Making Wishes

desarrollo aplicaciones web

Laura Martín Vicente

ies arcipreste de hita

Calle Poeta Manuel Martínez, 4,

Azuqueca de Henares

Contenido

[Contenido 1](#_Toc72701234)

[Ilustraciones: 3](#_Toc72701235)

[Tablas: 4](#_Toc72701236)

[1. Definición del Proyecto: 5](#_Toc72701237)

[1.1. Sectores Productivos. 5](#_Toc72701238)

[1.2. Descripción del problema. 5](#_Toc72701239)

[1.3. Objetivos. Especificación de Requisitos. 5](#_Toc72701240)

[1.4. Herramientas de diseño. 7](#_Toc72701241)

[1.5. Estudio de viabilidad: 8](#_Toc72701242)

[1.5.1 Viabilidad técnica: 8](#_Toc72701243)

[1.5.2 Viabilidad económica: 8](#_Toc72701244)

[1.5.3 Viabilidad operativa: 8](#_Toc72701245)

[1.5.4 Viabilidad legal: 8](#_Toc72701246)

[2. Planificación del Proyecto: 9](#_Toc72701247)

[2.1 Análisis inicial 9](#_Toc72701248)

[2.1.1 Requisitos funcionales 9](#_Toc72701249)

[2.1.2 Requisitos no funcionales 10](#_Toc72701250)

[2.2. Secuenciación de las fases del proyecto. 10](#_Toc72701251)

[2.3. Planificación de recursos y tiempos. 10](#_Toc72701252)

[3. Análisis del Proyecto: 12](#_Toc72701253)

[3.1. Casos de uso. 12](#_Toc72701254)

[3.2. Eventos de los casos de uso: 18](#_Toc72701255)

[3.2.2 Esquema de los eventos de los casos de uso: 23](#_Toc72701256)

[3.2. Modelo de datos. 26](#_Toc72701257)

[3.2.1. Esquema Entidad / Relación. 26](#_Toc72701258)

[3.2.2. Modelo relacional y normalización. 26](#_Toc72701259)

[3.2.3. Diccionario de datos. 28](#_Toc72701260)

[4. Diseño del Proyecto: 29](#_Toc72701261)

[4.1. Diseño físico de la BD. 29](#_Toc72701262)

[4.2. Estructura de los ficheros. 31](#_Toc72701263)

[4.3. Diseño de la estructura de clases y librerías (diagrama de clases). 32](#_Toc72701264)

[4.3. Diseño de la interfaz gráfica. 32](#_Toc72701265)

[5. Desarrollo y realización del Proyecto: 33](#_Toc72701266)

[5.1. Indicadores de calidad. 33](#_Toc72701267)

[5.2. Elaboración de una batería de pruebas para detectar errores. 33](#_Toc72701268)

[5.3. Evaluación y solución de incidencias. 33](#_Toc72701269)

[5.4. Evaluación y seguimiento de las actividades. 33](#_Toc72701270)

[6. Implantación del Proyecto: 34](#_Toc72701271)

[6.1. Plan de implantación. 34](#_Toc72701272)

[6.1.1 Back-end: 34](#_Toc72701273)

[6.1.2. Front-End 39](#_Toc72701274)

[6.2. Manual de instalación. 42](#_Toc72701275)

[6.2.1. Back-End: 42](#_Toc72701276)

[6.2.2. Front-End: 45](#_Toc72701277)

[6.3. Manual de usuario. 45](#_Toc72701278)

[7. Conclusiones Finales: 46](#_Toc72701279)

[7.1. Evaluación del proyecto: objetivos conseguidos y no alcanzados. 46](#_Toc72701280)

[7.2. Propuestas de mejora. 46](#_Toc72701281)

# **Ilustraciones**:

Ilustración 1 - Caso de uso usuario anónimo 12

Ilustración 2 - Caso de uso usuario logueado 13

Ilustración 3 - Caso de uso usuario administrador 13

Ilustración 4 - Evento landing 23

Ilustración 5 - Evento de registro. 23

Ilustración 6 - Evento de log in. 24

Ilustración 7- Evento de log out. 24

Ilustración 8 - Evento de modificar datos personales. 24

Ilustración 9 - Eventos de grupos. 24

Ilustración 10 - Evento administración de grupos. 25

Ilustración 11 - Evento de agregar contacto. 25

Ilustración 12 - Evento de agregar contacto. 25

Ilustración 13 - Evento seleccionar un deseo de un contacto 25

Ilustración 14 - Evento modificar la lista de deseos. 26

Ilustración 15 - Evento de administrar usuarios. 26

Ilustración 16 - Modelo relacional 27

Ilustración 17 - Esquema de la base de datos. 31

Ilustración 18 - Estructura de ficheros de back-end 32

Ilustración 19 - Estructura de ficheros de front-end 33

Ilustración 20 - Estudio de logotipo 34

Ilustración 21 - Estudio de color 34

Ilustración 22 - Logotipo definitivo 34

Ilustración 23 - Sketch landing 35

Ilustración 24 - Sketch log in 35

Ilustración 25 - Sketch registro 36

Ilustración 26 - Sketch home 36

Ilustración 27 - Sketch mis grupos 37

Ilustración 28 - Sketch modificar cuenta 37

Ilustración 29 – Sketch crear grupo 38

Ilustración 30 - Sketch texto legal 38

Ilustración 31 - Mock up landing 39

Ilustración 32 - Mock up log in 39

Ilustración 33 - Mock up regristro 40

Ilustración 34 - Mock up Inicio 40

Ilustración 35 - Mock up mis deseos 41

Ilustración 36 - Mock up Mis grupos 42

Ilustración 37 - Mock up configuración de la cuenta 43

Ilustración 38 - Mock up texto legal 44

Ilustración 39 - Mock up crear grupo 44

Ilustración 40 - Manual de implantación I 48

Ilustración 41 - Manual de implantación II 48

Ilustración 42 - Manual de implantación III 48

Ilustración 43 - Manual de implantación IV 49

Ilustración 44 - Manual de implantación V 49

Ilustración 45 - Manual de implantación VI 50

Ilustración 46 - Manual de implantación VII 50

Ilustración 47 - Manual de implantación VIII 51

Ilustración 48 - Manual de implantación IX 51

Ilustración 49 - Manual de implantación X 51

Ilustración 50 - Configuración de Apache. 52

Ilustración 51 - Plan de implantación front-end I 53

Ilustración 52 - Plan de implantación front-end II 53

Ilustración 53 - Plan de implantación front-end III 53

Ilustración 54 - Plan de implantación front-end IV 54

Ilustración 55 - Plan de implantación front-end V 54

Ilustración 56 - Plan de implantación front-end VI 54

Ilustración 57 - Plan de implantación front-end VII 55

Ilustración 58 - Plan de implantación front-end VIII 55

Ilustración 59 - Manual de instalación I 56

Ilustración 60 - Manual de instalación II 56

Ilustración 61 - Manual de instalación IV 56

Ilustración 62 - Manual de instalación V 57

Ilustración 63 - Manual de instalación VI 57

Ilustración 64 - Manual de instalación VII 57

Ilustración 65 - Manual de instalación VIII 58

Ilustración 66 - Manual de instalación IX 58

# Tablas:

Tabla 1 - Planificación de recursos y tiempos 11

Tabla 2 - Diccionario de datos. 29

# 1. Definición del Proyecto:

## 1.1. Sectores Productivos.

Esta aplicación está destinada a pequeños usuarios. Al ser una red social no depende de ninguna gran empresa ya que la idea, realización y demanda de la aplicación corre por mi cuenta. La idea surgió a raíz de unas Navidades en la que mi familia no sabía que regalos hacer a algunos miembros de la misma y pensé que estaría bien tener un lugar dónde poder registrar todos esos pequeños caprichos que nos haría mucha ilusión tener.

A raíz de esto surgió la idea y, en cierto modo, necesidad interna de desarrollar esta aplicación web.

## 1.2. Descripción del problema.

Making Wishes es una red social para conectar a gente. Se trata de una aplicación web en la que el usuario podrá guardar una lista de sus cosas más deseadas para que sus amigos sepan qué regalarle. Además de ello también habrá una sección en la que el usuario podrá crear grupos para eventos especiales y ahí postear una lista de deseos en común para muchas personas.

## 1.3. Objetivos. Especificación de Requisitos.

Los objetivos de Making Wishes es que los usuarios de esta aplicación tengan acceso a las listas de deseos de sus contactos para así siempre acertar con los regalos.

Los usuarios tendrán que configurar esas listas de regalos desde su perfil posteando los elementos con una serie de especificaciones tales como nombre, breve descripción y link.

A su vez, estos usuarios podrán crear grupos para eventos, como por ejemplo una boda, en la que postear su lista de regalos para que los invitados sepan que es lo que se necesita. Estos grupos tendrán un nombre, una imagen y una lista interna de deseos. Además, podrás ver los participantes que integren ese grupo.

Por parte del usuario no serán necesarios grandes requisitos de hardware porque la aplicación está diseñada para ser abierta en un navegador web. Por lo tanto, el usuario necesitará un equipo con capacidad para instalar un navegador web y conexión a internet.

Por parte del desarrollador, a nivel de hardware se necesitará un ordenador con unos requisitos de hardware medios, conexión a internet para sincronizar el proyecto en GIT, poder consultar los diferentes manuales oficiales de las tecnologías que se van a usar y poder desplegar la aplicación web.

Para el diseño de la web, el diseño del logotipo, imágenes y retoque de las mismas se va a combinar el uso del ordenador con software destinado a estas tareas y el uso de un IPad, que ya se posee. Por lo que no incrementará los costes.

La aplicación se va a desarrollar usando VUE.js para la parte de front-end y PHP con el framework de SLIM para la parte de back-end.

Se ha elegido VUE.js por la modularibilidad de componentes que aporta además de usar muchas de las librerías que tiene a nuestra disposición, como VUETIFY o VUELIDATE para simplificar la codificación.

Se ha elegido SLIM framework por la simplicidad que tiene para la realización de APIs REST además del conocimiento previo que se obtuvo de PHP por lo que la adaptación a este framework será mínima.

El motor de la base de datos será MySQL. Se ha elegido este motor por ser multiplataforma, es decir, funciona en la mayoría de sistemas operativos que hay en el mercado. Además de que el mantenimiento de una base de datos MySQL es muy sencilla y escalable, por lo que si se desea aumentar la funcionalidad a esta aplicación la base de datos no nos supondrá ningún problema, tan sólo habrá que añadir las tablas o los nuevos campos requeridos.

La aplicación se va a desplegar en XAMPP. Este software contiene todo lo necesario para desplegar el proyecto. Además, es compatible con todos los sistemas operativos.

Con respecto al tiempo de aprendizaje de las tecnologías nuevas que se van a usar especificamos lo siguiente:

* Para VUE.js será necesaria una previa preparación de tres semanas para la correcta interiorización de conceptos nuevos y aprender a usar el lenguaje y algunas de las librerías que se vayan a establecer.

Se estudiará este framework de JavaScript mediante el manual oficial que proveen ([*https://vuejs.org/*](https://vuejs.org/)), así como de un curso gratuito impartido mediante vídeos en YouTube.

* Para SLIM framework, se estima un tiempo de aprendizaje de al menos una semana para familiarizarse completamente con el uso y registro de rutas y usar esta tecnología correctamente.

Se estudiará este framework mediante el manual oficial que proveen *(*[*https://www.slimframework.com*](https://www.slimframework.com/)).

## 1.4. Herramientas de diseño.

La herramienta que se va a usar para realizar el código, tanto de front-end como de back-end será Visual Studio Code por ser una herramienta de acceso libre, la familiarización que se tiene con la misma, la facilidad que nos proporciona con las extensiones y la sincronización directa que ofrece con Git.

Para el los sketch de las páginas, el diseño y realización del logotipo se usará Adobe Sketch. Para los wireframe de las páginas se usará la web [*https://moqups.com/*](https://moqups.com/)*.* Y para tareas más complejas, así como realización o retoque fotográfico se usará Adobe Photoshop CS6.

Con respecto a la sincronización de la aplicación web para poder trabajar desde diferentes ordenadores se usará la herramienta Git. En este repositorio podremos tener un control de cómo va la aplicación, las actualizaciones o la adicción de distintas características mediante los commit o incluso se podrá volver atrás en las versiones si hay algún error imprevisto sin ninguna complicación.

## 1.5. Estudio de viabilidad:

### 1.5.1 Viabilidad técnica:

Actualmente se disponen de todo el hardware como software para comenzar a desarrollar el proyecto, sin embargo, los conocimientos sobre las tecnologías se van a ir adquiriendo conforme se avance con el proyecto. Por lo que se estima alrededor de cinco semanas más a la fecha de entrega para la adquisición de dichos conocimientos y la realización adecuada del proyecto. Además, teniendo en cuenta los posibles fallos que puedan aparecer en la aplicación web se sugiere ampliar el plazo de entrega dos semanas más dando dejando así el plazo de entrega máximo el 24 de junio de 2021.

### 1.5.2 Viabilidad económica:

En este proyecto va a estar ocupada una persona durante un mes y medio, esto supone un coste estimado de 5.000€. Esta persona ya dispone de material y licencias por lo que por esta parte no incrementará el coste.

Esta empresa no dispone de servidor, por lo que se alquilará un espacio en un servidor remoto. El espacio en un servidor remoto estima un coste de 75€/mes.

Al ser una aplicación de nueva creación no tiene disponible un dominio, por lo que habría que comprar uno. Un pack de dominios que contiene .eu y .com está valorado en torno a 20€ anuales.

### 1.5.3 Viabilidad operativa:

La aplicación va a estar preparada para asumir los requerimientos que pone la empresa y además estará preparada por si en un futuro la empresa quiere añadir nuevas funcionalidades.

### 1.5.4 Viabilidad legal:

La base de datos de la aplicación va a contener información de carácter sensible de los usuarios, por lo que se tendrá que tener en cuenta la ley de protección de datos.

# 2. Planificación del Proyecto:

## 2.1 Análisis inicial

### 2.1.1 Requisitos funcionales

Esta aplicación va a tener tres tipos de usuarios: usuario anónimo, usuario logueado y usuario administrador. Hay que diferenciar claramente las acciones que puede realizar cada uno de ellos:

1. Usuario anónimo:
   1. Podrá ver tres páginas. Nada más acceder al dominio se le mostrará una landing page. Esta página da una bienvenida a los usuarios y tiene dos botones, acceder a la aplicación (log in) y registrarse (sign in).
   2. Log in: Formulario de acceso a la aplicación. Pedirá el correo electrónico con el que te has registrado y la contraseña.
   3. Sign in: Formulario para darse de alta en la aplicación. Pedirá datos básicos como nombre, apellidos, correo electrónico, contraseña…
2. Usuario logueado:
   1. Podrá ver cinco páginas:
      1. Página de inicio, donde aparece un tablero con 2 tarjetas. Grupos y lista de deseos.
      2. Grupos a los que pertenece.
      3. Su lista de deseos.
      4. La página de un grupo en concreto.
      5. Panel de administración de su cuenta.
   2. Podrá desloguearse.
3. Usuario administrador:
   1. Tendrá acceso a dos páginas:
      1. Página de modificación o eliminación de usuarios.
      2. Página de modificación o eliminación de grupos.
   2. Hacer administrador a otra persona.

### 2.1.2 Requisitos no funcionales

La aplicación web deberá poder usarse en cualquier navegador web en cualquier dispositivo. Los requisitos mínimos de la aplicación web serán los requeridos por el navegador.

La interfaz deberá ser sencilla y clara para facilitar el manejo a todo tipo de usuarios, desde los usuarios más avanzados a los que tengan poca experiencia.

Se requiere un sistema gestor de bases de datos para almacenar los datos. Dichos datos podrán ser modificados o borrados por el administrador.

## 2.2. Secuenciación de las fases del proyecto.

## 2.3. Planificación de recursos y tiempos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tarea** | | **Duración** | **Inicio** | **Fin** |
| **Redacción del proyecto** | | 7 días | 06/05 | 13/05 |
| Recursos: | 1 persona, 1 ordenador, Microsoft Word. | | | |
| **Análisis y diseño de la base de datos** | | 1 día | 05/05 | 06/05 |
| Recursos: | 1 persona, 1 cuaderno, 1 bolígrafo | | | |
| **Creación de la base de datos** | | 1 día | 05/05 | 06/05 |
| Recursos: | 1 persona, 1 ordenador, Xampp, PhpMyAdmin | | | |
| **Análisis de la aplicación** | | 3 días | 07/05 | 09/05 |
| Recursos: | 1 persona, 1 ordenador, Microsoft Word | | | |
| **Diseño gráfico de la aplicación** | | 7 días | 09/05 | 15/05 |
| Recursos: | 1 persona, 1 ordenador, 1 iPad, PS CS6, Adobe Sketch | | | |
| **Desarrollo de la interfaz** | | 30 días | 15/05 | 15/06 |
| Recursos: | 1 persona, 1 ordenador, Visual Studio Code, Git Hub. | | | |
| **Implementación de funcionalidades** | | 30 días | 15/05 | 15/06 |
| Recursos: | 1 persona, 1 ordenador, Visual Studio Code, Xampp, PhpMyAdmin, Git Hub | | | |
| **Etapa de pruebas** | | 4 días | 15/06 | 19/06 |
| Recursos: | 1 persona, 1 ordenado, Visual Studio Code, Xampp, PhpMyAdmin | | | |
| **Finalización de la memoria** | | 1 día | 19/06 | 20/06 |
| Recursos: | 1 persona, 1 ordenador, Microsoft Word | | | |
| Entrega final: | **24/06/2021** | | | |

Tabla 1 - Planificación de recursos y tiempos

# 3. Análisis del Proyecto:

## 3.1. Casos de uso.

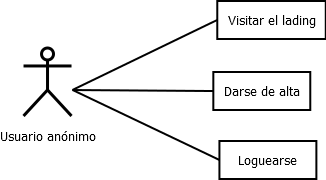


Ilustración 1 - Caso de uso usuario anónimo

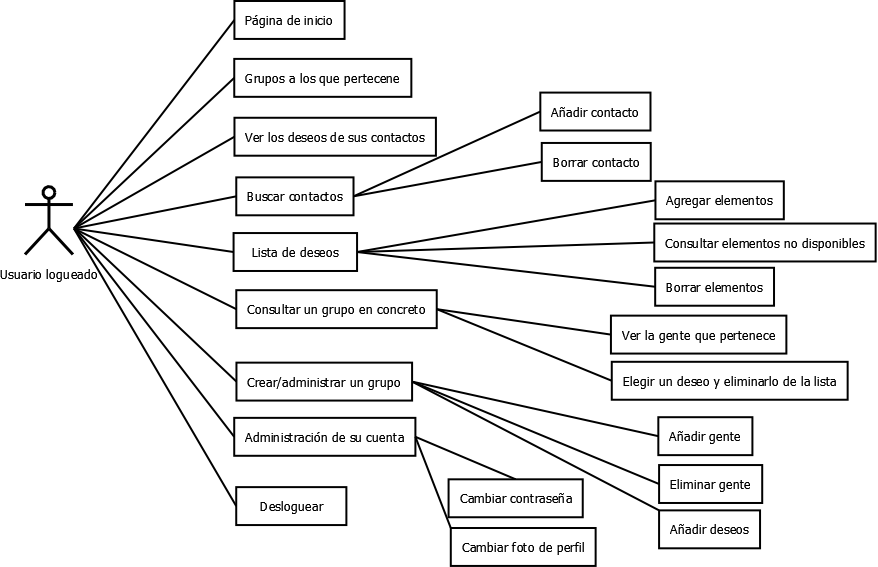


Ilustración 2 - Caso de uso usuario logueado

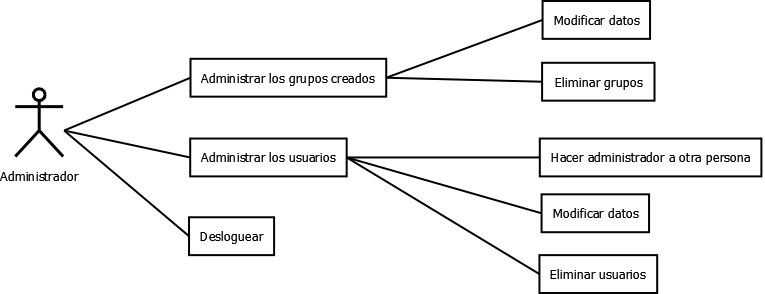


Ilustración 3 - Caso de uso usuario administrador

#### **Caso de uso 1 - Visitar landing page:**

Funcionalidad: Nada más acceder al dominio, los usuarios se encontrarán con una página de bienvenida. En ella habrá información sobre la web y además se encontrarán 2 botones. Estos botones redirigirán al usuario a acceder a la web o registrarse.

Actores: usuarios no logueados.

Precondiciones: no estar logueados en el sistema.

#### **Caso de uso 2 – Darse de alta:**

Funcionalidad: El usuario anónimo en esta página encontrará un formulario con un botón. Cuando rellene el formulario y todos los datos sean correctos podrá pulsar al botón para registrarse. \*\* Se enviará un correo electrónico de confirmación y podrá acceder a la plataforma una vez esté validado. \*\*

Actores: usuarios no logueados.

Precondiciones: no estar logueado ni estar dado de alta en el sistema.

#### **Caso de uso 3 - Loguearse:**

Funcionalidad: \*\* Cuando el usuario valide su cuenta mediante el correo electrónico que se le manda\*\* podrá entrar en la aplicación rellenando un formulario con el correo electrónico y la contraseña que introdujo cuando se registró.

Actores: usuario no logueado, usuario administrador.

Precondiciones: estar dado de alta en el sistema.

#### **Caso de uso 4 – Página de inicio:**

Funcionalidad: Una vez el usuario esté logueado este mismo entrará a una página de inicio. En esta página aparecerán los grupos a los que perteneces o un mensaje informativo de que aún no perteneces a ningún grupo.

Actores: usuario logueado.

Precondiciones: estar logueado.

#### **Caso de uso 5 – Ver grupos a los que pertenece:**

Funcionalidad: Cuando el usuario pinche en el botón de ver los grupos a los que pertenece se cargarán todos ellos en unas tarjetas. Cuando se pinche en una de estas tarjetas se redirigirá a una página específica de este grupo con la información correspondiente. Esta información se trata de la lista de deseos que el administrador de ese grupo ha establecido con los deseos que están disponibles y los que ya no, los usuarios que pertenecen a esa lista de deseos, una descripción del grupo y una foto representativa.

Si no se pertenece a ningún grupo todavía se informará al usuario y aparecerá un link para crear uno.

Actores: usuario logueado.

Precondiciones: estar logueado, pertenecer a un grupo.

#### **Caso de uso 6 – Buscar contactos:**

Funcionalidad: El usuario logueado tendrá en la parte superior una pestaña para listar sus contactos. Además, cuando pinche en esa pestaña, en la parte superior de la página habrá un buscador de personas. En ese buscador, se podrá escribir parte o el nombre completo de una persona y darle a buscar. Si aparece el contacto que deseamos podremos darle a añadir amigo por lo que se le mandará una petición de amistad.

Si por el contrario queremos eliminar un amigo, dentro de la pestaña de listar contactos aparecerá esta opción al lado de cada contacto.

Actores: usuario logueado.

Precondiciones: estar logueado.

#### **Caso de uso 7 – Ver los deseos de sus contactos:**

Funcionalidad: Cuando se tenga agregado un contacto, si pulsamos en la pestaña de “mis contactos” aparecerá un listado con todos tus contactos. Dentro de estos contactos se podrá seleccionar uno en concreto y ver su lista de deseos.

Actores: usuario logueado.

Precondiciones: estar logueado, tener contactos y que el contacto tenga lista de deseos.

#### **Caso de uso 8: Aceptar o denegar solicitud de amistad:**

Funcionalidad: Cuando un usuario mande a otro una petición de amistad se reflejará en su perfil. El usuario tendrá dos opciones:

1. Aceptar la petición de amistad y esos dos contactos tendrán acceso a sus datos personales en ambas cuentas.
2. Rechazar la petición de amistad. Los dos contactos seguirán sin ser amigos.

Actores: usuario logueado.

Precondiciones: estar logueado y tener pendiente una petición de amistad.

#### **Caso de uso 9 – Administrar su lista de deseos:**

Funcionalidad: El usuario podrá administrar su lista de deseos. Esto se refiere a que podrá añadir o borrar deseos de su lista para que aparezcan o no disponibles para sus contactos.

Actores: usuario logueado.

Precondiciones: estar logueado.

#### **Caso de uso 10 – Administrar un grupo:**

Funcionalidad: El usuario podrá crear un grupo. Cuando cree un grupo este usuario pasará a ser administrador del grupo. El grupo se conformará de 4 elementos: fotografía (opcional, si no se proporciona una imagen se pondrá una estándar), descripción, lista de deseos exclusiva del grupo y personas que lo conforman.

El administrador podrá modificar, actualizar o borrar cualquiera de estos datos en cualquier momento. – Caso de uso 14 –

Actores: usuario logueado.

Precondiciones: estar logueado.

#### **Caso de uso 11 – Consultar un grupo:**

Funcionalidad: Cuando un administrador de un grupo invite a un usuario a un grupo; el usuario podrá ver visualizar el avatar, la descripción, la lista de deseos y las personas que lo conforman.

* El usuario al pinchar en una persona de la lista que conforman el grupo podrá agregarlo de amigo.
* El usuario podrá interactuar con la lista de regalos. Tendrá un botón en el que al pinchar en él solicite dicho deseo como suyo y a los demás participantes les salga como no disponible ese deseo.

Actores: usuario logueado.

Precondiciones: estar logueado en el sistema y pertenecer a un grupo.

#### **Caso de uso 12 – Administrar su cuenta:**

Funcionalidad: El usuario podrá cambiar su avatar y contraseña.

Actores: usuario logueado, usuario administrador.

Precondiciones: estar logueado en el sistema.

#### **Caso de uso 13 - Desloguearse:**

Funcionalidad: El usuario podrá cerrar sesión.

Actores: usuario logueado y usuario administrador.

Precondiciones: estar logueado en el sistema.

#### **Caso de uso 14 – Administrar grupos:**

Funcionalidad: El usuario con poder para administrar un grupo podrá cambiar el avatar, cambiar la descripción, añadir o eliminar personas a ese grupo y añadir o eliminar deseos de la lista del grupo.

Además, el usuario administrador general podrá eliminar grupos.

Actores: usuario administrador de grupo y usuario administrador.

Precondiciones: estar logueado y haber creado un grupo o ser administrador general.

#### **Caso de uso 15 – Administrar usuarios:**

Funcionalidad: el usuario administrador podrá cambiar el nombre de otros usuarios o eliminar su cuenta.

Actores: usuario administrador.

Precondiciones: estar logueado, y ser administrador general.

## 3.2. Eventos de los casos de uso:

#### **Evento caso de uso 1 - Visitar landing page:**

Nada más acceder a la aplicación web se visualizará una web de bienvenida. Esta web lanzará dos eventos: redirección a la página de iniciar sesión y redirección a la página de registro. La redirección la realizará pulsando uno de los dos botones que se encuentran en la página.

#### **Evento caso de uso 2 – Darse de alta:**

El usuario rellenará un formulario en el que indicará su nombre, sus apellidos, su correo electrónico e introducirá la contraseña dos veces. Si las contraseñas son iguales se lanzará una llamada a la API que conectará con la base de datos.

La API comprobará que el correo electrónico de ese usuario no está registrado ya. Si no está registrado se insertará en la base de datos, si por el contrario sí que está registrado se lanzará una excepción y se retornará un código y mensaje de error a cliente para que se informe al usuario.

#### **Evento caso de uso 3 - Loguear:**

El usuario rellenará un formulario con su correo electrónico y la contraseña que eligió en el momento de registrarse y pulsará el botón de enviar. Este botón llamará a la API que contrastará con la base de datos si el correo electrónico y la contraseña pertenecen a algún usuario. Si valida se reenviará a la página de inicio y si no valida se mostrará un mensaje que indica que el usuario o la contraseña no son correctos.

#### **Evento caso de uso 4 – Página de inicio:**

Una vez el usuario esté logueado este mismo entrará a una página de inicio. En esta página aparecerá una bienvenida y en la parte superior un menú. Cada elemento del menú redirigirá a una página en concreto.

#### **Evento caso de uso 5 – Ver grupos a los que pertenece:**

Cuando el usuario pinche en el elemento del menú “Mis grupos” se mandará una petición a la API que devuelva todos los grupos a los que pertenece el usuario y se cargarán en unas tarjetas. Si no tiene ningún grupo todavía se informará.

**Evento de caso de uso 6 – Ver información de un grupo.**

Cuando el usuario pinche en una tarjeta se llamará a la API y se cargará toda la información de ese grupo. Toda la información incluye nombre, imagen, gente que lo compone y la lista de deseos.

**Evento de caso de uso 7 – Seleccionar un deseo.**

El usuario podrá seleccionar un deseo, tanto de la lista de deseos de un grupo como la de un usuario particular. Cuando pinche sobre ese deseo se llamará a la API y el deseo pasará a tener un estado de no disponible. El deseo aparecerá en la vista, pero en el lado de deseos no disponibles.

#### **Evento caso de uso 8 – Buscar contactos:**

En la parte superior de la página habrá un buscador de personas. En ese buscador, se podrá escribir parte o el nombre completo de una persona y darle a buscar.

Cuando se dé a buscar se llamará a la API y aparecerá una lista de todos los usuarios que contengan los caracteres buscados. Si no coindice ninguno se informará.

Si aparece el contacto que deseamos podremos darle a añadir amigo por lo que se le mandará una petición de amistad. Se llamará a la API y se actualizarán las notificaciones.

Si por el contrario queremos eliminar un amigo, dentro de la pestaña de listar contactos aparecerá esta opción al lado de cada contacto. Se llamará a la API y se eliminará la relación de los dos usuarios.

#### **Evento caso de uso 9 – Ver mis contactos:**

Cuando un usuario quiera ver la lista de deseos de otro tendrá que pinchar en mis contactos y seleccionar el usuario en concreto que quiere consultar. Se producirá una llamada a la API que traerá los datos del usuario y además, su lista de deseos.

#### **Evento caso de uso 10: Aceptar o denegar solicitud de amistad:**

Cuando un usuario tenga pendiente una petición de amistad se reflejará en su perfil. El usuario tendrá dos opciones:

1. Aceptar la petición de amistad y esos dos contactos tendrán acceso a sus datos personales en ambas cuentas. Se producirá llamada a la API y se actualizará la tabla de contactos con esta nueva relación y la notificación pasará a vista.
2. Rechazar la petición de amistad. Los dos contactos seguirán sin ser amigos. Se producirá una llamada a la API y la notificación pasará a vista.

#### **Evento caso de uso 11 – Administrar su lista de deseos:**

El usuario tendrá un apartado en el que podrá gestionar su perfil. Una de las cosas que podrá hacer es gestionar su lista de deseos. En ella podrá añadir o borrar elementos.

Cuando quiera añadir elementos tendrá que rellenar un formulario. La web llamará a la API y escribirá estos nuevos datos en la base de datos.

Por el contrario, si lo que quiere es eliminar un elemento de la lista de deseos se llamará a la API y se borrará ese deseo de la base de datos.

#### **Evento caso de uso 12 – Administrar un grupo:**

Cuando un usuario quiera crear un grupo este mismo tendrá que rellenar un formulario con el nombre, descripción, lista de deseos, personas que lo conforman e imagen. La web llamará a la API para que lo inscriba en la base de datos.

Además, el usuario administrador de ese grupo en cualquier momento podrá añadir o borrar deseos e integrantes. Se llamará a la API y se actualizará la base de datos.

#### **Evento caso de uso 13 – Consultar un grupo:**

Cuando un usuario pertenezca a un grupo podrá pinchar en la tarjeta de un grupo para visualizar su contenido. Se realizará una llamada a la API y se cargarán todos los datos de ese grupo (avatar, descripción, lista de deseos y personas que lo conforman).

Además, el usuario podrá pinchar en una persona de la lista que conforman el grupo para agregarlo como amigo. Se conectará con la API y se mandará y se escribirá en la tabla de notificaciones la petición de amistad si no existe ya.

El usuario podrá interactuar con la lista de regalos. Tendrá un botón en el que al pinchar en él solicite dicho deseo como suyo y a los demás participantes les salga como no disponible ese deseo. Se conectará con la API y se actualizará la base de datos poniendo el regalo a no disponible.

#### **Evento caso de uso 14 – Administrar su cuenta:**

El usuario podrá cambiar su avatar y contraseña. Tendrá que rellenar un formulario y se realizará una llamada a la API y se actualizará uno o los dos campos.

#### **Evento caso de uso 15 - Desloguearse:**

El usuario podrá cerrar sesión. Se realizará una llamada a la API y se borrará la sesión quedando así deslogueado.

#### **Evento caso de uso 16 – Administrar grupos:**

El usuario con poder para administrar un grupo podrá cambiar el avatar, cambiar la descripción, añadir o eliminar personas a ese grupo y añadir o eliminar deseos de la lista del grupo. Se realizará una llamada a la API y se actualizarán los cambios en la base de datos.

Además, el usuario administrador general podrá eliminar grupos. Se realizará una llamada a la API y se borrará el grupo de la base de datos.

#### **Evento caso de uso 17 – Administrar usuarios:**

Al usuario administrador le saldrán todos los miembros de la plataforma y podrá cambiar los datos de otros usuarios o eliminar su cuenta. Se realizará una llamada a la API y se actualizarán los cambios reflejados.

### 3.2.2 Esquema de los eventos de los casos de uso:

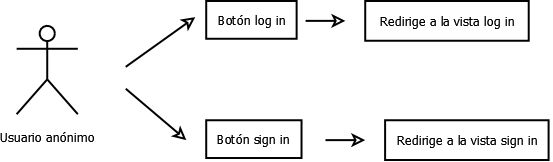


Ilustración 4 - Evento landing



Ilustración 5 - Evento de registro.

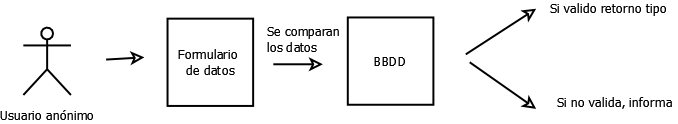


Ilustración 6 - Evento de log in.

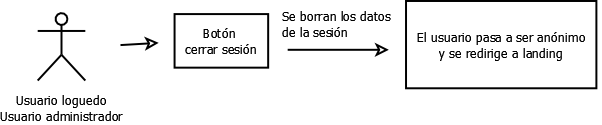


Ilustración 7- Evento de log out.

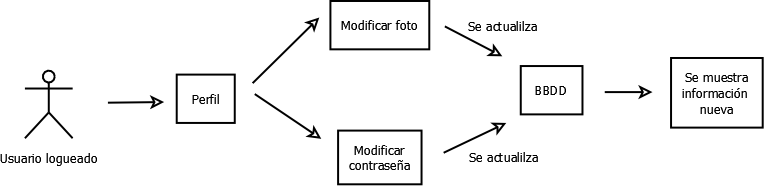


Ilustración 8 - Evento de modificar datos personales.

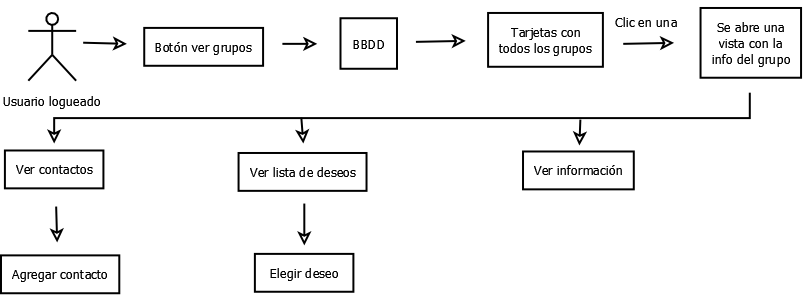


Ilustración 9 - Eventos de grupos.

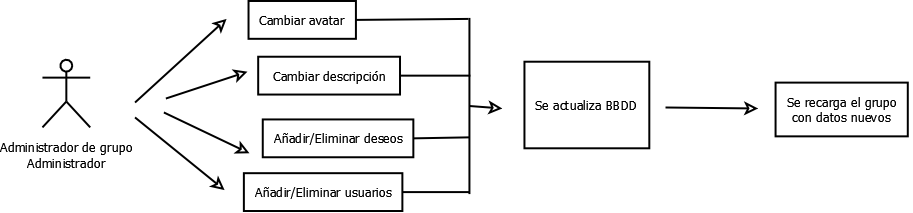


Ilustración 10 - Evento administración de grupos.

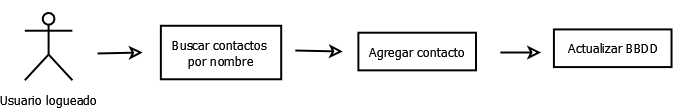


Ilustración 11 - Evento de agregar contacto.

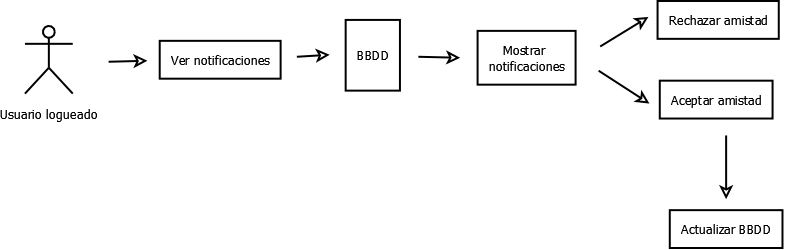


Ilustración 12 - Evento de agregar contacto.

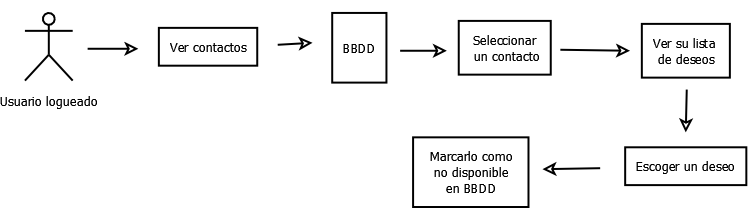


Ilustración 13 - Evento seleccionar un deseo de un contacto

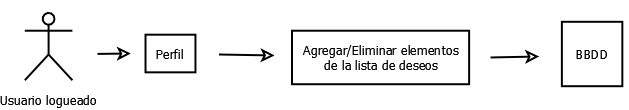


Ilustración 14 - Evento modificar la lista de deseos.

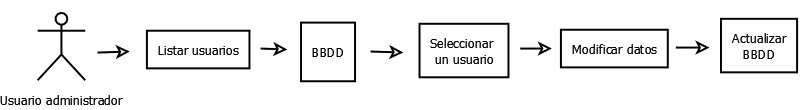


Ilustración 15 - Evento de administrar usuarios.

## 3.2. Modelo de datos.

### 3.2.1. Esquema Entidad / Relación.



### 

### 3.2.2. Modelo relacional:

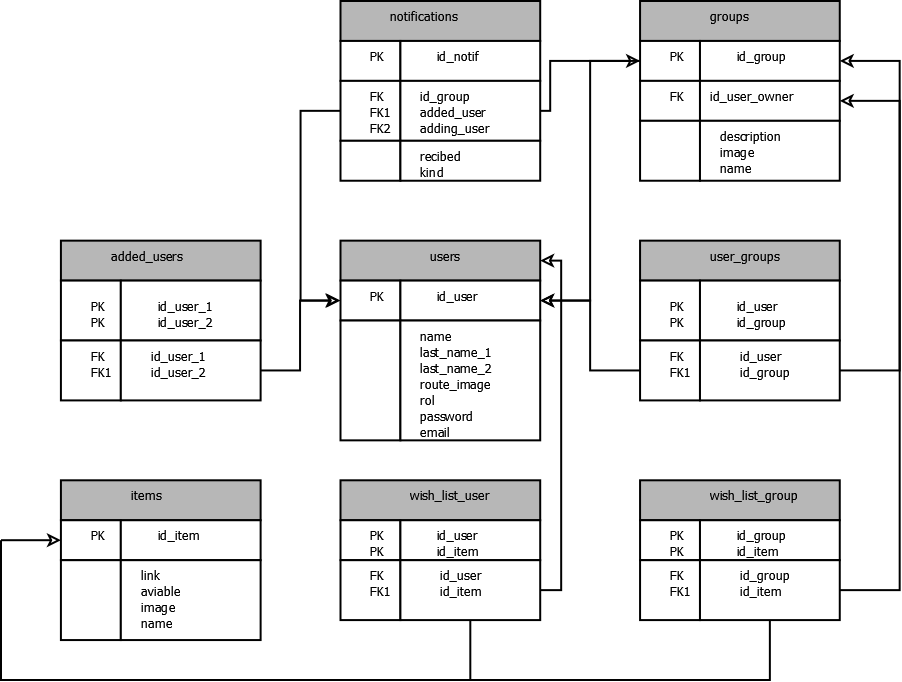


Ilustración - Modelo relacional

### 3.2.3 Normalización:

Se han realizado 8 tablas estructuradas de la siguiente forma:

Tabla **USERS** con los atributos: id\_user, name, last\_name\_1, last\_name\_2, email, password, route\_image, rol.

Clave primaria: **id\_user.**

Tabla **GROUPS** con los atributos: id\_group, id\_user\_owner, name, route\_image, description.

Clave primaria: id\_group.

Clave ajena: id\_user\_owner referencia a la tabla USERS.

Tabla **NOTIFICATIONS** con los atributos: id\_notif, added\_user, adding\_user, id\_group, kind, recibed.

Clave primaria: id\_notif.

Claves ajenas:

* added\_user: referencia a la tabla USERS.
* adding\_user: referencia a la tabla USERS.
* id\_group: referencia a la tabla GROUPS.

Tabla **ADDED\_USERS** con los atributos: id\_user\_1, id\_user\_2.

Clave primaria: id\_user\_1, id\_user\_2.

Claves ajenas:

* id\_user\_1: referencia a la tabla USERS.
* id\_user\_2: referencia a la tabla USERS.

Tabla **USERS\_GROUPS** con los atributos: id\_user, id\_group.

Clave primaria: id\_user, id\_group.

Claves ajenas:

* id\_user: referencia a la tabla USERS.
* id\_group: referencia a la tabla GROUPS.

Tabla **ITEMS** con los atributos: id\_item, name, description, link, avaiable.

Clave primaria: id\_item.

Tabla **WISH\_LIST\_USERS** con los atributos: id\_user, id\_item.

Clave primaria: id\_user, id\_item.

Claves ajenas:

* id\_user: referencia a la tabla USERS.
* id\_item: referencia a la tabla ITEMS.

Tabla **WISH\_LIST\_GROUPS** con los atributos: id\_group, id\_item.

Clave primaria: id\_group, id\_item.

Claves ajenas:

* id\_group: referencia a la tabla GROUPS.
* id\_item: referencia a la tabla ITEMS.

### 3.2.3. Diccionario de datos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| **id\_notif** | Number | 7 | Clave única que identifica numéricamente las notificaciones.  (Autoincremental) |
| **added\_user** | Number | 7 | ID del usuario que agrega a otro. |
| **adding\_user** | Number | 7 | ID del usuario agregado. |
| **ig\_group** | Number | 7 | Clave numérica que identifica numéricamente los grupos.  (Autoincremental) |
| **rol** | Enum | 2 valores:  Admin|wisher | Tipo de usuario. Puede ser admin o wisher. |
| **kind** | Enum | 2 valores:  User|Group | Tipo de notificación, puede ser de grupos o de usuarios |
| **recibed** | Enum | 2 valores:  yes|no | Indica si la notificación ha sido vista. |
| **id\_user\_owner** | Number | 7 | ID del usuario propietario del grupo. |
| **name** | Varchar | 30 | Nombre del grupo o del usuario. |
| **route\_image** | Varchar | 40 | Ruta donde está guardada la foto |
| **description** | Varchar | 1000 | Breve descripción del grupo o del deseo. |
| **id\_user** | Number | 7 | Clave numérica que identifica numéricamente los usuarios.  (Autoincremental) |
| **last\_name\_1** | Varchar | 30 | Primer apellido del usuario. |
| **last\_name\_2** | Varchar | 30 | Segundo apellido del usuario. |
| **email** | Varchar | 40 | Correo electrónico del usuario. |
| **password** | Varchar | 30 | Contraseña cifrada del usuario. |
| **id\_user\_1** | Number | 7 | ID del un usuario que es amigo de otro. |
| **id\_user\_2** | Number | 7 | ID del un usuario que es amigo de otro. |
| **id\_item** | Number | 7 | Clave numérica que identifica numéricamente los deseos.  (Autoincremental) |
| **link** | Varchar | 255 | Link que redirige al deseo. |
| **available** | Enum |  | Indica si el deseo sigue disponible. |

Tabla - Diccionario de datos.

# 4. Diseño del Proyecto:

## 4.1. Diseño físico de la BD.

|  |
| --- |
| {{Script de la base de datos}} |

Después de la creación, la base de datos quedará de la siguiente manera:

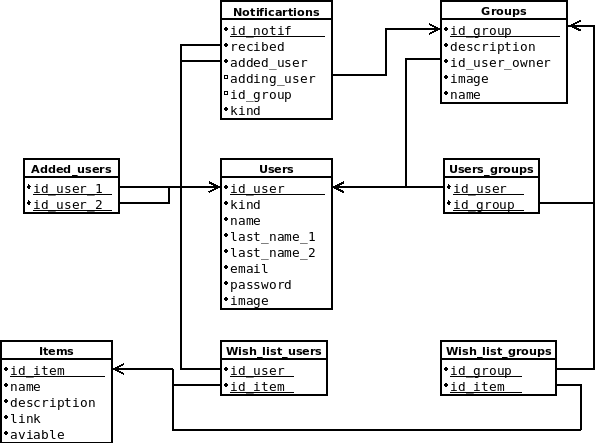


Ilustración - Esquema de la base de datos.

## 

## 4.2. Estructura de los ficheros.

|  |  |
| --- | --- |
| Ilustración - Estructura de ficheros de back-end | La estructura de ficheros la de parte de back-end se estructura de la siguiente manera:  En la carpeta “public” hay un archivo llamado index.php en el que se establecen todas rutas con sus correspondientes funciones. Como iba a ser un archivo muy extenso y difícil de mantener se ha creado una carpeta “routes” en la que se dividirán las rutas en diferentes archivos pequeños. Después estas rutas se cargarán todas en el index.php consiguiendo así un código estructurado y mantenible.  Además, en la dentro de la carpeta public encontramos otra carpeta llamada upload dónde se guardan las imágenes que subirán los usuarios. |
| En la carpeta “vendor” se incluyen los archivos necesarios que conforman el framework de SLIM. | |
| Ilustración - Estructura de ficheros de front-end | La estructura de ficheros de la parte de front-end se estructura de la siguiente manera:  En la carpeta “node\_modules” se encuentran todas las librerías necesarias para que el proyecto funcione.  En la carpeta “public” se encuentra “index.html” que es el archivo dónde se monta la aplicación.  En la carpeta “scr” se encuentran los archivos sobre los que he trabajado. Dentro de “src” encontramos “assets”. En esta carpeta es dónde se incluyen las imágenes, fuentes y archivos css.  En la carpeta “components” están todos los micro-componentes de los que se conforma la aplicación. Como las tarjetas, el footer y otros elementos que se repiten.  En la carpeta “pluggins” encontramos una librería, en este caso “vuetify.js” que se usa para hacer los estilos, ya que tiene muchos componentes ya hechos que facilitan el diseño. |
| En la carpeta “router” encontramos las rutas de front tales como “login”, “register” entre muchas otras.  En la carpeta “store” se encuentra un archivo de JS en el que están almacenados los datos que tienen que existir entre componentes. Como el usuario, sus grupos y demás.  En la carpeta “views” se encuentran todas las vistas en HTML junto con su script en JS para que tenga una funcionalidad plena. | |

## 4.3. Diseño de la estructura de clases y librerías (diagrama de clases).

## 4.4 Diseño de marca

El logotipo de la marca se ha inspirado en una caja de regalo.

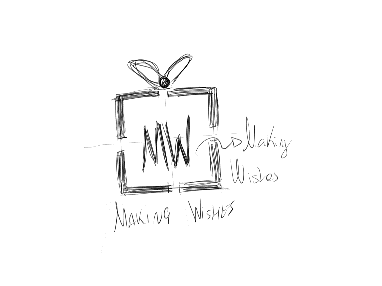


Ilustración - Estudio de logotipo

Cuando ya se tuvo claro el logotipo se pasó a un estudio de color.

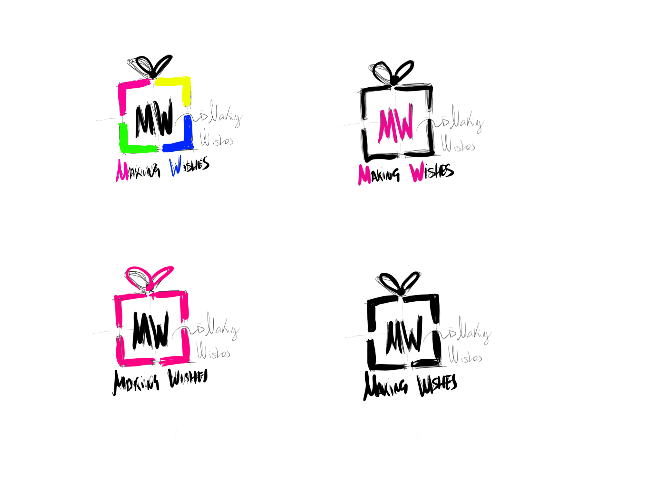


Ilustración - Estudio de color

Al final me decanté por dejar el logo sólo en negro, porque se ve más limpio y sofisticado. Se utilizó la tipografía “Noteworthy-Bold” resultando el logotipo de la siguiente manera:



Ilustración - Logotipo definitivo

## 4.5. Diseño de la interfaz gráfica.

### 4.5.1 Sketch

|  |
| --- |
|  |

Ilustración - Sketch landing

|  |
| --- |
|  |

Ilustración - Sketch log in

|  |
| --- |
|  |

Ilustración - Sketch registro

|  |
| --- |
|  |

Ilustración - Sketch home

|  |
| --- |
|  |

Ilustración - Sketch mis grupos

|  |
| --- |
|  |

Ilustración - Sketch modificar cuenta

|  |
| --- |
|  |

Ilustración – Sketch crear grupo

|  |
| --- |
|  |

Ilustración - Sketch texto legal

### 4.5.2. Mock-ups



Ilustración - Mock up landing

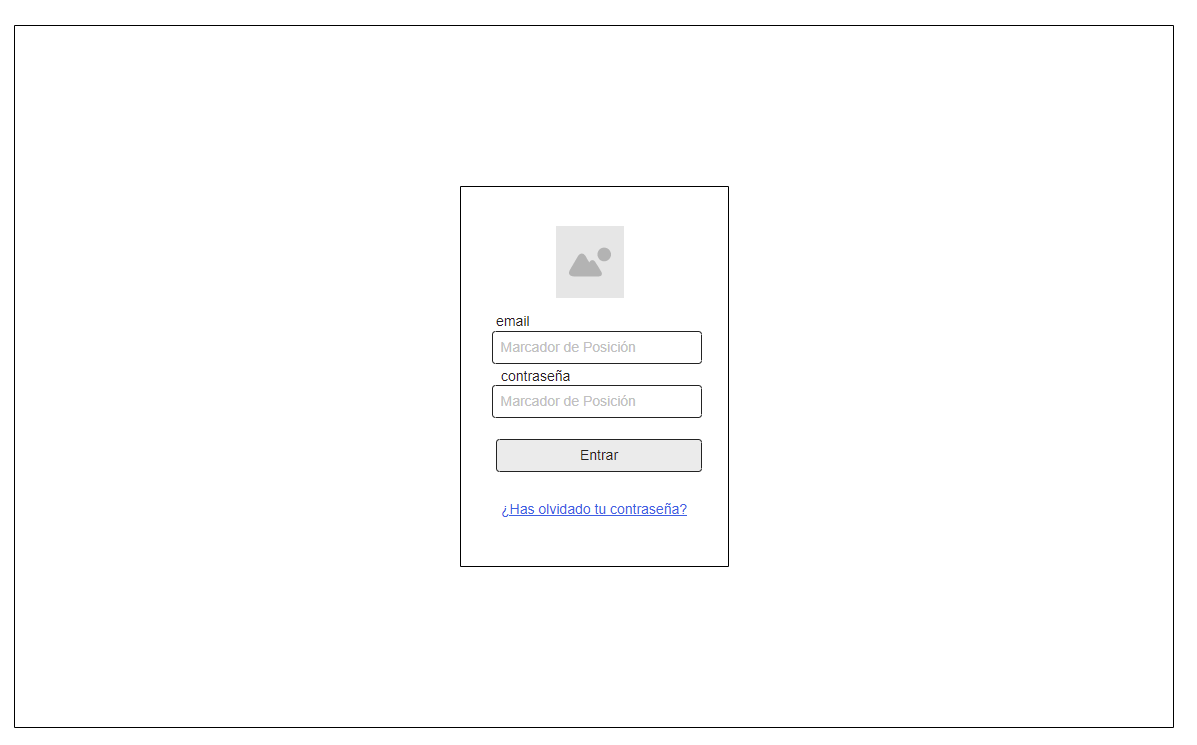


Ilustración - Mock up log in

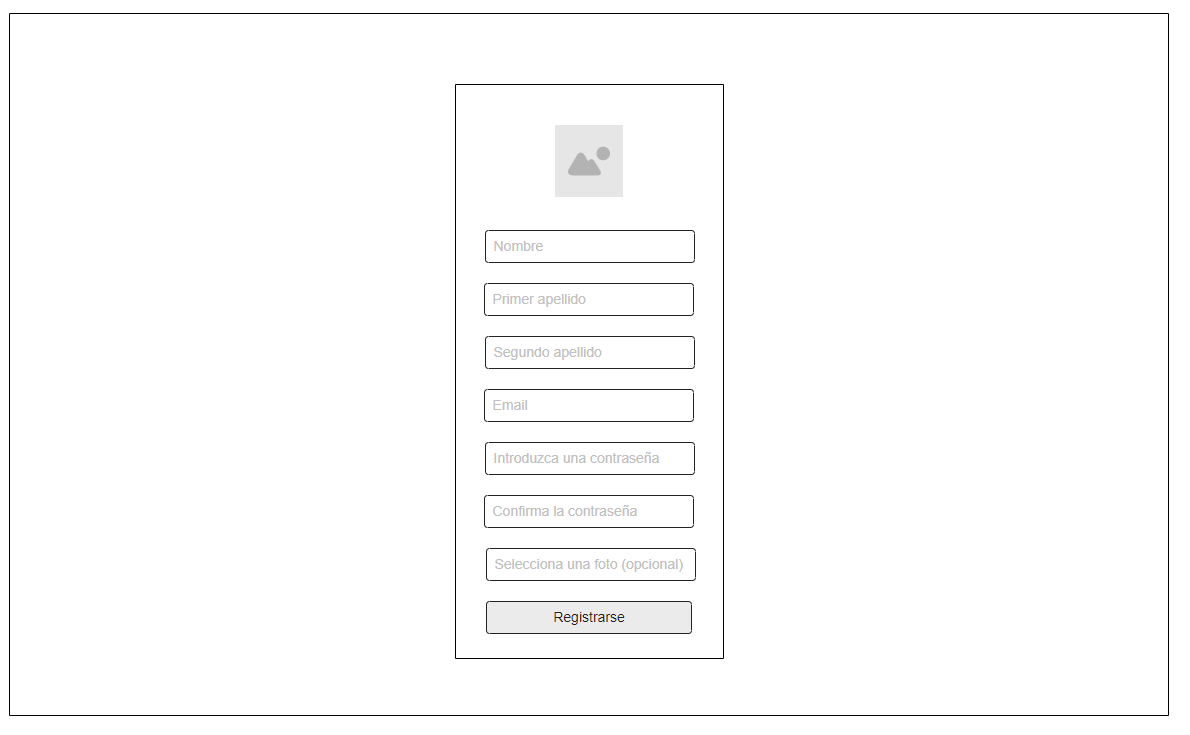


Ilustración - Mock up regristro

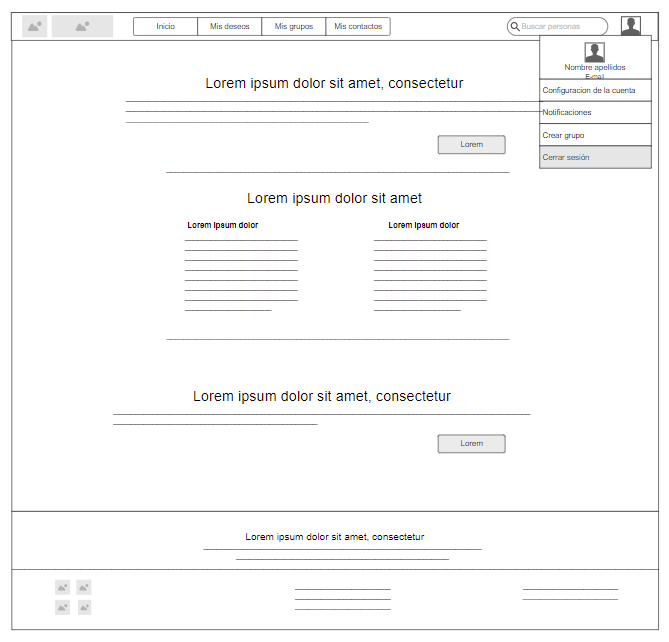


Ilustración - Mock up Inicio



Ilustración - Mock up mis deseos

Aquí mis contactos

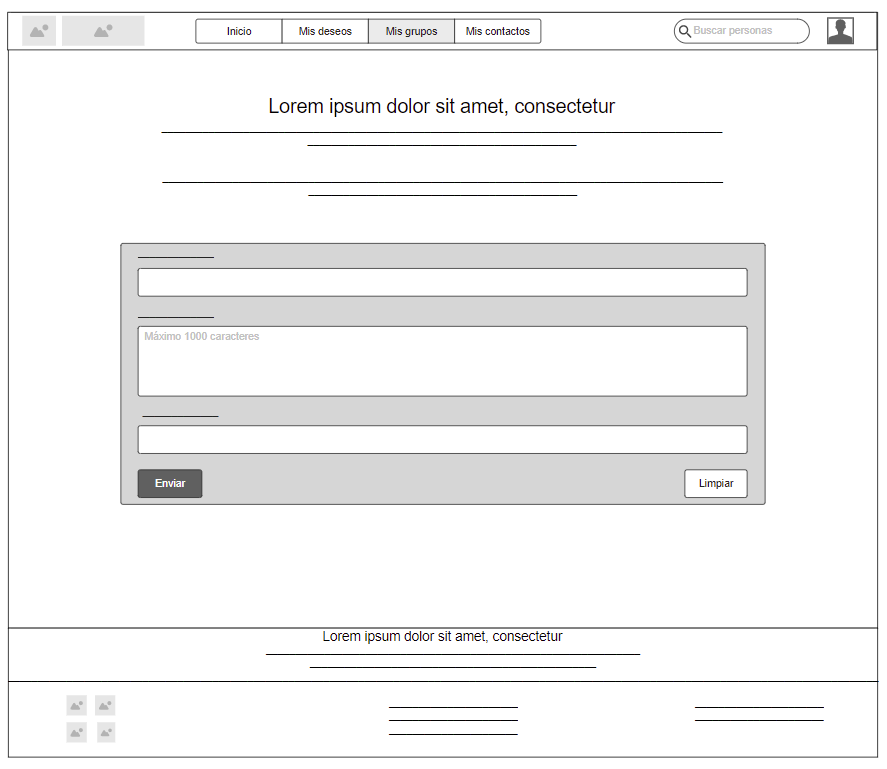


Ilustración - Mock up Mis grupos

# 

Ilustración - Mock up configuración de la cuenta

Aquí notificaciones

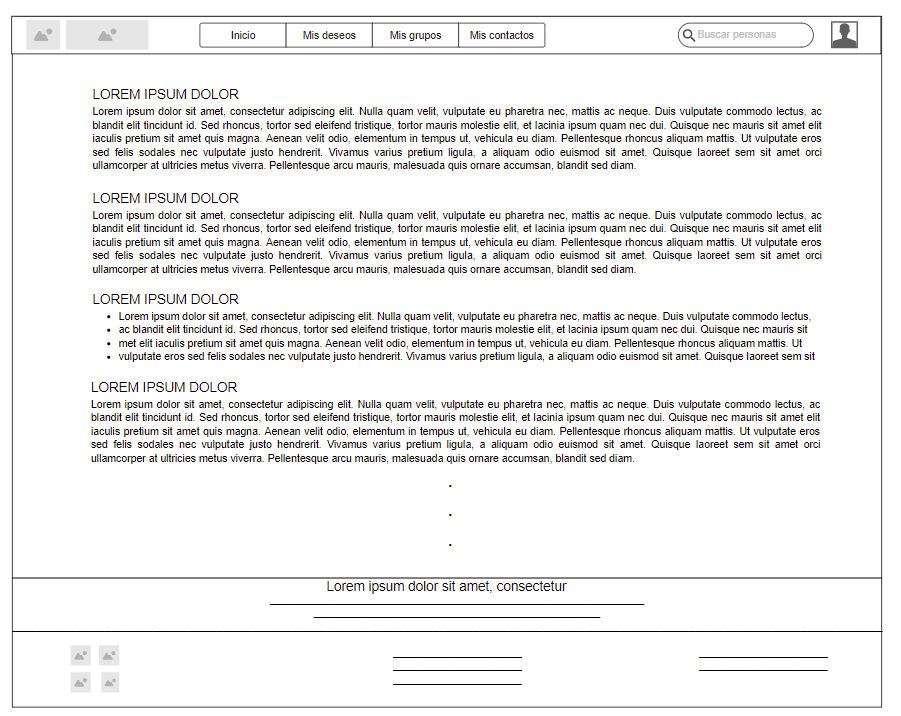


Ilustración - Mock up texto legal

# 

Ilustración - Mock up crear grupo

# 5. Desarrollo y realización del Proyecto:

## 5.1. Indicadores de calidad.

## 5.2. Elaboración de una batería de pruebas para detectar errores.

|  |  |  |
| --- | --- | --- |
| Pruebas realizadas | | |
| Tipo | Resultado esperado | Resultado conseguido |
| Registro | | |
| Se ha intentado registrar con las credenciales mal escritas | Fallo | Éxito |
| Se ha intentado registrar con las contraseñas diferentes | Fallo | Fallo |
| Se ha intentado poner caracteres no permitidos | Fallo | Fallo |
| Se ha intentado subir un archivo que no es una foto | Fallo | Fallo |
| Se ha intentado registrar sin subir un avatar | Éxito | Éxito |
| Se ha intentado registrar con uno o más campos vacíos. | Fallo | Fallo |
| Loguin | | |
| Loguin con usuario erróneo | Fallo | Fallo |
| Loguin con usuario correcto, clave errónea | Fallo | Fallo |
| Loguin con usuario correcto, clave correcta | Éxito | Éxito |
| Se ha revisado que “cerrar sesión” cierre realmente la sesión. | Éxito | Éxito |
| Navegación entre ventanas | | |
| Acceder sin loguear a la web de modificar usuarios del administrador | Fallo | Fallo |
| Acceder sin loguear a la web de información | Éxito | Éxito |
| Acceder sin loguear a la web de consultar contactos usuarios del consultor | Fallo | Fallo |
| Acceder sin loguear a la web de seleccionar contactos del usuario consultor | Fallo | Fallo |
| Acceder sin loguear a la web de ver notificaciones del usuario consultor | Fallo | Fallo |
| Acceder logueado de administrador a la web de información | Éxito | Éxito |
| Acceder logueado de administrador a la web de modificar usuarios del administrador | Éxito | Éxito |
| Acceder logueado de administrador a la web de consultar contactos usuarios del consultor | Fallo | Fallo |
| Acceder logueado de administrador a la web de seleccionar contactos del usuario consultor | Fallo | Fallo |
| Acceder logueado de administrador a la web de ver notificaciones del usuario consultor | Fallo | Fallo |
| Acceder logueado de consultor a la web de información | Éxito | Éxito |
| Acceder logueado de consultor a la web de m dificar usuarios del administrador | Fallo | Fallo |
| Acceder logueado de consultor a la web de consultar contactos usuarios del consultor | Éxito | Éxito |
| Acceder logueado de consultor a la web de seleccionar contactos del usuario consultor | Éxito | Éxito |
| Acceder logueado de consultor a la web de ver notificaciones del usuario consultor | Éxito | Éxito |
| Revisión de funciones | | |
| Se ha revisado que aparezcan correctamente los usuarios que se siguen con todos sus datos. | Éxito | Éxito |
| Se ha revisado que se inserte de manera correcta cuando un usuario agrega a otro | Éxito | Fallo |
| Se ha revisado que los usuarios reciban las notificaciones correctamente. | Éxito | Éxito |
| Mandar más de una notificación a un usuario | Fallo | Éxito |
| Se ha revisado que tras leer una notificación no vuelva a aparecer. | Éxito | Éxito |
| Se ha revisado que tras leer la notificación y darle a rechazar no se inserte como contacto. | Éxito | Éxito |
| Se ha revisado que tras leer la notificación y darle a aceptar se inserte como contacto. | Éxito | Éxito |
| Se ha revisado que si buscas usuarios en una provincia que no te quedan por seguir que no salte error. | Éxito | Éxito |
| Se ha revisado que se inserte correctamente en la base de datos tanto los usuarios nuevos como la modificación de los datos de usuario de los mismos realizada por el administrador. | Éxito | Éxito |
| Registro | | |
| Registro con credenciales vacíos | Fallo | Éxito |
| Registro con credenciales rellenos | Éxito | Éxito |

Tabla - Batería de pruebas

## 5.3. Evaluación y solución de incidencias.

Se ha añadido un campo más a la tabla notifications que indica si esta ha sido leída o no.

Se ha añadido un campo más a la tabla de usuarios que determina si es administrador o no.

## 5.4. Evaluación y seguimiento de las actividades.

# 6. Implantación del Proyecto:

## 6.1. Plan de implantación.

### 6.1.1 Back-end:

Este proyecto está pensado para implantarlo y correrlo en un servidor Apache y que se sirva de una base de datos MySql. A continuación, se mostrará como desplegarlo, como instalarlo y cómo usarlo.

En esta ocasión vamos a instalar el servidor Apache y la base de datos MySql a través de la herramienta XAMPP. Para comenzar primero deberemos de descargar el producto en la web oficial, por favor siga el siguiente enlace: <https://www.apachefriends.org/es/download.html>

A continuación, tenemos que ir a la carpeta dónde lo hemos guardado, normalmente suele ser en C:\Users\miUsuario\Descargas. También podemos usar el acceso directo que encontramos en el explorador de carpetas:

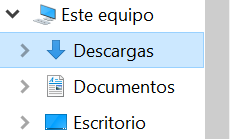


Ilustración - Manual de implantación I

Se recomienda desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación.

A continuación, tenemos que ejecutar el archivo xampp-windows-x64-8-0-2-0-VS16- installer.exe, para ello damos doble clic sobre el archivo.

Si no hemos desactivado el antivirus nos aparecerá la siguiente ventana diciendo que puede interferir en la instalación. Desactivamos el antivirus y continuamos con la instalación dándole a “Yes”.



Ilustración - Manual de implantación II

Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en “Next”.

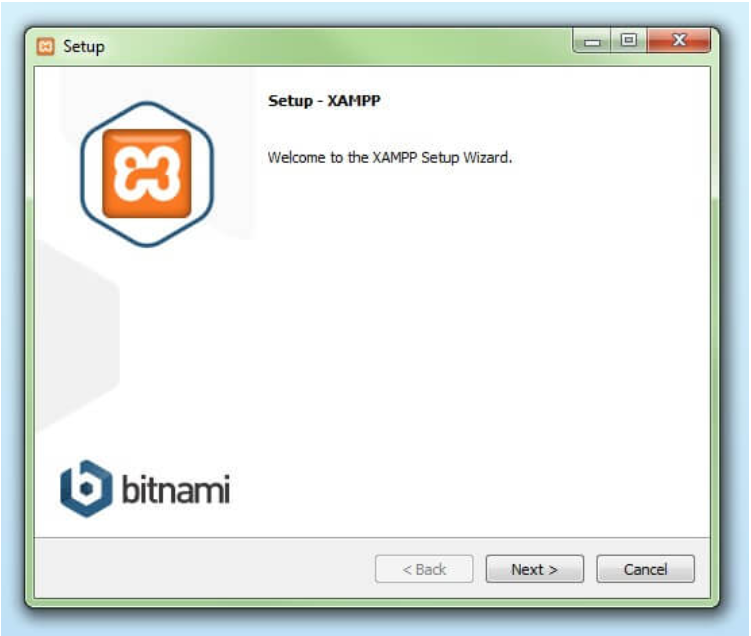


Ilustración - Manual de implantación III

En la rúbrica “Select components” se pueden excluir de la instalación componentes aislados del paquete de software de XAMPP. Se recomienda la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Confirma la selección haciendo clic en “Next”.

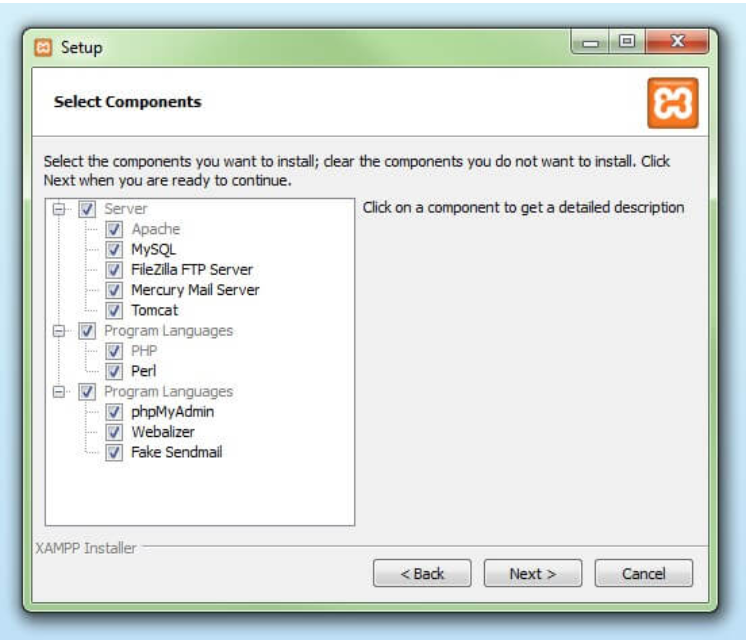


Ilustración - Manual de implantación IV

En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:\. Confirma la selección haciendo clic en “Next”.

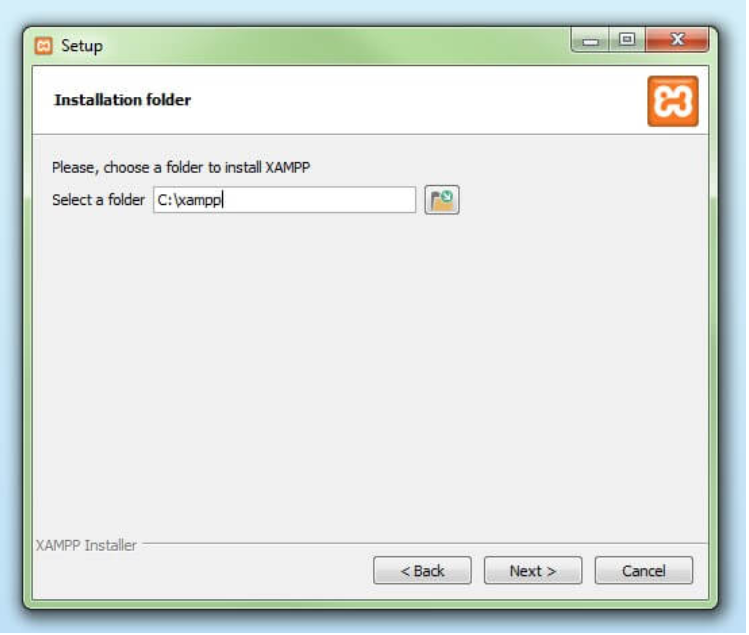


Ilustración - Manual de implantación V

El asistente extrae los componentes seleccionados y los guarda en el directorio escogido en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.



Ilustración - Manual de implantación VI

Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública. Pulsamos en “permitir acceso”

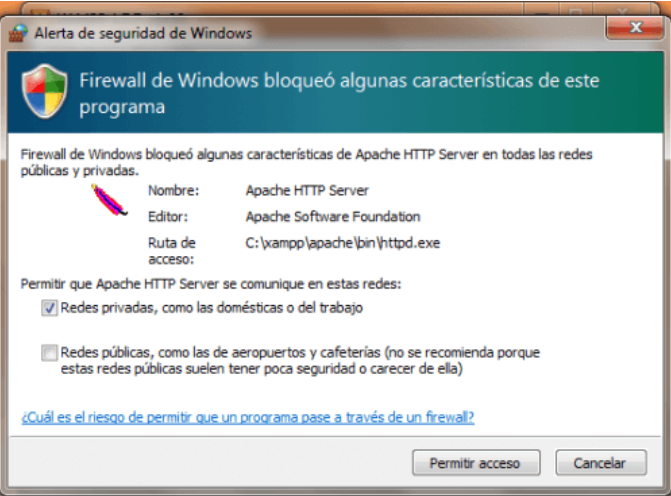


Ilustración - Manual de implantación VII

Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla “Finish”. Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.

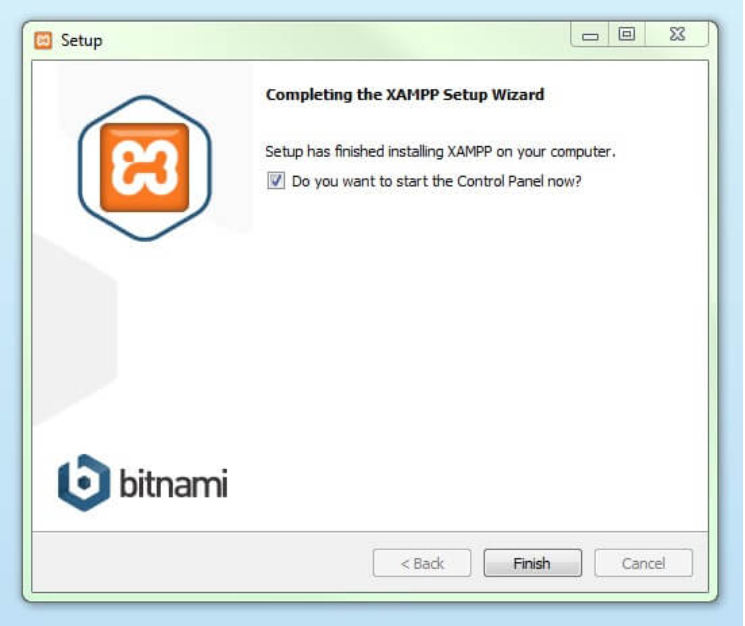


Ilustración - Manual de implantación VIII

En la interfaz de panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic.

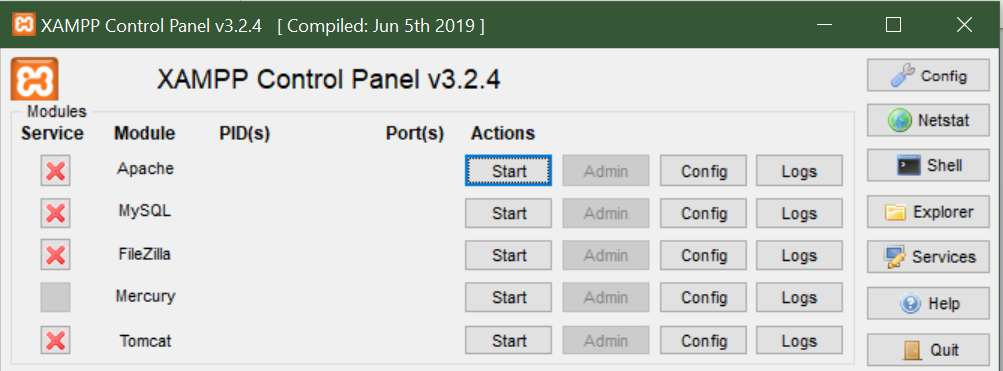


Ilustración - Manual de implantación IX

Le damos a “Start” en Apache y en MySQL. Si todo va bien tiene que aparecer lo siguiente:

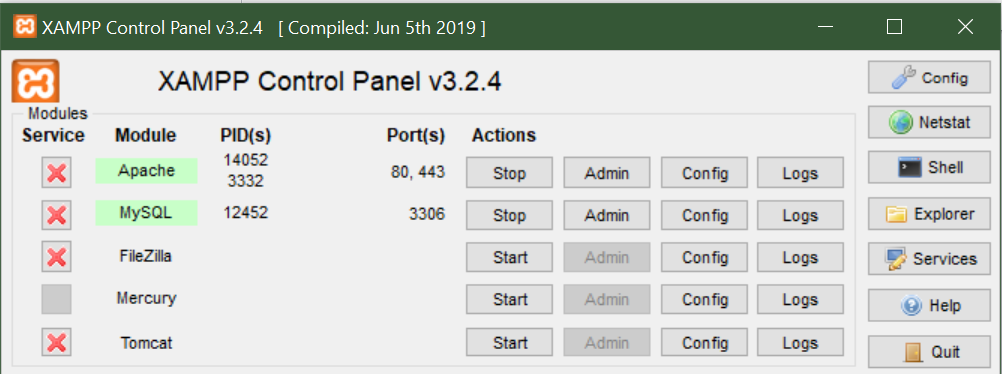


Ilustración - Manual de implantación X

Si los servicios no se han podido iniciar habrá que cambiar los puertos o bien finalizar las aplicaciones que están impidiendo que el servidor se levante correctamente. Esta es la opción que se recomienda. Cuando el servidor está levantado correctamente puede abrir la aplicación que estaba dando conflicto.

A continuación, vamos a pulsar sobre config del servidor apache y vamos a pulsar sobre “Apache (httpd.conf). Se abrirá un documento de texto plano, en él vamos a pegar las siguientes líneas al final del todo:

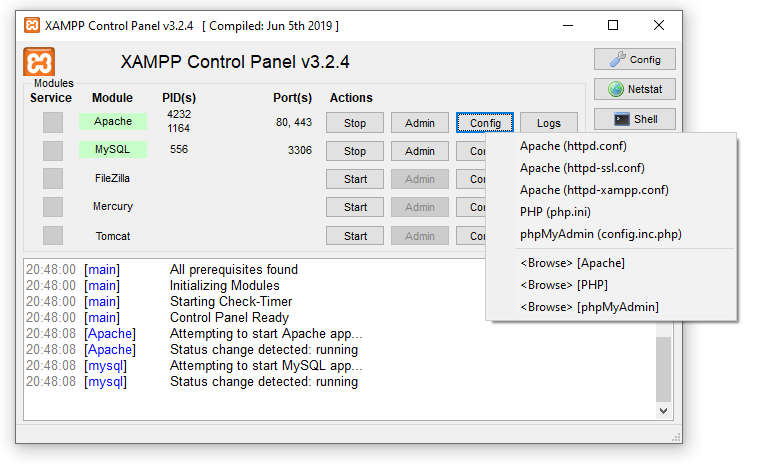


Ilustración - Configuración de Apache.

Header set Access-Control-Allow-Origin "\*"

Header set Access-Control-Allow-Methods "GET, POST, PUT, DELETE, OPTIONS"

Header set Access-Control-Allow-Headers "X-Requested-With, Content-Type, X-Token-Auth, Authorization"

Estas líneas nos van a permitir hacer peticiones get, post, put, delete y options desde cualquier dominio.

### 6.1.2. Front-End

Para que el proyecto creado en VUE.js funcione correctamente en nuestro terminal es necesario instalar NODE.js.

Primero accedemos a la página de descargas de node y hacemos clic en descargar el instalador: <https://nodejs.org/es/download/>

Abrimos el archivo que se nos ha descargado y hacemos clic en siguiente:

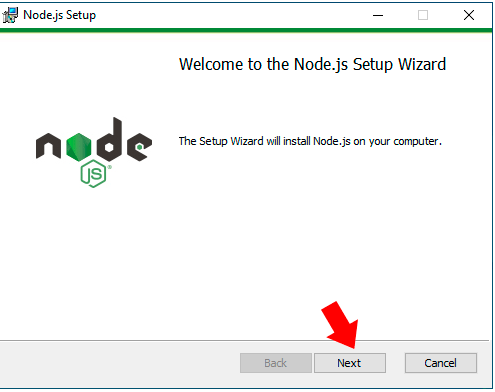


Ilustración - Plan de implantación front-end I

A continuación, se mostrará un mensaje con las condiciones de NODE.js. Acéptalas y haz clic en siguiente.



Ilustración - Plan de implantación front-end II

Selecciona la carpeta en la que quieres guardarlo y haz clic en siguiente.

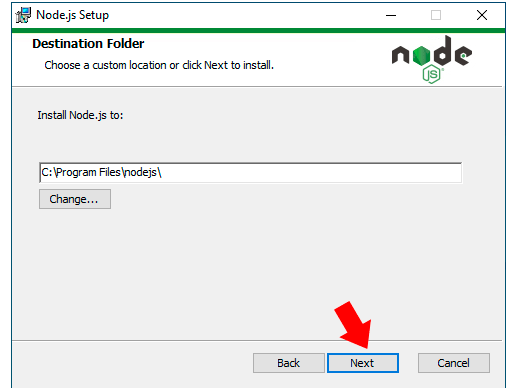


Ilustración - Plan de implantación front-end III

A continuación, aparecerán los distintos componentes que se van a instalar. Deja la configuración recomendada y haz clic en siguiente.



Ilustración - Plan de implantación front-end IV

Seguidamente se te preguntará si quieres instalar Python y las VS Build Tools para así poder compilar ciertos módulos. Es recomendable que aceptes su instalación. Luego haz clic en Next para continuar.



Ilustración - Plan de implantación front-end V

Finalmente, haz clic en Install para iniciar la instalación de npm, de Node.js y de las herramientas adicionales.

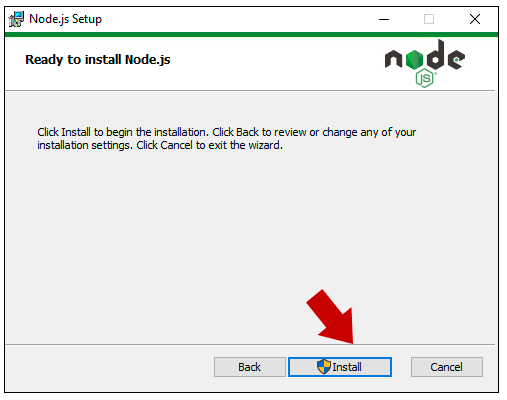


Ilustración - Plan de implantación front-end VI

Durante la instalación se abrirá una ventana de símbolo de sistema o de PowerShell, desde donde se instalarán tanto Python como Chocolatey, siempre que hayas aceptado su instalación. Pulsa cualquier tecla para iniciar la instalación:

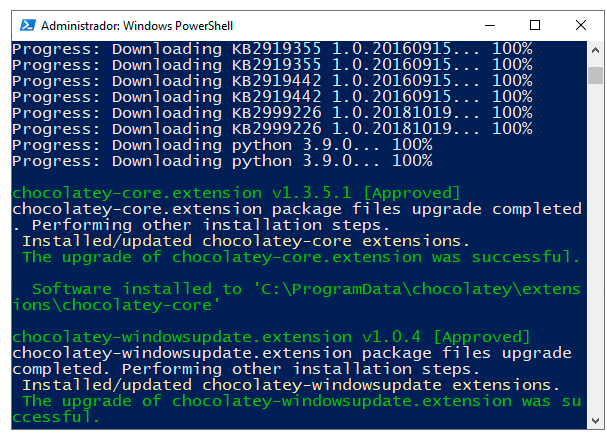


Ilustración - Plan de implantación front-end VII

Cuando la instalación termine, haz clic en Finish.

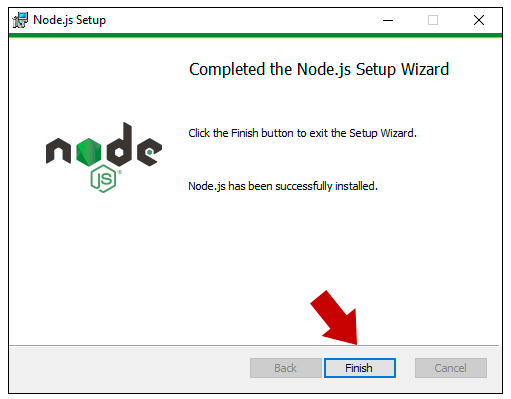


Ilustración - Plan de implantación front-end VIII

## 6.2. Manual de instalación.

### 6.2.1. Back-End:

Para instalar la aplicación hay que seguir unos pasos muy sencillos. Primero se recomienda hacer un subdirectorio dentro de la carpeta htdocs, esta carpeta se encuentra en la siguiente ruta: C:\xampp\htdocs.

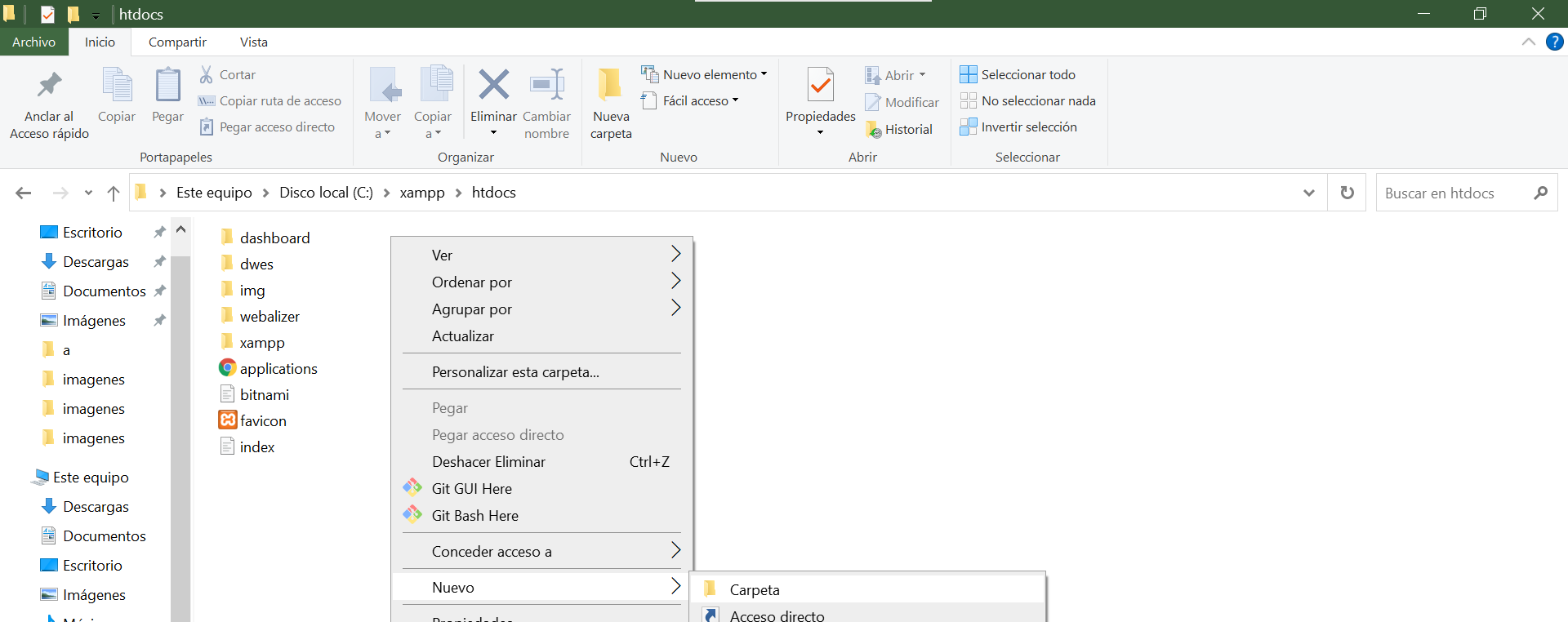


Ilustración - Manual de instalación I

Copie los archivos dispuestos en el pendrive o dvd y peguelos en la siguiente ruta: C:\xampp\htdocs\phoneBook

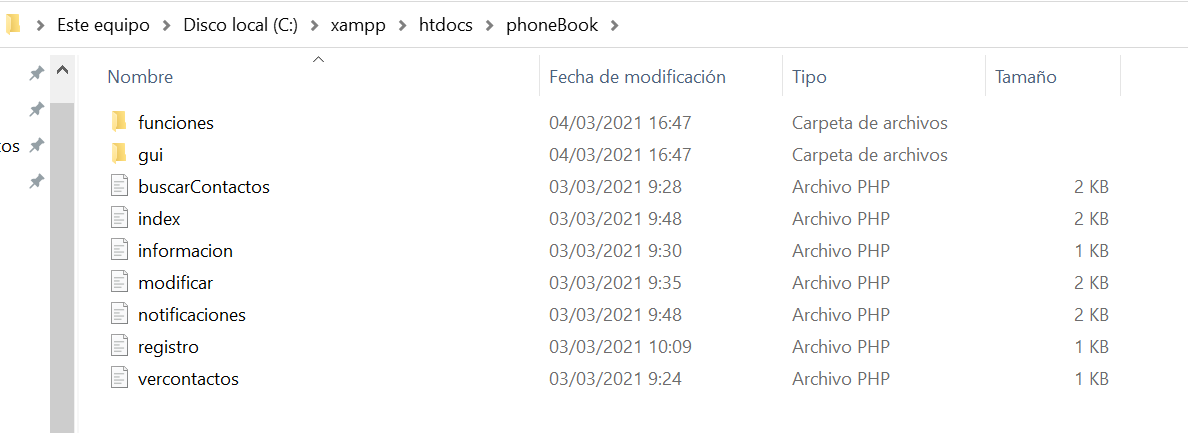


Ilustración - Manual de instalación II

A continuación, tiene que cargar la base de datos. Para ello nos dirigimos a la siguiente dirección: <http://localhost/>



Ilustración - Manual de instalación IV

Pinchamos en el botón de la parte superior derecha que pone phpMyAdmin. Encontraremos una web similar a la siguiente:

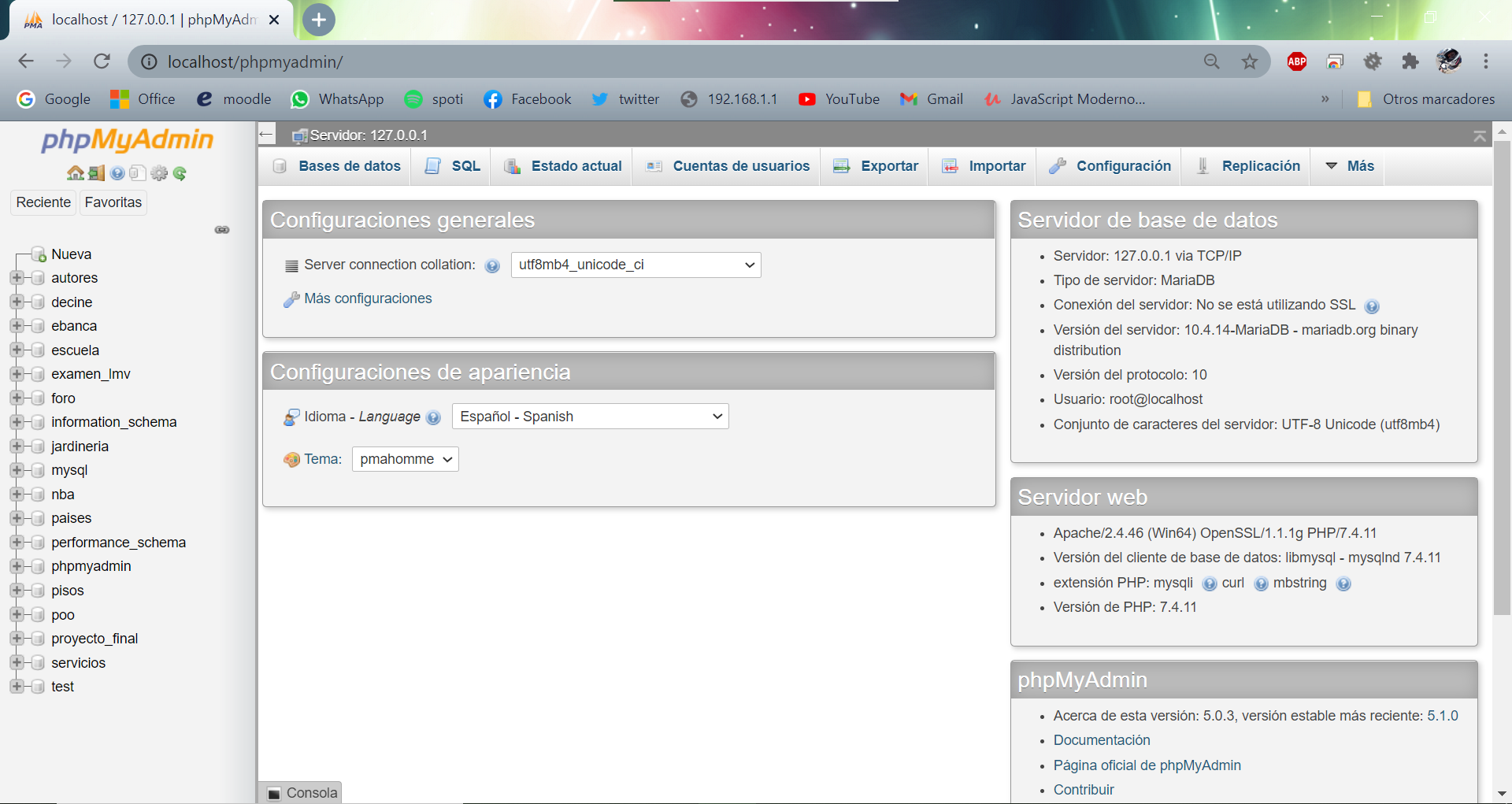


Ilustración - Manual de instalación V

Pinchamos en importar:

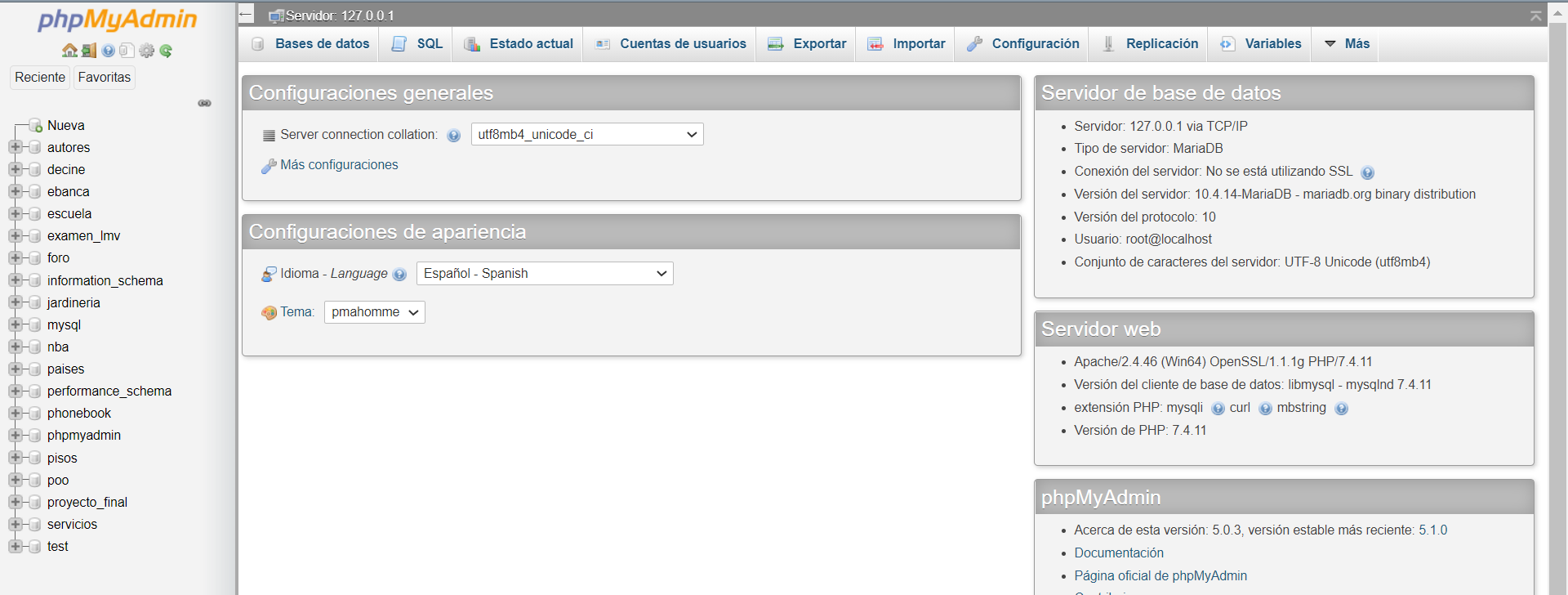


Ilustración - Manual de instalación VI

Le damos a “Seleccionar archivo”.

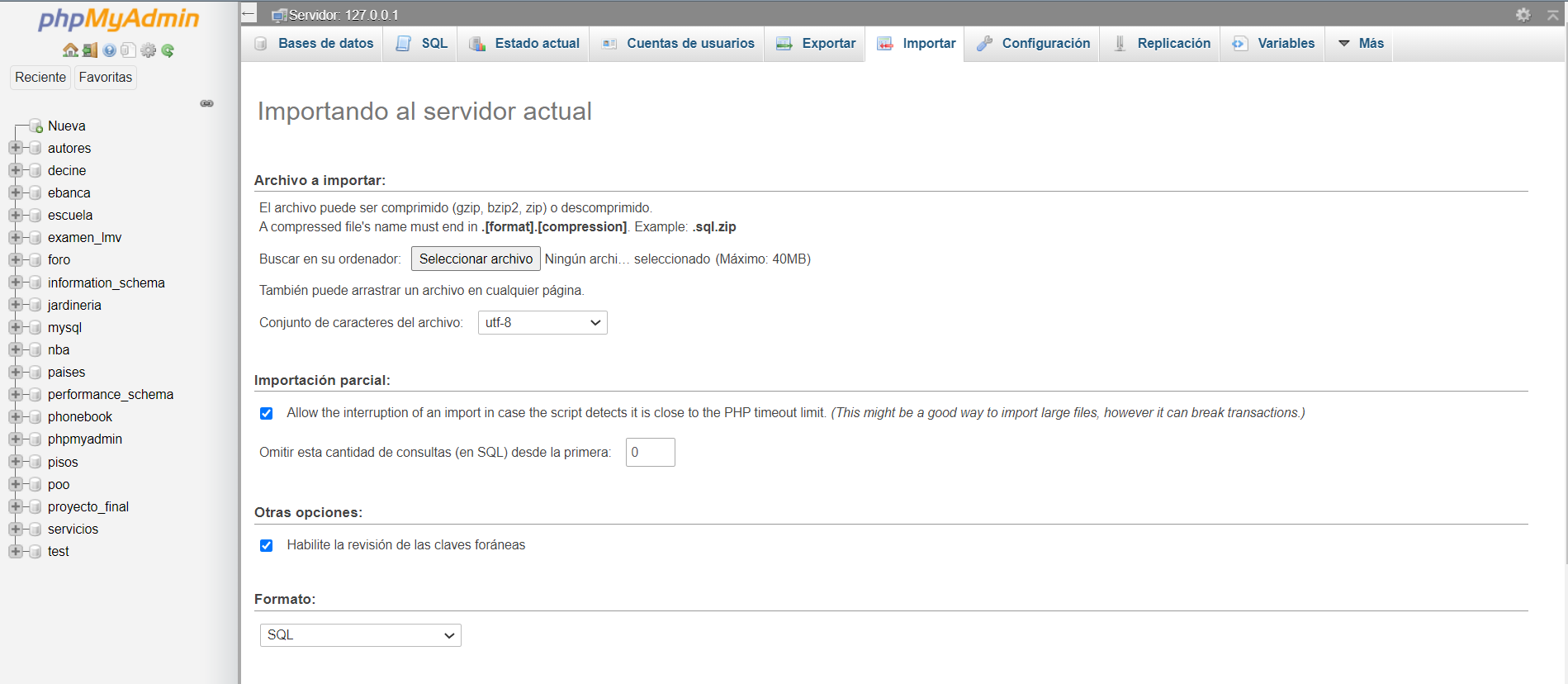


Ilustración - Manual de instalación VII

A continuación, nos saldrá una ventana emergente y seleccionaremos el archivo “MySql-Making-Wishes.txt”

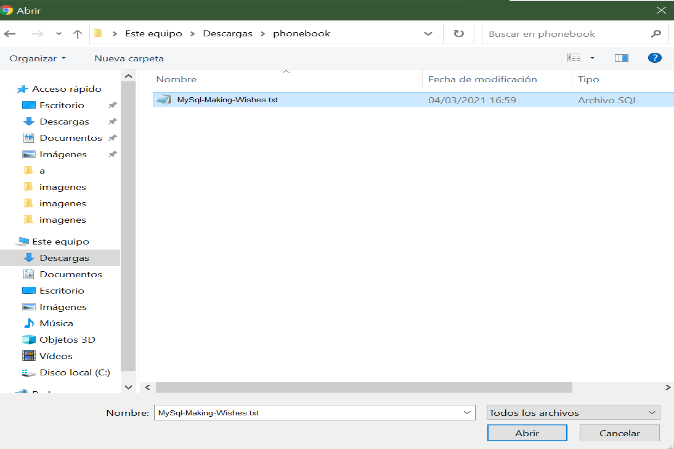


Ilustración - Manual de instalación VIII

Dejamos todo como en la imagen y le damos a continuar:

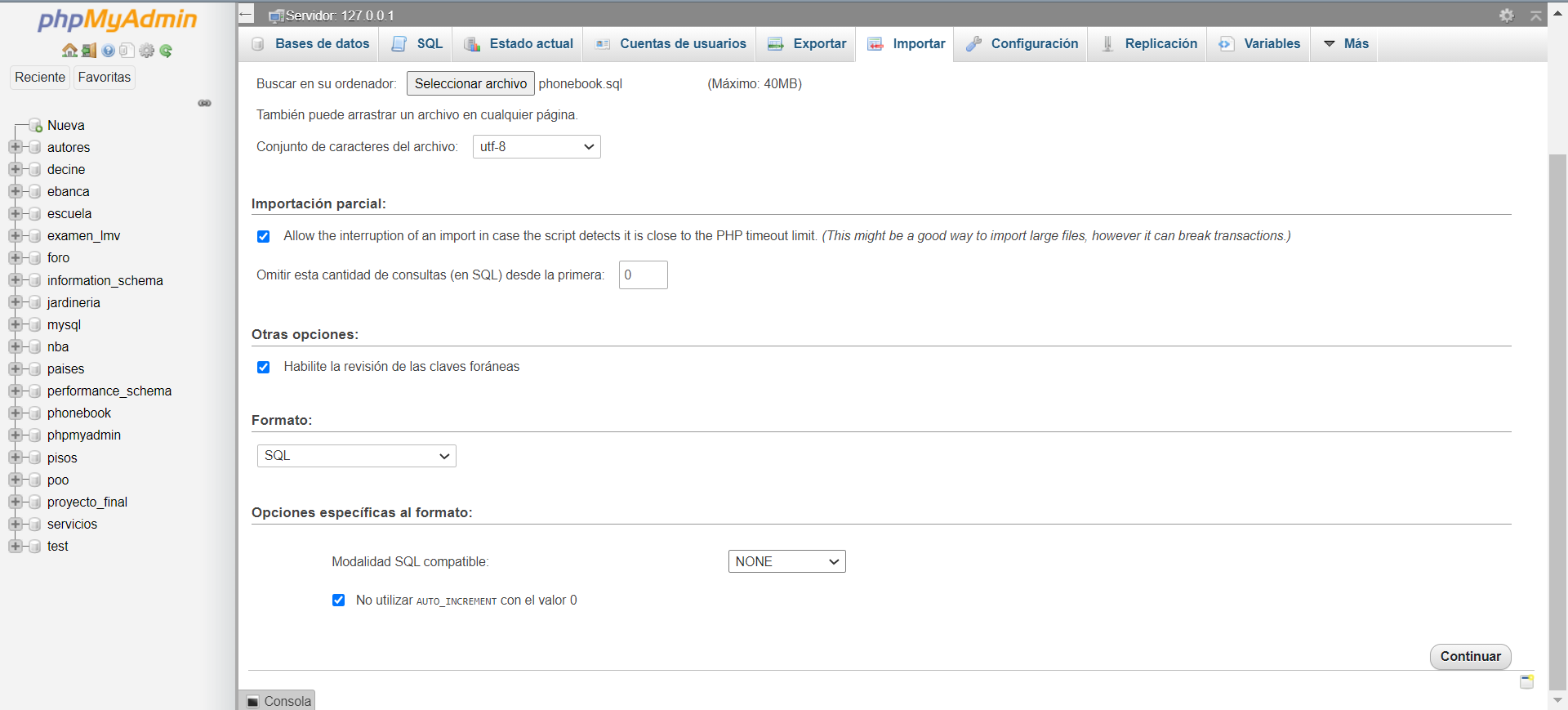


Ilustración - Manual de instalación IX

### 6.2.2. Front-End:

Para instalar la parte de front-end hay que seguir unos pequeños pasos muy sencillos:

Primeramente, copie la carpeta MakingWishes que se adjunta en el pendrive o CD y péguela en el directorio deseado. Seguidamente abra una terminal, posiciónese en la carpeta dónde ha copiado los archivos usando el comando cd y ejecute el siguiente comando: npm run serve.



Por defecto se abrirá en el navegador predeterminado el proyecto, pero si no lo hace copie la siguiente dirección en la barra de navegación de su navegador favorito: [**http://localhost:8080/**](http://localhost:8080/)o la que se muestre en su terminal para visualizar el proyecto en el navegador.

## 6.3. Manual de usuario.

# 7. Conclusiones Finales:

## 7.1. Evaluación del proyecto: objetivos conseguidos y no alcanzados.

## 7.2. Propuestas de mejora.