamente relación con la tabla usuarios 1:1.

## 3.2.3 Tabla Permisos\_Usuarios.

Esta tabla permite realizar la relación entre la tabla permisos y la tabla usuarios. Esta tabla es importante para que la aplicación sepa a qué secciones tendrá acceso el usuario. Estará relacionada con la tabla usuarios y con la tabla permisos.

## 3.2.4 Tabla Permisos.

Esta tabla guardará la información relevante a los permisos disponibles en la aplicación. Tendrá una relación 1:n con la tabla permisos\_usuarios.

## 3.2.5. Tabla Transacciones.

Junto con la de usuarios es una de las tablas más importantes de la aplicación. Esta tabla nos permitirá tener registrada la actividad de las cuentas de un usuario, tanto gastos como ingresos. Esta tabla mantendrá relaciones con otras 4 tablas, como son las de usuarios, cuentas y tipos de ingresos gastos.

## 3.2.6 Tabla tipo\_ingreso\_gasto.

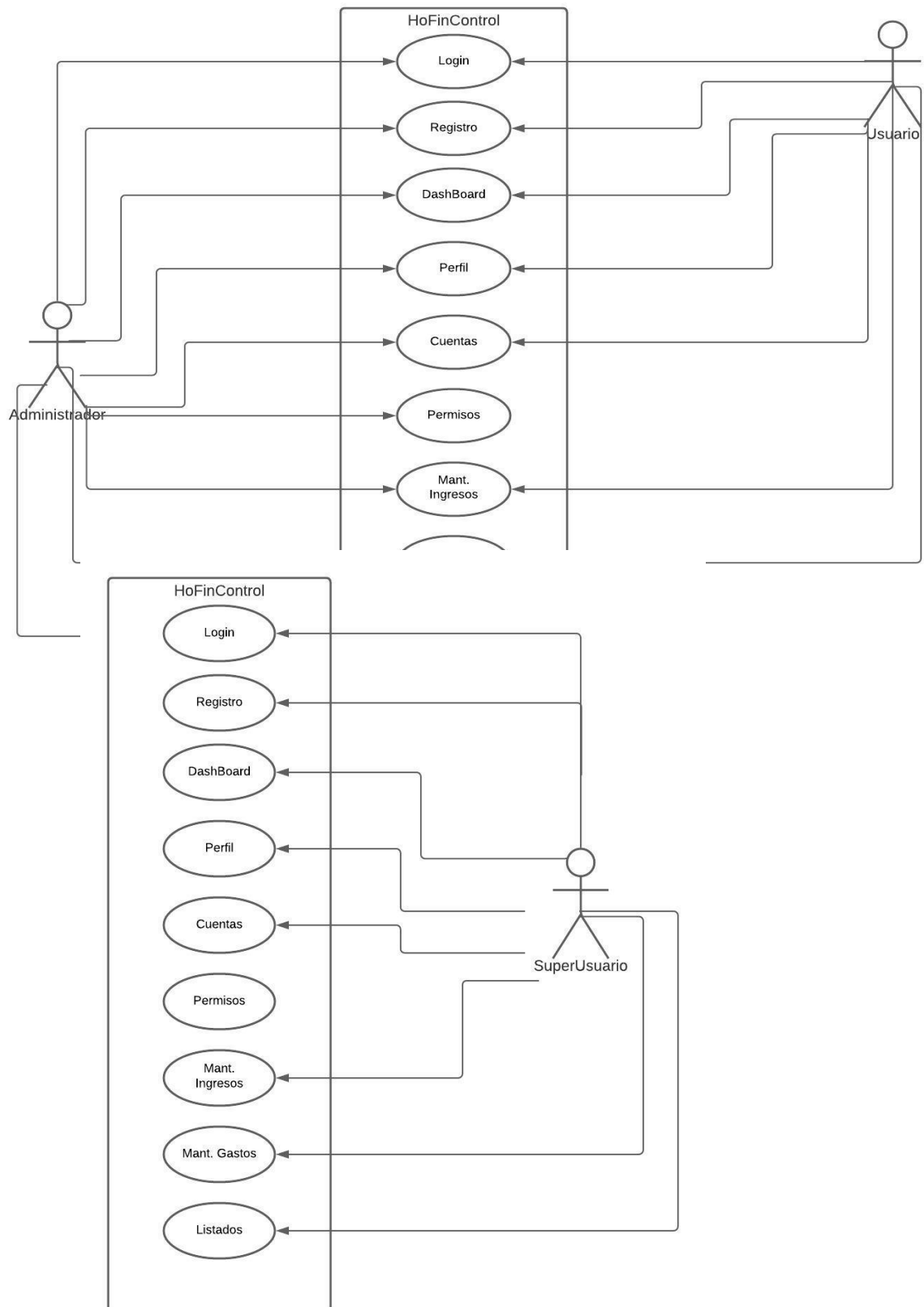
Esta tabla tendrá la información relevante a los tipos de ingresos y gastos que el usuario podrá elegir en la aplicación. Como en esta tabla se guardan tanto las denominaciones de los ingresos y gastos, se ha añadido una columna llamada isIngreso, la cual nos permitirá distinguir cual denominación es ingreso y cual es gasto. Esta tabla solamente se relacionará 1:n con la tabla de transacciones.

## 3.2.7 Tabla Cuentas.

Esta tabla llevará la gestión de las cuentas de un determinado usuario. Esta tabla mantendrá relación 1:n con la tabla de usuarios y 1:n con la tabla de transacciones.

## 3.3. Diagrama de casos de usos.

A continuación pasamos a indicar los diagramas de casos de uso de los que dispone la aplicación.



Actualmente nuestra aplicación dispondrá de 3 roles diferenciados.

Rol de Administrador. Este rol tendrá acceso a todas las partes de la aplicación.

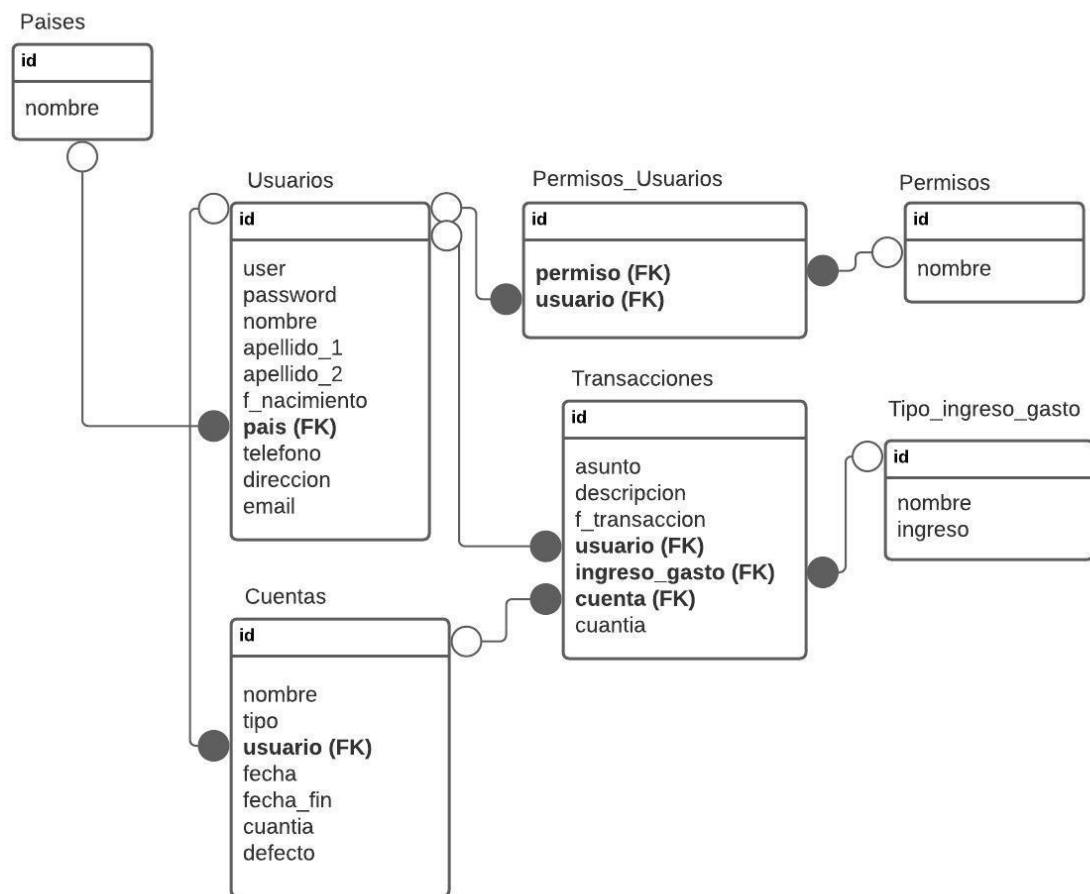
Rol de Usuario. Este rol tendrá acceso a todas las funcionalidades de la aplicación, excepto a la gestión de permisos y listados. Este rol se aplica a todo usuario nuevo que se dé de alta en la aplicación.

Rol de SuperUsuario. Este rol solamente podrá ser adquirido en el caso de que el administrador de la aplicación dé acceso al resto de las opciones disponibles y las cuales no tiene acceso el rol de Usuario. Para que un usuario pase a tener rol de superusuario el administrador de la aplicación deberá darle permisos en todas las opciones de la aplicación excepto en la de permisos.

### 3.4. Modelo lógico.

A continuación se adjunta la representación del modelo lógico de la aplicación.





## 3.5. Lenguajes de desarrollo.

En este apartado se enumerarán los diferentes lenguajes de desarrollo utilizados para la implementación de la aplicación.

Cabe mencionar que todo el software utilizado es libre, con lo que conllevará un considerable ahorro en concepto de licencias. El software seleccionado no lo ha sido sólo por gratuito, sino porque además es una de las tecnologías más utilizada en la actualidad en el desarrollo de aplicaciones web debido a su fiabilidad y a su versatilidad.

### 3.5.1 PHP

El lenguaje PHP es un lenguaje de código abierto normalmente dedicado al desarrollo web y que puede ser incrustado en lenguaje HTML. Utilizado también en servidores para gestionar llamadas http, lo que es perfecto para alcanzar los objetivos de este proyecto.

### 3.5.2 JavaScript

Es un lenguaje de programación interpretado, dialecto del estándarECMAScript. Es un lenguaje de programación que se ejecuta del lado del cliente, permitiendo mejoras en la interfaz del usuario y en páginas web de tipo dinámicas.

## 3.6 Frameworks.

En este apartado se indicarán los frameworks utilizados para el desarrollo de la aplicación.

### 3.6.1 AngularJs

AngularJS es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications.

Como este Framework no ha sido estudiado en el ciclo, se explicará con más detenimiento y sus ventajas.

Como hemos comentado con anterioridad, angular es un Framework javascript que incluye prácticamente todos los aspectos que puedas necesitar para la creación de una aplicación cliente en javascript, esto incluye la generación de vistas, uso del databinding, comunicación con el servidor... Una cosa importante a puntualizar, es que angular está centrado en la creación de aplicaciones web y no páginas web.

Las cosas buenas de utilizar este Framework es su sistema de databinding que es muy completo y potente. Otra ventaja es que al ser un Framework y no una biblioteca, este incluye muchas más funcionalidades y opciones de serie que una simple biblioteca, lo que provoca que la curva de aprendizaje sea más rápida.

Otro punto a favor es que es Open Source, con lo cual permite que todas las personas en el mundo emprendan y colaboren sin restricciones tecnológicas.

### 3.6.2 Bootstrap

Bootstrap es el Framework HTML, CSS y JavaScript de código abierto más popular para desarrollar sitios web responsive y móviles.

## 3.7. Librerías.

En este apartado se pretende enumerar las diferentes librerías utilizadas en el desarrollo de la aplicación.

### 3.7.1. JQuery.

jQuery es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

### 3.7.2 ChartJs.

Chart.js es una biblioteca JavaScript gratuita de código abierto para la visualización de datos, que admite 8 tipos de gráficos: Barra, línea, área, circular, burbuja, radar, polar, dispersión.

## 3.8. Otros lenguajes

### 3.8.1 HTML

HyperText Markup Language, aunque lo hayamos metido en un apartado de “otros lenguajes”, en realidad HTML no es un lenguaje de programación sino que es un lenguaje de marcado para la elaboración de páginas y aplicaciones web.

## 3.9 Guía de estilos.

### 3.9.1 Introducción.

La guía de estilos de la Web para la aplicación HoFiControl tiene como objetivo indicar las pautas a seguir por el equipo de desarrollo para la elaboración del sitio web de un modo coherente y relativo y que permitirá en un futuro un fácil mantenimiento de dicha Web.

### 3.9.2 Navegación.

La navegación elegida para el sitio web será de forma jerárquica. Desde la página de Inicio navegaremos a las diferentes secciones de la página web y desde estas navegaremos a la página de Inicio, bien por un botón de volver o seleccionando la opción del menú pertinente.

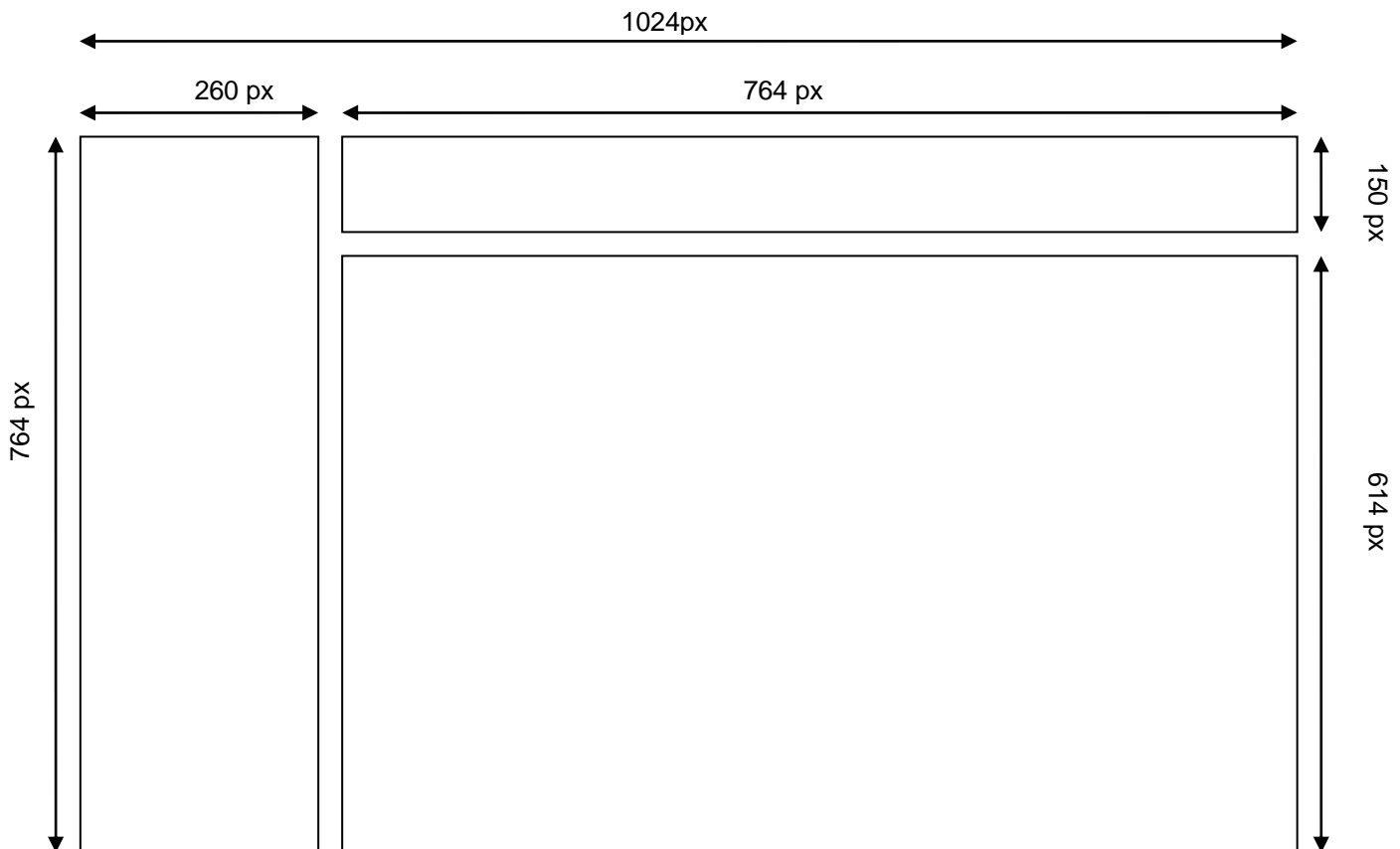
### 3.9.3 Estructura.

La interfaz de la página web estará formada por una columna dividida en 2 bloques claramente diferenciados. Estableceremos en el bloque izquierdo el menú de navegación de la aplicación, para que el usuario pueda navegar entre las diferentes secciones. En el bloque derecho a su vez estará dividido en dos bloques, un bloque superior donde estableceremos la cabecera, seguido por un bloque donde mostraremos el contenido de las páginas a las cuales el usuario accede a través del menú.

La interfaz permitirá:

- Adaptación a varias resoluciones.
- Interfaz escalable.

A continuación se muestra un gráfico de la estructura de la aplicación.



### 3.9.4. Elementos de ordenación.

En este apartado se describirán las diferentes áreas que tendrán las páginas de la aplicación y los elementos que aparecerán en ellos.

#### 3.9.4.1 Área superior páginas

Encabezado: Espacio destinado a la colocación del logotipo de la asociación, además se establecerá un elemento de información denominado rastro de migas, donde el usuario pueda identificar donde se encuentra en cada instante de la aplicación. También en este espacio se encontrará un elemento identificativos donde aparecerá el usuario logado en la aplicación y otro elemento interactivo, identificado por un enlace y el texto salir donde el usuario al pulsar puede salir de la aplicación.

### 3.9.4.2 Área de navegación páginas

Navegación: se establecerá un sistema de navegación con un menú vertical, donde el usuario podrá seleccionar entre las opciones de Dashboard, Perfil, Cuentas, Gestión de permisos, Mantenimiento de gastos, Mantenimiento de ingresos y Listados. Dentro de la opción Listados tendrá dos subopciones que serán Listados de ingresos y Listado de gastos.

### 3.9.4.3 Área central páginas.

Contenido: Área destinada al contenido de la web, para la página de Dashboard mostraremos 4 etiquetas con diversos contenidos además de 2 gráficos y en la parte inferior un listado.

## 3.9.5 Elementos básicos.

### 3.9.5.1 Tipografía.

Para el sitio web se empleará un tipo de fuente Arial, una de las tipografías estándar de Html y que será visible prácticamente por la totalidad de navegadores web. Usada en su versión regular y con un tamaño mínimo de 14px.

Arial  
abcdefghijklmnopqrstuvwxyzáéíóü  
ABCDEFGHIJKLMNOPQRSTUVWXYZÁÉÍÓÜ  
1234567890 \$%&(',;:'"!?

### 3.9.5.2 Colores.

La página web destaca por una gama cromática fría, con un predominio del color blanco, azul y gris. Como colores para la fuente elegiremos un color más contrastado como el negro, dado por la imagen corporativa.

A continuación indicaremos la gama cromática para los contenidos de la página.





A continuación indicaremos la gama cromática para la fuente.



### 3.9.5.3 Imágenes.


Las imágenes e iconografía dentro de la web no deben de ser considerados como elementos de ornamentación, sino como apoyo a que el usuario tenga más fácil el manejo e interacción con la interfaz de la web.


### 3.9.5.4 Iconografía.

Para los iconos hemos utilizado Font Awesome, una lista de más de 400 iconos gratuitos que, al ser tratados como una fuente normal de letra, son fácilmente escalables y editables mediante CSS. Además, tardan menos tiempo en cargar que si usáramos imágenes.







Se establecerá una iconografía en la zona de navegación la cual ayudará al usuario a recordar e interactuar de una manera más agil e intuitiva por la aplicación. A continuación se indican los iconos y su tamaño. Como hemos dicho los iconos vendrán definidos por la

librería fontAwesome, por lo tanto su tamaño no estará en pixeles por alto/ancho, sino por tamaño de fuente font-size.



Icono	Definición	Propiedades
	Icono menú vertical para la opción inactiva de Dashboard.	Icono perteneciente a <b>fa fa-desktop</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Dashboard.	Icono perteneciente a <b>fa fa-desktop</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Perfil.	Icono perteneciente a <b>fa fa-config</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Perfil.	Icono perteneciente a <b>fa fa-config</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Cuentas.	Icono perteneciente a <b>fa fa-braille</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Cuentas.	Icono perteneciente a <b>fa fa-braille</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Gestión de permisos.	Icono perteneciente a <b>fa fa-database</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Gestión de permisos.	Icono perteneciente a <b>fa fa-database</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Mantenimiento de gastos.	Icono perteneciente a <b>fa fa-area-chart</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Mantenimiento de gastos.	Icono perteneciente a <b>fa fa-area-chart</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Mantenimiento de ingresos.	Icono perteneciente a <b>fa fa-signal</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Mantenimiento de ingresos.	Icono perteneciente a <b>fa fa-signal</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva de Listados.	Icono perteneciente a <b>fa fa-list</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Listados.	Icono perteneciente a <b>fa fa-list</b> , tamaño 14 px.
	Icono menú vertical para la opción inactiva Listados de gastos y para la opción de Listado de ingresos.	Icono perteneciente a <b>fa fa-folder</b> , tamaño 14 px.
	Icono menú vertical para la opción activa de Listados de gastos y para la opción de Listado de ingresos.	Icono perteneciente a <b>fa fa-folder</b> , tamaño 14 px.
	Icono para cerrar o comprimir el menú vertical.	Icono perteneciente a <b>fa fa-times</b> , tamaño 14 px.
	Icono para cerrar o descomprimir el menú vertical.	Icono perteneciente a <b>fa fa-align-justify</b> , tamaño 14 px.
	Icono para indicar el usuario logado.	Icono perteneciente a <b>fa fa-user</b> , tamaño 14 px.

	Icono para deslogarnos de la aplicación	Icono perteneciente a <b>fa fa-sign-out</b> , tamaño 14 px.
---	---	---

A continuación pasaremos a detallar la iconografía de las páginas.

Icono	Definición	Propiedades
	Icono perteneciente a la pantalla de dashboard. Indicará el nombre de la cuenta.	Icono perteneciente a <b>fa fa-list-alt</b> , tamaño 30 px, color #04B404
	Icono perteneciente a la pantalla de dashboard. Indicará el total disponible en cuenta.	Icono perteneciente a <b>fa fa-bar-chart</b> , tamaño 30 px, color #fca504
	Icono perteneciente a la pantalla de dashboard. Indicará el total de gasto anual.	Icono perteneciente a <b>fa fa-arrow-down</b> , tamaño 30 px, color #d61317
	Icono perteneciente a la pantalla de dashboard. Indicará el total de ingreso anual.	Icono perteneciente a <b>fa fa-arrow-up</b> , tamaño 30 px, color #008386
	Icono perteneciente a la pantalla de dashboard. Indicará el lugar del gráfico de fluctuaciones anual.	Icono perteneciente a <b>fa fa-line-chart</b> , tamaño 14 px, color #eeeeee
	Icono perteneciente a la pantalla de dashboard. Indicará el lugar del gráfico de gastos anual.	Icono perteneciente a <b>fa fa-pie-chart</b> , tamaño 14 px, color #eeeeee

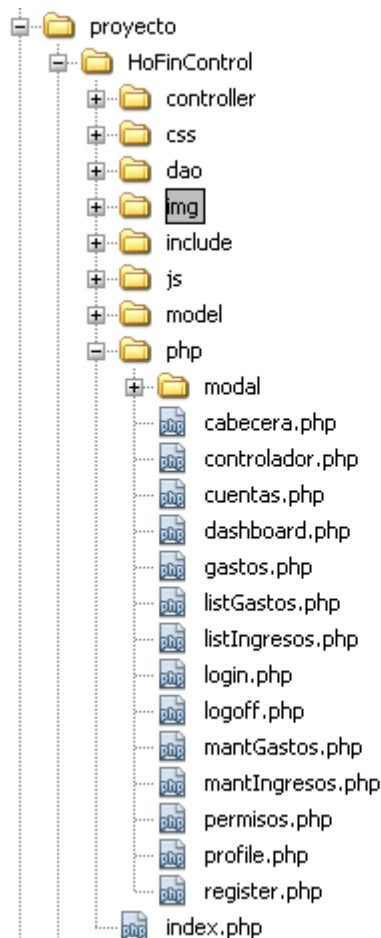
En cada pantalla de gestión o mantenimiento se incluirán los siguientes iconos.

Icono	Definición	Propiedades
	Icono que nos indicará que si pulsamos sobre el, el registro puede ser editado.	Icono perteneciente a <b>fa fa-pencil</b> , tamaño 30 px, color #000000
	Icono que nos indicará que si pulsamos sobre el, el registro puede ser eliminado.	Icono perteneciente a <b>fa fa-times</b> , tamaño 14 px, color #DF0101

## 3.10. Estructura del proyecto.

Los siguientes apartados nos aportarán la información necesaria para comprender la estructura del proyecto. A continuación se muestra una captura de pantalla donde se muestra la estructura de carpetas.





Enumeraremos las diferentes carpetas indicando su significado.

### 3.10.1 Carpeta php

Dentro de esta carpeta se ubicarán todos los ficheros php necesarios para pintar el front de la aplicación. Se añade también otra carpeta donde se ubicarán los php necesarios para pintar las ventanas modales de la aplicación.

### 3.10.2 Carpeta model

Dentro de esta carpeta estarán todas las clases php del modelo. Habrá una por cada entidad de base de datos.

### 3.10.3 Carpeta js

Carpeta necesaria para incluir librerías y útiles JavaScript necesarias para el funcionamiento de la aplicación.

## 3.10.4 Carpeta include

Esta carpeta incluye un fichero php, que irá incluido con una etiqueta include dentro de cada fichero php de la carpeta php. Además hay un fichero con extensión html donde están añadidas todas las librerías, css y otros elementos necesarios para el funcionamiento de la aplicación. Este fichero irá incluido en el fichero php controller, mediante una etiqueta include.

## 3.10.5 Carpeta img

Dentro de esta carpeta se ubicarán todas aquellas imágenes necesarias para la aplicación.

## 3.10.6 Carpeta dao.

En esta carpeta estarán ubicadas todas aquellas clases que nos ayudarán con el acceso a datos de la aplicación. Estará dividida en dos subcarpetas, service y resources.

Dentro de la carpeta service se encontrarán clases php que harán la función de servicios, ayudándonos a manejar la información entre las peticiones que se realizan desde la parte front a la parte back. Dentro de la carpeta resources se encontrarán las clases encargadas de lanzar las consultas a base de datos y que reportarán la información a las clases ubicadas en la carpeta service, cuando estas lo soliciten.

## 3.10.7 Carpeta css.

Dentro de esta carpeta se encontrarán todos los ficheros css necesarios de aportar los estilos para nuestra aplicación.

## 3.10.8 Carpeta controller.

Esta carpeta tendrán ficheros js controladores, encargados de recoger la información de pantalla y realizar las peticiones a las clases ubicadas en la carpeta service, mencionada en puntos anteriores, además de realizar diferentes interacciones con la parte front o manejar los diferentes elementos del DOM.

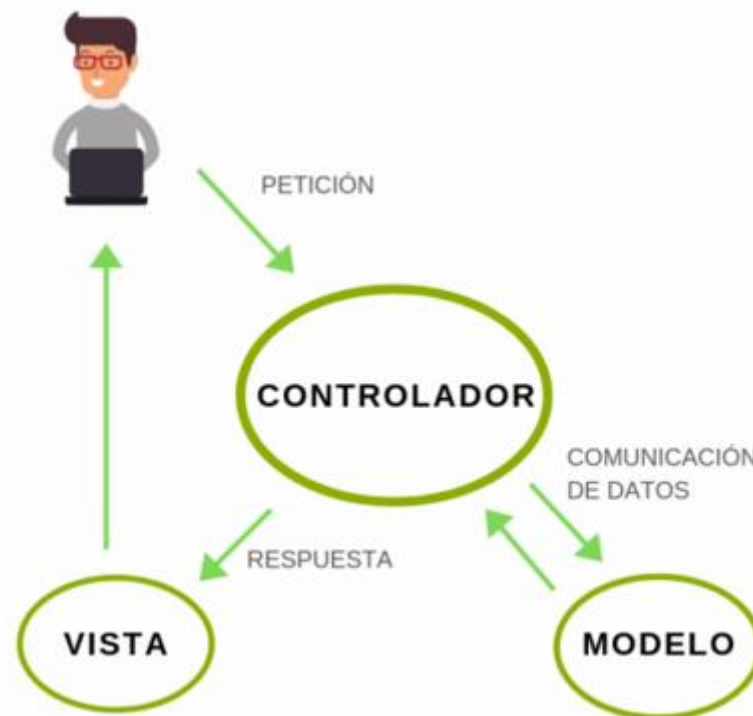
# 3.11. Proceso de desarrollo

En los siguientes puntos se explicará el proceso de desarrollo seguido para la implementación de la aplicación.

## 3.11.1 Implementación.

La aplicación se ha desarrollado siguiendo el diseño Modelo - Vista – Controlador. El diseño Modelo - Vista - Controlador, o también llamado diseño MVC, está orientado a objetos dentro

de la programación de las aplicaciones. Este permite separar la programación de la aplicación en tres campos. El Modelo contiene la información de la aplicación, como pueden ser: las variables, las especificaciones, etc. El Controlador se encarga de como presentar el Modelo al usuario en una interfaz gráfica. Finalmente, la Vista es la parte de la interfaz de usuario y la encargada de mostrar lo que le ordene el Controlador.

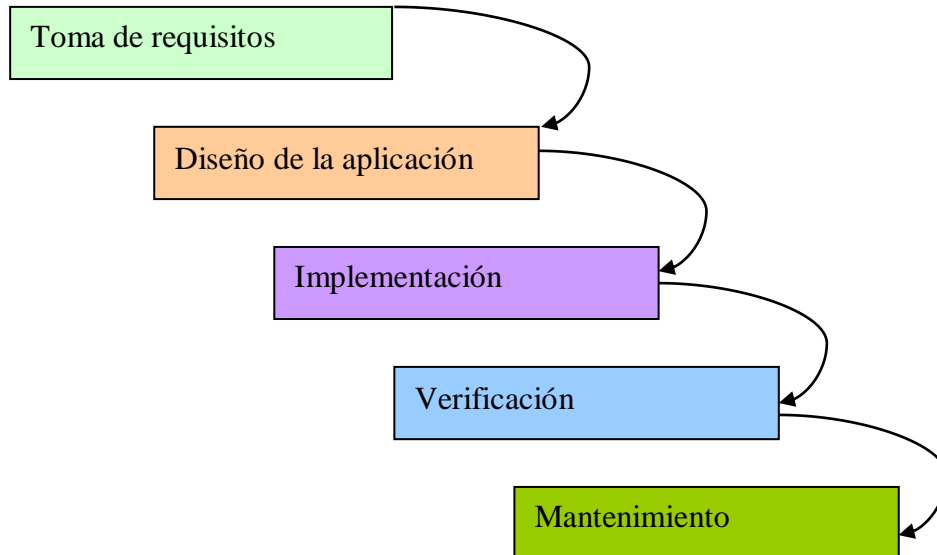


Para nuestra aplicación tenemos estas tres partes claramente diferenciadas según la estructura de carpetas mostrada en el punto 2.7. En la carpeta PHP se establecen todo aquel elemento necesario para la vista de la aplicación. Todo aquel código necesario para implementar la parte controladora se establecerá dentro de la carpeta controller. Mientras que para el modelo se utilizarán las clases dentro de la carpeta model y dao.

Nuestra parte vista estará desarrollado íntegramente 95% con etiquetas html y etiquetas angularJs, encargadas de hacer el binding con la parte controladora, solamente el resto será código php. El controlador de la aplicación está desarrollado con AngularJs, que como hemos indicado en apartados anteriores es un Framework JavaScript. Para finalizar la parte del modelo, son clases entitys que están dentro de la carpeta model y clases que se encargarán de realizar las peticiones a base de datos y obtener la información necesaria que se pasará a la parte controladora y esta a su vez tratar esos datos para ser pintados en la parte de la vista. Esta parte del modelo está desarrollado íntegramente en php.

## 3.11.2 Ciclo de vida de la aplicación.

Para la realización de la aplicación se ha utilizado un ciclo de vida en modo cascada. A continuación se muestra un esquema de las fases que conlleva el ciclo de vida de tipo cascada.



Este modelo se basa en un proceso de desarrollo de forma secuencial, en el que la siguiente tarea no comienza hasta que la anterior se ha completado correctamente.

La elección de este modelo se realiza en base a las siguientes ventajas encontradas:

- El tiempo que se invierte en las primeras fases en el diseño de la aplicación nos evitó problemas a la hora de solucionar problemas o rediseñar el producto estando ya en fases avanzadas de la codificación.
- Al ser un proyecto muy estructurado y con sus fases bien definidas, será más fácil comprender el proyecto.
- Es un modelo ideal, para proyectos que van a ser estables y donde sus requisitos son claros y estos no van a cambiar a lo largo del proceso de desarrollo.
- La cantidad de recursos necesarios para realizar este modelo son mínimos.
- Un punto muy a favor es la documentación que se va realizando en cada etapa del desarrollo, la cual será de gran utilidad para desarrolladores que se unan a posteriori, pudiendo comprender rápidamente el proyecto leyendo la documentación.

## 4. Despliegue de la aplicación

En el presente punto se pretende explicar como realizar el despliegue de la aplicación en un servidor web.

### 4.1 Preparativos previos

En el caso de que vayamos a instalar la aplicación en un entorno local, previamente tendremos que tener instalado el siguiente software:

- PHP, mínimo la versión 5.0.
- Base de datos MySQL mínimo la versión 5.5.
- Servidor web Apache.
- Conexión a Internet.

Para el caso de desplegar nuestra aplicación en un servidor remoto, tendremos que verificar que el servidor tenga instalado el anterior software.

## 4.2 Instalación de XAMP

Como hemos comentado en el punto anterior, previamente tendremos que tener instalado PHP, MySQL y Apache.

En nuestro caso, al trabajar bajo un sistema operativo Windows, nos decantamos por utilizar la distribución XAMPP, la cual incluye Apache, MariaDB, PHP y Perl. El paquete de instalación es bastante fácil de instalar y podemos descargarlo desde la siguiente dirección: <https://www.apachefriends.org/es/download.html> Una vez descargado, procedemos a instalarlo. La instalación es sumamente sencilla y sólo tendrás que ser consciente del directorio donde quedará instalado para poder acceder a él posteriormente. Tras la instalación, abrimos XAMPP y nos aparecerá el panel de control. Debemos iniciar los dos servicios necesarios para que funcione nuestra aplicación web. El servidor web Apache y el servidor MySQL, tras iniciarlos la interfaz de XAMPP debería quedar así:



Podemos probar si todo funciona correctamente, abriendo el navegador de internet e introducir la siguiente dirección: <http://localhost> o [127.0.0.1](http://127.0.0.1)

Para comprobar que MySQL está levantado podemos introducir en el navegador la siguiente url.

<http://localhost/phpmyadmin/>

## 4.3 Instalación de la base de datos.

Se aportará junto con el documento, un zip en el cual se encuentra la generación del modelo de base de datos.

Paso previo tendremos que generar un nuevo schema de nombre “hoficontrol” sobre el cual podemos lanzar los scripts de generación del modelo de base de datos.

Una vez generado el schema, tendremos que ir lanzando los scripts por el siguiente orden:

1. hoficontrol\_paises.sql
2. hoficontrol\_permisos.sql
3. hoficontrol\_tipo\_ingreso\_gasto.sql
4. hoficontrol\_usuarios.sql
5. hoficontrol\_permisos\_usuarios.sql
6. hoficontrol\_cuentas.sql
7. hoficontrol\_transacciones.sql

Lanzado los scripts, tendremos el modelo de base de datos generado con alguna información adicional en las tablas, necesarias para su puesta en funcionamiento, como un usuario por defecto, administrador de la aplicación y los permisos asociados a dicho usuario.

## 4.4 Paso previo instalación del software.

Para la instalación del software tendremos que verificar que tenemos php versión 5.0 o superior en nuestro servidor además de tener instalado un servidor web como apache que permita el despliegue y ejecución de aplicaciones con tecnología php. Podemos verificar este punto dirigiéndonos a la carpeta htdocs de nuestro servidor web apache. Nos crearemos una página php, la cual podemos llamar phpInfo.php y dentro estableceremos el siguiente contenido.


```
1  <?php
2
3  echo phpinfo();
4
5  ?>
6
```

Guardamos el contenido e introducimos en el navegador:

<http://localhost/phpInfo.php>

localhost/phpInfo.php

PHP Version 5.4.7



System	Windows NT ALEJANDRO 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
Build Date	Sep 12 2012 23:44:56
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	csconfig /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=.\obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgsql"
Server API	Apache 2.4 Handler Apache Lounge
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files (loaded)	

Una vez verificado que tenemos la versión correcta de php pasaremos a la instalación de la aplicación.

## 4.5 Instalación software

Crearemos dentro de la carpeta htdocs un directorio llamado HoFinControl. Dentro de este directorio tendremos que descomprimir el código aportado junto con el documento. Una vez descomprimido pasaremos a la configuración del acceso a la base de datos de la aplicación.

## 4.6 Configuración del acceso a la base de datos HoFinControl

Para configurar el acceso a la base de datos creada en el paso 2.9.2, nos dirigiremos al fichero Configuracion.php, ubicado en la ruta HoFinControl\dao. Dentro de este fichero tendremos que configurar los siguientes parámetros:

```
define("DB_HOST", "127.0.0.1");
define("DB_NAME", "HoFiControl;charset=utf8");
define("DB_USER", "dwes");
define("DB_PASSWORD", "abc123.");
```

Siendo:

**DB\_HOST.** Dirección ip del servidor de base de datos. Normalmente será localhost o 127.0.0.1

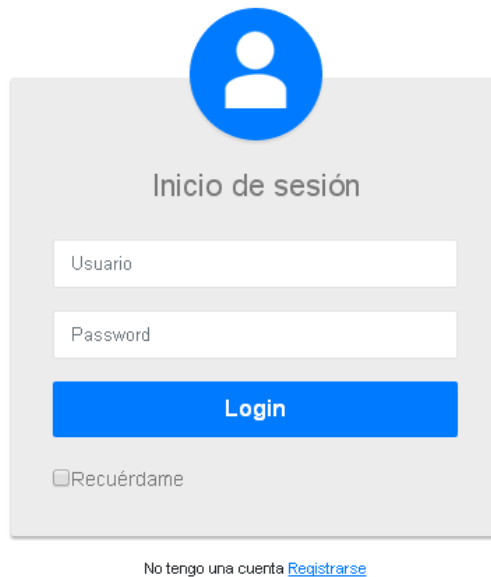
**DB\_NAME.** Nombre de la base de datos.

**DB\_USER.** Nombre de usuarios de la base de datos.

**DB\_PASSWORD.** Contraseña de la base de datos.

Una vez modificado dichos parámetros comprobaremos que podemos acceder correctamente a la aplicación. Introduciremos en un navegador la siguiente url,

<http://localhost/HoFinControl/php/index.php> y debería de aparecernos la pantalla de login de la aplicación.



Inicio de sesión

Usuario

Password

Login

☐ Recuérdame

No tengo una cuenta [Regístrase](#)

## 5. Propuesta de mejora o trabajos futuros.

Dentro del trabajo futuro en este proyecto pueden destacarse varias vertientes.

La primera de ellas es la mejora de la interfaz gráfica, con ayuda de una persona especialista puede hacerse que la aplicación tenga un acabado visual más profesional. Además de hacerla más responsive, dando la posibilidad a que se vea de manera correcta en resoluciones inferiores o en dispositivos móviles.

Otra mejora que cabe destacar es la posibilidad de que el usuario pueda recuperar sus credenciales en caso de olvido. Para ello sería necesario desarrollar una pantalla de recuperación de credenciales además de configurar un servidor smtp el cual se encargue de realizar el envío del correo de recuperación de credenciales al usuario.

Actualmente la aplicación tiene tablas maestras las cuales tienen que ser administradas a través de una gestión manual de los datos en base de datos. Para evitar esto, en una propuesta de mejora sería añadir unas pantallas de gestión de estas tablas, para que un usuario con permisos de administrador pueda tener la posibilidad de gestionarlas, sin la necesidad de tener que acceder a la base de datos y gestionarlas manualmente.



Por último puede ser interesante que los usuarios quieran sacar listados en otros formatos como pdf. Sería un punto a favor como trabajo futuro dar la posibilidad de que se puedan obtener dichos distados en otros formatos.

## 6. Bibliografía.

- <https://www.php.net/manual/es/>
- <https://www.w3schools.com/php/default.asp>
- <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html>
- <https://www.w3schools.com/sql/>
- <https://jquery.com/>
- <https://www.w3schools.com/jquery/>
- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
- <https://lucid.app>
- <https://es.wikipedia.org/wiki/AngularJS>
- <https://docs.angularjs.org/tutorial>
- <https://www.w3schools.com/angular/>
- <https://www.guidacode.com/2017/angularjs/explicacion-del-patron-mvc-en-angularjs/>
- <https://www.chartjs.org/docs/latest/>

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: HoFiControl



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo