



## DIQSI, tu web de boda

**Autor: Roberto Pérez Marín**

Grado Multimedia

Ingeniería Web

Consultor: Ignasi Lorente Puchades

Profesor: Carlos Casado Martínez

Fecha de entrega: 08/01/2018



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>DIQSI, tú web de boda</i>
<b>Nombre del autor:</b>	<i>Roberto Pérez Marín</i>
<b>Nombre del consultor/a:</b>	<i>Ignasi Lorente Puchades</i>
<b>Nombre del PRA:</b>	<i>Carlos Casado Martínez</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>01/2018</i>
<b>Titulación o programa:</b>	<i>Grado Multimedia</i>
<b>Área del Trabajo Final:</b>	<i>Ingeniería Web</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Aplicación, desarrollo, web de boda</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>Este trabajo de final de grado aborda las fases de diseño y desarrollo de una aplicación web que permitirá a futuras parejas que quieren casarse, ofrecer a sus invitados toda la información posible del evento. En este documento se detallarán todos los procesos realizados en los que se han puesto en práctica los conocimientos adquiridos a lo largo del grado, especialmente los que pertenecen al ámbito de la Ingeniería web. El objetivo principal del proyecto es obtener un primer prototipo funcional que sea operativo y que sirva para mostrar, sobretodo, su funcionalidad básica, con el objetivo de continuar mejorando la aplicación en fases posteriores.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>This final project tackles the design and development phases of a web application that will allow to future couples that want to get married, the possibility to offer all possible information of the event to their guests. In this document, I will describe all the processes made where I have put into practice the knowledge learned during the degree, especially those that belong to the field of web engineering. The main objective of the project is to obtain a first functional prototype. That prototype will be operative and will be able to show, mainly, its basic functionality, with the objective of continue with the improvement of the application in later phases.</p>	

# Índice

<b>1. Introducción</b> .....	<b>4</b>
1.1 Contexto y justificación del Trabajo .....	4
1.2 Objetivos del Trabajo .....	4
1.3 Enfoque y método seguido .....	5
1.4 Planificación del trabajo .....	5
1.5 Breve resumen de productos obtenidos .....	7
1.6 Breve descripción de los otros capítulos de la memoria.....	7
<b>2. Análisis e implementación</b> .....	<b>8</b>
2.1 Usabilidad/UX.....	8
2.1.1 Perfiles de usuario.....	8
2.1.2 Historias de usuario .....	10
2.1.3 Prototipo Lo-Fi .....	15
2.2 Arquitectura de la aplicación .....	21
2.2.1 Cliente.....	22
2.2.2 Servidor.....	22
2.2.3 Base de datos.....	23
2.2.3.1 Modelo Entidad Relación .....	23
2.2.4 APIs y librerías utilizadas .....	24
2.3 Casos de uso.....	25
2.3.1 Diagrama UML de casos de uso.....	25
2.3.2 Descripción de los casos de uso .....	26
<b>3. Instalación</b> .....	<b>30</b>
3.1 Requisitos de instalación .....	30
3.1 Instrucciones de instalación.....	31
3.2 Material para el uso de la aplicación .....	32
<b>4. Desarrollo</b> .....	<b>33</b>
4.1 Extractos de código .....	33
4.2 Seguridad.....	40
4.3 Tests .....	41
4.4 Versiones de la aplicación .....	43
4.1 Bugs.....	43
<b>6. Conclusiones</b> .....	<b>43</b>
<b>7. Líneas de trabajo futuro</b> .....	<b>44</b>
<b>8. Bibliografía</b> .....	<b>45</b>
<b>9. Recursos</b> .....	<b>45</b>

## Figuras y tablas

### Índice de tablas

<b>Tabla 1: Planificación del proyecto .....</b>	<b>6</b>
<b>Tabla 2: Perfil de usuario PU_001 .....</b>	<b>9</b>
<b>Tabla 3: Perfil de usuario PU_002 .....</b>	<b>9</b>
<b>Tabla 4: Perfil de usuario PU_003 .....</b>	<b>9</b>
<b>Tabla 5: Historia de usuario HU_001.....</b>	<b>10</b>
<b>Tabla 6: Historia de usuario HU_002.....</b>	<b>10</b>
<b>Tabla 7: Historia de usuario HU_003.....</b>	<b>11</b>
<b>Tabla 8: Historia de usuario HU_004.....</b>	<b>11</b>
<b>Tabla 9: Historia de usuario HU_005.....</b>	<b>12</b>
<b>Tabla 10: Historia de usuario HU_006.....</b>	<b>12</b>
<b>Tabla 11: Historia de usuario HU_007.....</b>	<b>13</b>
<b>Tabla 12: Historia de usuario HU_008.....</b>	<b>13</b>
<b>Tabla 13: Historia de usuario HU_009.....</b>	<b>14</b>
<b>Tabla 14: Historia de usuario HU_010.....</b>	<b>14</b>
<b>Tabla 15: Historia de usuario HU_011.....</b>	<b>15</b>
<b>Tabla 16: Caso de uso CU_001 .....</b>	<b>26</b>
<b>Tabla 17: Caso de uso CU_002 .....</b>	<b>27</b>
<b>Tabla 18: Caso de uso CU_003 .....</b>	<b>28</b>
<b>Tabla 19: Caso de uso CU_004 .....</b>	<b>28</b>
<b>Tabla 20: Caso de uso CU_005 .....</b>	<b>29</b>
<b>Tabla 21: Caso de uso CU_006 .....</b>	<b>30</b>
<b>Tabla 22: Caso de uso CU_007 .....</b>	<b>30</b>

**Índice de figuras**

<b>Ilustración 1. Diagrama de Gantt</b>	<b>7</b>
<b>Ilustración 2. Lo-Fi página de registro</b>	<b>15</b>
<b>Ilustración 3. Lo-Fi modal datos sobre la boda</b>	<b>15</b>
<b>Ilustración 4. Lo-Fi login</b>	<b>16</b>
<b>Ilustración 5. Lo-Fi editar perfil de usuario</b>	<b>16</b>
<b>Ilustración 6. Lo-Fi listado de álbumes</b>	<b>17</b>
<b>Ilustración 7. Lo-Fi modal crear álbum</b>	<b>17</b>
<b>Ilustración 8. Lo-Fi editar álbum</b>	<b>18</b>
<b>Ilustración 9. Lo-Fi página de plantillas</b>	<b>18</b>
<b>Ilustración 10. Lo-Fi lista de invitados</b>	<b>19</b>
<b>Ilustración 11. Lo-Fi modal crear/editar invitados</b>	<b>19</b>
<b>Ilustración 12. Lo-Fi ejemplo plantilla web de boda</b>	<b>20</b>
<b>Ilustración 13. Lo-Fi página de registro (móvil)</b>	<b>20</b>
<b>Ilustración 14. Lo-Fi login (móvil)</b>	<b>20</b>
<b>Ilustración 15. Lo-Fi editar perfil de usuario (móvil)</b>	<b>21</b>
<b>Ilustración 16. Lo-Fi página de plantillas (móvil)</b>	<b>21</b>
<b>Ilustración 17. Lo-Fi listado de álbumes (móvil)</b>	<b>21</b>
<b>Ilustración 18. Lo-Fi lista de invitados (móvil)</b>	<b>21</b>
<b>Ilustración 19. Arquitectura cliente/servidor</b>	<b>22</b>
<b>Ilustración 20: Estructura estática de aplicación Yii2</b>	<b>23</b>
<b>Ilustración 21: Modelo Entidad Relación</b>	<b>24</b>
<b>Ilustración 22: Diagrama UML de casos de uso</b>	<b>25</b>

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Quién iba a decir hace 50 o 60 años que tendríamos móviles, *tablets* o Internet. Está claro que, en los tiempos que corren, hay que adaptarse a las nuevas tecnologías ya que, cada vez se tiende más a hacerlo todo por Internet. Y cómo no, las bodas no iban a ser una excepción.

Al igual que muchas otras cosas, podemos decir que las bodas ya no son como antes. Cada vez están más adaptadas a los gustos y aficiones de los novios, los cuales, ahora buscan una forma más original y divertida de mostrar su amor.

Coincidiendo con mi etapa final en el Grado Multimedia, he querido desarrollar mi TFG en el ámbito de la Ingeniería web y poder dar pie a esta evolución, ofreciendo a las miles de parejas que se casan cada año, la oportunidad de crear su propia web de boda de forma fácil e intuitiva. Para ello se pretende realizar un análisis, diseño e implementación de una aplicación web que pueda ofrecer este servicio.

La aplicación quiere poder ayudar a los futuros novios a dar toda la información posible a sus invitados mediante servicios y funciones específicas, para que ese momento tan especial quede registrado.

Además, como en la actualidad la interacción con nuestros teléfonos móviles es diaria, se dará importancia al diseño *responsive* de las diferentes plantillas con las que pueda contar la aplicación. De esta manera se asegurará el buen funcionamiento y visualización de la web de boda en diferentes dispositivos. Ofrecer una grata experiencia de usuario y plantillas de calidad que puedan ser personalizadas por el usuario, será lo que nos diferencie de los principales competidores.

## 1.2 Objetivos del Trabajo

Los principales objetivos para este trabajo son los siguientes:

- El objetivo general es el desarrollo de un primer prototipo funcional de DIQSI, que ayude a los futuros novios a poder dar toda la información posible a sus invitados mediante servicios y funciones específicas.
- El desarrollo de una aplicación web *responsive*, de forma que se pueda asegurar su utilización y visualización en diferentes dispositivos.
- El desarrollo de una aplicación web intuitiva y de fácil uso para el usuario.

- La implementación de tecnologías modernas que garanticen la viabilidad del proyecto.
- La puesta en práctica todos los conocimientos adquiridos a lo largo del grado, especialmente los que pertenecen al ámbito de la Ingeniería web.
- El aprendizaje de un framework PHP para el desarrollo de la aplicación.
- El aprendizaje de un sistema de control de versiones.

### 1.3 Enfoque y método seguido

Cuando pensé por primera vez la opción de desarrollar una aplicación web para la creación de webs de bodas online, en el mercado aún no existía este tipo de servicio. En la actualidad, sin embargo, ya hay varias opciones.

Hace poco más de un año me casé e intenté utilizar uno de estos sitios para crear mi propia web de boda, pero me encontré con que todas ofrecían una mala experiencia de usuario y las plantillas que tenían eran de poca calidad. Al final se me hacía muy aburrido y difícil crear mi propia página web.

Ésta es una de las razones por las cuales considero que la mejor estrategia a seguir en la realización de este proyecto es la de desarrollar un producto nuevo. Esto me permitirá definir la usabilidad de la aplicación web de forma que a los usuarios les sea muy fácil crear y gestionar su propia web de boda, sin contaminar la aplicación con lo que considero podrían ser los errores de la competencia. Además de poder tener libertad a la hora de elegir las tecnologías que considere más adecuadas para el desarrollo de la aplicación.

### 1.4 Planificación del trabajo

Para el desarrollo de este proyecto serán necesarios los siguientes recursos:

- Ordenador de sobremesa o portátil.
- Editor de código: Atom
- Servidor HTTP Apache
- Sistema gestor de Base de Datos: Mysql 5.6+
- Lenguaje de programación: PHP 7.1+
- Sistema de control de versiones: Git
- Aplicación para la administración de la Base de Datos: Sequel Pro

El proyecto se llevará a cabo en poco más de tres meses de trabajo y constará de 3 entregables en forma de PACs y otro como entrega final del proyecto:

- PAC 1 - 03/10/2017
- PAC 2 - 01/11/2017
- PAC 3 - 10/12/2017
- Entrega Final - 08/01/2018

La planificación del trabajo se propone de la siguiente manera:



Nombre de la tarea	Duración	Fecha de Inicio	Fecha final
<b>Propuesta del proyecto</b>	<b>5d</b>	<b>20/09/17</b>	<b>24/09/17</b>
Búsqueda de información	4d	20/09/17	23/09/17
Resumen ámbito del proyecto	1d	24/09/17	24/09/17
<b>Entrega PAC 1</b>	<b>7d</b>	<b>27/09/17</b>	<b>03/10/17</b>
Contexto y justificación del trabajo	1d	27/09/17	27/09/17
Objetivos del trabajo	1d	28/09/17	28/09/17
Enfoque y método seguido	1d	29/09/17	29/09/17
Planificación del trabajo	4d	30/09/17	03/10/17
<b>Entrega PAC 2</b>	<b>29d</b>	<b>04/10/17</b>	<b>01/11/17</b>
Breve resumen de productos obtenidos	1d	04/10/17	04/10/17
Breve descripción de los otros capítulos de la memoria	1d	05/10/17	05/10/17
Diseño de la aplicación	16d	06/10/17	21/10/17
Diagramas UML	6d	06/10/17	11/10/17
Usabilidad / UX	4d	12/10/17	15/10/17
Arquitectura de la aplicación	6d	16/10/17	21/10/17
Instalación del entorno de trabajo	2d	22/10/17	23/10/17
Desarrollo de la aplicación	9d	24/10/17	01/11/17
<b>Entrega PAC 3</b>	<b>39d</b>	<b>02/11/17</b>	<b>10/12/17</b>
Desarrollo de la aplicación	34d	02/11/17	05/12/17
Realizar vídeo de demostración del funcionamiento de la aplicación	5d	06/12/17	10/12/17
<b>Entrega Final</b>	<b>29d</b>	<b>11/12/17</b>	<b>08/01/18</b>
Finalizar el desarrollo de la aplicación	8d	11/12/17	18/12/17
Testeo y Pruebas	1d	20/12/17	20/12/17
Extractos de código	1d	21/12/17	21/12/17
Versiones de la aplicación/servicios	1d	22/12/17	22/12/17
Seguridad y Bugs	1d	23/12/17	23/12/17
Redactar conclusiones	2d	24/12/17	25/12/17
Finalizar la memoria	4d	26/12/17	29/12/17
Realizar presentación del proyecto para el público general	3d	30/12/17	01/01/18
Realizar presentación en vídeo del proyecto para su defensa	5d	01/01/18	05/01/18
Autoinforme de evaluación	3d	06/01/18	08/01/18

**Tabla 1: Planificación del proyecto**

El siguiente diagrama de Gantt nos permite planificar y programar las tareas gracias a su fácil y cómoda visualización de las acciones previstas. Para su lectura, hay que tener en cuenta que he calculado que un día laborable equivale aproximadamente unas 3 horas de trabajo.

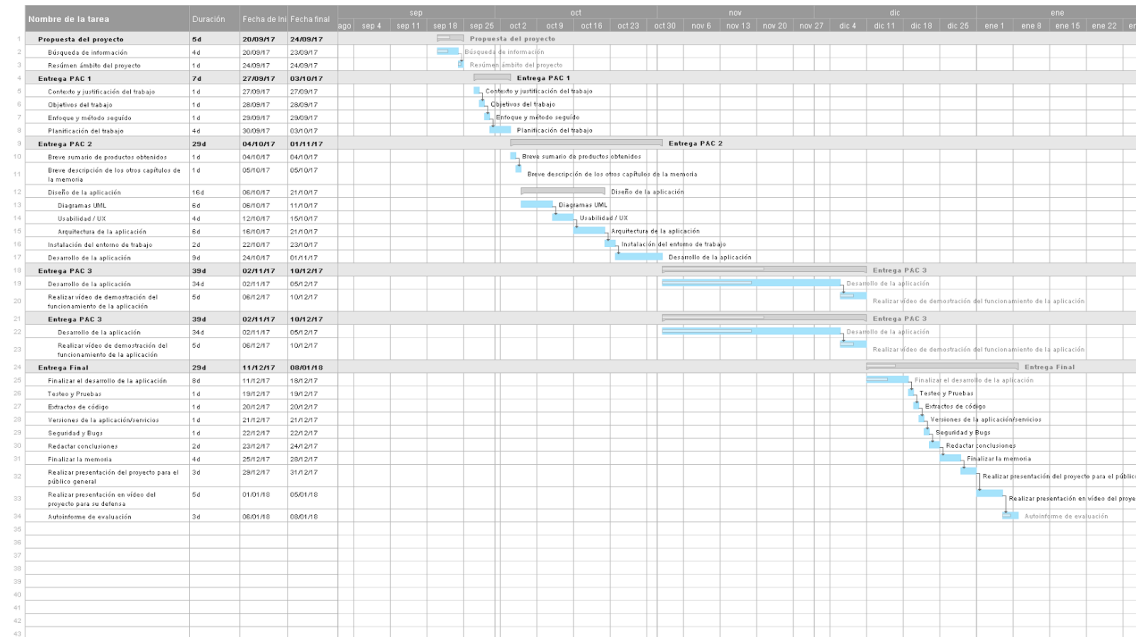


Ilustración 1. Diagrama de Gantt

Para la entrega de la PAC 3, he decidido dedicar la mayor parte de su realización al desarrollo de la aplicación y dejar para la entrega final la realización del testeo, las pruebas, el extracto de código, las versiones de la aplicación, la seguridad y los bugs. Considero que para comenzar su documentación, la aplicación debe estar finalizada.

### 1.5 Breve resumen de productos obtenidos

Los productos que se pretenden obtener durante la realización de este proyecto son los siguientes:

- La aplicación web *responsive* DIQSI.
- La memoria del proyecto.
- Un vídeo de presentación del proyecto para su defensa delante del tribunal de evaluación.
- Una presentación del proyecto para el público general.

### 1.6 Breve descripción de los otros capítulos de la memoria

En el resto de capítulos de la memoria se tratarán los siguientes temas:

- Diseño de la aplicación centrado en la usabilidad.
  - Definición de los perfiles de usuario.
  - Realización de las historias de usuario.
  - Diseño de los prototipos Lo-fi.

- Definición de la arquitectura de la aplicación (Cliente, servidor y Base de datos).
- Realización de los casos de uso.
  - Realización de los diagramas UML.
  - Descripción de los casos de uso.
- Desarrollo de la aplicación.

## 2. Análisis e implementación

### 2.1 Usabilidad/UX

#### 2.1.1 Perfiles de usuario

He realizado un estudio de los diferentes perfiles de usuario que utilizarán la aplicación web DIQSI. En términos generales, destacar que los principales usuarios posiblemente sean mujeres jóvenes de entre 24 y 32, solteras, con un nivel medio en el uso de las nuevas tecnologías y que buscan una herramienta novedosa para compartir información de la boda con sus invitados.

Entre la infinidad de posibilidades destaco, sobre todo, tres tipos de perfiles: uno familiarizado con las nuevas tecnologías y las redes sociales; otro perfil no tan familiarizado con las nuevas tecnologías y que no utiliza con asiduidad las redes sociales; y un último perfil poco familiarizado con las nuevas tecnologías y que no utiliza las redes sociales.

PU_001			
Nombre	Arantxa	Sexo	Mujer
Edad	29	Profesión	Profesora
Estado civil	Soltera	Hijos	0
Idioma	Español	Nivel económico	Medio
Descripción personal	Persona que está organizando su boda y que busca una herramienta novedosa para compartir información del evento con sus invitados.		
Nivel tecnológico	Familiarizada con las nuevas tecnologías y las redes sociales. Tiene experiencia utilizando aplicaciones móviles. Navega con frecuencia por Internet.		

Tabla 2: Perfil de usuario PU\_001

PU_002			
Nombre	Alba	Sexo	Mujer
Edad	32	Profesión	Bióloga
Estado civil	Casada	Hijos	1
Idioma	Catalán	Nivel económico	Medio-Alto
Descripción personal	Persona que no está organizando su propia boda, pero que en su momento ya lo hizo y ahora ayuda a organizar la de una amiga.		
Nivel tecnológico	Familiarizada con las nuevas tecnologías y las redes sociales, aunque no las utiliza a diario para estar conectada con amigos y familiares. Tiene experiencia utilizando aplicaciones móviles. Navega con frecuencia por Internet.		

Tabla 3: Perfil de usuario PU\_002

PU_003			
Nombre	Guillem	Sexo	Hombre
Edad	55	Profesión	Operario
Estado civil	Casado	Hijos	2
Idioma	Español	Nivel económico	Medio
Descripción personal	Persona que no está organizando su boda y que accede a la aplicación como invitado.		
Nivel tecnológico	Poco familiarizada con las nuevas tecnologías y que no utiliza las redes sociales. Tiene muy poca experiencia utilizando aplicaciones móviles. Navega con poca frecuencia en Internet.		

Tabla 4: Perfil de usuario PU\_003

### 2.1.2 Historias de usuario

A continuación se muestran una serie de historias de usuario de la aplicación, con varios criterios de aceptación.

HU_001	
Cómo	Usuario
Quiero	Acceder a la aplicación mediante Facebook.
Para	Acceder a todas las funcionalidades sin tener que recordar siempre el usuario y contraseña.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>Se debe verificar el funcionamiento de la librería que permite acceder a la aplicación mediante esta red social.</li> </ul>	

**Tabla 5: Historia de usuario HU\_001**

HU_002	
Cómo	Usuario
Quiero	Acceder a la aplicación mediante usuario y contraseña.
Para	Acceder a todas las funcionalidades si no dispongo de una cuenta en Facebook.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>Se debe verificar el funcionamiento que permite acceder a la aplicación mediante usuario y contraseña.</li> <li>Se debe verificar el funcionamiento de “recordar contraseña”.</li> </ul>	

**Tabla 6: Historia de usuario HU\_002**

HU_003	
Cómo	Usuario validado
Quiero	Poder obtener un listado de invitados.
Para	Tener una lista actualizada de los invitados con los nombres y el número de personas que necesitaré para gestionar el banquete, invitaciones, etc.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• El listado contendrá los campos:               <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Asistencia</li> <li>• Menú</li> <li>• Mesa</li> <li>• Opciones</li> </ul> </li> </ul>	

Tabla 7: Historia de usuario HU\_003

HU_004	
Cómo	Usuario validado
Quiero	Poder filtrar los invitados según diferentes criterios.
Para	Obtener una lista en función de mis necesidades.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• El listado de invitados debe poder ser filtrado por:               <ul style="list-style-type: none"> <li>• Asistencia                   <ul style="list-style-type: none"> <li>• Confirmados</li> <li>• Pendientes</li> <li>• Cancelados</li> </ul> </li> <li>• Menú</li> <li>• Mesa</li> <li>• Todos</li> </ul> </li> <li>• El sistema buscará correctamente por la categoría seleccionada.</li> </ul>	

Tabla 8: Historia de usuario HU\_004

<b>HU_005</b>	
Cómo	Usuario validado
Quiero	Poder confirmar la asistencia de los invitados.
Para	Tener un control de las personas que asistirán a la boda.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• Desde el propio listado de invitados se debe poder seleccionar el estado del usuario como confirmado, pendiente o cancelado.</li> <li>• Se comprobará que al hacer clic sobre el nuevo estado del invitado se registrará la solicitud en la aplicación y se hará visible para el usuario.</li> </ul>	

Tabla 9: Historia de usuario HU\_005

<b>HU_006</b>	
Cómo	Usuario validado
Quiero	Poder asignar un menú a los invitados.
Para	Tener un control de los menús que tomarán las personas que asistirán a la boda.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• Desde el propio listado de invitados se debe poder seleccionar el menú del usuario.</li> <li>• Se comprobará que al hacer clic sobre el menú se registrará la solicitud en la aplicación y se hará visible para el usuario.</li> </ul>	

Tabla 10: Historia de usuario HU\_006

<b>HU_007</b>	
Cómo	Usuario validado
Quiero	Poder asignar una mesa a los invitados.

Para	Tener un control de las mesas que ocuparán las personas que asistirán a la boda.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>Desde el propio listado de invitados se debe poder seleccionar la mesa del usuario.</li> <li>Se comprobará que al hacer clic sobre la mesa se registrará la solicitud en la aplicación y se hará visible para el usuario.</li> </ul>	

**Tabla 11: Historia de usuario HU\_007**

<b>HU_008</b>	
Cómo	Usuario validado
Quiero	Poder crear invitados.
Para	Añadirlos a la lista de la boda.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>Cuando se haga clic sobre el botón de "Aceptar o guardar" se registrará la solicitud en la aplicación y se hará visible para el usuario.</li> <li>Habrà una validación de los campos y se comprobará que todos los que son obligatorios se han introducido</li> </ul>	

**Tabla 12: Historia de usuario HU\_008**

<b>HU_009</b>	
Cómo	Usuario validado
Quiero	Poder enviar una invitación de boda a los invitados.
Para	Avisarles y recordarles que contamos con ellos para un día tan especial.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>Solo se podrá enviar la invitación si el campo email del invitado no está vacío.</li> </ul>	



- No se podrá volver a enviar una invitación hasta pasado 7 días

**Tabla 13: Historia de usuario HU\_009**

HU_010	
Cómo	Usuario validado
Quiero	Seleccionar, al menos, una plantilla para mostrar la web de boda
Para	Personalizar la web de la boda con el diseño que más guste.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• Debe aparecer una lista de las plantillas disponibles junto con una miniatura.</li> <li>• Debe poder verse en todo momento la web de la boda.</li> <li>• Al seleccionar una plantilla, automáticamente se aplica a la web de boda y se visualizan los cambios al instante.</li> </ul>	

**Tabla 14: Historia de usuario HU\_010**

HU_011	
Cómo	Usuario validado
Quiero	Crear un álbum de fotos
Para	Subir las fotografías que quiera visualizar en mi web de boda.
Criterios de aceptación	
<ul style="list-style-type: none"> <li>• Cuando se haga clic sobre el botón de "Aceptar o guardar" se registrará la solicitud en la aplicación y se hará visible para el usuario.</li> <li>• Habrá una validación de los campos y se comprobará que todos los que son obligatorios se han introducido</li> <li>• Se deben de poder subir de forma simultánea tantas imágenes como se quieran.</li> <li>• En el momento de la subida de la imagen se visualizará una barra de porcentaje para saber cuanto le queda a la imagen por subirse o cuando ha terminado.</li> </ul>	

- Al subirse la imagen, automáticamente aparecerá una miniatura de la misma.

Tabla 15: Historia de usuario HU\_011

### 2.1.3 Prototipo Lo-Fi

A continuación se presentan una serie de prototipos Lo-Fi (*Low Fidelity*) que representan el diseño general de las diferentes pantallas de la aplicación. He decidido presentar solo prototipos de baja fidelidad ya que éstos se caracterizan por ser económicos y rápidos de construir (y arreglar).

Lo-Fi prototipo de la página de registro. El formulario contiene los siguientes elementos:

- Botón de Facebook
- Campo de texto: Nombre y apellidos
- Campo de texto: Email
- Campo de texto: Contraseña
- Campo de texto: Fecha boda
- Campo de texto: Teléfono
- Radio buttons: Yo soy,  Novio,  Novia,  Otro
- Botón: Registrarme

Ilustración 2. Lo-Fi página de registro

Lo-Fi prototipo del modal de datos sobre la boda. El formulario está dividido en secciones:

- FECHA Y LOCALIZACIÓN:**
  - Campo de texto: Fecha de la boda
  - Campo de texto: Población
- MI PAREJA:**
  - Campo de texto: Nombre
  - Campo de lista desplegable: Mi pareja es
- PRESUPUESTO:**
  - Campo de texto: Presupuesto aproximado
  - Campo de texto: N.º de invitados
- Botón: Guardar

Ilustración 3. Lo-Fi modal datos sobre la boda

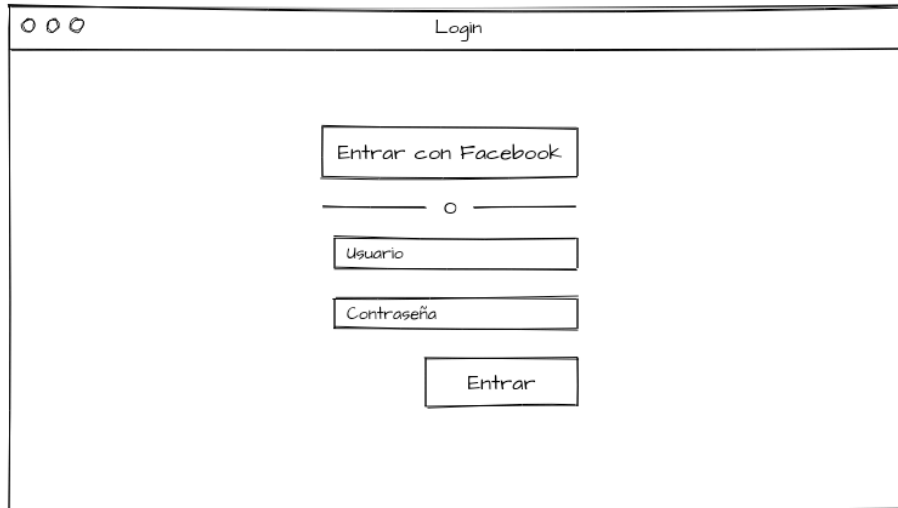


Ilustración 4. Lo-Fi login

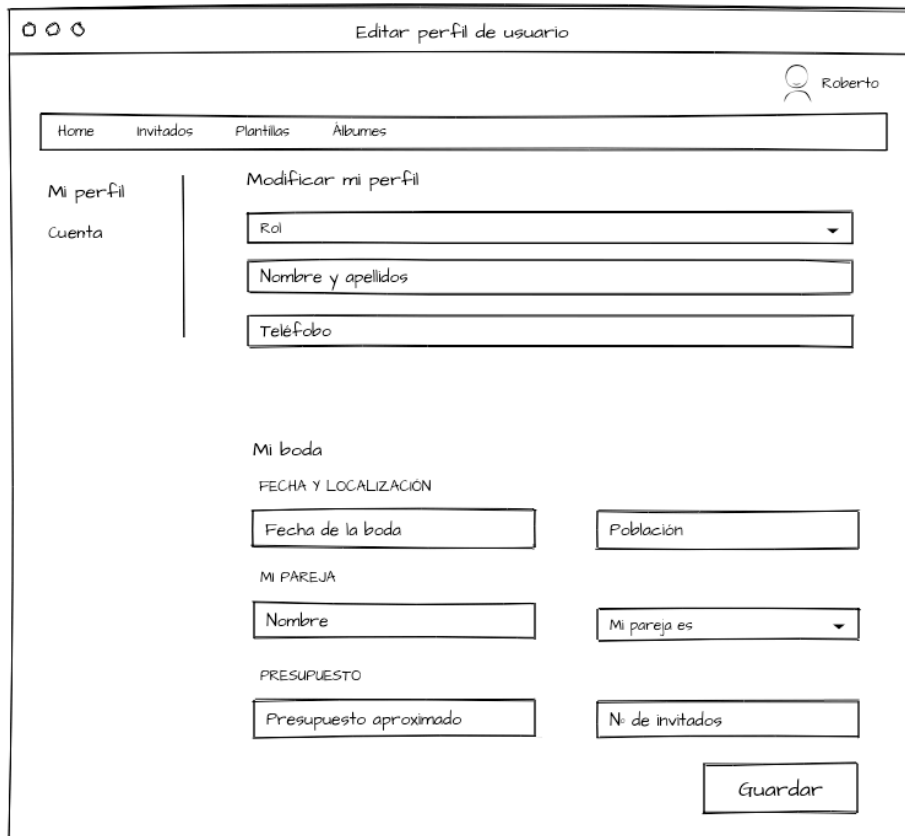


Ilustración 5. Lo-Fi editar perfil de usuario

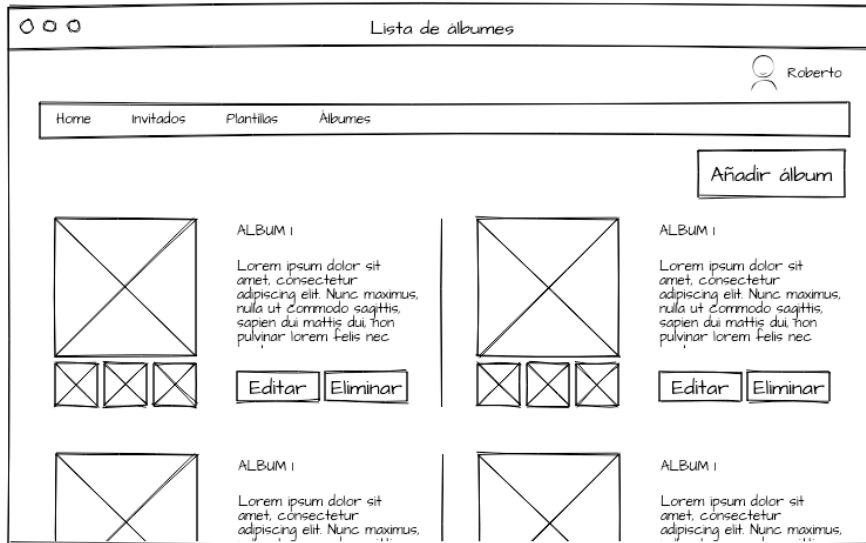


Ilustración 6. Lo-Fi listado de álbumes

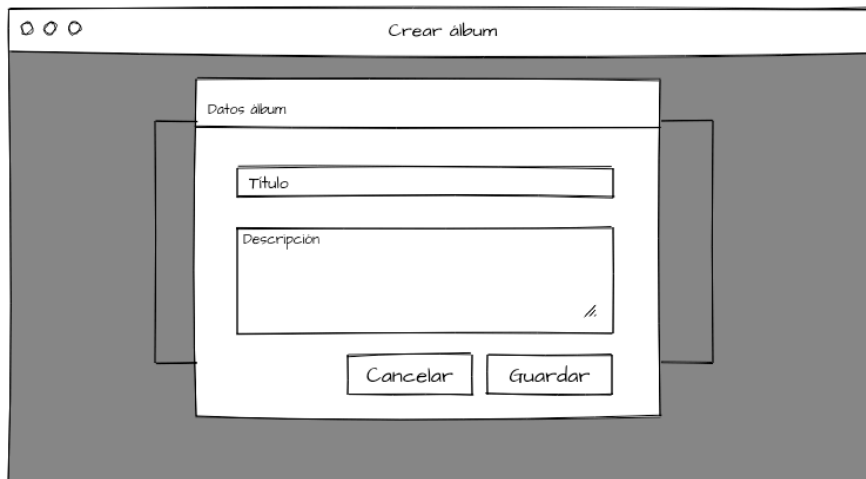


Ilustración 7. Lo-Fi modal crear álbum

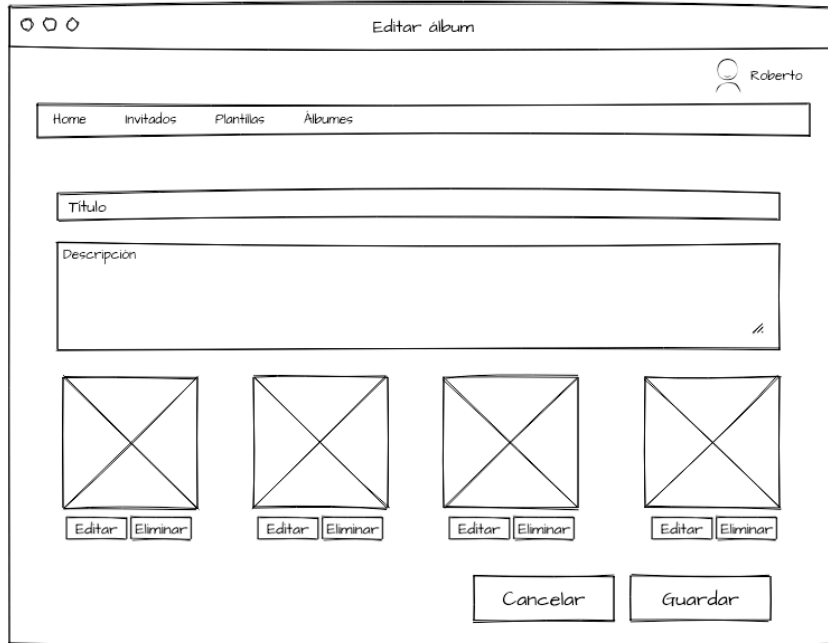


Ilustración 8. Lo-Fi editar álbum

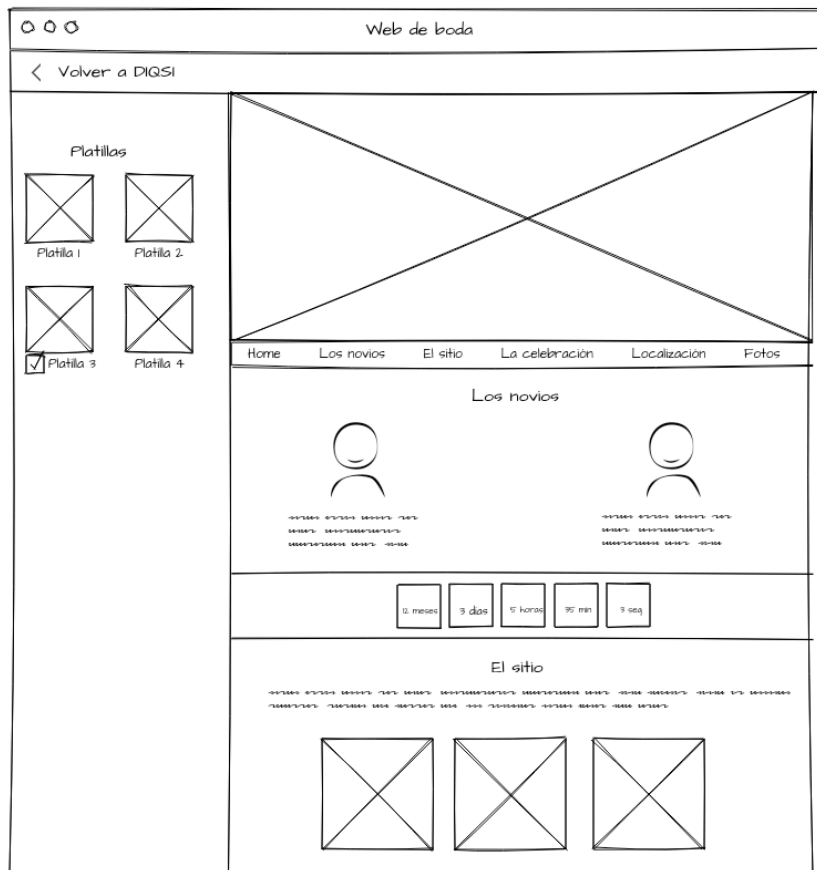


Ilustración 9. Lo-Fi página de plantillas

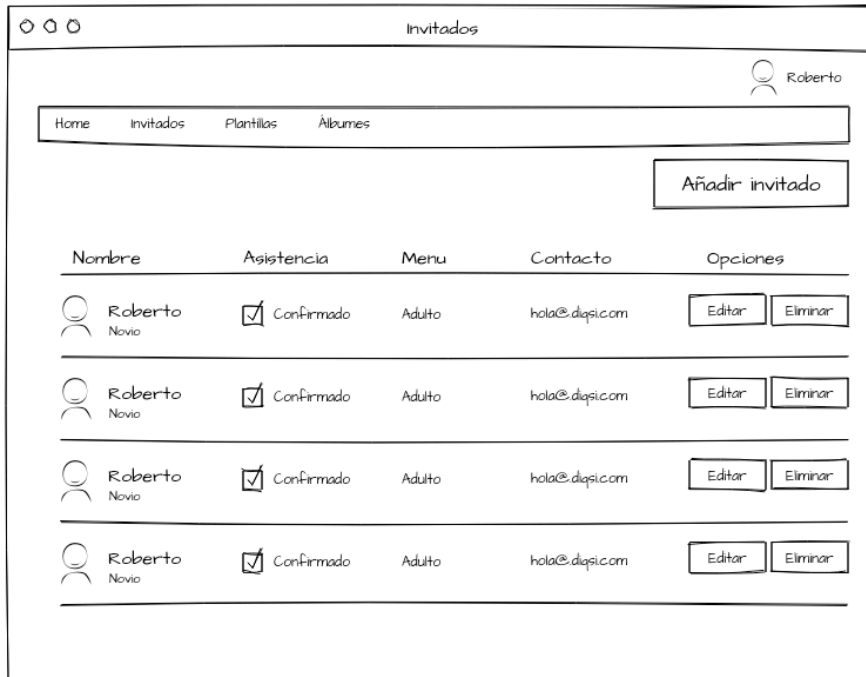


Ilustración 10. Lo-Fi lista de invitados

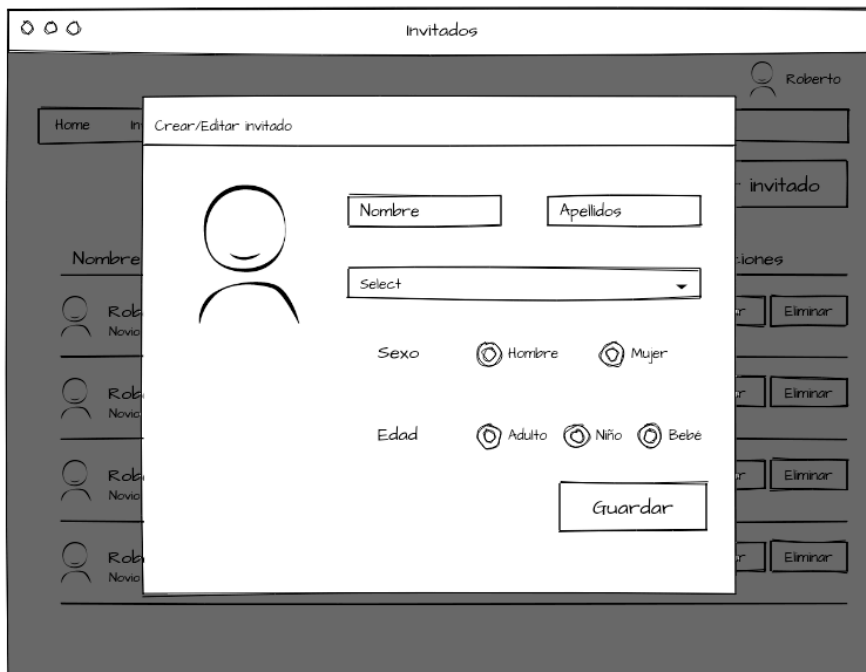


Ilustración 11. Lo-Fi modal crear/editar invitados

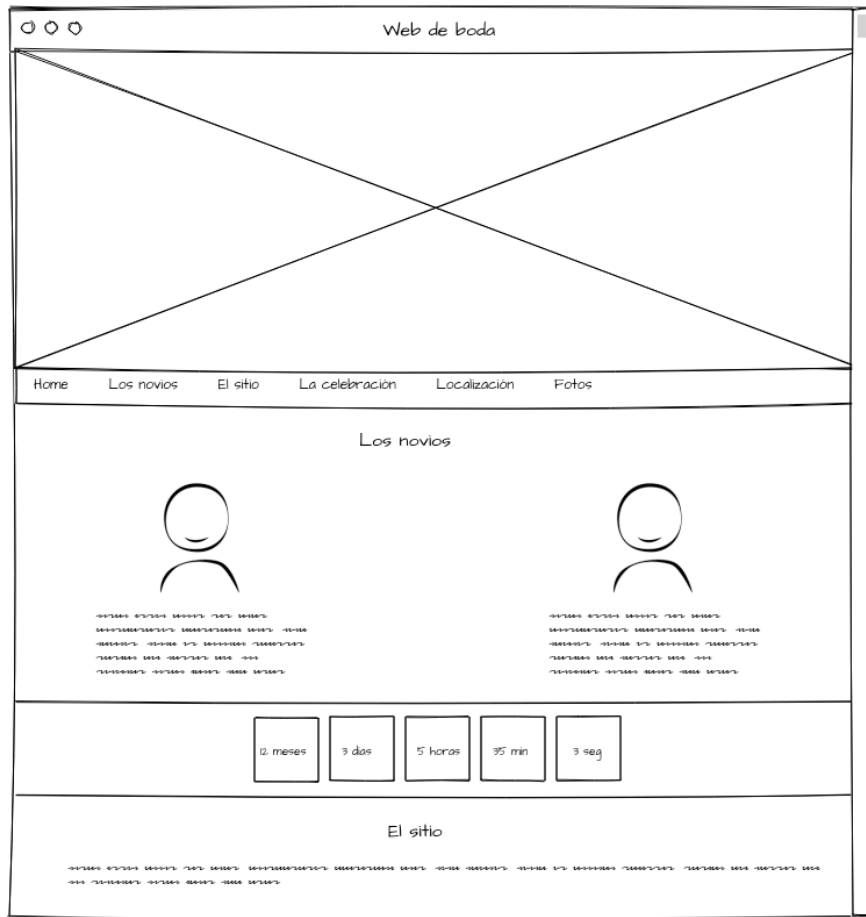


Ilustración 12. Lo-Fi ejemplo plantilla web de boda

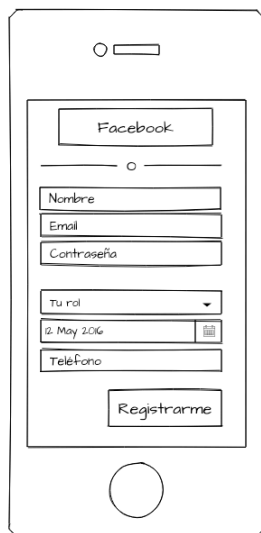


Ilustración 13. Lo-Fi página de registro (móvil)

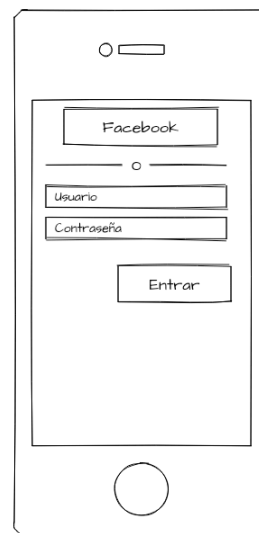
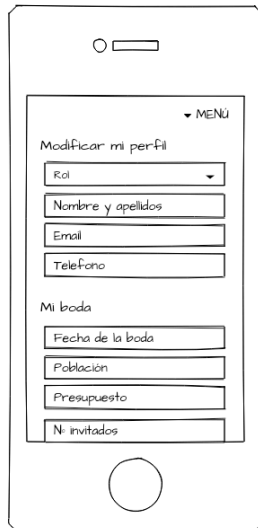
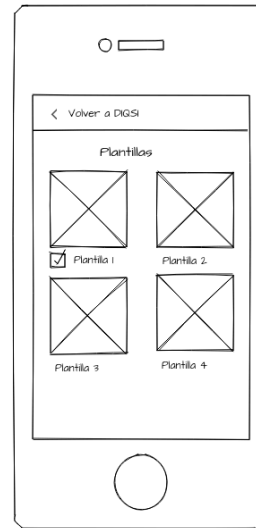


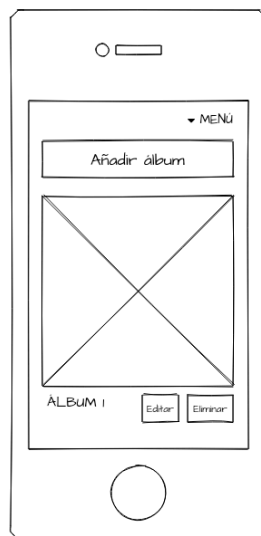
Ilustración 14. Lo-Fi login (móvil)



**Ilustración 15. Lo-Fi editar perfil de usuario (móvil)**



**Ilustración 16. Lo-Fi página de plantillas (móvil)**



**Ilustración 17. Lo-Fi listado de álbumes (móvil)**



**Ilustración 18. Lo-Fi lista de invitados (móvil)**

## 2.2 Arquitectura de la aplicación

El desarrollo de esta aplicación está centrado en una arquitectura que se basa en un modelo Cliente/Servidor. La aplicación web es proporcionada por un servidor y utilizada, a su vez, por usuarios que se conectan desde cualquier lugar mediante un navegador Web.



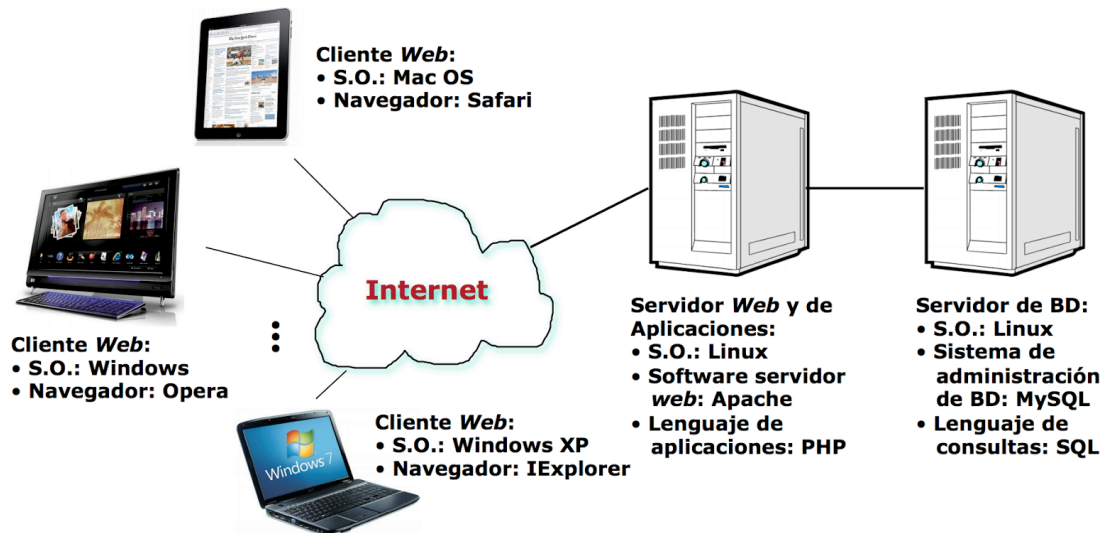


Ilustración 19. Arquitectura cliente/servidor

### 2.2.1 Cliente

Del lado del cliente se utiliza HTML5 (HyperText Markup Language versión 5), la última versión del lenguaje de marcado utilizado en el desarrollo web. Permite definir la estructura de una página web y es el componente estructural clave de todas los sitios de internet.

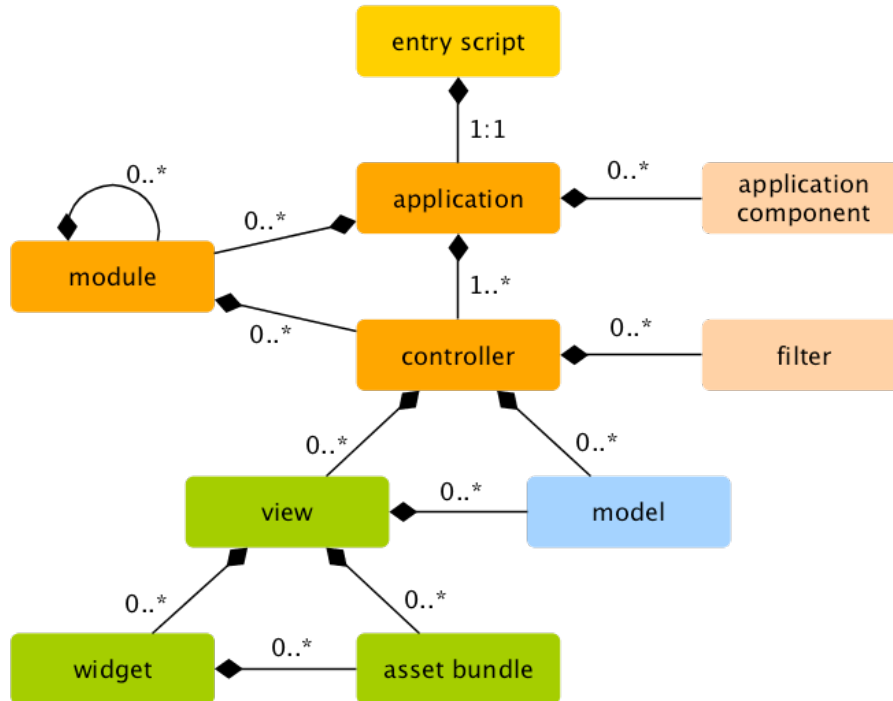
CSS3 (Cascading Style Sheets level 3), que es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en HTML. Básicamente describe cómo se deben representar los elementos en la pantalla.

JavaScript, un lenguaje de programación ligero e interpretado que se ejecuta en el navegador del cliente y que permite realizar actividades complejas.

### 2.2.2 Servidor

El desarrollo de la aplicación web está basado en un patrón de arquitectura de software llamado MVC (Modelo Vista Controlador), el cual es adoptado ampliamente en la programación Web. Para su implementación, en el lado del servidor, he decidido utilizar un framework de PHP de alto rendimiento llamado Yii2.

El siguiente diagrama muestra la estructura estática de una aplicación Yii2.



**Ilustración 20: Estructura estática de aplicación Yii2**

Se puede observar como el diagrama muestra una división entre el modelo, que representa la información (los datos) y las reglas del negocio; el controlador, que administra la comunicación entre la vista y el modelo; y la vista, que contiene elementos de la interfaz de usuario.

### 2.2.3 Base de datos

La capa de persistencia de la aplicación dispone de un sistema de gestión de bases de datos relacional llamado MySQL, que permite organizar los datos en tablas y establecer relaciones entre ellas. Garantiza la integridad funcional de los datos y el correcto almacenamiento de éstas.

#### 2.2.3.1 Modelo Entidad Relación

A continuación se exponen el modelo entidad relación de la base de datos de la aplicación.

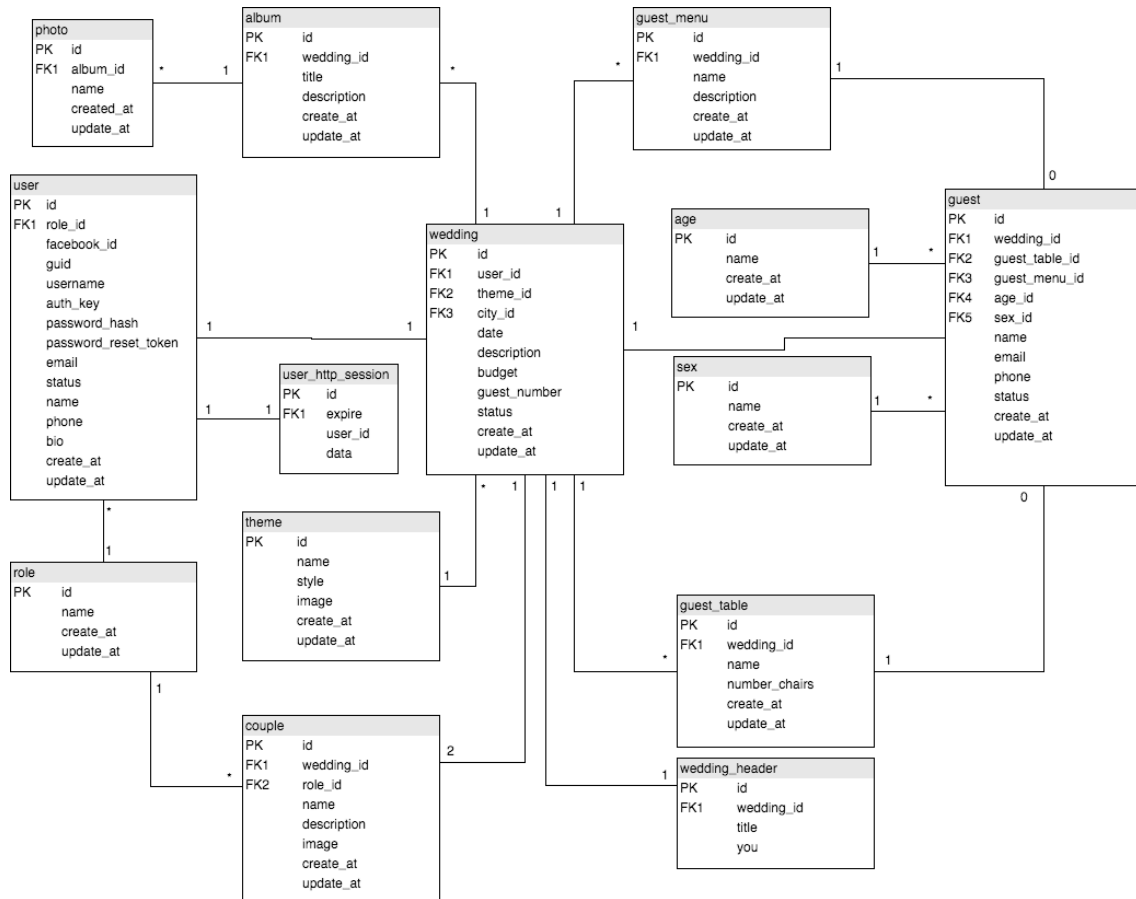


Ilustración 21: Modelo Entidad Relación

### 2.2.4 APIs y librerías utilizadas

Para llevar a cabo el proyecto se han utilizado las siguientes APIs y librerías:

1. jQuery

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

2. Google Maps

API que ofrecen a los desarrolladores varias formas de incorporar Google Maps a páginas web o recuperar datos de Google Maps.

3. Google Fonts

Es un directorio interactivo de uso libre que programa interfaces para fuentes web. Permite cargar tipografías no existentes en la librería personal del usuario.

#### 4. Font Awesome

Biblioteca que mediante el uso de la propiedad font-face de CSS3 y la etiqueta de HTML5 nos da acceso a una gran variedad de iconos totalmente escalables y en constante actualización.

#### 5. Bootstrap

Framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web.

### 2.3 Casos de uso

#### 2.3.1 Diagrama UML de casos de uso

A continuación se exponen los principales casos de uso de la aplicación.

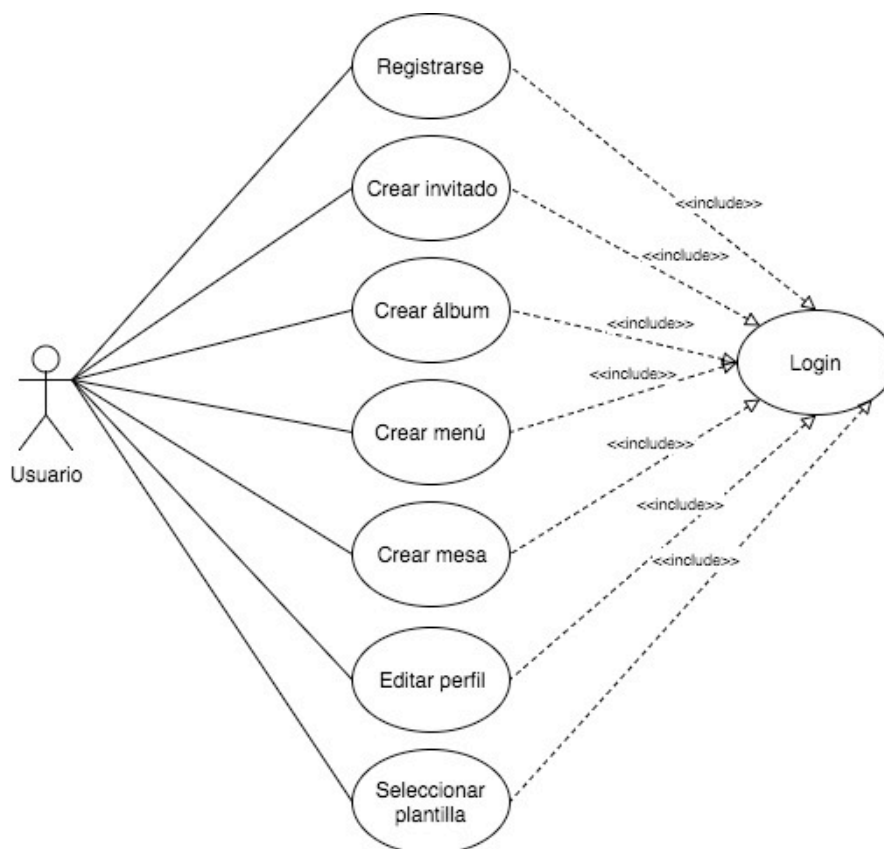


Ilustración 22: Diagrama UML de casos de uso

Para poder hacer uso de las funcionalidades de la aplicación , el usuario debe haber iniciado sesión previamente. Los casos de uso se detallan en el siguiente apartado.

### 2.3.2 Descripción de los casos de uso

CU_001	
Nombre	Login
Descripción	Permite al usuario identificarse en la aplicación
Actores	Usuario
Flujo normal	
<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla de “Login”.</li> <li>2. El usuario introduce los datos de identificación (todos obligatorios) o hace click para identificarse mediante Facebook.</li> <li>3. La aplicación comprueba que los datos sean correctos e identifica al usuario en su sesión.</li> </ol>	
Flujo alternativo	
<p>3a. Los datos proporcionados por el usuario no son correctos o existe algún problema con la identificación con Facebook.</p> <p>3a1. La aplicación muestra la pantalla “Login” con un mensaje de error.</p> <p>3a2. Ir al punto 2.</p>	

Tabla 16: Caso de uso CU\_001

CU_002	
Nombre	Registro
Descripción	Permite al usuario registrarse en la aplicación para poder utilizar todas las funcionalidades
Actores	Usuario

Casos de uso relacionados	Extiende el caso de uso "Login" en el paso 2
Flujo normal	
<ol style="list-style-type: none"> <li>1. La aplicación muestra al usuario la pantalla de "Registro".</li> <li>2. El usuario introduce los datos de identificación (todos obligatorios) o hace click para identificarse mediante Facebook.</li> <li>3. La aplicación comprueba que los datos sean correctos y registra al usuario.</li> </ol>	
Flujo alternativo	
<p>2a. El usuario hace click en "Cancelar".</p> <p>2a1. El caso de uso finaliza.</p> <p>3a. Los datos proporcionados por el usuario no son correctos o existe algún problema con la identificación con Facebook.</p> <p>3a1. La aplicación muestra la pantalla "Registro" con un mensaje de error.</p> <p>3a2. Ir al punto 2.</p>	

**Tabla 17: Caso de uso CU\_002**

CU_003	
Nombre	Editar perfil
Descripción	Permite al usuario modificar la información de su perfil y los datos de su boda.
Actores	Usuario validado
Precondiciones	El usuario ha iniciado sesión en la aplicación
Flujo normal	
<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil.</li> <li>2. La aplicación muestra la información de perfil.</li> <li>3. El usuario hace click en "Editar perfil".</li> <li>4. La aplicación muestra la pantalla de "Datos de perfil".</li> <li>5. El usuario rellena los datos y hace click en "Guardar" para almacenar los cambios</li> <li>6. El sistema guarda las modificaciones realizadas</li> </ol>	
Flujo alternativo	

<p>5a. Los datos proporcionados por el usuario no son correctos.                  5a1. La aplicación muestra la pantalla de "Datos de perfil" con un mensaje de error                  5a2. Ir al punto 4</p>
---

**Tabla 18: Caso de uso CU\_003**

<b>CU_004</b>	
Nombre	Crear invitado
Descripción	Permite al usuario crear invitados para añadirlos a la lista de la boda
Actores	Usuario validado
Precondiciones	El usuario ha iniciado sesión en la aplicación
Flujo normal	
<ol style="list-style-type: none"> <li>1. El usuario hace click en "Añadir invitado".</li> <li>2. La aplicación muestra la pantalla "Datos del invitado". El usuario rellena los datos y hace click en "Añadir".</li> <li>3. La aplicación comprueba que los datos sean correctos y los guarda en la base de datos.</li> </ol>	
Flujo alternativo	
<p>2a. El usuario hace click en "Cancelar"                  2a1. El caso de uso finaliza</p> <p>3a. Los datos proporcionados por el usuario no son correctos.                  3a1. La aplicación muestra la pantalla de "Datos del invitado" con un mensaje de error                  3a2. Ir al punto 2</p>	

**Tabla 19: Caso de uso CU\_004**

<b>CU_005</b>	
Nombre	Crear menú
Descripción	Permite al usuario crear menús que luego podrá asignar a los invitados.

Actores	Usuario validado
Precondiciones	El usuario ha iniciado sesión en la aplicación
Flujo normal	
<ol style="list-style-type: none"> <li>1. El usuario hace click en "Crear menú".</li> <li>2. La aplicación muestra la pantalla "Añadir nuevo menú". El usuario rellena los datos y hace click en "Añadir".</li> <li>3. La aplicación comprueba que los datos sean correctos y los guarda en la base de datos.</li> </ol>	
Flujo alternativo	
<ol style="list-style-type: none"> <li>2a. El usuario hace click en "Cancelar" <ol style="list-style-type: none"> <li>2a1. El caso de uso finaliza</li> </ol> </li> <li>3a. Los datos proporcionados por el usuario no son correctos. <ol style="list-style-type: none"> <li>3a1. La aplicación muestra la pantalla de "Añadir nuevo menú" con un mensaje de error</li> <li>3a2. Ir al punto 2</li> </ol> </li> </ol>	

Tabla 20: Caso de uso CU\_005

<b>CU_006</b>	
Nombre	Crear álbum
Descripción	Permite al usuario crear álbumes de fotos que luego se podrán ver en la web de boda.
Actores	Usuario validado
Precondiciones	El usuario ha iniciado sesión en la aplicación
Flujo normal	
<ol style="list-style-type: none"> <li>1. El usuario hace click en "Añadir álbum".</li> <li>2. La aplicación muestra la pantalla "Añadir nuevo álbum". El usuario rellena los datos y añade las fotos que quiera. Luego el usuario hace click en "Guardar".</li> <li>3. La aplicación comprueba que los datos sean correctos y los guarda en la base de datos.</li> </ol>	
Flujo alternativo	



<p>2a. El usuario hace click en "Cancelar"</p> <p>2a1. El caso de uso finaliza</p> <p>3a. Los datos proporcionados por el usuario no son correctos.</p> <p>3a1. La aplicación muestra la pantalla de "Añadir nuevo álbum" con un mensaje de error.</p> <p>3a2. Ir al punto 2</p>
--

**Tabla 21: Caso de uso CU\_006**

<b>CU_007</b>	
Nombre	Seleccionar plantilla
Descripción	Permite al usuario seleccionar una plantilla para el diseño de su web de boda.
Actores	Usuario validado
Precondiciones	El usuario ha iniciado sesión en la aplicación
Flujo normal	
<ol style="list-style-type: none"> <li>1. La aplicación muestra al usuario la pantalla de "Diseño".</li> <li>2. El usuario selecciona la plantilla.</li> <li>3. La aplicación almacena automáticamente en base de datos el cambio solicitado y actualiza la pantalla de "Diseño" con los nuevos cambios.</li> </ol>	
Flujo alternativo	
<p>2a. La aplicación no ha podido almacenar el cambio de diseño.</p> <p>2a1. Se muestra un aviso al usuario para que vuelva a realizar la petición</p>	

**Tabla 22: Caso de uso CU\_007**

### 3. Instalación

#### 3.1 Requisitos de instalación

Para la implementación de la aplicación web DIQSI será necesario disponer de un servidor que cumpla con los siguientes requisitos: disponer de un servidor web (Apache), disponer de un sistema relacional de administración de bases de datos (MySQL) así como de un interprete de un lenguaje de programación del lado del servidor (PHP).

### 3.1 Instrucciones de instalación

Una vez disponemos de un servidor web, a continuación se detallan los pasos a seguir para la correcta instalación:

1. Subir los archivos de la carpeta "diqsi" al servidor manteniendo la misma estructura de carpetas.

Este proceso se puede realizar mediante el uso de un gestor FTP para conectarse al servidor y subir todos los archivos a la carpeta raíz del servidor web.

2. Configuración de la base de datos

Mediante el uso del archivo SQL disponible en la carpeta "bbdd", se podrá crear toda la estructura de la base de datos. Este archivo contiene todas las tablas, campos y relaciones necesarias para el funcionamiento de la web.

Para la carga del archivo se puede utilizar un gestor MySQL como PhpMyAdmin, que permitirá seleccionar el archivo SQL proporcionado y revisar la estructura creada de forma gráfica.

3. Configurar el archivo de conexión de la aplicación web.

Cada servidor donde se quiera instalar la aplicación dispone de unos datos de conexión a la base de datos diferente, por ese motivo será necesario editar los datos de conexión ubicados en "common/config/db.php".

El contenido de este archivo deberá ser editado con los datos de conexión del servidor en particular.

```
return [  
    'class' => 'yii\db\Connection',  
    'dsn' => 'mysql:host=SERVIDOR_BBDD;dbname= NOMBRE_BBDD;',  
    'username' => USUARIO_BBDD,  
    'password' => 'CONTRASEÑA_BBDD ',  
    'charset' => 'utf8',  
];
```

Donde **SERVIDOR\_BBDD** sería el nombre del servidor donde se encuentra alojada la base de datos (En el caso de una instalación localhost deberá de modificarse por el valor 127.0.0.1), **NOMBRE\_BBDD** sería el nombre de la base de datos, **USUARIO\_BBDD** sería el nombre de usuario de la base de datos y **CONTRASEÑA\_BBDD** sería la contraseña de la base de datos.

Finalmente el archivo deberá de tener un aspecto parecido al siguiente:

```
return [  
    'class' => 'yii\db\Connection',  
    'dsn' => 'mysql:host=127.0.0.1;dbname=diqsi',  
    'username' => 'root',  
    'password' => 'root',  
    'charset' => 'utf8',  
];
```

Es posible que en algunos servidores se deba proporcionar la configuración de un puerto para el correcto funcionamiento. En ese caso el archivo de conexión deberá verse tal y como se muestra a continuación:

```
return [  
    'class' => 'yii\db\Connection',  
    'dsn' => 'mysql:host=127.0.0.1;dbname=diqsi;port=8889',  
    'username' => 'root',  
    'password' => 'root',  
    'charset' => 'utf8',  
];
```

Para visualizar la aplicación en el navegador, se deberá acceder a la URL de nuestro localhost (en mi caso <http://localhost:8888/>) y añadir `'diqsi/frontend/web/` al final.

### 3.2 Material para el uso de la aplicación

Se incluye una carpeta “material” dentro del directorio raíz con fotografías y textos de ejemplo para poder ser utilizados en la aplicación.

Para las pruebas de envío de email he utilizado *Mailtrap*, un servidor SMTP falso para que los equipos de desarrollo puedan probar, ver y compartir correos electrónicos enviados desde los entornos de desarrollo y sin clientes reales.

Podremos acceder al servicio desde la siguiente URL:

<https://mailtrap.io/inboxes/300431/messages>

Y con las siguientes credenciales:

Usuario: **robert-z@hotmail.es**

Contraseña: **123456**

## 4. Desarrollo

### 4.1 Extractos de código

Como se ha explicado con anterioridad, el proyecto se ha desarrollado mediante un patrón de arquitectura de software llamado MVC (Modelo Vista Controlador). Al no considerar puntos más importantes que otros en el desarrollo de la aplicación, se muestra a modo de ejemplo las vistas, modelos y controladores que forman parte del proceso de login para explicar como funciona este patrón.

- Controlador AuthController.php

```
<?php
namespace frontend\modules\user\controllers;

use Yii;
use yii\base\InvalidParamException;
use yii\web\BadRequestHttpException;
use yii\web\Controller;
use yii\filters\VerbFilter;
use yii\filters\AccessControl;
use frontend\modules\user\models\LoginForm;
use frontend\modules\user\models>PasswordResetRequestForm;
use frontend\modules\user\models\ResetPasswordForm;
use frontend\modules\user\models\SignupForm;
use common\models\Role;
use common\models\Wedding;
use frontend\modules\guests\models\Guest;
use frontend\modules\guests\models\GuestTable;
use frontend\modules\guests\models\GuestMenu;
use common\models\Couple;

/**
 * Auth controller
 */
class AuthController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['logout', 'signup'],
                'rules' => [
                    [
                        'actions' => ['signup'],
                        'allow' => true,

                        'roles' => ['?'],
                    ],
                    [
                        'actions' => ['logout'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
        ],
    }
}
```

```

    ],
    'verbs' => [
        'class' => VerbFilter::className(),
        'actions' => [
            'logout' => ['post'],
        ],
    ],
];
}

/**
 * @inheritdoc
 */
public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
    ];
}

public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        return $this->goBack();
    } else {
        return $this->render('login', [
            'model' => $model,
        ]);
    }
}

public function actionLogout()
{
    Yii::$app->user->logout();

    return $this->goHome();
}

public function actionSignup()
{
    $model = new SignupForm();
    if ($model->load(Yii::$app->request->post())) {

        if ($user = $model->signup()) {

            $wedding = new Wedding();

            if($wedding->createWedding($user, $model)){

                if($model->role == 1 || $model->role == 2) {
                    //Crear novios
                    $couple = new Couple();
                    $couple->name = $user->name;
                    $couple->wedding_id = $wedding->id;
                    $couple->role_id = $model->role;
                }
            }
        }
    }
}

```

```

        $couple->save();

        $couple = new Couple();
        $couple->wedding_id = $wedding->id;
        $couple->save();

        $presidentialTable = new GuestTable();
        $presidentialTable->wedding_id = $wedding->id;
        $presidentialTable->name = 'Presidencial';
        $presidentialTable->save();

        $presidentialMenu = new GuestMenu();
        $presidentialMenu->wedding_id = $wedding->id;
        $presidentialMenu->name = 'Adulto';
        $presidentialMenu->save();

        //Crear invitados
        $guest = new Guest();
        $guest->wedding_id = $wedding->id;
        $guest->guest_table_id = $presidentialTable->id;
        $guest->guest_menu_id = $presidentialMenu->id;
        $guest->age_id = 1;
        $guest->sex_id = ($model->role == 1) ? 2 : 1;
        $guest->name = $user->name;
        $guest->email = $user->email;
        $guest->editable = Guest::STATUS_NO_EDITABLE;
        $guest->status = Guest::STATUS_CONFIRMED;
        $guest->save();
    }

    if (Yii::$app->getUser()->login($user)) {
        return $this->goHome();
    }
}

}
}

$roles = Role::find()->all();

return $this->render('signup', [
    'model' => $model,
    'roles' => $roles
]);
}

/**
 * Requests password reset.
 *
 * @return mixed
 */
public function actionRequestPasswordReset()
{
    $model = new PasswordResetRequestForm();
    if ($model->load(Yii::$app->request->post()) && $model->validate()) {
        if ($model->sendEmail()) {

            Yii::$app->session->setFlash('success', 'Revise su correo electrónico para obtener
            más instrucciones.');
```

```

            return $this->goHome();
        } else {
            Yii::$app->session->setFlash('error', 'Lo sentimos, no podemos restablecer la

```

```

contraseña de la dirección de correo electrónico proporcionada.});

    }
}

return $this->render('requestPasswordResetToken', [
    'model' => $model,
]);
}

/**
 * Resets password.
 *
 * @param string $token
 * @return mixed
 * @throws BadRequestHttpException
 */
public function actionResetPassword($token)
{
    try {
        $model = new ResetPasswordForm($token);
    } catch (InvalidParamException $e) {
        throw new BadRequestHttpException($e->getMessage());
    }

    if ($model->load(Yii::$app->request->post()) && $model->validate() && $model->resetPassword()) {
        Yii::$app->session->setFlash('success', 'Nueva contraseña guardada');

        return $this->goHome();
    }

    return $this->render('resetPassword', [
        'model' => $model,
    ]);
}
}
}
}

```

Quando se intenta acceder a una URL de la aplicación que requiere del inicio de una sesión con usuario y contraseña, la aplicación automáticamente se redirige a `user/auth/login`. Esta URL llama al método público `actionLogin()` (correspondiente al tercer parámetro de la URL), que se encuentra en el controlador `AuthController` (correspondiente al segundo parámetro de la URL), que a su vez se encuentra dentro de la carpeta del módulo de `user` (correspondiente al primer parámetro de la URL).

Podemos ver como en las primeras líneas del método `actionLogin()` se comprueba si el usuario actual es un usuario logeado o, por el contrario, si se trata de un usuario invitado.

```

if (!Yii::$app->user->isGuest) {
    return $this->goHome();
}

```

En las siguientes líneas (en el caso de ser un usuario invitado), se hace una llamada al modelo *LoginForm* y vemos una estructura condicional donde se comprueba si se están recibiendo parámetros tipo POST de un formulario y si el método *login()* del modelo *LoginForm* devuelve *true*.

```
if ($model->load(Yii::$app->request->post()) && $model->login()) {
    return $this->goBack();
}
```

En el caso de cumplir la condición, se redirecciona al usuario a la última URL que se ha intentado acceder o a la página de inicio de la aplicación.

En caso contrario, se renderiza la vista de *login*, pasándole como parámetro la variable del modelo *LoginForm*, que se encargará de realizar las correspondientes validaciones del formulario.

```
return $this->render('login', [
    'model' => $model,
]);
```

- Vista login.php

```
<?php

use yii\helpers\Html;
use yii\bootstrap\ActiveForm;

$this->title = 'Login';
$this->params['breadcrumbs'][] = $this->title;
?>

<div class="layout-auth">
    <div class="row">
        <div class="col-lg-12">
            <div class="layout-facebook">
                <?= Html::a('<i class="fa fa-facebook-square"></i> Inicia sesión con Facebook', ['#],
                ['class' => 'btn btn-facebook disabled']); ?>
                <small class="no-publicar">No publicaremos nada sin tu permiso</small>
            </div>
            <?php $form = ActiveForm::begin(['id' => 'form-login', 'options' => ['class' => 'auth-form']]);
            ?>
            <div class="row">
                <div class="col-lg-12">
                    <?=
                        $form->field($model, 'email')->textInput([
                            'autofocus' => true,
                            'placeholder' => 'Email',

                            ]->label(false);
                    ?>
                    <?=
                        $form->field($model, 'password')->passwordInput([
                            'placeholder' => 'Contraseña',
                            ]->label(false);
```



```

?>

    <?= $form->field($model, 'rememberMe')->checkbox() ?>
    <small style="color:#999;margin:1em 0">
        Si olvidó su contraseña, puede <?= Html::a('restablecerla', ['auth/request-password-reset']) ?>.
    </small>
</div>
</div>
<div class="registrate text-center">
    <?= Html::submitButton('Acceder', ['class' => 'btn btn-pink auth-submit-button', 'name' => 'login-button']) ?>
    <div class="ya-cuenta">
        <span class="layout-action">¿Aún no tienes cuenta?</span>
        <?= Html::a('Registrate', ['auth/signup']); ?>
    </div>
</div>
<?php ActiveForm::end(); ?>
</div>
</div>
</div>

```

En esta vista, y gracias a las herramientas que ofrece el framework Yii2, podemos generar un formulario de inicio de sesión que automáticamente validará todos sus campos, según las reglas de validación del modelo que hemos pasado anteriormente como parámetro a esta vista.

```

$form->field($model, 'email')->textInput([
    'autofocus' => true,
    'placeholder' => 'Email',
])->label(false);

```

En este caso se está imprimiendo en la vista un input tipo texto correspondiente al campo email de la tabla *User* de la base de datos.

- Modelo LoginForm.php

```

<?php
namespace frontend\modules\user\models;

use Yii;
use yii\base\Model;

class LoginForm extends Model
{
    public $email;
    public $password;
    public $rememberMe = true;

    private $_user;

    public function rules()
    {

```

```

return [

    [['email'], 'required', 'message' => Yii::t('app', 'El email no puede estar vacío)],
    [['password'], 'required', 'message' => Yii::t('app', 'La contraseña no puede estar vacía)],
    ['rememberMe', 'boolean'],
    ['password', 'validatePassword'],
];
}

public function attributeLabels()
{
    return [
        'email' => 'Email',
        'password' => 'Contraseña',
        'rememberMe' => 'Recuerdame',
    ];
}

public function validatePassword($attribute, $params)
{
    {
        if (!$this->hasErrors()) {
            $user = $this->getUser();
            if (!$user || !$user->validatePassword($this->password)) {
                $this->addError($attribute, 'Email o contraseña incorrectos.');
            }
        }
    }
}

public function login()
{
    {
        if ($this->validate()) {
            return Yii::$app->user->login($this->getUser(), $this->rememberMe ? 3600 * 24 * 30 :
0);
        } else {
            return false;
        }
    }
}

protected function getUser()
{
    {
        if ($this->_user === null) {
            $this->_user = User::findByEmail($this->email);
        }

        return $this->_user;
    }
}
}

```

Como se ha comentado anteriormente, este modelo contiene las reglas de validación del formulario que se renderiza en la vista *login*.

```

public function rules()
{
    return [

        [['email'], 'required', 'message' => Yii::t('app', 'El email no puede estar vacío)],
        [['password'], 'required', 'message' => Yii::t('app', 'La contraseña no puede estar vacía)],
    ];
}

```

```
        ['rememberMe', 'boolean'],  
  
        ['password', 'validatePassword'],  
    ];  
}
```

Se puede observar como el campo *email* y *password* del formulario deberán ser un campo obligatorio.

Como se ha mencionado anteriormente, el método *login()* que aparecía en el controlador *AuthController*, se encarga de validar el formulario mediante las reglas de validación para posteriormente, realizar un inicio de sesión.

```
public function login()  
{  
    if ($this->validate()) {  
        return Yii::$app->user->login($this->getUser(), $this->rememberMe ? 3600 * 24 * 30 : 0);  
    } else {  
        return false;  
    }  
}
```

Para ello se le ha de pasar como parámetro el usuario logeado. Es por ello que se llama al método *getUser()* del mismo modelo, que a su vez se encarga de llamar al método *findByEmail()* del modelo *User*. El método *findByEmail()* buscará en la base de datos un usuario activo con el email introducido en el input tipo texto del formulario. Este input se visualiza en la vista *login.php*.

```
protected function getUser()  
{  
    if ($this->_user === null) {  
        $this->_user = User::findByEmail($this->email);  
    }  
  
    return $this->_user;  
}
```

## 4.2 Seguridad

La aplicación se beneficia de la utilización de la arquitectura de autenticación, seguridad y autorización que ofrece el framework Yii2, y que aporta muchas medidas de seguridad para ayudar a prevenir ataques como inyección SQL, cross-site scripting (XSS), cross-site request forgery (CSRF), y manipulación de cookies.

- Autenticación

El modelo de usuario de Yii se integra con la interfaz de identidad para manejar el estado de autenticación de usuario dentro de la aplicación, permitiendo también acceder fácilmente a la información de perfil del usuario.

Yii2 proporciona integración pre-construida del modelo de usuario con ActiveRecord y la base de datos. De esta forma la aplicación ofrece autenticación manejada por base de datos desde el principio.

El modelo de usuario, además, permite loguear y desloguear al usuario de forma programática.

- Autorización

Yii2 nos proporciona soporte ACL (Access Control List), ideal para aplicaciones que solo necesitan un control de acceso simple, como en el caso de nuestra aplicación. El control de acceso se integra con las características de autenticación para permitir o restringir el acceso a características o páginas específicas de la aplicación.

- Contraseñas

Yii2 facilita encriptar y verificar las contraseñas de forma segura utilizando, por ejemplo, *bcrypt* para generar hashes que permiten encriptar nuestras contraseñas.

- Seguridad en las vistas

Yii2 ofrece también una serie de métodos que limpian cualquier información que se vaya a mostrar a los usuarios en las vistas, previniendo de esta forma ataques como la inyección SQL.

### 4.3 Tests

En la parte final del proceso de desarrollo de esta primera versión, se ha sometido a la web DIQSI a una serie de pruebas (técnicas y de usabilidad) para poder determinar algunas mejoras de cara a nuevas versiones, con el objetivo de corregir los puntos a mejorar en siguientes fases del proyecto.

- Test de usabilidad

Se han realizado una serie de test de usabilidad con usuarios reales para poder comprobar si la usabilidad de la aplicación es la correcta. Para ello, y basándonos en los perfiles de usuario descritos en el

capítulo 2.1.1, se han definido una serie de tareas que los usuarios han tenido que realizar:

- Dar de alta un usuario nuevo.
- Iniciar sesión una sesión nueva con un usuario existente.
- Modificar el email de la cuenta.
- Modificar la contraseña de la cuenta.
- Modificar los datos personales del usuario.
- Modificar la información general sobre la boda.
- Crear un álbum de la boda.
- Añadir fotografías al álbum de la boda.
- Seleccionar una plantilla para la web de la boda.
- Editar la información de la cabecera de la web de la boda.
- Editar la información de los novios de la web de la boda.
- Crear un invitado para la boda.
- Editar un invitado de la boda.
- Crear un menú nuevo.
- Crear una mesa nueva.
- Enviar una invitación de la boda a un invitado.
- Exportar un listado de invitados.

De las diferentes tareas propuestas anteriormente, todos los usuarios han logrado finalizar con éxito cada una de ellas. En uno de los casos, el usuario ha mencionado la falta de avisos cuando se realiza una acción satisfactoriamente. Finalmente se opta por repasar todas las acciones de la web y añadir dicho aviso que informa al usuario de la acción realizada.

- Test técnicos

La web ha presentado algunos problemas de compatibilidad en algunas versiones antiguas de navegadores, pero que quedan fuera del alcance de este proyecto.

Además, se ha comprobado el correcto funcionamiento en los navegadores Chrome, Firefox y Safari para sistema operativo Mac OS X, y Chrome, Firefox y Edge para sistema operativo Windows.

También se ha comprobado la correcta adaptación de la web de boda para dispositivos con diferentes resoluciones de pantalla. Para los dispositivos basados en iOS, las pruebas se han realizado en Safari y para los dispositivos basados en Android, las pruebas se han realizado en Chrome.

Se han realizado validaciones de HTML y CSS mediante los validadores de la W3C, los cuales no han pasado satisfactoriamente. Estos problemas se solucionarán en próximas fases del proyecto.

#### 4.4 Versiones de la aplicación

A continuación se muestran las diferentes versiones de la aplicación junto con las implantaciones que lleva cada una.

##### Versión Alpha (Primera entrega)

- Login de usuario
- Registro de usuario
- Editar perfil de usuario
- Editar datos de la cuenta de usuario
- Seleccionar plantilla
- Editar cabecera de la plantilla
- Editar sección “novios” de la plantilla
- Crear álbum
- Añadir fotos al álbum

##### Versión Alpha (Entrega final)

- Crear y editar invitados.
- Crear mesas de invitados.
- Crear menús para invitados.

Durante el desarrollo de esta primera versión, no se ha considerado que aporte valor el hecho de que el usuario pueda crear más de un álbum de fotos. Es por ello que únicamente se ha desarrollado la posibilidad de crear y editar un solo álbum para la web de boda.

#### 4.1 Bugs

Se ha encontrado un error en el login de la aplicación que redirecciona al usuario a un error 404 al loguearse. Este error apareció en una única ocasión. Se ha intentado, sin éxito, reproducir el error con el objetivo de solucionarlo antes de la entrega del proyecto. Se buscará una solución al mismo en futuras versiones de la aplicación.

## 6. Conclusiones

La realización del Trabajo Final de Grado ha sido una experiencia gratificante que me planteé como un reto personal y como una oportunidad de plasmar los conocimientos adquiridos durante todos estos años en el grado. Era una idea que hacía tiempo que tenía en mente y realmente me ha gustado desarrollarla. Creo que esto ha sido un plus de motivación que me ha hecho realizar un esfuerzo adicional para obtener un resultado bastante satisfactorio.

A nivel de desarrollo, he podido terminar casi todas las funcionalidades que me había marcado como objetivo. No obstante, en este proceso he priorizado los

aspectos relacionados con la programación y finalización de las funcionalidades básicas. Es por ello que el diseño ha quedado en un segundo lugar, dejando la posibilidad de mejorarlo en futuras versiones.

En cuanto a la planificación del trabajo, he seguido estrictamente la planificación propuesta al principio de esta memoria. La he seguido bien hasta el momento en el que tenía que programar la aplicación. Creo que fui bastante optimista en cuanto a tiempos de desarrollo y pensé que podría realizar todas las funcionalidades previstas. Finalmente tuve que tomar la decisión de no incluir, por ejemplo, el registro de usuarios mediante Facebook que considero imprescindible para la versión final de la aplicación pero prescindible para el buen funcionamiento de este primer prototipo. Gracias a ello, pude ganar unos días de trabajo que decidí destinar al desarrollo de la edición de algunos elementos de la plantilla de la web de boda, cuya funcionalidad no estaba inicialmente prevista.

Adicionalmente, durante el mismo desarrollo de las funcionalidades de la aplicación, me hubiese gustado haber tenido una planificación más detallada de cada una de ellas, en lugar de tener una planificación general para todas. Creo que esto me hubiese ayudado mejor a organizar mi tiempo y quizá podría haber implementado perfectamente el registro de usuario mediante Facebook.

Respecto a los objetivos iniciales planteados en la memoria, considero que han sido completados con el trabajo realizado. El objetivo principal era desarrollar un primer prototipo funcional de la aplicación que sirviera para mostrar, sobre todo, su funcionalidad básica. Pero además quería aprender a programar en un lenguaje orientado a objetos utilizando un framework PHP, como Yii2, para el desarrollo de la aplicación.

Para finalizar, y como he dicho al principio de mis conclusiones, estoy muy satisfecho del trabajo realizado hasta ahora. Es evidente que el trabajo puede mejor considerablemente y lo hará en futuras versiones, porque pienso seguir trabajando en él, pero cuando empiezas un proyecto de estas características siempre tienes la duda de si estarás a la altura, o si realmente has aprendido lo suficiente durante todos estos años de estudio y esfuerzo. Y es al ver el resultado final cuando te das cuenta que ha merecido la pena.

## 7. Líneas de trabajo futuro

A continuación se muestra un listado de los desarrollos y mejoras que se realizarán en futuras versiones de la aplicación:

- Registro y login de usuario mediante Facebook.
- Realizar página home.
- Realizar página de FAQ, contacto y aviso legal.
- Editar todos los contenidos de la web de boda.
- Crear otros temas diferentes para la web de boda.
- Mejorar el diseño responsive de la parte del administrador del usuario.

- Poder crear diferentes secciones para la web de boda.
- Secciones privadas para la web de boda.
- Confirmación de asistencias.
- Editar los datos de plantilla de la invitación que se manda por email.
- Restricción de características según el plan (básico, estándar, premium) escogido por el usuario.

## 8. Bibliografía

- **Yii Framework.** (2017) Modelo-Vista-Controlador (Model-View-Controller MVC) [online] Yii Framework.  
<http://www.yiiframework.com/doc/guide/1.1/es/basics.mvc>
- **Claudio Cubillos.** (2017) Arquitectura Cliente/Servidor [online] PUCV.  
<http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>
- **Jmbeas.** (2017) Historias de usuario [online] JMBEAS.  
<http://jmbeas.es/guias/historias-de-usuario/>
- **Wikipedia.** (2017) jQuery [online] Wikipedia.  
<https://es.wikipedia.org/wiki/JQuery/>
- **Danilo Gonzalez Reyes.** (2017) Telecomunicaciones: Arquitectura cliente/servidor [online] Monografias.  
<http://www.monografias.com/docs114/telecomunicaciones-arquitectura-cliente-servidor/telecomunicaciones-arquitectura-cliente-servidor.shtml>
- **Google Maps JavaScript API.** (2017) [online] Google.  
<https://developers.google.com/maps/documentation/javascript/>

## 9. Recursos

- **Bootstrap.** Documentación de Bootstrap.  
<http://getbootstrap.com>
- **Stackoverflow.** Foro de desarrolladores.  
<http://stackoverflow.com>
- **W3C.** Referencia sobre HTML, CSS, Javascript, etc.



<http://www.w3schools.com>

- **Fontawesome**. Repositorio de iconos.  
<https://fontawesome.io>
- **Yii PHP Framework Version 2**. Documentación de Yii2.  
<http://www.yiiframework.com/doc-2.0/index.html>
- **jQuery API**. Documentación de jQuery.  
<http://api.jquery.com>