



Web para fiestas, eventos y bodas

Memoria de Proyecto Final de Grado

Grau de Multimèdia

Gestión y publicación de contenidos

Autor: Joaquín Gállego Bernués

Consultor: Joan Giner Miguelez

Profesor: César Pablo Córcoles Briongos

Fecha de entrega: 19/06/2019

Créditos/Copyright



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España \(CC BY-NC-ND 3.0 ES\)](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Abstracto

Este TFG se basa en la creación de un entorno de trabajo sobre el CMS (*content management system*) Drupal 8¹ para la elaboración rápida de páginas web a bajo coste.

Para ello se crean una serie de componentes que quedan disponibles para su posterior inserción en futuros proyectos a demanda del cliente. Una vez creados estos componentes, pueden ser recombinados o modificados de múltiples maneras para aumentar la oferta de variaciones a disposición de clientes futuros. La elaboración de e estos componentes se lleva a cabo principalmente utilizando en conjunto Tipos de Contenido, Vistas y Paragraphs, así como con diferentes módulos contribuidos de Drupal disponibles en su web.

Para poder ofrecer distintas opciones a nivel estético se crean también tres subtemas (partiendo del tema Bootstrap), con sus respectivas variaciones en plantillas Twig y múltiples hojas de estilo CSS (*Cascading Style Sheets*).

Esto permite la creación de páginas web con muy bajo coste para un sesgo de mercado concreto, eventos, fiestas y bodas, siendo una herramienta muy útil para un pequeño proyecto empresarial.

Palabras clave: CMS, Drupal, Twig, CSS, fiesta, evento, boda, web, personalización.

¹ Drupal - Open Source CMS. Drupal.org [en línea], 2018. [Consulta: 2 mayo 2019]. Disponible en: <https://www.drupal.org/home>.

Abstract (english version)

This TFG is based on the creation of a working environment on the CMS (content management system) Drupal 8 for the quick development of web pages at low cost.

To do this, a series of components are created that are available for their subsequent insertion in future projects at the customer's request. Once these components are created, they can be recombined or modified in multiple ways to increase the supply of variations available to future customers. The elaboration of these components is carried out mainly using Content Types, Views and Paragraphs, as well as with different contributed Drupal modules available on their website.

In order to offer different options at an aesthetic level, three sub-themes are also created (based on the Bootstrap theme), with their respective variations in Twig templates and multiple CSS stylesheets (Cascading Style Sheets).

This allows the creation of web pages at very low cost for a specific market segment: events, parties and weddings, being a very useful tool for a small business project.

Keywords: CMS, Drupal, Twig, CSS, party, event, wedding, web, customization.

Notaciones y convenciones

Todo el texto con fuente *Courier New* y *cursiva* hace referencia a código. Por ejemplo:

```
{% if language=='es' %}  
    {% set msg = 'Sube' %}  
    {% elseif language=='ca' %}  
        {% set msg = 'Puja' %}  
{% endif %}
```

Índex

1. Introducción	9
2. Definición	11
3. Objetivos	12
3.1 Principales.....	12
3.2 Secundarios	12
4. Contenidos.....	13
4.1 Los Componentes	14
4.2 Los Temas	26
4.3 Los Usuarios	30
5. Metodología	33
6. Plataforma de desarrollo	35
7. Planificación.....	37
8. Prototipos.....	38
9. Proceso de trabajo.....	44
10. Usabilitat/UX	49
11. Proyección de futuro.....	51
12. Presupuesto.....	51
13. Análisis de Mercado	51
14. Márquetin y ventas	54
15. Conclusiones	54
16. Requisitos de instalación.....	55
17. Instrucciones de instalación	56
18. Instrucciones de uso.....	59
Anexos	60
Annex 1. Entregables del proyecto.....	60
Annex 2- Librerías y código externo utilizado	60
Annex 3. Bibliografía.....	61

Figuras y tablas

Índice de figuras

Figura 1: Resultado de una vista.....	13
Figura 2: Esquema de estructura de relaciones entre contenidos en Drupal 8	14
Figura 3: Tipos de cuenta atrás seleccionables desde un paragraph	15
Figura 4: Filtros del bloque Map	16
Figura 5: Paragraphs seleccionables cuando se crea contenido con Content type Foto	17

Figura 6: Detalle de filtro avanzado mediante Relationship en vista "Pictures"	17
Figura 7: Detalle de selección de campos dentro de Paragraphs en vista "Pictures"	17
Figura 8: Detalle de opción necesaria para mostrar individualmente fotos dentro de un campo en vista "Pictures"	18
Figura 9: Paragraphs seleccionables cuando se crea contenido con Content type Home.....	18
Figura 10: Detalle de formulario de creación del Paragraph "Home2", ya rellenado, en Content Type "Home".....	19
Figura 11: Detalle de los campos de taxonomía añadidos al perfil de usuario.....	20
Figura 12: Detalle de código en "node--foto.html.twig" – fecha personaliza.....	21
Figura 13: Detalle de código en "node--foto.html.twig" – impresión por pantalla.....	21
Figura 14: Detalle de código en "node--foto.html.twig" – Comentarios personalizados	23
Figura 15: Detalle de código en "node--foto--teaser.html.twig" – Recuento de comentarios y enlace	24
Figura 16: Detalle de Blocks add_foto y add_video en Block Layout.....	24
Figura 17: Configuración "Pages" del Block add_foto	25
Figura 18: Detalle de código en "node--dd_content_button_.html.twig – URL	25
Figura 19: Detalle de código en "node--dd_content_button_.html.twig - Botón.....	25
Figura 20 Sección de página "Home" con Classy Theme	27
Figura 21 Ejemplo de imágenes de perfil de usuario redondeadas. – Classy Theme.....	27
Figura 22 Sección de página "Home" con Party Theme	28
Figura 23 Ejemplo de imágenes de perfil de usuario recortadas. - Party Theme.....	29
Figura 24 Sección de página "Home" con Simple Theme	29
Figura 25 Ejemplo de imágenes de perfil de usuario cuadradas. – Simple Theme.....	30
Figura 27 Detalle de pestañas de edición cuando el contenido ha sido creado por el usuario "Participant"	31
Figura 26 Detalle de "add_content_button" en los resultados de una de las categorías de foto	31
Figura 28 Detalle de pantalla de administración para usuario "Participant"	32
Figura 29 Detalle de pantalla de administración>People para usuario "Organizer"/"The Couple"	33
Figura 30 Diagrama de Gantt: Principales hitos.....	37
Figura 31 Diagrama de Gantt: actividades desglosadas.....	38
Figura 32 Página principal- Versión 1 - 2 columnas.....	39
Figura 33 Página principal- Versión 2 - Todo en una columna.....	39
Figura 34 Página principal- Versión 3 - 2 columnas- Ubicación no está en la página principal, sino que tiene sección propia en el menú	39
Figura 35 Página principal- Navegación principal con menús desplegables.....	39
Figura 36 Galerías de fotos / vídeos Versión 1- Lista thumbnails- 1 columna.....	40
Figura 37 Galerías de fotos / vídeos Versión 2- Lista thumbnails- 2 columnas.....	40
Figura 38 Galerías de fotos / vídeos Versión 3- Grid/ Solo thumbnails/ Mayor tamaño	40
Figura 39 Fotos- Versión 1 – thumbnails- La galería se visualiza con colorbox o PhotoSwipe	41
Figura 40 Fotos- colorbox – Pop up al hacer clic en un thumbnail	41
Figura 41 : Fotos- PhotoSwipe – Pantalla completa al hacer clic en un thumbnail	41
Figura 42 : Fotos- PhotoSwipe – Pantalla completa al hacer clic en un thumbnail	41
Figura 43 : Usuarios- Versión 1 – Grid simple, foto y nombre.....	42
Figura 44 : Usuarios- Versión 2 – Dos columnas- Foto, nombre y relación con los novios.....	42
Figura 45 : Usuarios- Versión 3 – 3 columnas separades con el estado de asistencia.....	43
Figura 46 : Artículo. Con foto (opcional)- Una sola columna.....	43
Figura 47 : Artículo. Dos columnas. Otros artículos aparecerán en la columna derecha	43
Figura 48 : Artículo. Dos columnas. Fotos aparecerán en la columna derecha	44

Figura 49 : Estructura de carpetas	47
Figura 50 : Detalle de party_theme.libraries.yml - Lista de librerías CSS y fuentes.....	48
Figura 51 : Detalle de typography.css para el tema Classy Theme	48
Figura 52 : Respectivamente los Logos de Simple, Classy y Party	49
Figura 53 : Editor WiX- ejemplo de interfaz con muchas opciones	52
Figura 54 : The knot, página web.....	52
Figura 55 : Zankyou, página web.....	53
Figura 56 : WiX, página web	54
Figura 57 : Instalación Dev Desktop -descompresión de archivos.....	56
Figura 58 : Instalación Dev Desktop- finalización de instalación.....	56
Figura 59 : Importación con Dev Desktop	57
Figura 60 : Importación con Dev Desktop-selección de archivos.....	57
Figura 61 : Importación con Dev Desktop-selección de base de datos.....	58
Figura 62 : Importación con Dev Desktop-selección de base de datos.....	59

1. Introducción

Durante años he ido estudiando poco a poco el Grado de Multimedia, una o dos asignaturas por cuatrimestre, a veces tres, aprendiendo poco a poco HTML (*HyperText Markup Language*) y CSS, JavaScript, Java, PHP (*PHP: Hypertext Preprocessor*), Processing, etc.... a la par que, aprendiendo principios de diseño, audio, vídeo, interactividad, usabilidad, trabajo en red, entre otros temas necesarios para poder llevar a cabo un trabajo completo. Esto me hace consciente de la increíble cantidad de piezas móviles que componen una web, o un producto interactivo y/o audiovisual, por qué están ahí y que aporta cada una de ellas a la obra final.

Por supuesto, una ventaja de la UOC es haber podido llevar a cabo este aprendizaje a mi propio ritmo, mientras me ganaba la vida en un trabajo que, por desgracia, tenía muy poco que ver con lo que estaba estudiando. Por desgracia, precisamente este lento ritmo de aprendizaje, unido al hecho de que mi necesaria fuente de ingresos, que ocupaba la mayor parte de mi tiempo, no me permitía practicar los conocimientos adquiridos, han hecho que no haya podido profundizar y consolidado en forma de una progresión continua, el dominio de todos estos lenguajes y habilidades.

Este largo periodo me ha dado perspectiva y el tiempo para interiorizar y procesar los conceptos generales de cada disciplina, pero no tanto para especializarme y ser brillante en ninguna de ellas en concreto.

En varias ocasiones me he encontrado con la perspectiva de, tras más de un año habiendo matriculado asignaturas que no requerían, por ejemplo, HTML y CSS o JavaScript, al año siguiente tener que recordar y reaprender partes de alguna de estas disciplinas para una nueva asignatura para la que se suponía un dominio previo, lógico, si hubieran sido cursadas el cuatrimestre anterior.

Aproximándome al final de este viaje de 9 años, me sentía algo decepcionado con que, a pesar de mis buenas calificaciones, asignatura por asignatura, no sentía que hubiera hecho, y dudaba fuera capaz de

hacer por mí mismo, un producto acabado y competente de la complejidad imaginaba cuando comencé el grado.

Esta situación es producto de muchas más variables, por su puesto, empezando por mi propia falta de valentía para asumir el riesgo de dejar mi trabajo para convertir aquello que he estado aprendiendo en la UOC en mi trabajo, lo que me hubiera dado el tiempo y el incentivo para perfeccionar y nunca desconectar de aquellas habilidades en las que había sido introducido en las asignaturas del grado. Así mismo, mi propia falta de enfoque en mi proyecto educativo y, paradójicamente, mi obsesión con los productos multimedia que me han llevado a interesarme por este grado, desde cine y televisión a videojuegos y redes sociales, me han hecho dedicar menos de mi tiempo libre del que debería a profundizar y mantener lo aprendido. Aquí es donde la asignatura de Sistemas de Gestión de Contenidos entra en la historia. Aunque tenía un conocimiento muy superficial de WordPress y había usado (con el rol de editor, no de administrador) un CMS (bastante terrible) en mi trabajo en call center, desconocía el potencial real de los CMS.

Los CMS que he visto por ahora, especialmente WordPress y Drupal, tienen el punto justo de accesibilidad suficiente para conseguir resultados de forma relativamente rápida y de forma individual, a la par que una complejidad tal que permite aplicar los principios de programación, diseño, usabilidad, comportamiento de usuarios y todas las demás habilidades y conceptos aprendidos durante el grado, de forma transversal, permitiendo desarrollar a alguien como yo un producto final superior al de otra persona que simplemente haya aprendido a usar la interfaz de uno de estos CMS para probar por ensayo y error a desarrollar una web, sin el bagaje de todo lo aprendido durante el grado.

En cuanto a la elección de una web para eventos, fiestas y bodas, la idea parte del hecho de que, cuando este trabajo haya sido finalmente entregado en junio, me habré casado. He estado trabajando durante el último mes en una versión reducida de la web que tengo en mente para este trabajo, pero para mi propia boda. Llevar a cabo un proyecto así, sin ayuda y en un plazo de tiempo limitado, ha sido posible para mí con relativo éxito, lo que me ha animado a seguir adelante ampliando esta idea.

De este modo, no llevo a cabo simplemente una web, si no una serie de tipos de contenidos, vistas, plantillas y estilos que pueden ser re combinados entre sí de múltiples formas para poder ofrecer varias opciones de web, compatibles con múltiples tipos de eventos y ampliables en el futuro con cada sucesiva iteración personalizada que me fuerce a desarrollar nuevas variaciones reutilizables.

La idea es plantar las bases de un proyecto laboral a nivel local, que me permita dedicarme activamente a poner en práctica todo lo aprendido durante el grado, desarrollando en este Trabajo de Final de Grado un muestrario de lo que puedo llegar a hacer con Drupal 8.

2. Definición

Creación de un boilerplate o herramienta útil para agilizar la creación de páginas webs personalizadas según las necesidades de un sesgo concreto del mercado: fiestas, eventos y bodas.

Esta herramienta se basa en la creación de una serie de componentes que quedarán “pre-hechos” para su posterior reutilización y recombinación en futuros proyectos, de forma modular. De esta manera se puede ofrecer a los clientes tantas potenciales opciones de página como combinaciones posibles entre los componentes previamente desarrollados.

Estos componentes se desarrollan combinando Tipos de Contenido, Vistas y Paragraphs del CMS Drupal.

Los componentes previstos a desarrollar son:

- Reloj: Un reloj con cuenta regresiva hasta el comienzo del evento con diferentes estilos seleccionables
- Insertador de imágenes: que permita seleccionar entre distintos 3 modos de visualización de imagen
- Insertador de vídeos: que permita seleccionar entre subir un archivo o vídeo de URL
- 3 tipos de página Home con distintas disposiciones de contenidos
- 3 tipos de página Blog con maquetación diferente
- 3 tipos de página Users con diferentes funcionalidades y presentaciones

Así mismo se crean tres subtemas para poder variar la parte estética de la página. Estos subtemas tienen estilos muy diferenciados para desde ellos hacer variaciones que aumenten la cantidad de oferta de temas visuales en el futuro. Todos cuentan con logos y favicon personalizados, lo que sería un servicio que se ofrecería a los clientes.

Los temas previstos a desarrolla son:

Classy Theme: Tema con estilo clásico y elegante, pensado para bodas o eventos que requieran de una imagen más cuidada.

Party Theme: Tema con estilo divertido y colorido. Pensado para fiestas, cumpleaños, etc.

Simple Theme: Tema con un estilo sobrio y funcional. Pensado para reuniones de trabajo o clientes que no estén interesados en la parte estética de la página.

3. Objetivos

3.1 Principales

Los objetivos principales del proyecto son los siguientes:

Para el Back End, la generación de los bloques y funcionalidades reutilizables entre sitios web

- Desarrollo de bloques y tipos de Paragraph como el Reloj con cuenta regresiva, los insertadores de imagen y vídeo.
- La creación de los distintos roles de usuario y la asignación de permisos para que vean y puedan editar su contenido. Asignación de permisos para los usuarios organizadores de modo que puedan gestionar a otros usuarios, incluso dando acceso o no a ciertas partes de la web.
- La creación de las secciones principales de la web con sus respectivos accesos en el menú de navegación principal. Estos bloques son: Blog, Fotos, Vídeo, Música, Usuarios.

Para el Front-end, la generación de 3 subtemas a partir del tema Bootstrap. Las funcionalidades creadas en Back-end se mantienen para todos ellos, pero se puede pasar de uno a otro cambiando la estética de la web. Se definirá y ejecutará una estrategia de organización CSS que permita la fácil modificación de los temas y la creación de posteriores subtemas a partir de ellos.

Para el Back-office, se definirá y configurará una interfaz para que los usuarios puedan añadir contenido y para que los administradores puedan gestionar la página

3.2 Secundarios

- Desarrollo de más subtemas si fuera posible
- Implementación de módulos adicionales no originalmente previstos que mejoren la experiencia de la web
- Añadir nuevas funcionalidades para tipos de evento no previstos
- Integración de google Analytics
- Análisis heurísticos y pruebas de usuarios (opcional)

4. Contenidos

Para comprender adecuadamente el contenido de la web, primero es necesario entender cómo se relacionan todos los elementos que lo conforman en Drupal 8.

Nodo/Node - cada una de las páginas dentro de la web.

Por ejemplo, en mi web `/foto/category-1/foto-test-5` La página foto-test-5 es el nodo

Campos/Fields- Cada una de las partes de un nodo.

Por ejemplo, dentro del nodo anterior, el campo de comentarios, el campo de fotos, el campo de etiquetas, etc.

Tipo de Contenido/Content Type- Cada nodo pertenece a un tipo de contenido, éste define qué campos y de qué tipo tiene o puede tener dicho nodo.

Por ejemplo, en el nodo foto-test-5 es del Tipo de Contenido Foto, por lo que, al crear un nuevo nodo, en el formulario de creación habrá un campo para adjuntar fotos, pero no vídeo o contenido de otros tipos.

Vistas/Views- Son filtros, que permiten mostrar en pantalla contenido ya existente en nodos de la página, configurando diferentes criterios de filtrado y de presentación. Pueden ser de tipo Page, una página entera, que se vería como un Nodo o Block, un bloque insertable dentro de otro contenido. Las Views por si mismas no sirven de mucho si no hay contenido que puedan encontrar. Es decir, una vista que filtre contenido del Content type Foto, no devolverá ningún resultado visible en pantalla si no se ha generado ningún nodo del Content type Foto primero.

Por ejemplo, en mi página, la vista "Users_pic_and_name_3cols" filtra contenido de tipo User (usuario), y muestra por cada resultado encontrado los campos Name y Picture. La vista está configurada para que muestre la información en 3 columnas.



Figura 1: Resultado de una vista

Párrafos/Paragraphs- Los Paragraphs se pueden añadir a cualquier tipo de contenido como si fueran un campo más. Sin embargo, los Paragraphs pueden contener múltiples campos, vistas o incluso otros Paragraphs. Es una manera de poder tener estructuras preconfiguradas para insertar fácilmente en un tipo de contenido sin necesidad de añadir los campos manualmente a éste. Es también una forma de dar opciones a los editores o usuarios, ya que no todos los campos dentro del paragraph tiene que ser obligatorios, por lo que aparece como un menú desplegable con distintas opciones a elegir. El sistema de Paragraphs permite anidar múltiples niveles de Paragraphs en un solo tipo de contenido, por lo que es muy versátil y añade una gran profundidad a la organización del contenido dentro de los nodos. No obstante, esta profundidad añade complejidad al sistema, al deber tener siempre presente de que manera se relacionan los elementos entre sí y en qué niveles se encuentran.

CONTENT TYPE

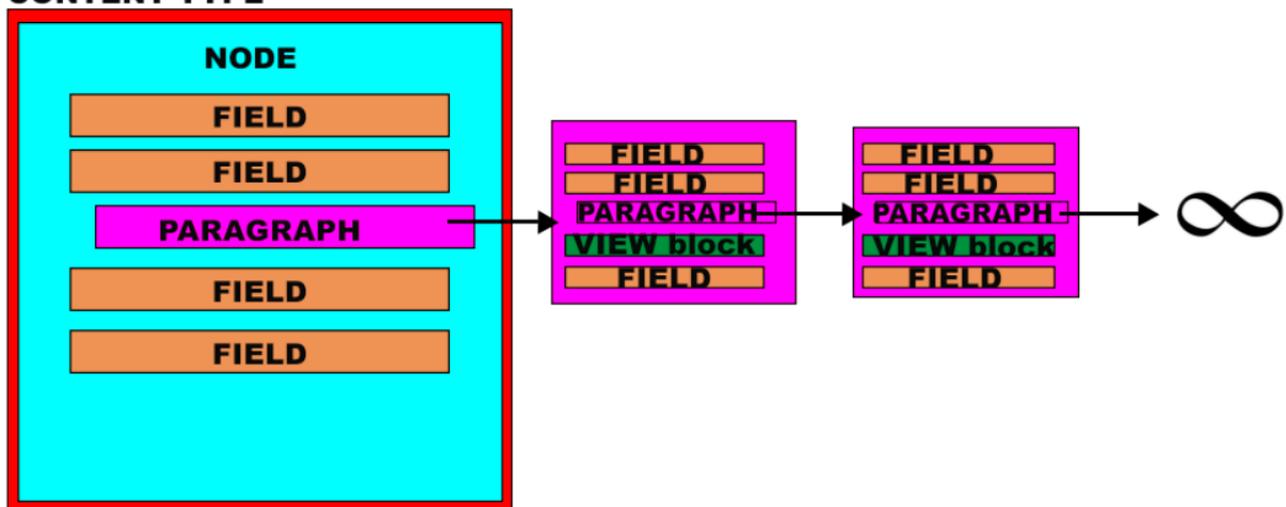


Figura 2: Esquema de estructura de relaciones entre contenidos en Drupal 8

4.1 Los Componentes

Para clarificar el proceso primero explico, indicando todas las rutas, como se crea el primer Paragraph. El resto de los contenidos son explicados con un nivel menor de detalle, asumiendo que no es necesario indicar las mismas rutas de forma reiterada:

Clock

1. Creo un nuevo tipo de Paragraph al que llamo "Clock"
 - Ruta: Home>Admin>Structure>Paragraphs type>Add
2. Añado 4 campos para los diferentes tipos de reloj, que serán del tipo Date
 - Ruta: Home>Admin>Structure>Paragraphs type>Clock>Fields>Add Field

3. En Manage Display selecciono los 4 formatos de reloj posibles que el módulo “Timer Field” permite.

Ruta: Home>Admin>Structure>Paragraphs type>Clock>Display

4. En el Content Type “Home” añadimos un field de tipo Paragraph

Ruta: Home>Admin>Structure>Types>Home>Fields>Add field

5. Ahora se puede elegir fácilmente entre los 4 tipos de reloj al crear un nuevo contenido de tipo HOME, insertando un paragraph de tipo Clock

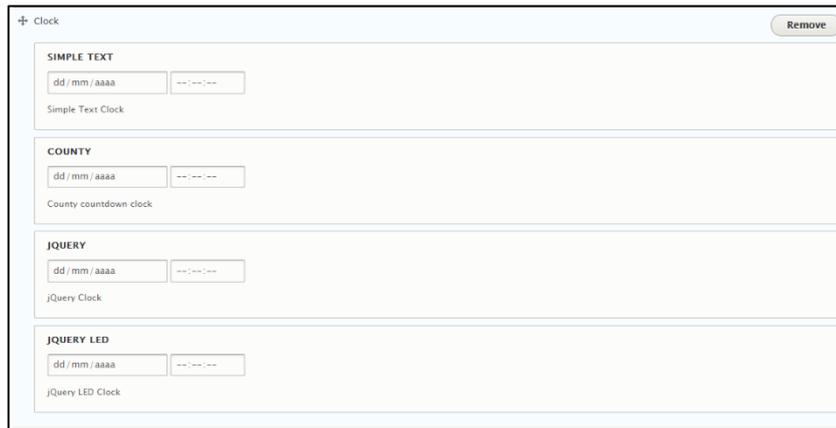


Figura 3: Tipos de cuenta atrás seleccionables desde un paragraph

Alternativamente, genero una View de tipo Block de Clock, ya que será útil para poder ser insertada en Paragraphs compuestos de múltiples columnas.

- Importancia para el negocio: Es importante tener diferentes opciones de reloj de cuenta atrás ya que es un elemento indispensable para el tipo de evento al que va dirigida la herramienta. Un evento, boda o fiesta, por definición, ocurre en un momento determinado y en un lugar determinado. Tener accesibles los diferentes estilos de cuenta atrás agiliza el trabajo y permite mostrar al cliente de forma rápida lo que tiene disponible.

Carousel

Utilizo la función ya existente en Bootstrap Paragraphs llamada Carousel. Machine name: bp_carousel

Posteriormente se le aplican estilos CSS para conseguir distintos resultados.

- Importancia para el negocio: Es importante ya que hace más llamativa la página principal, añade dinamismo y es un elemento que normalmente podría costar un tiempo configurar e implementar correctamente, pero, mediante un paragraph preconfigurado es cuestión de segundos.

Map

- 1.- Utilizo el módulo Simple Google Maps, que permite crear in iframe incrustado a partir de una dirección almacenada en un campo de texto.
- 2.- Creo un contenido que tenga la dirección deseada en un campo de texto que formateo desde Manage Display para que sea reconocido por Simple Google Maps
- 3.- Creo un View de tipo Block, que filtre de modo que muestre únicamente el campo que contiene el mapa en el nodo anteriormente creado

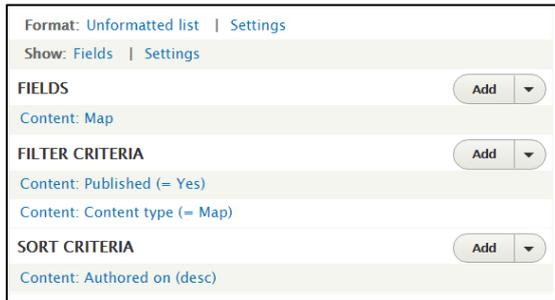


Figura 4: Filtros del bloque Map

- 4.- De este modo el bloque puede ser insertado dentro de cualquier Paragraph, ya que éstos permiten utilizar cualquier vista/bloque disponible.
- Importancia para el negocio: Al igual que en caso del reloj, el evento ocurrirá en un lugar determinado y tener fácil acceso a crear y modificar el mapa es fundamental

Funcionalidades de tipo de foto- Genero tres Paragraphs con distintos estilos de imagen del mismo modo que se ha llevado a cabo el paragraph Clock

PhotoSwipe pictures- Campo imagen con el formato PhotoSwipe, seleccionado en Manage Display. Mediante el módulo PhotoSwipe se crea una galería de fotos en la que se puede deslizar de una a la otra, hacer zoom, compartir o descargar la foto.

ColorBox pictures- Campo imagen con el formato CorlorBox, seleccionado en Manage Display. Mediante el módulo ColorBox se crea una galería con botones de navegación, algo más limitada que PhotoSwipe.

SlideShow pictures- Campo imagen con el formato CorlorBox, seleccionado en Manage Display. Mediante el módulo Imagefield Slideshow se puede crear una presentación de diapositivas con pager y botones de navegación si se desea.

Plain pictures- Campo imagen, configurado para aceptar una cantidad ilimitada. Aunque sería sencillo tener esto mismo sin necesidad de un Paragraph, al tenerlo en esta forma puedo elegirlo justo con las demás opciones en un solo clic.

De este modo cuando crea un nodo de Content Type foto se puede elegir una de las cuatro opciones para la presentación de estas. Por supuesto, el usuario final elegiría que tipo quiere y se

configuraría el Content Type para incluir solo una de las tres, de modo que todos los usuarios no tuviesen que elegir uno de los tres Paragraphs cada vez que quiera crear un nuevo nodo

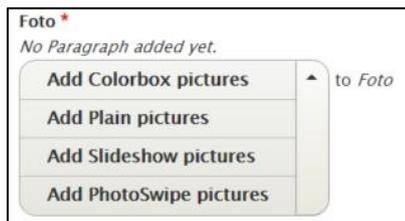


Figura 5: Paragraphs seleccionables cuando se crea contenido con Content type Foto

- Importancia para el negocio: La compartición de fotos es la sección de la página que normalmente debería generar más actividad. Tener disponibles todas las opciones de manera rápida para mostrar al cliente como se vería el resultado final puede ser muy útil. Una vez el cliente elija cual quiere se puede configurar para que se muestre exclusivamente ese tipo.

Funcionalidades de tipo de Video- Siguiendo el mismo proceso que en el punto anterior, genero dos Paragraphs, uno con un campo Vídeo y otro con un campo Video Embed, para poder contar con las dos opciones en forma de menú desplegable en el Content Type "Video"

- Importancia para el negocio: Los motivos son los mismos que en el caso de Foto.

Funcionalidad Pictures- View de tipo Block configurada para mostrar las últimas imágenes añadidas a la página.No obstante, como se pueden incluir múltiples fotos en un solo campo en todas los Paragraphs indicados en el punto 9.1.3 es necesario configurar la vista de un modo especial.

1.-Después de crear el Block se establece una Relationship con el campo foto de Paragraph



Figura 6: Detalle de filtro avanzado mediante Relationship en vista "Pictures"

2.-Tras este paso, es posible seleccionar estos campos específicos en Fields

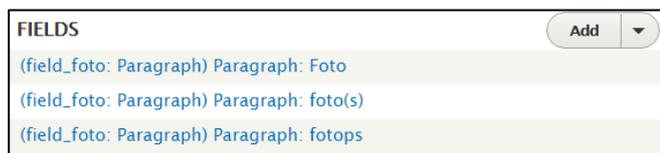


Figura 7: Detalle de selección de campos dentro de Paragraphs en vista "Pictures"

3.-Por último, además de seleccionar el formateador de imagen deseado (PhotoSwipe), es necesario desmarcar “Multiple Field Settings”> “Display all values in the same row” para que la vista devuelva cada foto dentro del campo en lugar del campo en si mismo



Figura 8: Detalle de opción necesaria para mostrar individualmente fotos dentro de un campo en vista “Pictures”

- Importancia para el negocio: Ofrecer contenidos dinámicos resulta muy vistoso y puede ser atractivo para los clientes.

Funcionalidad Other blog entries- View de tipo Block configurada para mostrar las últimas entradas del blog. Se crea igual que el anterior, pero como está pensado para insertarse junto a una entrada de blog, en el mismo nodo que ésta, se le configura un filtro avanzado que no muestre en sus resultados la entrada actualmente en pantalla.

- Importancia para el negocio: Ofrecer contenidos dinámicos resulta muy vistoso y puede ser atractivo para los clientes.

Tipos de Home

Al crear un nuevo nodo de tipo Home, lo hago eligiendo entre 3 paragraphs, lo que agiliza enormemente la tarea.



Figura 9: Paragraphs seleccionables cuando se crea contenido con Content type Home

Home- Incluye: Paragraph Carousel, Paragraph Clock (creado previamente) y un Bootstrap Paragraph de dos columnas desiguales (“Columns (Two Uneven)”). Este Paragraph permite incluir un tanto texto como una View/Block. En este caso se usará para incluir el Block “Map”

Home2- Incluye un Bootstrap Paragraph para generar contenido en dos columnas. Mediante esto puedo incluir un Column Wrapper para cada columna y en cada uno de estos

añadir el contenido deseado. Columna izquierda: Carousel, Plain (texto), Clock. Columna derecha: Map, Plain.

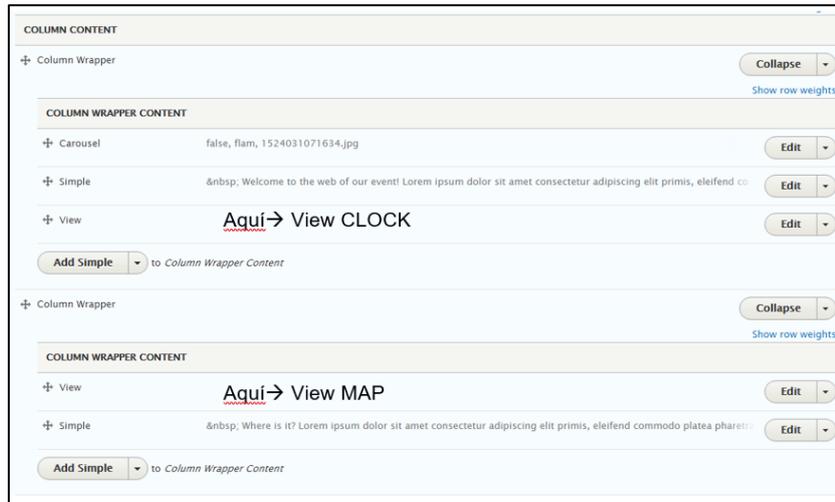
The image shows a screenshot of a Drupal content editor interface. The main heading is 'COLUMN CONTENT'. Below it, there are two 'Column Wrapper' sections. The first wrapper contains three items: a 'Carousel' block with a file path 'false, flam, 1524031071634.jpg', a 'Simple' text block containing ' Welcome to the web of our event! Lorem ipsum dolor sit amet consectetur adipiscing elit primis, elefend co', and a 'View' block with the text 'Aquí → View CLOCK'. Below these is an 'Add Simple' button and a link 'to Column Wrapper Content'. The second wrapper contains two items: a 'View' block with the text 'Aquí → View MAP' and a 'Simple' text block containing ' Where is it? Lorem ipsum dolor sit amet consectetur adipiscing elit primis, elefend commodo platea phare...'. Below these is another 'Add Simple' button and a link 'to Column Wrapper Content'. The interface includes 'Collapse' and 'Show row weights' buttons for each wrapper, and 'Edit' buttons for each content block.

Figura 10: Detalle de formulario de creación del Paragraph "Home2", ya rellenado, en Content Type "Home".

Home3- Incluye cuatro campos: Un Paragraph de tres columnas que se usa para incluir tres imágenes en fila, Paragraph Clock, Paragraph Carousel y un campo de texto.

- Importancia para el negocio: La página principal es la primera impresión que los usuarios tienen de la web. Tener diferentes opciones fácilmente configurables y editables para esta página es esencial. Diferentes eventos se pueden beneficiar de distintas configuraciones de página, aumentando los clientes potenciales.

Vistas para la lista de usuarios.

Usando una metodología muy similar a la utilizada para la generación de los distintos tipos de Paragraph Home, genero tres tipos de Paragraph para el Content Type "Users". Se generan Views de tipo Block para el filtrado de los campos que posteriormente se insertan en Bootstrap Paragraphs del tipo Columns (Equal) para distribuirlos en columnas.

View Users_pic_and_name_3cols- View que simplemente filtra por usuario y muestra la fotografía y el nombre

Views The Groom, The Bride, Families_left_col y Families_right_col

1.-Desde Home>Admin>Config>People>Accounts añado nuevos campos a los perfiles de usuario para poder establecer con ellos los filtros necesarios en en las vistas

Attendance status	field_attendance_status	Entity reference	Edit
Bride_Or_Groom	field_bride_or_groom	Entity reference	Edit
Picture	user_picture	Image	Edit
Relation	field_relation	Entity reference	Edit
Side	field_side	Entity reference	Edit

Figura 11: Detalle de los campos de taxonomía añadidos al perfil de usuario.

2.-Utilizo el campo “Bride_or_Groom” para filtra, por un lado al novio en el Block “The Groom” y por otro a la novia en el Block “The Bride”; Utilizo el campo Side para filtrar que invitados pertenecen a que familia y filtrarlos respectivamente en los Blocks “Families_left_col” y “Families_right_col” respectivamente.

3.- Inserto estas vistas en un Paragraph de dos columnas con un columna wrapper en cada una, de modo que El novio quede en la parte superior derecha y sus invitados debajo de él y lo mismo para la novia en la izquierda,

Views status_center_col, status_left_col, status_right_col – Con la misma metodología que en punto anterior, pero usando el campo taxonomía como “Attendance Status”, filtro a los usuarios en tres Blocks, en función de si han confirmado su asistencia, no han confirmado o han declinado la invitación. Después incrusto los Blocks en un párrafo de 3 columnas.

- Importancia para el negocio: De nuevo, al tener distintos tipos de eventos es importante poder mostrar diferentes categorías de invitados. Mediante el uso de taxonomías resulta relativamente sencillo poder clasificar a los usuarios en diferentes grupos y mostrarlos en pantalla según los requerimientos del usuario.

Override de templates Twig

Además de las configuraciones que se pueden llevar a cabo a través de la, como las indicadas más arriba, también es posible modificar el comportamiento de Drupal mediante código personalizado que sobrescribe al del tema principal o al del núcleo de Drupal. Aunque los templates son parte del subtema, y por tanto se pueden considerar también parte del Front-end (punto 9.2, más adelante), se han usado para conseguir funcionalidades que alteran el comportamiento de la web, no únicamente su estética. Los templates son idénticos entre temas, ya que he querido mantener una coherencia en cuanto al funcionamiento de la web y dejar los temas para modificar de forma rápida y sencilla la estética de esta.

1) node--video.html.twig // node--entrada_blog.html.twig // node--foto--teaser.html.twig

Estos tres templates sobrescriben el funcionamiento de los Content types de los mismos nombres, “Video”, “Foto” y “Entrada Blog”. Son muy similares entre sí, con las adaptaciones pertinentes para adaptarse a los diferentes campos con diferentes nombres que se deben de llamar en los respectivos templates mediante `{{}}`

Tomando como ejemplo “node--foto.html.twig”, éstas son sus funcionalidades:

Fecha personalizada

Toma la fecha de creación, le da el formato deseado y la asigna a la variable `createdDate`

Imprime en pantalla únicamente nombre y fecha con el formato correcto

```
{% set createdDate = node.getCreatedTime|date('d/m/y - H:i') %}
```

```
91 <div{{ author_attributes.addClass('author') }}>
92     {% set createdDate = node.getCreatedTime|date('d/m/y - H:i') %}
93     {{ author_name }} - {{ createdDate }}
94     {{ metadata }}
95 </div>
```

Figura 12: Detalle de código en “node--foto.html.twig” – fecha personaliza

Impresión por pantalla de los campos deseados con saltos de línea

Imprime de forma controlada los campos del tipo de contenido en el orden deseado e incluyendo saltos de línea

```
{{content.field_foto}}
{{content.field_etiquetas}}
</br></br>
{{ content.addtoany}}
</br></br>
{{ content.likeit}}
</br>
```

```
99 <div{{ content_attributes.addClass('content') }}>
100     {{ content.field_foto}}
101     {{content. field_etiquetas}}
102     </br></br>
103     {{ content. addToany}}
104     </br></br>
105     {{ content. likeit}}
106     </br>
```

Figura 13: Detalle de código en “node--foto.html.twig” – impresión por pantalla

Campo de comentarios colapsable, con recuento de comentarios y comprueba si el usuario está logado para poder comentar.

Genera un enlace llamado "Comments" con un botón. El botón tiene como target el div con la id "comment" que contiene el campo de comentario. De este modo este div está colapsado y solo al pulsar el botón queda expandido.

La instrucción `{% if not logged_in %}` comprueba si el usuario está logado (y por tanto es un usuario con cuenta). Si no está logado, se muestra un botón que permite acceder a la página de login.

Por último, se imprime el campo de comentario.

Podría haberse evitado que los usuarios no logados vieran los comentarios, simplemente relacionando `{{ content.field_comment}}` a la estructura condicional de modo que no se imprimiese a no ser que `{% if not logged_in %}` fuese falso. No obstante, no veo daño alguno en que los no usuarios puedan leer los comentarios, ya que igualmente no tienen permiso para escribir ni responder a no ser que entren con su cuenta.

```
<a name="comments"></a>
<button type="button" class="btn btn-danger" data-toggle="collapse" data-
target="#comment">Comments{{ ' ('~ content.field_comment['#items'].comment_count~
') ' }}</button>
<div id="comment" class="collapse">
{% if not logged_in %}
<br>
<p id="nologado_msg">You must be logged to comment</p>
<a class="btn btn-default" id="login_link"
href="{{ language.getId() }}/user/login">Log In</a><br>
{% endif %}
</br>
{{ content.field_comment}}
</div>
</div>
```

```

1108 <a name="comments"></a>
1109 <button type="button" class="btn btn-danger" data-toggle="collapse" data-target="#comment">
Comments{{' ('~ content.field_comment['#items'].comment_count~ ') ' }}</button>
1110 <div id="comment" class="collapse">
1111     {% if not logged_in %}
1112         <br>
1113         <p id="nologado_msg">You must be logged to comment</p>
1114         <a class="btn btn-default" id="login_link" href="{{ language.getId() }}/user/login">
Log In</a><br>
1115     {% endif %}
1116     <br>
1117     {{ content.field_comment }}
1118 </div>

```

Figura 14: Detalle de código en "node--foto.html.twig" – Comentarios personalizados

- Importancia para el negocio: El botón colapsable ayuda a que la apariencia de la web sea más limpia ya que si hay muchos comentarios estos pueden impactar negativamente en la estética de la web. Además, la funcionalidad de reconocimiento de si el usuario está o no logado ayuda a los usuarios menos familiarizados con este tipo de webs a detectar el error si han olvidado logarse.

2) node--entrada_blog--teaser.html.twig // node--video--teaser.html.twig // node--entrada_blog--teaser.html.twig

Estos templates afectan a los teasers de los Content Types indicados anteriormente y también son virtualmente idénticos, con las adaptaciones necesarios debidos a los distintos nombres de campo entre Content Types.

Siguiendo la misma metodología del punto anterior también imprimen la fecha personalizada y los diferentes campos del content uno a uno.

Recuento de comentarios y enlace directo al nodo con ancla al a comentario.

La instrucción `{{' ('~ content.field_comment['#items'].comment_count~ ') ' }}` cuenta e imprime la cantidad de comentarios dentro del botón.

La instrucción `` es el enlace al nodo, que tiene el ancla al objeto con id `"tease_ccom"`

```

<div{{ content_attributes.addClass('content') }}>
{{ content.field_etiquetas }}
{{ content.addtoany }}
<a href="{{ url }}#comments" id="teaser_com">Comments{{' ('~
content.field_comment['#items'].comment_count~ ') ' }}</a>

```

```
<br><br>
</div>
```

```
99 <div{{ content_attributes.addClass('content') }}>
100   {{ content.field_etiquetas}}
101   {{ content.addtoany}}
102   <a href="{{ url }}#comments" id="teaser_com">Comments{{' ('~
    content.field_comment['#items'].comment_count~ ') ' }}</a>
103   <br><br>
104 </div>
```

Figura 15: Detalle de código en “node--foto--teaser.html.twig” – Recuento de comentarios y enlace

- Importancia para el negocio: El recuento de comentarios permite que los usuarios puedan ver desde la pantalla en la que se muestran los teasers aquellas entradas de blog, foto o video que tiene más y, por lo tanto, es más interesante de visitar. En enlace directo a la sección de comentarios en la parte baja de la pantalla es un extra de usabilidad que siempre es beneficiosos para que los usuarios estén satisfechos.

3) dd-node--dd_content_button_.html.twig

Este template sobrescribe el funcionamiento del content type Add content button, creado ex profeso para poder generar un botón que selectivamente permita acceder a la sección de creación de contenido, fotos o vídeo, según corresponda. Para entender su funcionamiento, primero hay que explicar el Content Type.

Add_content_buton tiene dos campos:, “Type”, una taxnomía que permite elegir entre “Video” y “Foto” y “url”, un campo de texto que permite introducir la url del lugar de la web al que vamos a dirigir al usuario al pulsar el botón.

1.- Una vez creados los nodos con las urls y tipos necesarios, creo una View- Block por cada uno y los sitúo en la región “Content” del Block Layout de todos los temas.



Figura 16: Detalle de Blocks add_foto y add_video en Block Layout.

2.- Los configuro para que aparezcan en las páginas deseadas. En este caso el Block aparecerá solo en página que contengan en su URL la palabra “foto”

Content type	Pages
Webform	/foto /foto/*
Content types Not restricted	Specify pages by using their paths. Enter one path per line. The "*" character is a wildcard. An example path is <code>/user/*</code> for every user page. <code><front></code> is the front page.
Pages Restricted to certain pages	<input checked="" type="radio"/> Show for the listed pages <input type="radio"/> Hide for the listed pages

Figura 17: Configuración "Pages" del Block add_foto

El template `node--dd_content_button_.html.twig` hace dos cosas:

- 1.- Escanea la URL. Dependiendo de la cadena de texto que detecte, modifica la variable que genera el texto dentro del botón para que éste tenga sentido contextual

```
{% set url = url('<current>') %}
{% if '/video' in url|render|render %}
{% set msg = 'add new video' %}
{% elseif '/foto' in url|render|render %}
{% set msg = 'add new foto' %}
{% endif %}
```

```
80 {% set url = url('<current>') %}
81
82 {% if '/video' in url|render|render %}
83     {% set msg = 'add new video' %}
84 {% elseif '/foto' in url|render|render %}
85     {% set msg = 'add new foto' %}
86 {% endif %}
```

Figura 18: Detalle de código en "node--dd_content_button_.html.twig – URL

1. Genera un botón que lleva a la URL almacenada en el campo llamado "URL". En este caso a `/node/add/foto` para poder acceder directamente a generar contenido del tipo foto

```
<div class="add_button">
  <a href={{content.field_url.0}} class="btn btn-danger btn-
  block">{{msg}}</a>
</div>
```

```
89 <div class="add_button">
90   <a href={{content.field_url.0}} class="btn btn-danger btn-block">{{msg}}</a>
91 </div>
```

Figura 19: Detalle de código en "node--dd_content_button_.html.twig - Botón

4.- Aunque el campo URL no era estrictamente necesario, ya que la URL se podría haber compuesto utilizando el código del punto 1 para detectar si se tiene que enlazar a “/node/add/foto” o “/node/add/video”, he preferido hacerlo así porque el mismo Content Type se puede llegar a usar en otras circunstancias en las que no se pueda extraer la información de a donde dirigir al usuario de la URL.

- Importancia para el negocio: Soy consciente de que el acceso a la creación de contenidos se puede llevar a cabo de formas mucho más simples, pero el desarrollo de este método queda justificado si pensamos en la web de una boda, por ejemplo, donde puede haber gente mayor que quiere participar, pero les puede resultar difícil utilizar un sistema de menú de administración más complejo. Un solo botón grande para hacer lo que interesa más al usuario me parece la opción que genera más fricciones y facilita el uso de la web a todo los usuarios independientemente de su nivel en el uso de interfaces.

4.2 Los Temas

Classy Theme

Tema diseñado para eventos del estilo de bodas o similares, con énfasis en una estética que denote elegancia, sin grandes estridencias ni en colores ni en formas, optando por imágenes redondeadas y una tipografía también redondeada que recuerda a la escritura a mano alzada del siglo XIX. Se sacrifica un poco de legibilidad en los títulos a cambio de una letra con más estilo.

Se ha elegido el turquesa como color principal ya que transmite sensaciones de calma y representa la comunicación abierta y claridad de pensamiento²

Se ha optado por sobrescribir el CSS de los contenedores principales para conseguir un estilo visual “Edge to Edge”, utilizando 100% para que se llene toda la pantalla.

```
.container {width: 100%;}  
.navbar.container {max-width: 100%;}
```

Se han dejado las líneas grises que el tema de Bootstrap ya incorpora, pero eliminando los colores de fondo de la barra de navegación, breadcrumbs y footer, para aumentar la sensación de limpieza y claridad.

² Teal Meaning, Combinations and Hex Code - Canva Colors. Canva's Design Wiki [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://www.canva.com/colors/color-meanings/teal/>.

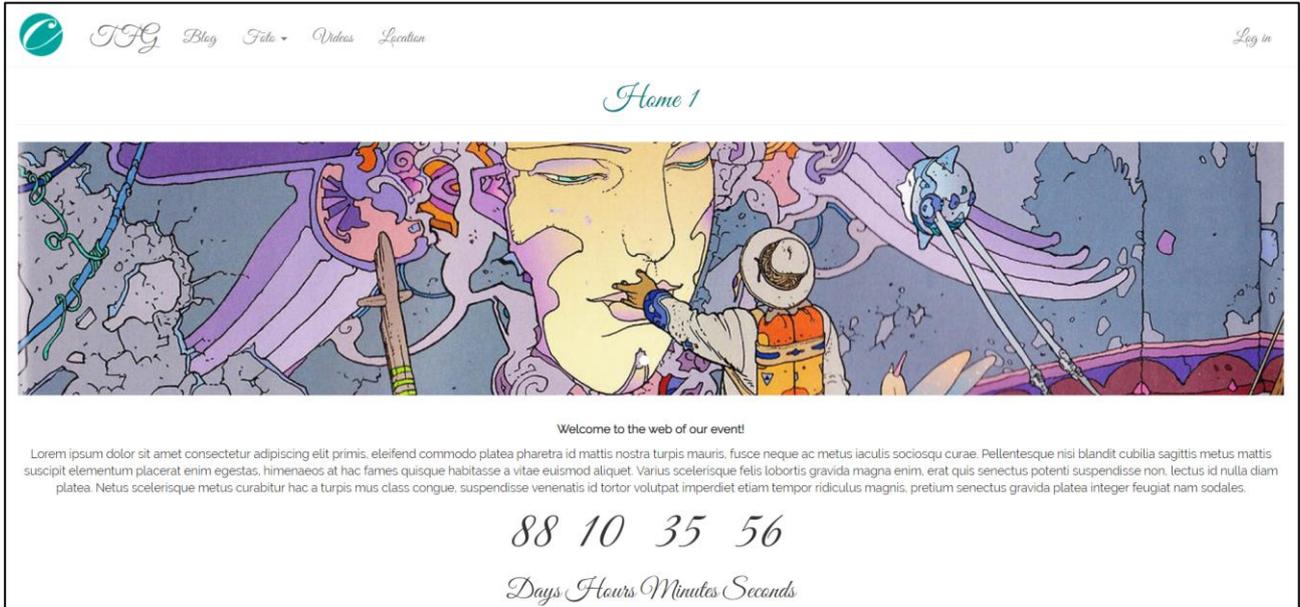


Figura 20 Sección de página "Home" con Classy Theme

Se ha optado por que todas las fotos de usuario sean redondeadas mediante la propiedad "*border-radius: 50%;*"

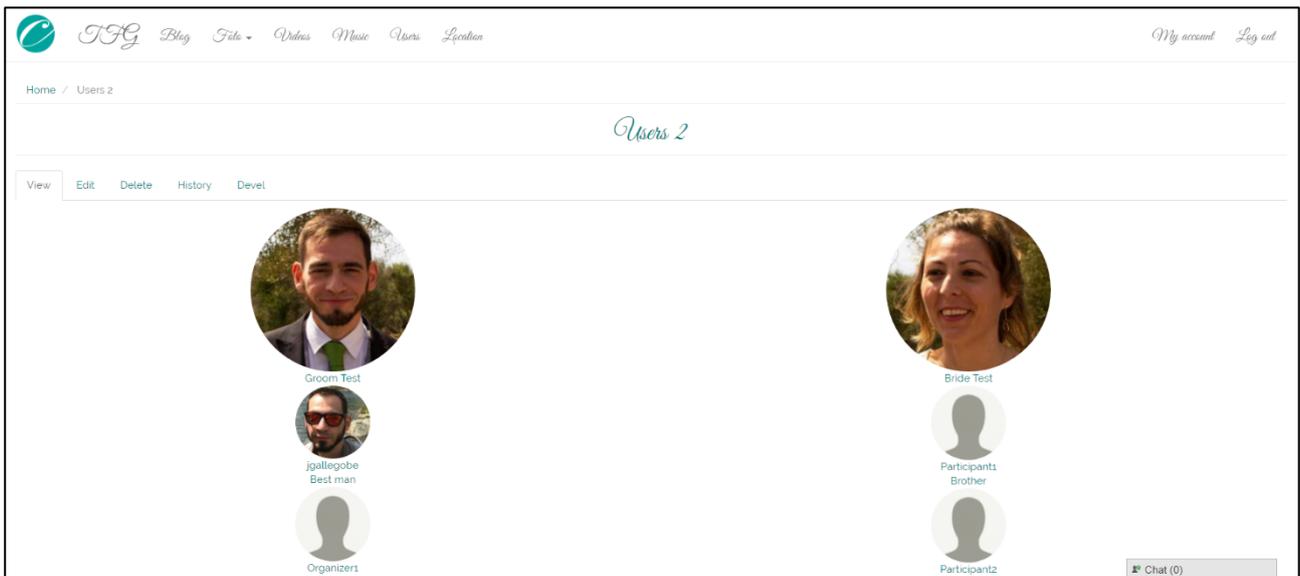


Figura 21 Ejemplo de imágenes de perfil de usuario redondeadas. – Classy Theme

Party Theme

Al contrario que eçn en Classy Theme, prima la potencia visual sobre otros factores.

Se recurre a colores chillones y contrastados, color degradado para el fondo y una tipografía decorativa muy expresiva que, si bien no es la más legible, imprime un sentido de urgencia y diversión a la página.

Se utiliza el color borgoña, que simboliza poder³, para la tipografía más grande y llamativa, así como el degradado para el fondo que contrasta con el blanco puro del tema anterior.

Para el degradado se ha usado el siguiente código⁴:

```
background: #A1FFCE; /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #FAFFD1, #A1FFCE); /* Chrome 10-25, Safari 5.1-6 */
background: linear-gradient(to right, #FAFFD1, #A1FFCE); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
```

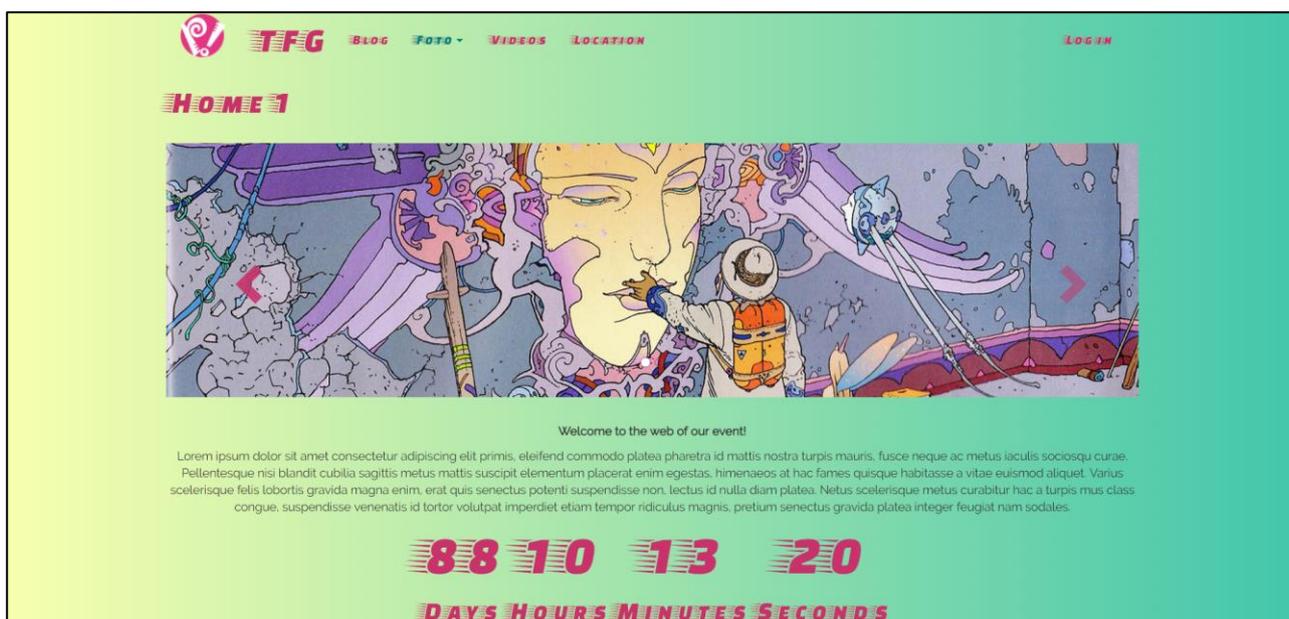


Figura 22 Sección de página "Home" con Party Theme

Las imágenes en lugar de redondeadas se han recortado siguiendo un patrón trapezoidal asimétrico lo que les da un aire divertido y moderno que se ajusta al concepto party. Para ello se ha usado el siguiente código⁵

```
clip-path: polygon(11% 0, 100% 32%, 82% 100%, 1% 81%);
```

3 Burgundy Meaning, Combinations and Hex Code - Canva Colors. Canva's Design Wiki [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://www.canva.com/colors/color-meanings/burgundy/>.

4 uiGradients - Beautiful colored gradients. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://uigradients.com/#Limeade>.

5 Clippy — CSS clip-path maker. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://bennettfeely.com/clippy/>.

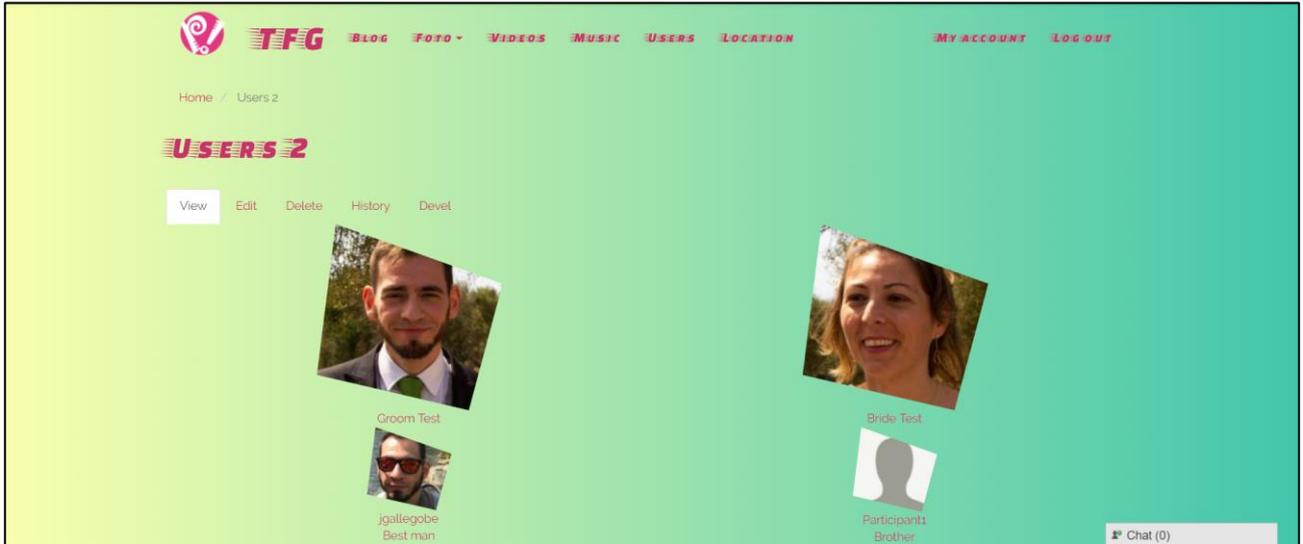


Figura 23 Ejemplo de imágenes de perfil de usuario recordadas.- Party Theme

Simple Theme

El tema simple es el que menos variaciones recibe con respecto a el tema básico de Bootstrap. La idea es ofrecer un tema sin florituras para posibles clientes que solo estén interesados en la funcionalidad sin prestar atención a la parte estética.

La tipografía de palo seco ha sido elegida en aras de la legibilidad por encima de todo.

He evitado los colores saturados y he dado protagonismo a las líneas por encima del color. He reducido el tamaño de del botón de añadir nuevas fotos o vídeo.

He dejado las imágenes cuadradas por defecto y, en general, optado por la sobriedad. Como su propio nombre indica, el tema destaca por su simpleza y no requiere más explicación.

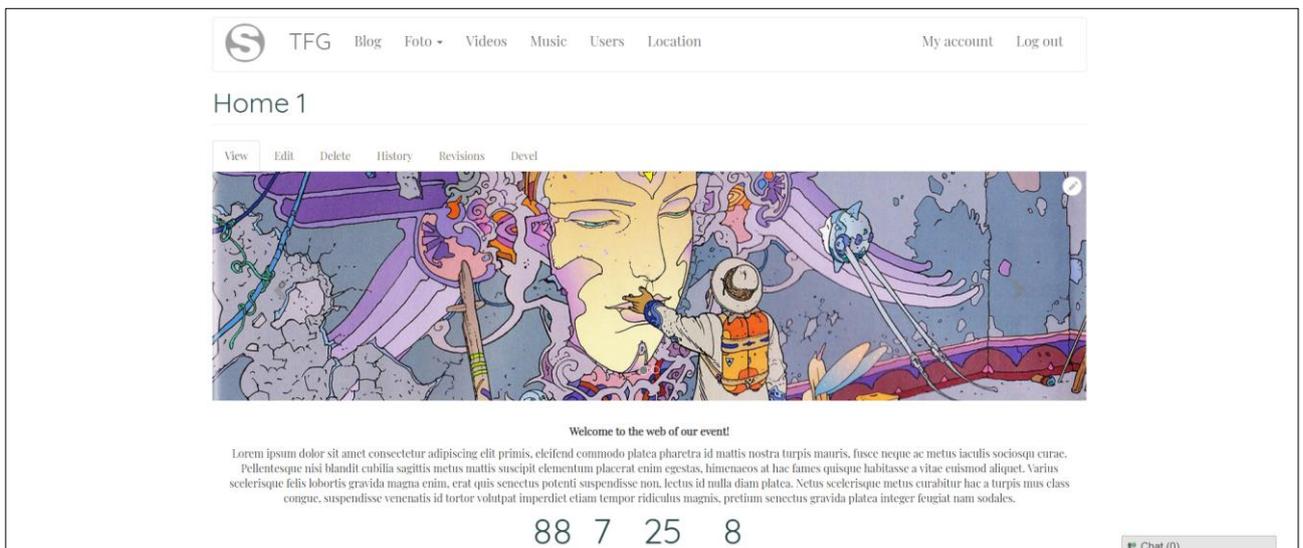


Figura 24 Sección de página "Home" con Simple Theme

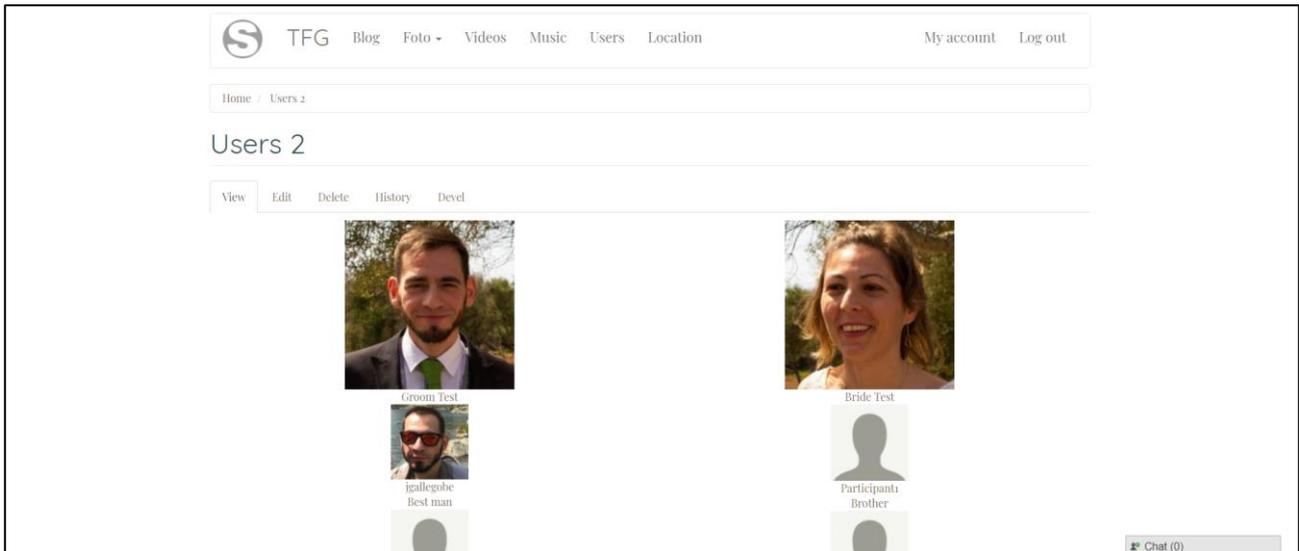


Figura 25 Ejemplo de imágenes de perfil de usuario cuadradas. – Simple Theme

4.3 Los Usuarios

El back-office se compone de los menús de gestión necesarios para publicar, editar y eliminar contenido, así como para editar las fichas de usuario si fuese necesario.

El servicio ofrecido tiene como objetivo ser algo lúdico, centrado en el evento y en compartir contenido multimedia, por lo que desde el principio quedaron descartados workflows complejos con sistemas de borradores, revisión y publicación.

Lo interesante es que todos los usuarios puedan publicar su contenido con el mínimo indispensable de fricción y con la máxima inmediatez posible.

Por este motivo, todos los usuarios tienen acceso a publicar directamente sin necesidad de validación de otro usuario de rango superior o del administrador.

El sistema de roles está dividido de la siguiente manera mediante “Home”>”Admin”>”People”>”Permissions”:

- **Usuarios anónimos y autenticados:** Pueden ver todos los nodos y paragraphs y vistas, excepto:
 - Usuarios- No tiene sentido que posibles extraños vean quien va o no va al evento
 - Permiso relacionado → “User” > “View user information” = Desactivado
 - Módulo Content Access → Permite limitar acceso a Content types concretos filtrando a través de rol.
 - Para estos usuarios la opción “Usuarios” ni siquiera aparece en el menú de navegación

- Música- La sección música contiene links abiertos a Spotify para listas colaborativas. No tendría sentido que cualquiera pueda colaborar.
 - Módulo Content Access para el Content Type "Music"
- Comentarios- estos usuarios pueden leer los comentarios, pero no responder ni escribir nuevos comentarios
 - Como se explica en el punto 9.1.7.1.3, los comentarios están colapsados por defecto. Al hacer clic en el botón para expandir se detectará si el usuario está logado y se le darán un mensaje junto con el botón que conduce a la página de acceso para logarse.
 - Permiso relacionado→ "Comments" > "Post comments" = desactivado
- Contenido de ningún tipo. Los usuarios anónimos y autenticados no tienen acceso a la "Administration toolbar" ni pueden ver el botón "add_content_button" ni crear contenido de ningún modo.
 - Permiso relacionado→ "Toolbar" > "Use the administration toolbar" =desactivado
 - Todos los permisos que contienen "Create new content" = desactivado
 - Módulo Content Access para el Content Type "add_content_button"→ No aparece en pantalla para estos usuarios.
- **"Participant"**: Es el con login activo con permisos más básicos. Puede hacer:
 - Crear contenido de tipo "Foto" y "Video". Puede editar y borrar su propio contenido.
 - Permiso relacionado→"Node">"Administer content"= activado
 - (opcional) "Use the administration toolbar" =activado

- La activación de la barra de administración es opcional, dependiendo de los requerimientos de los usuarios. Con el botón "add_content_button" el usuario "Participant": puede crear nuevos nodos (de vídeo desde las páginas de vídeo y de foto desde las páginas de foto. Después puede buscar su contenido navegando por la página y editarlo o borrarlo.

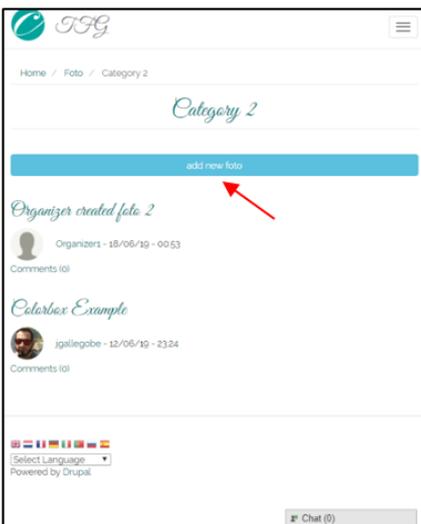


Figura 26 Detalle de pestañas de edición cuando el contenido ha sido creado por el usuario "Participant"

Figura 27 Detalle de "add_content_button" en los resultados de una de las categorías de foto

- Alternativamente, al usuario se le puede dar permiso a la administration toolbar.

- Desde ahí puede acceder a la sección “Content” donde verá un listado con todo el contenido. Solo podrá crear, modificar o borrar el suyo propio. Puede crear contenido a través de la propia barra de herramientas o del botón “add content” que aparece en esa pantalla.
- Ambos sistemas pueden coexistir o funcionar de forma independiente, dependiendo del nivel de los requerimientos del cliente.
- **“Participant_minus”**: Igual que Participant pero sin acceso a poder ver la sección “Music” para que no colabore con las listas de reproducción de Spotify
 - Módulo Content Access para el Content Type “Music” → La sección no aparece en pantalla para estos usuarios.
- **“Participant_plus”**: Igual que Participant pero puede también añadir contenidos del tipo Blog entry
 - Permiso relacionado → “Node” > “Blog Entry” todos activados.

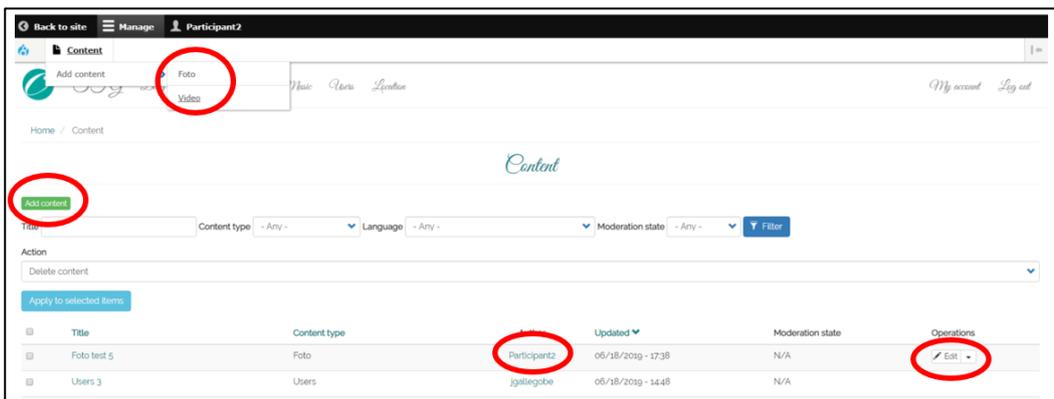


Figura 28 Detalle de pantalla de administración para usuario “Participant”

- **“Organizer” y “The Couple”** : Son equivalentes en permisos, la única diferencia es que en caso de que el evento sea una boda, los novios sería usuarios del tipo “The Couple” Pueden hacer lo mismo que los usuarios “Participant” y además:
 - Editar Content Type “Music”- Para añadir sus propias listas de reproducción
 - “Node” > “Music: Edit any content”= Activado
 - Crear, editar, eliminar contenido de tipo “Blog Entry”
 - Todos los permisos en “Node” para “Blog Entry”
 - Administrar Usuarios- Puede editar las fichas de usuario para actualizar información, borrar, suspender, cancelar, añadir nuevos. La única excepción es la ficha de usuario del “Administrator”
 - Permiso → “Administer users”= Activado

- Módulo “User Protect”- Se configura para que la ficha del “Administrator” sea solo modificable por él mismo.

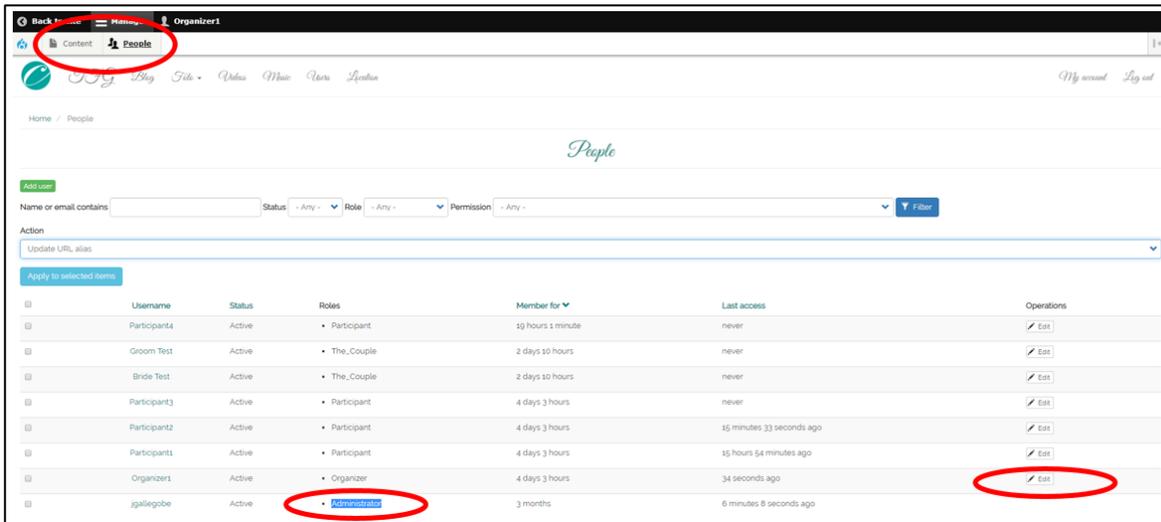


Figura 29 Detalle de pantalla de administración>People para usuario “Organizer”/”The Couple”

- Cambiar de rol a Usuarios- Mediante el módulo “Role Delegation” pueden dar o quitar a los usuarios de con rol “Participant” el rol “Participant_Plus” o “Participant_Minus”

“Administrator”: Es el rol reservado para mí que contaría con todos los permisos y accesos activos

5. Metodología

La metodología de trabajo consiste en una adaptación de la metodología Waterfall⁶ o en cascada, adaptada a las herramientas usadas durante el desarrollo del TFG.

La metodología Waterfall fue creada por Dr. Winston W. Royce en 1970 y, de forma general, consiste en los siguientes pasos:

1. Análisis de requisitos
2. Diseño del sistema
3. Diseño del programa
4. Codificación
5. Pruebas
6. Implementación o verificación del programa
7. Mantenimiento

⁶ ROYCE, Winston W. Managing the development of large software systems: concepts and techniques. En Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, 1987. p. 328-338.

En este caso se opta por una simplificación de la metodología previamente expuesta ya que no se está codificando un programa desde 0 si no que se usa Drupal 8 como CMS.

Esto implica que se pueden llevar a cabo pruebas de forma casi inmediata sin necesidad de programar y con muy poco trabajo previo de diseño.

Experiencias previas en el uso de Drupal 8 me han demostrado que esto puede convertirse en un arma de doble filo. Por un lado, esta posibilidad de ver el resultado deseado en funcionamiento (o una aproximación simplificada/incompleta) de una característica, puede resultar muy útil cuando no se sabe del todo bien como funciona un determinado módulo o una determinada configuración.

Por otra parte, es fácil caer en tentación, y posterior rutina, de probar cada pequeño cambio e ir diseñando de forma orgánica a base de estos pequeños cambios. Aunque es posible conseguir buenos resultados de este modo, muchas veces se destina una gran cantidad de tiempo a hacer y deshacer cambios, con los consiguientes tiempos de espera, limpiezas de caché y posibilidades de crear un problema de forma involuntaria donde antes no lo había.

En un proyecto con plazos estrictos de desarrollo y entrega como es éste, y como lo son la gran mayoría de proyectos no personales, tiene más sentido dedicar un tiempo razonable a la planificación en los puntos del 1 al 3 de la lista y ceñirse posteriormente al plan desarrollado durante el punto 4.

En resumen, cuantos más posibles comportamientos de usuarios, variaciones y malos usos de la herramienta se hayan tenido en cuenta en la fase de planificación, menos posibilidades hay de que éstos aparezcan durante la fase de pruebas forzando a buscar soluciones en ese momento y volver a la fase de diseño de nuevo para ello, con el consiguiente retraso.

Ahora bien, desde 1970 el mundo del desarrollo de software a evolucionado mucho y hoy en día, la metodología Waterfall se considera que representa una gran rigidez propia del desarrollo corporativo en el que prima la rentabilidad por encima de la creatividad y es muy poco adaptable a los cambios que puedan surgir en el proceso. Prueba de este cambio de paradigma entre los desarrolladores es el Manifiesto por el Desarrollo Ágil de Software ⁷, publicado en 2001.

Además, una crítica frecuente a la metodología Waterfall es que la fase de testeo se sitúa al final del proceso, por lo que, si un error grave es detectado, corregirlo puede ser muy difícil y costoso (en tiempo y dinero) forzando en ocasiones a tener que deshacerse de mucho de lo ya hecho.⁸

Como se puede ver en la sección 8 de esta memoria, cada elemento de la web tiene su tiempo de desarrollo planificado. Esto implica que para cada uno de estos elementos hay que aplicar individualmente la metodología Waterfall, a la par que ésta se aplica al proyecto en su totalidad.

Esto quiere decir que, en la fase de diseño inicial de proyecto hay un diseño general de los elementos que la componen y de las interacciones entre ellos, pero al llegar al punto 4, el de codificación, dentro de cada sub fase de desarrollo del proyecto queda anidado otro proceso con metodología waterfall más concreta para cada una de ellas que incluirá los puntos del 2 al 6.

⁷ BECK, Kent, et al. Manifiesto for agile software development. 2001.

⁸ PERIYASAMY, J., [sin fecha]. What is Waterfall model and list its advantages, disadvantages. [en línea]. [Consulta: 5 marzo 2019]. Disponible en: <http://jobsandnewstoday.blogspot.com/2013/04/what-is-waterfall-model.html>.

El objetivo de esto es contar con más flexibilidad para añadir los cambios necesarios a cada una de las partes del proyecto durante el proceso de desarrollo de este, sin tener que esperar al final para hacer estos cambios cuando puede no haber tiempo suficiente para llevarlos a cabo.

Finalmente, en el punto 6 del proceso general para todo el proyecto, éste es sometido a un análisis heurístico basado en los 10 principios de usabilidad de Nielsen⁹ y a una prueba de usabilidad con usuarios, de modo que se puedan detectar espacios de mejora y aplicar los consiguientes cambios a la web para que el resultado final sea óptimo.

Cada uno de estos cambios se desarrolla siguiendo a su vez la metodología Waterfall en sus puntos del 2 al 6.

Soy consciente de que la elección de esta metodología Waterfall está algo pasada de moda, pero para el alcance de este proyecto me parece adecuada. Los requerimientos están claros y no se esperan grandes cambios durante el proceso. La tecnología es conocida y no se pretende inventar nada nuevo.

Así mismo, en este caso no se está en contacto con un cliente que pueda exigir cambios constantes durante el proyecto, por lo que el desarrollo de este es relativamente predecible, como se puede ver en el punto 9 de esta memoria.

El mayor riesgo de esta metodología es encontrarse en la fase final del proyecto con la necesidad de un cambio profundo, pero al hacer el trabajo con Drupal 8 se tiene la ventaja de que los resultados van a ir siendo visibles durante el proceso, por lo que se pueden ir solicitando feedback a distintas personas en diferentes estadios de desarrollo del proyecto, incorporando mejoras tras la fase pruebas de cada una de las subsecciones a desarrollar.

6. Plataforma de desarrollo

La plataforma escogida es Drupal 8. Me decanté por Drupal tras comparar funcionalidades con WordPress¹⁰ y PicoCMS¹¹ en la asignatura *Sistemes de gestió de continguts*. Otra posibilidad habría sido probar Joomla!¹², pero nunca lo he usado y aprender desde cero como funciona un nuevo CMS me pareció fuera del alcance propuesto para el TFG, que debería basarse en profundizar conocimientos adquiridos durante el grado.

PicoCMS quedó descartado casi directamente. Aunque usa templates twig y archivos yaml (*YAML Ain't Markup Language*) de un modo similar al que son usados en Drupal, Drupal cuenta con estas funcionalidades, pero además una mayor cantidad de módulos, temas y, sobre todo, información y soporte online. Para un proyecto de una envergadura relativamente grande como éste, desarrollado por una sola persona, la información online es un factor determinante.

⁹ NIELSEN, Jakob. Ten usability heuristics. 2005.

¹⁰ WordPress.com: crea un sitio web o blog gratuito. WordPress.com [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <https://es.wordpress.com/>.

¹¹ Pico - A stupidly simple, blazing fast, flat file CMS. [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <http://picocms.org/>.

¹² Joomla Content Management System (CMS) - try it for free! Joomla! [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <https://www.joomla.org/>.

PicoCMS me parece más adecuado para un proyecto pequeño. No me cabe duda de que es perfectamente posible desarrollar mi proyecto con PicoCMS, pero hubiese requerido desarrollo de módulos y modificación extensiva de templates twig, lo que escapa a mis posibilidades por tiempo y alcance.

La decisión final fue entre WordPress y Drupal, ya que son los CMS en los que más se profundizó en la asignatura *Sistemes de gestió de continguts*. A continuación una comparación de varias características de ambos sistemas.

1. Definición de tipos de contenidos, campos y categorías.

- Con Drupal es más sencillo generar diferentes tipos de contenido sin necesidad de utilizar código para crear nuevas plantillas. Las plantillas, equivalentes a las de WordPress, se generan muy fácilmente desde Administration>Structure>Add content type. Desde ahí, se pueden añadir/eliminar o reorganizar los campos de estas mediante menús. Esto nos permite muy fácilmente filtrar posteriormente los contenidos de forma dinámica o hacer cambios en las plantillas que se apliquen de forma dinámica a todo el contenido (existente o futuro). Si comparamos Wordpress y Drupal sin extensiones, el sistema de Drupal resulta más flexible a la hora de crear contenido personalizado.
- Con respecto a las taxonomías, Wordpress permite el uso de Categorías y Tags en su configuración por defecto, pero Drupal, además de los tipos de contenido antes mencionados, permite generar múltiples tipos de vocabularios controlados desde Administration> Structure> Taxonomy > + Add vocabulay. Estos se pueden usar para generar subcategorías para los tipos de contenido, de modo que los usuarios seleccionen, por ejemplo, la taxonomía “los novios” en el tipo de contenidos “Foto” y se pueda configurar una vista que muestre únicamente los contenidos que cumplan con estos dos parámetros y para la que no aparezcan otros tipos de foto.

2. Plugins o extensiones

- Tanto Drupal como Wordpress cuentan con gran cantidad de plugins, pero me da la sensación de que Drupal, al ser software libre, cuenta con una comunidad de desarrolladores (no de usuarios, al ser más complejo) más numerosa. Es relativamente fácil encontrar módulos específicos casi para cualquier necesidad que se pueda plantear, completamente gratuitos. En el caso de Wordpress, al ser un CRM muy extendido también hay gran variedad de extensiones y plugins, pero muchas de ellas de pago. Además, hay gran cantidad de información sobre como instalar y usar los módulos en los foros oficiales de Drupal, que considero es más desarrollada y mejor que la que se puede encontrar en los de Wordpress.

3. Blocs/Widgets

- El sistema de bloques y vistas de Drupal es más potente que las opciones por defecto disponibles en Wordpress. Muy rápidamente se puede anidar un tipo de contenido dentro de otro, de modo que se pueden tener múltiples secciones de una página con comportamientos muy distintos. Es más fácil asignar determinados bloques en determinado orden a unas vistas y

no a otras, de modo que se puede hacer que la apariencia y contenido de las diferentes partes de la web de forma más sencilla que con los widgets en Wordpress.

Conclusiones:

Drupal 8 es el CMS que permite el nivel de personalización profunda y relaciones lógicas complejas y dinámicas entre contenidos, campos y taxonomías, que mejor se adapta a mi proyecto. El componente dinámico y configurable es especialmente interesante para mí, ya que al desarrollar una estructura que posteriormente debe adaptarse a los requerimientos variados de múltiples usuarios, Drupal permite llevar a cabo estas adaptaciones rápidamente.

Además, al ser tanto Drupal como sus módulos software libre, hay gran cantidad de material accesible (módulos, plugins) y gratuito que me permite acelerar el desarrollo sin incrementar el presupuesto y usando código que ha sido validado por la comunidad y en el que puedo confiar.

Por último, al tratarse de un proyecto que podría convertirse en una empresa con potencial de crecimiento, necesito un sistema que sea escalable¹³ y robusto¹⁴ para proyectos complejos y Drupal es reconocido como el más adecuado en esos ámbitos en todas las comparativas que he podido encontrar en la red.

7. Planificación

Diagrama de Gantt

Principales hitos

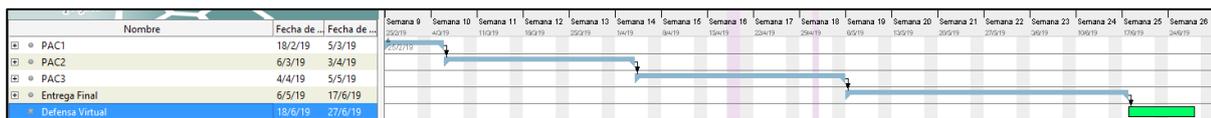


Figura 30 Diagrama de Gantt: Principales hitos

Actividades desglosadas

¹³ Why do people think Drupal is better than WordPress for scalability? - Quora. [en línea], [sin fecha]. [Consulta: 3 mayo 2019]. Disponible en: <https://www.quora.com/Why-do-people-think-Drupal-is-better-than-WordPress-for-scalability>.

¹⁴ WordPress vs. Drupal? Which is better. [en línea], [sin fecha]. [Consulta: 3 mayo 2019]. Disponible en: <https://www.commonplaces.com/blog/wordpress-drupal>.

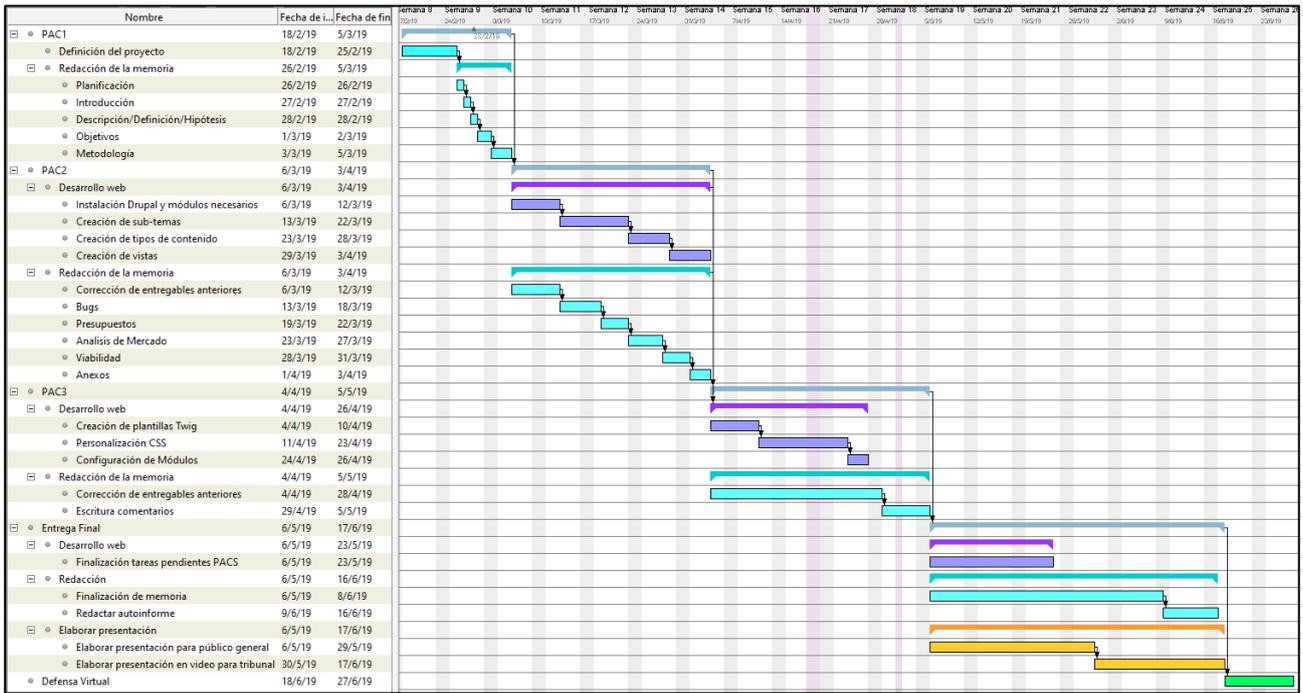


Figura 31 Diagrama de Gantt: actividades desglosadas

[Archivo Diagrama de Gantt editable con GanttProject \(Software Libre\):](#)



8. Prototipos

3 versiones de cada sección de la página. Todos los componentes están desarrollados como paragraphs por lo que se pueden diseñar rápidamente todas las permutaciones posibles de los elementos mostrados a continuación

Página principal

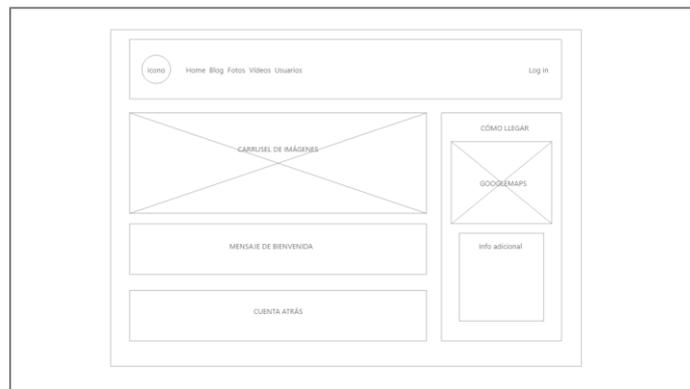


Figura 32 Página principal- Versión 1 - 2 columnas

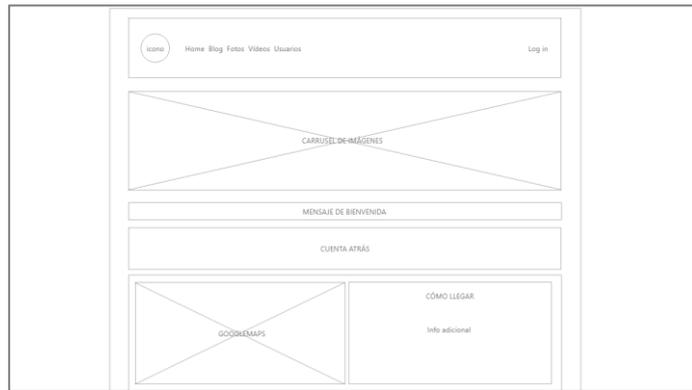


Figura 33 Página principal- Versión 2 - Todo en una columna

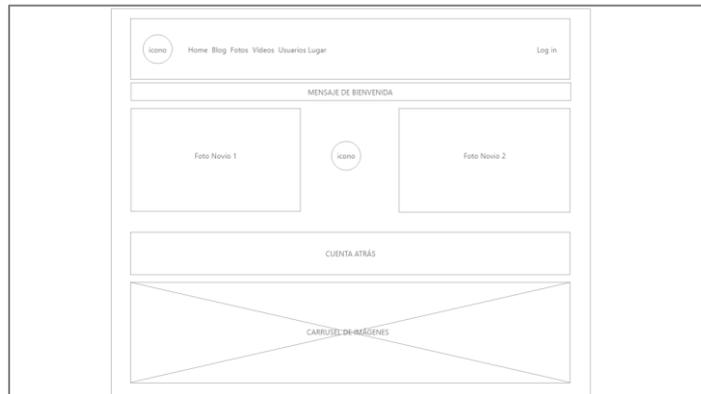


Figura 34 Página principal- Versión 3 - 2 columnas- Ubicación no está en la página principal, sino que tiene sección propia en el menú



Figura 35 Página principal- Navegación principal con menús desplegables

Galerías de fotos / vídeos

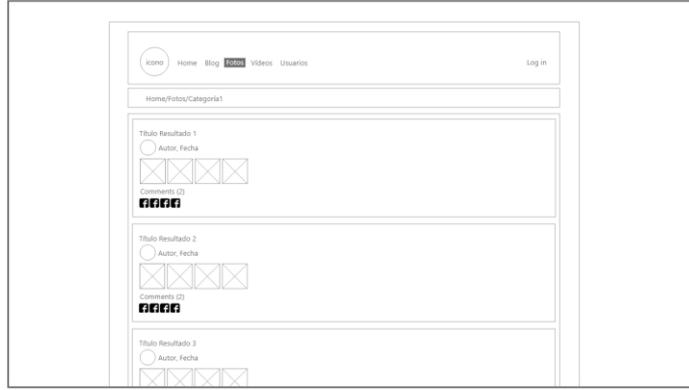


Figura 36 Galerías de fotos / vídeos Versión 1- Lista thumbnails- 1 columna

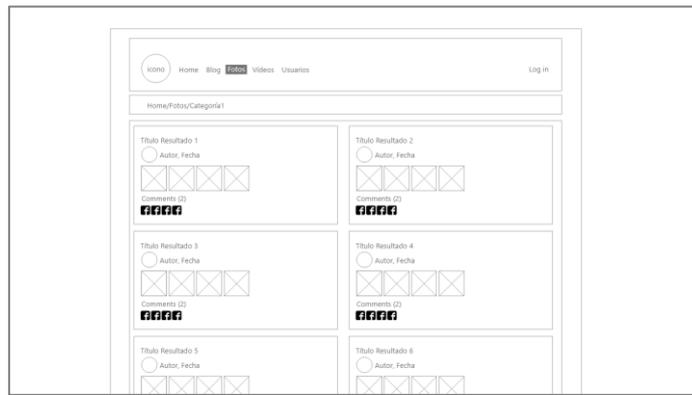


Figura 37 Galerías de fotos / vídeos Versión 2- Lista thumbnails- 2 columnas

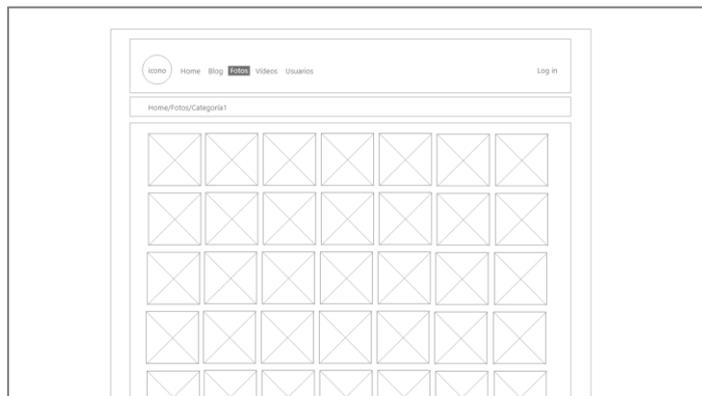


Figura 38 Galerías de fotos / vídeos Versión 3- Grid/ Solo thumbnails/ Mayor tamaño

Fotos

Dos versiones, ya que la versión 1 tiene dos “subversiones”: Una usando colorbox y otra usando PhotoSwipe

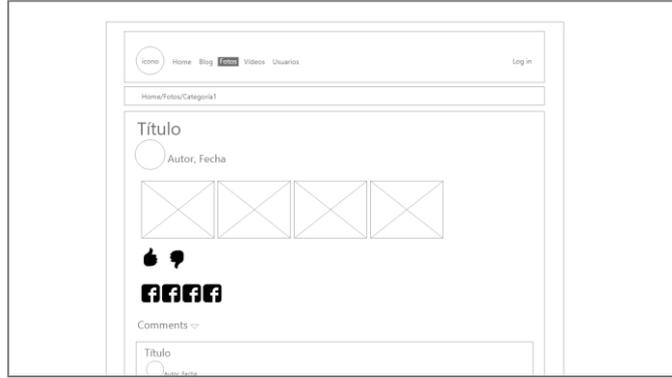


Figura 39 Fotos- Versión 1 – thumbnails- La galería se visualiza con colorbox o PhotoSwipe

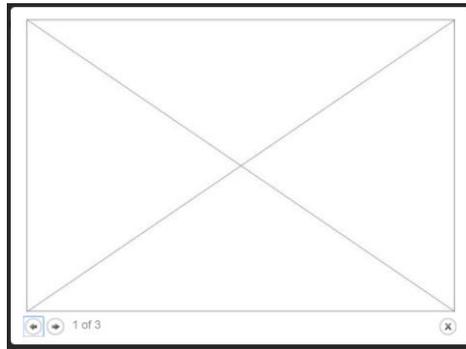


Figura 40 Fotos- colorbox – Pop up al hacer clic en un thumbnail

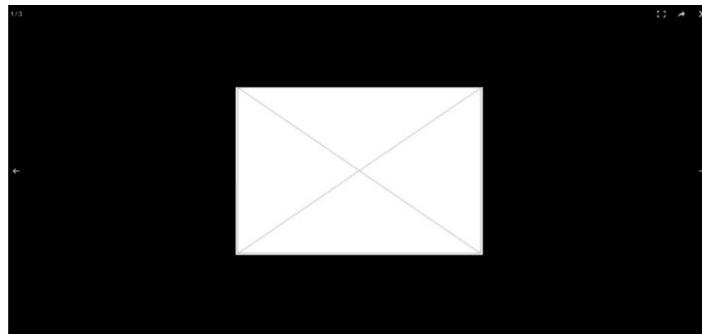


Figura 41 : Fotos- PhotoSwipe – Pantalla completa al hacer clic en un thumbnail

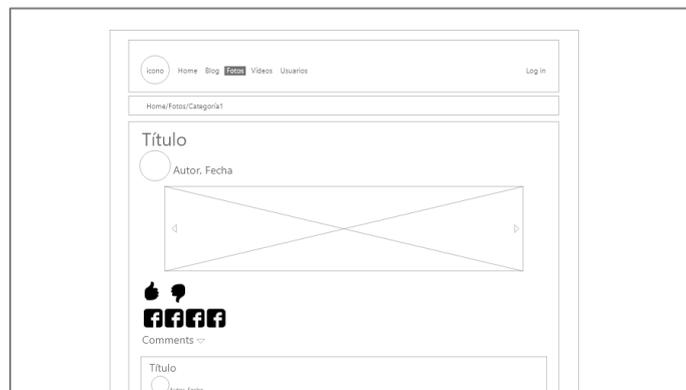


Figura 42 : Fotos- PhotoSwipe – Pantalla completa al hacer clic en un thumbnail

Videos

Para los videos solo se permite un video por post, por lo que la presentación de este es siempre igual. Se puede variar el tamaño del reproductor a petición del usuario, pero todos lo demás (Comments, votos) permanecerá igual que en fotos

Usuarios

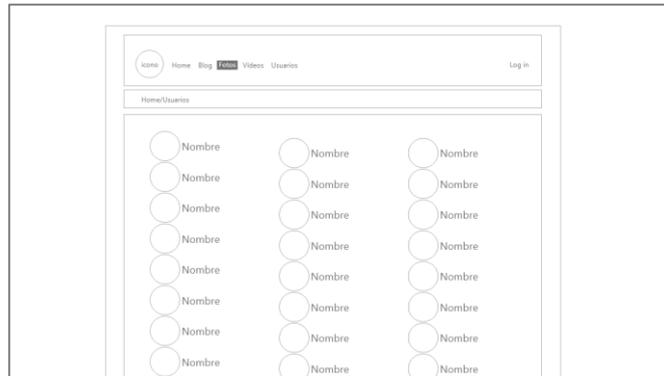


Figura 43 : Usuarios- Versión 1 – Grid simple, foto y nombre

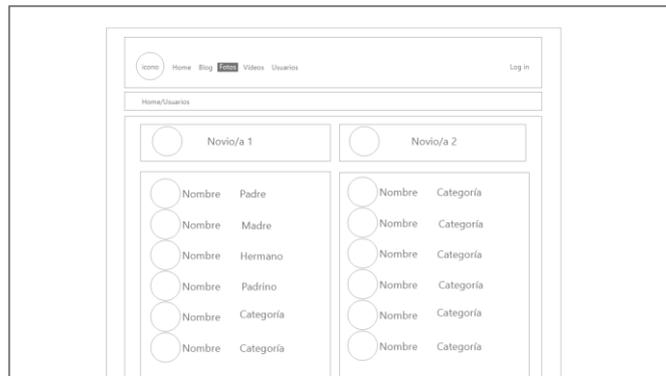


Figura 44 : Usuarios- Versión 2 – Dos columnas- Foto, nombre y relación con los novios

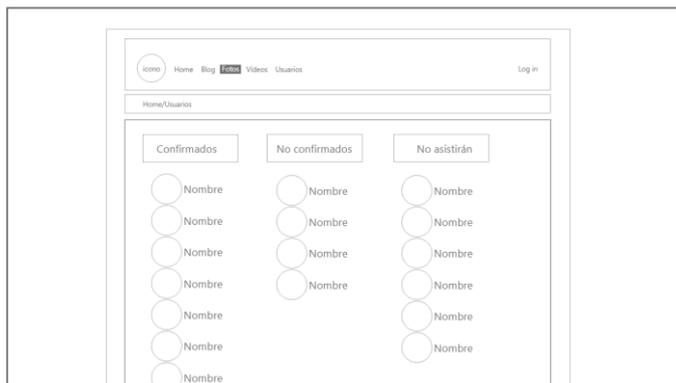


Figura 45 : Usuarios- Versión 3 – 3 columnas separades con el estado de asistencia

Artículos Blog

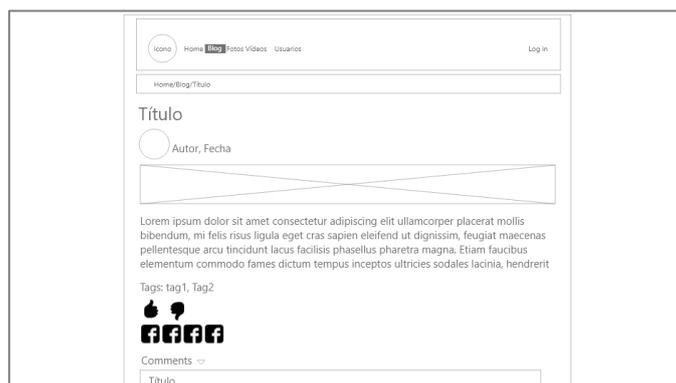


Figura 46 : Artículo. Con foto (opcional)- Una sola columna

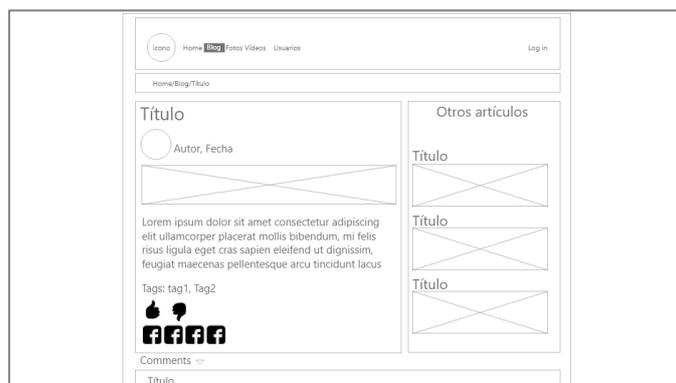


Figura 47 : Artículo. Dos columnas. Otros artículos aparecerán en la columna derecha



Figura 48 : Artículo. Dos columnas. Fotos aparecerán en la columna derecha

9. Proceso de trabajo

El proceso de trabajo detallado en la planificación en el punto de planificación (7) está enfocado al desarrollo de componentes reutilizables que permiten combinarse entre sí de múltiples maneras. De este modo, se pueden ofrecer soluciones personalizadas para el cliente a partir de una amplia gama de opciones desarrolladas previamente, haciendo este proceso de personalización tan rápido como lo que cueste recombinar los elementos deseados ya disponibles.

Todas las secciones disponibles en el menú de navegación de la página se han llevado a cabo mediante Views de tipo Page. La única excepción son las subsecciones que se corresponden a las diferentes categorías de las fotos. En este caso se accede a ellas a través de la vista Taxonomy term, ya que la forma en la que se distribuyen los nodos en las categorías es a través de taxonomías seleccionadas durante su creación.

Para llevar a cabo muchos de estos componentes se ha usado el módulo Paragraphs, que permite crear múltiples variaciones que funcionan como bloques insertables dentro de un nodo en un tipo de contenido.

También he utilizado el módulo Bootstrap Paragraphs, que ya incluye una selección preconfigurada, especialmente útil para la creación de contenido en múltiples columnas.

9.1 - Funcionalidades (back-end)

En el back-end se concentra la estructura y la lógica de la página, que no cambiará independientemente de la estética elegida posteriormente.

La ventaja de Paragraphs es que me permite tener configurados de antemano muchas variaciones diferentes de un mismo contenido, que de otro modo tendría que reconfigurar cada vez que quisiera usar desde Manage Display. Así mismo, permite limitar la cantidad de tipos de contenido necesarios, ya que a un mismo tipo de contenido se le pueden insertar diferentes Paragraphs para conseguir variantes.

Para la separación entre estética y estructura los Paragraphs son fundamentales, ya que mucho de lo que puede hacerse con ellos solo sería posible a través de modificaciones en “Block Layout” y éste depende del tema. Por lo tanto, si cambiáramos de tema, podríamos cambiar la estructura de la página involuntariamente.

Además, Paragraphs sirve para tener facilitar la creación de un muestrario de variaciones, aunque en el caso de este proyecto, no siempre es viable que el usuario final al que va dirigido inserte Paragraphs al crear contenido; por ejemplo, cuando se requiere crear contenido con múltiples columnas, columna wrappers y distintos sub-paragraphs y vistas dentro de cada una.

9.2 – Front-End

En front-end se engloba la parte estética de la web, y para su diseño, posterior mantenimiento y futuras ampliaciones se usan, por una parte, subtemas de Bootstrap diferenciados, de modo que la disposición de bloques (opcionalmente), tipografía, colores y demás elementos visuales cambien cada vez que se cambie de tema. No obstante, para tener un control preciso de cada uno de los componentes y elementos se usa una interpretación ligera de la metodología BEM (*Block Element Modifier*)¹⁵ y SMACSS (Scalable and Modular Architecture for CSS)¹⁶.

BEM- Se basa en la noción de que una web está compuesta por bloques, elementos dentro de los bloques y estados de estos elementos en función de las distintas modificaciones que pueden aplicárseles.

Siguiendo esta filosofía, estas diferentes partes se nombran en HTML siguiendo una serie de convenciones¹⁷, que en el caso de Bootstrap en Drupal 8 se ajustan hasta cierto punto al estilo “*Two Dashes style*”

Por ejemplo, de un paragraph, como clock, el paragraph es el bloque (o componente) y los campos de este son los elementos. Los diferentes estados de los campos (hidden, por ejemplo) serían los modificadores.

En un caso concreto de clase CSS:

```
.field--name-field-jquery {text-align: center;}
```

¹⁵ Methodology / BEM. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <https://en.bem.info/methodology/>.

¹⁶ Home - Scalable and Modular Architecture for CSS. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <http://smacss.com/>.

¹⁷ Naming convention / Methodology / BEM. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <https://en.bem.info/methodology/naming-convention/#two-dashes-style>.

Dentro de la clase indicada arriba, *field* es un elemento dentro del componente paragraph y su modificador (lo que lo diferencia del resto de fields), separado por guiones dobles (--) , es su nombre *name-field-jquery* (las partes del nombre se separan con guiones simples (-)).

No obstante, he indicado que es una interpretación ligera porque el nombre de las clases que genera tanto Bootstrap/Drupal como los módulos descargados no siguen necesariamente estas convenciones. Por lo tanto, aunque sigue siendo útil interpretar el contenido de la web con la jerarquía y subdivisiones de BEM, no siempre es aplicable.

Muchas veces es preciso concatenar nombres de clases para los modificadores, por ejemplo:

field-timer-jquery-countdown.is-countdown

SMACSS- Es una guía de organización y estilo que ayuda a organizar el CSS de la página. Aunque es más complejo, he usado únicamente la división de CSS en diferentes archivos siguiendo las pautas de SMACSS. De este modo, en lugar de incluir todo el CSS en un archivo *styles.css*, lo reparto en múltiples archivos en diferentes carpetas, cada una con su misión específica.

En la carpeta "base" almaceno diferentes archivos CSS que contienen etiquetados sobre tipografía o sobre los elementos básicos de la web (h1, p, a, etc)

En la carpeta "components" almaceno los estilos de cada uno de los componentes, como clock.

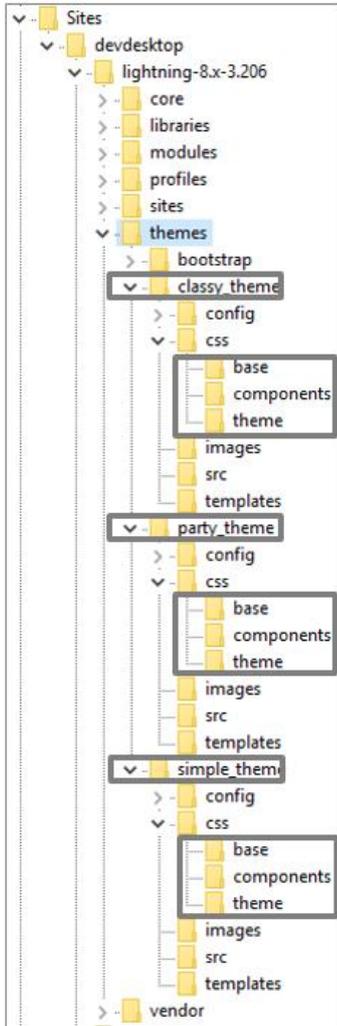


Figura 49 : Estructura de carpetas

En la carpeta “theme” están los estilos referentes a colores, bordes, sombras y aquellos puramente estéticos.

Podría haber otras carpetas llamadas “layout”, para la distribución de elementos en sistemas de columnas o rejilla y “state”, para los estados que modifican a los componentes.

En mi caso, para simplificar el proceso de trabajo, limitando algo la claridad, he decidido usar el sistema de paragraphs, bloques y vistas siempre que haga falta dividir el contenido en columnas o rejillas (“layout”) e incluir la información sobre las modificaciones (“state”) en “components”.

Igualmente, se puede ver en la figura a la izquierda que cada uno de los tres temas cuenta con esta división de carpetas para el CSS, haciendo posible, en caso de ajustarse a la petición del cliente, pasar el archivo un estilo de componente de un tema a otro.

Esto permite utilizar el estilo de un componente en diferentes temas o ir desarrollando múltiples variaciones de estilo por tema conforme se vayan completando proyectos.

Todos los temas, además de la estructura indicada para el CSS cuentan con sus templates en la carpeta “templates”.

Para que Drupal reconozca el CSS es preciso indicar en el archivo “libraries.yml” del tema correspondiente todos los nombres de archivos CSS y sus rutas correspondientes, así como enlazar a todas las fuentes que se quieran tener disponibles.

```

1 framework:
2   css:
3     base:
4       css/base/typography.css: {}
5       css/base/elements.css: {}
6       css/base/container.css: {}
7
8     components:
9       #home
10        css/components/home/clock.css: {}
11        css/components/home/slideshow.css: {}
12        css/components/home/navbar.css: {}
13        css/components/home/home3_center icon.css: {}
14       #fotos
15        css/components/foto/items.css: {}
16        css/components/foto/foto_slideshow.css: {}
17
18       #general
19        css/components/map.css: {}
20        css/components/breadcrumbs.css: {}
21        css/components/paragraph.css: {}
22        css/components/like.css: {}
23        css/components/comment.css: {}
24        css/components/fotoarticulo.css: {}
25        css/components/fotouser.css: {}
26        css/components/tags.css: {}
27        css/components/users.css: {}
28        css/components/addform.css: {}
29
30     theme:
31       css/theme/color.css: {}
32
33 fonts:
34   css:
35     titles:
36       '//fonts.googleapis.com/css?family=Faster+One|Lobster|Press+Start+2P|Princess+Sofia|Rock+Salt|Lato|Open+Sans|Roboto|Paytone+One|Raleway': { type: external }
37
38

```

Figura 50 : Detalle de party_theme.libraries.yml - Lista de librerías CSS y fuentes

En todos los temas se han separado todas las instrucciones que tienen que ver con color en el archivo “color.css” de modo que mediante un simple “Find and Replace” en el texto se puedan cambiar los colores de la página a voluntad del usuario.

Así mismo, todas las instrucciones que tiene que ver con cambios de fuente de texto, se han centralizado en el archivo “typography.css”, de modo que, de nuevo, sea muy sencillo cambiar la tipografía del mismo modo que el color. En cada tema se ha incluido en texto comentado la información necesaria de todas las tipografía incluidas en su respectivo “libraries.yml”, para poder copiar y pegar.

```

1 /**
2  * CLASSY
3  */
4
5
6 /**
7  * Titles:
8     font-family: 'Delius Swash Caps', cursive;
9     font-family: 'Tangerine', cursive;
10    font-family: 'Pinyon Script', cursive;
11    font-family: 'Bilbo', cursive;
12    font-family: 'Amita', cursive;
13    font-family: 'Great Vibes', cursive;
14
15
16
17  * Body:
18    font-family: 'Lato', sans-serif;
19    font-family: 'Open Sans', sans-serif;
20    font-family: 'Roboto', sans-serif;
21    font-family: 'Merriweather', serif;
22    font-family: 'Raleway', sans-serif;
23  */
24

```

Figura 51 : Detalle de typography.css para el tema Classy Theme

Además, se han generado Logos y favicons personalizados para cada tema, siendo la idea de negocio generar bajo demanda iconos para los clientes.



Figura 52 : Respectivamente los Logos de Simple, Classy y Party

Técnicas aplicadas para conseguir comportamiento “responsive”

Al utilizar subtemas de Bootstrap ya hay grandes ventajas desde este punto de vista. Bootstrap incluye por defecto clases “responsive” como “img-responsive, que ayudan a que en todo momento la visualización sea correcta con diferentes tamaños de pantalla y en distintos dispositivos.

Se han aplicado siempre que ha sido posible medidas basadas en valores relativos en porcentaje y no en medidas en píxel. Además, para los carruseles e imágenes disponibles en las entradas del blog y las páginas principales, se ha utilizado la propiedad “object-fit” que ayuda a evitar distorsiones de relación de aspecto en las imágenes.

10. Usabilidad/UX

El sitio se ha desarrollado teniendo en cuenta los principios básicos de usabilidad de Nielsen¹⁸, lo que resulta relativamente sencillo ya que Drupal y el tema de Bootstrap ya han sido claramente diseñados con estos principios en mente.

1. Visibilidad del estado del sistema

He tenido especial cuidado tanto con las breadcrumbs como en en usar URLs limpias de modo que el usuario siempre sepa dónde está y pueda volver atrás o navegar por la jerarquía de la web sin perderse en ningún momento. Para ello he usado alias de URL mediante el módulo Pathauto que son las que proporcionan la información al módulo Easy Breadcrumb para crear las breadcrumbs correctamente

2. Adecuación entre el sistema y el mundo real

Por este motivo, a pesar de que el módulo Like It no incorpora ninguna imagen por defecto (solo un link con la palabra like o dislike) busqué un icono con la forma del pulgar, y sustituí la palabra por una imagen mediante CSS (background-image url) ya que es una convención social del mundo real su uso para indicar una respuesta positiva o negativa.¹⁹

3. Libertad y control por el usuario

18 NIELSEN, Jakob. Ten usability heuristics. 2005.

19 iOS Glyph Icons. [en línea], [sin fecha]. [Consulta: 17 junio 2019]. Disponible en: <https://icons8.com/icon/set/like/ios-glyphs>.

Nunca hay una situación en la página en la que el usuario se ve forzado a continuar un proceso sin poder volver atrás. Se puede cancelar los procesos de subida de imágenes y vídeos. Todos los contenidos propios se pueden editar en caso de error gracias a los permisos otorgados de acuerdo con lo indicado en el punto anterior

4. Consistencia y estándares

Me he preocupado de estilar los enlaces de modo que siempre reaccionen a las acciones del usuario, como es el estándar para el correcto reconocimiento de sus funciones como enlace. La web, en todos sus subtemas cumple las convenciones más clásicas de diseño web, con la barra de navegación la parte superior, botón “home” en la esquina superior izquierda, parte central y “footer”.

5. Prevención de errores

Los permisos otorgados cuidadosamente a cada uno de los roles imposibilitan que haya usuarios que puedan borrar o modificar contenidos de otros por error. Los campos de formulario obligatorios están adecuadamente marcados para que sea evidente que deben ser rellenados.

6. Reconocer mejor que recordar.

De nuevo las breadcrumbs ayudan a que el usuario reconozca dónde está y a dónde puede ir de forma visual, sin necesidad de tener que recordar una ruta determinada.

7. Flexibilidad y eficiencia de uso

Se han generado dos métodos alternativos para crear nuevos nodos de tipo “Foto” y tipo “Video” de modo que los usuarios puedan elegir lo que se adapta mejor a sus necesidades.

Se han habilitado los “Shortcuts” (Atajos) para todos los usuarios que tenga a su vez habilitada la barra de administración.

8. Estética y diseño minimalista

Incluso en el tema Party, aunque la tipografía no se puede considerar minimalista, he procurado dar suficiente espacio a todos los componentes de la página y no sobrecargar visualmente al usuario.

Un ejemplo de ello es la versión personalizada del “submitted by” generada a través de plantilla TWIG y que es mucho más corta y visualmente armoniosa que la versión por defecto. Los logotipos diseñados para cada tema son también minimalistas, con colores planos y formas sencillas.

9. Ayudar a los usuarios a reconocer, diagnosticar y solucionar los errores

Además de los mensajes de error por defecto que pueda devolver Drupal o el tema de Bootstrap, por ejemplo, en el caso de los comentarios, he generado una comprobación lógica, de modo que si el usuario accede a los comentarios sin estar logado el sistema lo detecta y facilita la información junto con un enlace hacia la página de acceso.

10. Ayuda y documentación

La sección “Music” es básicamente un manual sobre como utilizar listas colaborativas de Spotify. En un entorno real, el primer artículo del blog hubiera sido una guía con la información necesaria para poder empezar a subir contenido (Aunque en mi opinión es suficientemente intuitivo sin necesidad de ella, nunca está de más incluir información de consulta). En esta memoria se incluye una guía de instalación y uso.

11. Proyección de futuro

A raíz de lo aprendido durante la elaboración del proyecto pretendo ampliar la cantidad de contenidos y plantillas y ofrecer el servicio de forma local en mi entorno. Si consigo llevar a cabo al menos cinco páginas con éxito de forma gratuita o cobrando muy poco, pretendo hacer una inversión en publicidad y comenzar a ofrecer el servicio de forma profesional.

12. Presupuesto

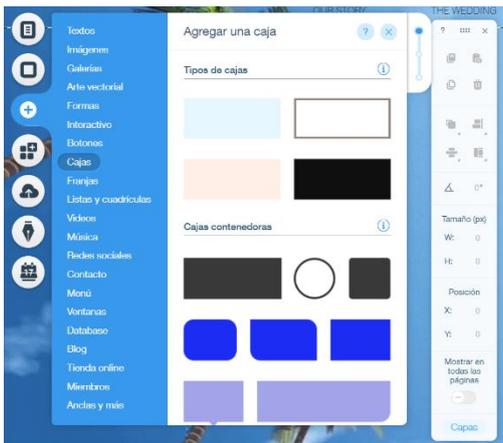
- Equipo humano
 - Proyecto unipersonal
- Equipamiento técnico
 - Un ordenador personal
 - Acceso a internet doméstico
- Hosting y dominio
 - Al tratarse de páginas web para eventos, el tiempo de hosting varía dependiendo de la tipología de página
 - Boda
 - Hosting de 1 año incluido en el precio para cliente - 5 euros * 12 meses = 60 euros
 - Dominio: 10 euros.
 - Eventos de otro tipo
 - Hosting- Variable en función del tiempo que la web tiene que estar activa
 - Dominio: 10 euros.

13. Análisis de Mercado

Público objetivo

Existe una cantidad considerable de páginas web que ofrecen asistentes para la creación de una web personalizada para BODAS, y algunas, como Wix, que ofrecen todo tipo de plantillas, entre las que hay bodas y eventos de diferentes sitios.

La propuesta consiste no en ofrecer asistentes para que el usuario haga su propia página, si no generar la página adecuada a las especificaciones del usuario (ofreciéndole una serie de temas que se irán ampliando conforme el proyecto se consolide y avance) y proporcionarle acceso a un panel de control desde el que pueda gestionar permisos y actualizar el contenido.



En este sentido, el perfil de usuario es éste que está dispuesto a pagar por un producto personalizado y que no desea, aunque sea con un asistente, llevar a cabo el trabajo de diseño de la página web. Un editor como Wix, aunque muy flexible y, sin duda más intuitivo que llevar a cabo la página desde 0 o con Drupal, igualmente requiere una curva de aprendizaje considerable y la cantidad de opciones puede ser muy elevada. Además, aunque es gratis comenzar a hacer la página, conforme se quieren añadir funcionalidades, se va encontrando que ciertas opciones son de pago

Figura 53 : Editor Wix- ejemplo de interfaz con muchas opciones

El perfil es el de usuarios con pocos conocimientos, poco tiempo o que simplemente quieren delegar el trabajo de diseño de la web en un profesional, siempre reservándose el papel de administrador de contenidos y permisos para el resto de los usuarios de la web.

En el caso de la sección de Bodas, el público objetivo es:

- Parejas muy jóvenes (18-26). Poder adquisitivo bajo o medio
- Parejas jóvenes (27-35). Poder adquisitivo medio o medio-alto.

En el caso de fiestas y eventos:

- Cualquier persona de 18 a 70 años. Poder adquisitivo de bajo a alto.

Competidores



The Knot

Wedding Websites - Free Wedding Websites - The Knot. [en línea], [sin fecha].

[Consulta: 11 marzo 2019]. Disponible en:

<https://www.theknot.com/gs/wedding-websites>.

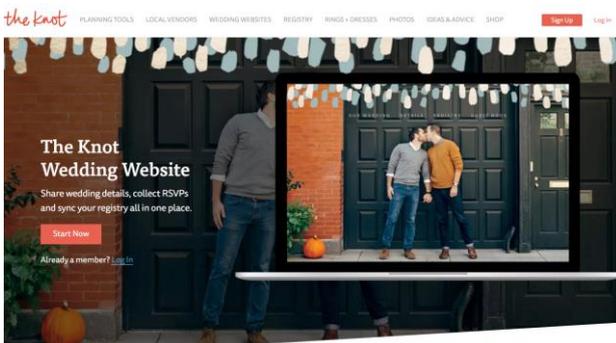


Figura 54 : The knot, página web

Página de organización de bodas con una sección para la creación de páginas web.

(+) Positivo:

- Más de 100 diseños
- Invitaciones a juego con la web o viceversa
- Web builder sencillo e intuitivo

(-) Negativo:

- Compartición de fotos y vídeos a través de aplicación externa
- No hay lista colaborativa para la música
- Web builder limitado

ZANKYOU Web de Boda Gratis. [en línea], [sin fecha]. [Consulta: 11 marzo 2019].
Disponibile en: <https://www.zankyou.es/web-de-bodas>.



Figura 55 : Zankyou, página web

Página de organización de bodas con una sección para la creación de páginas web.

(+) Positivo:

- Más 50 diseños
- Web builder sencillo e intuitivo

(-) Negativo:

- Promociones y sorteos integrados en el editor de la web
- No hay lista colaborativa para la música
- Web builder limitado
- Funcionalidades bloqueadas tras “muros de pago”
 - Cuentas de usuario y acceso restringido para los invitados



Wix

Plantillas Web HTML de Eventos | Wix. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <https://es.wix.com/website/templates/html/events>.



Figura 56 : Wix, página web

Página asistente de creación de páginas web, con sección para eventos, incluyendo bodas.

(+) Positivo:

- 34 diseños
- Web builder con muchas opciones
- Gran cantidad de módulos descargables

(-) Negativo:

- Web builder limitado puede resultar demasiado complejo para un usuario
- Muchos módulos son de pago

14. Márquetin y ventas

El proyecto queda circunscrito al ámbito local, como base para un pequeño negocio de diseño de páginas web. Por lo tanto, el márketing queda, en una primera fase, limitado al boca a boca y a anuncios en papel en lugares públicos (biblioteca, ayuntamiento, escaparates de negocios de amigos y conocidos). Si el proyecto crece y la cantidad de encargos y beneficios crecen, los canales de márketing se escalarán proporcionalmente. Por ejemplo, si la cantidad de ingresos lo permite, se incluirán banners en otras webs, anuncios radiofónicos en emisoras locales, anuncios en televisión local.

15. Conclusiones

Mi idea original era generar exclusivamente sub-temas con CSS y Twig, tipos de contenido y vistas para este proyecto. Antes de comenzar desconocía la existencia de Paragraphs y todas sus posibilidades.

Al terminar este proyecto me doy cuenta de que he arañado la superficie de lo que se puede hacer con los paragraphs, ya que el nivel de personalización mediante CSS de estos que he llevado a cabo no es muy elevado.

Por falta de tiempo se han quedado fuera del trabajo el análisis heurístico, más allá de la breve justificación de los 10 criterios de Nielsen, y las pruebas con usuarios, que me hubiera gustado poder hacer para mejorar el diseño de la página.

La conclusión es satisfactoria, porque he podido terminar el trabajo de forma funcional, con las características deseadas que funcionan con se espera de ellas, pero no óptima porque no he exprimido al máximo el potencial de las herramientas con las que he trabajado y detecto un potencial de mejora que espero cubrir en los próximos meses ampliando este proyecto con fines comerciales-

16. Requisitos de instalación

Con migración a servidor, únicamente haría falta acceso de desde un ordenador con internet.

No obstante, ante la imposibilidad de migrar mi desarrollo local en Acquia Dev Desktop 2 al servidor habilitado a tales efectos por UOC, para poder utilizar la página, la opción recomendada es importarla a otro entorno local también con Acquia Dev Desktop 2.

Los sistemas operativos soportados para Acquia son:

macOS

- OS X 10.12 Sierra

- OS X 10.11 El Capitan

- OS X 10.10 Yosemite

- OS X 10.9 Mavericks

- OS X 10.8 Mountain Lion

- OS X 10.7 Lion

Windows (32- and 64-bit)

- Windows 10

- Windows 8.0 and 8.1

- Windows Vista

- Windows 7

El proceso de importación de los archivos locales a Acquia Dev Desktop 2 no requiere de ningún hardware especial ni de ningún conocimiento previo específico.

Se adjunta en el pie de página un enlace a la guía oficial de instalación (en inglés)²⁰, así como instrucciones para llevar a cabo la importación del archivo tras instalar Acquia.

17. Instrucciones de instalación

1.- Descargue el archivo comprimido :

<https://drive.google.com/drive/folders/1bUTdWMLUINUAGCQdXRqVK0VbUKHSS21T?usp=sharing>

2.- Descomprima los archivos en una carpeta, de modo que, una vez finalizado el proceso de descompresión, dentro de esa carpeta se vean: otra carpeta llamada “lightning_8_x_3_206”, un archivo .sql llamado “lightning_8_x_3_206”, y un .ini llamado “MANIFEST”

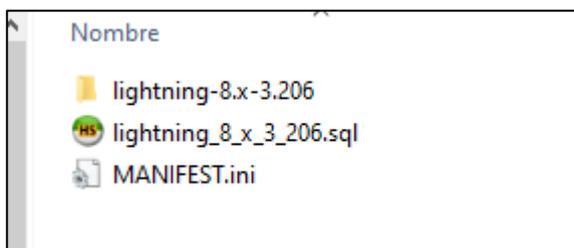


Figura 57 : Instalación Dev Desktop -descompresión de archivos

3.- Descargue Acquia Dev Desktop desde el siguiente enlace: <https://dev.acquia.com/downloads>

4.- Instale el programa siguiendo las indicaciones este enlace: <https://docs.acquia.com/dev-desktop/install/>.

5.- Ejecute el programa tras la instalación.

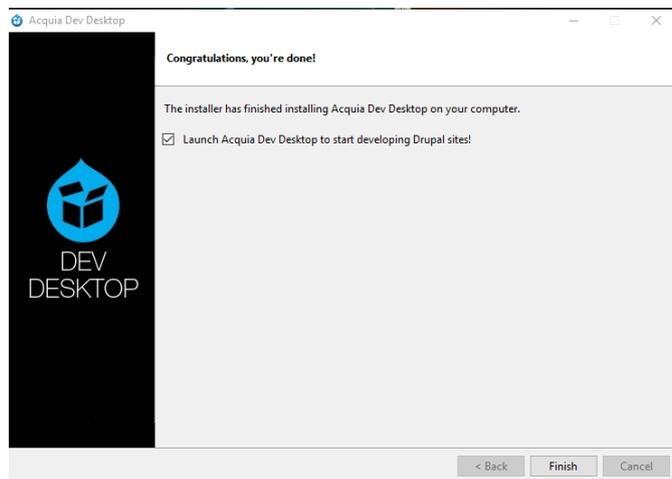


Figura 58 : Instalación Dev Desktop- finalización de instalación

20 Installing Acquia Dev Desktop — Acquia Product Documentation. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://docs.acquia.com/dev-desktop/install/>.

5.- En la esquina inferior izquierda de la ventana del programa haga clic sobre el símbolo “+”

7.- Seleccione la opción “Import local Drupal site”

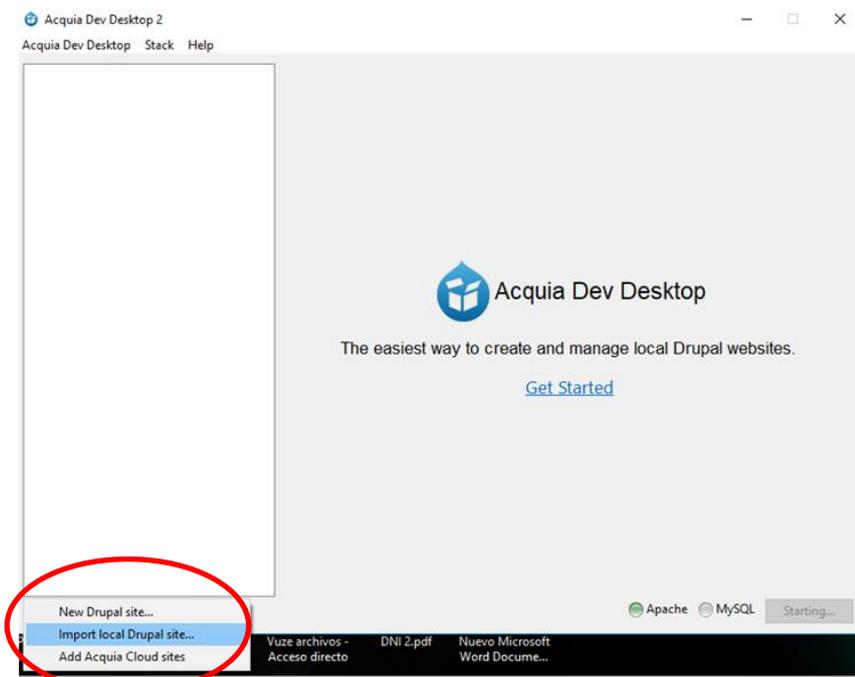


Figura 59 : Importación con Dev Desktop

8.- En la ventana que aparecerá, haga clic en el botón “Change” para seleccionar el directorio donde se encuentran los archivos locales. En este caso: lightning_8_x_3_206. Una vez localizado haga clic en “Seleccionar carpeta”

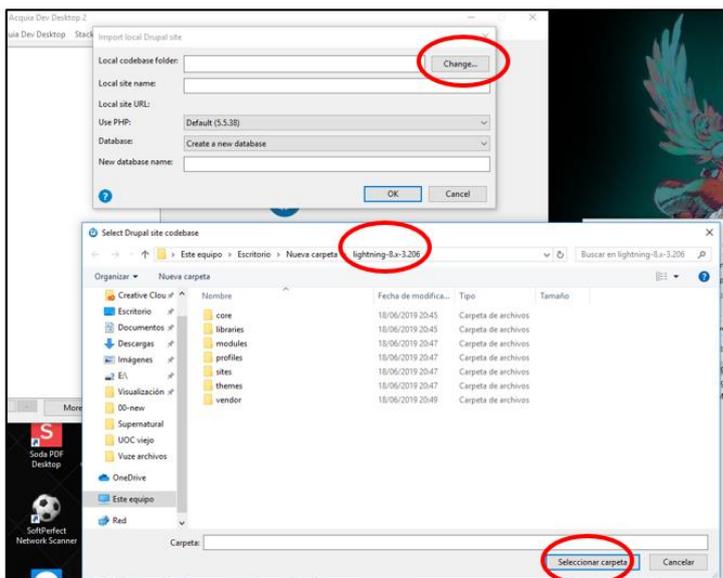


Figura 60 : Importación con Dev Desktop-selección de archivos

9.- En la sección “Database” seleccione la opción “Start with MySQL database dump file” y haga clic en “browse” para seleccionar el archivo de base de datos.

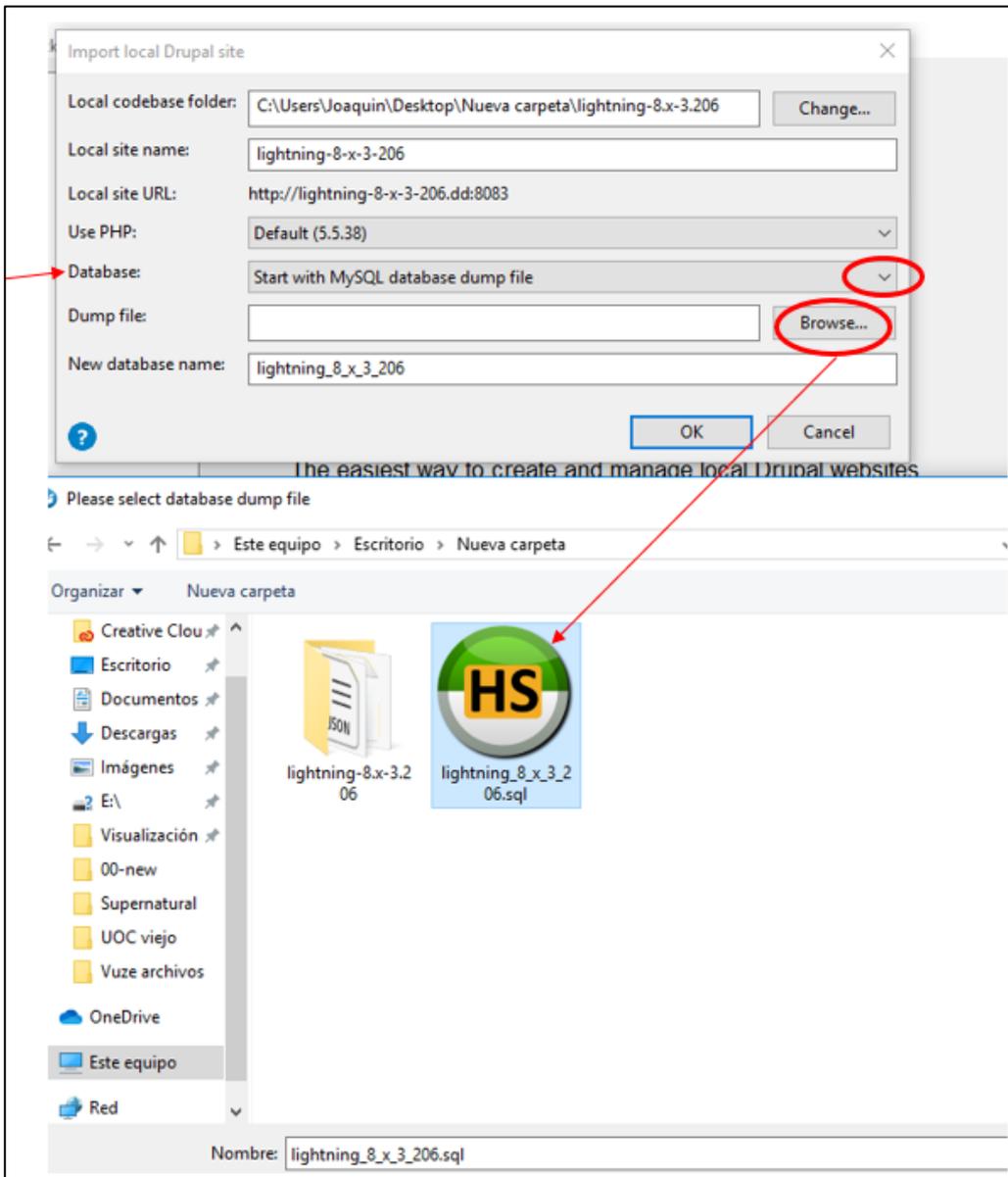


Figura 61 : Importación con Dev Desktop-selección de base de datos

9.- Por último, seleccione la última versión de PHP (7.2.13) y haga clic en “OK” para que comience la importación.

10.- Una vez finalizada la importación, podrá acceder a la página web desde el enlace etiquetado como “Local site:”

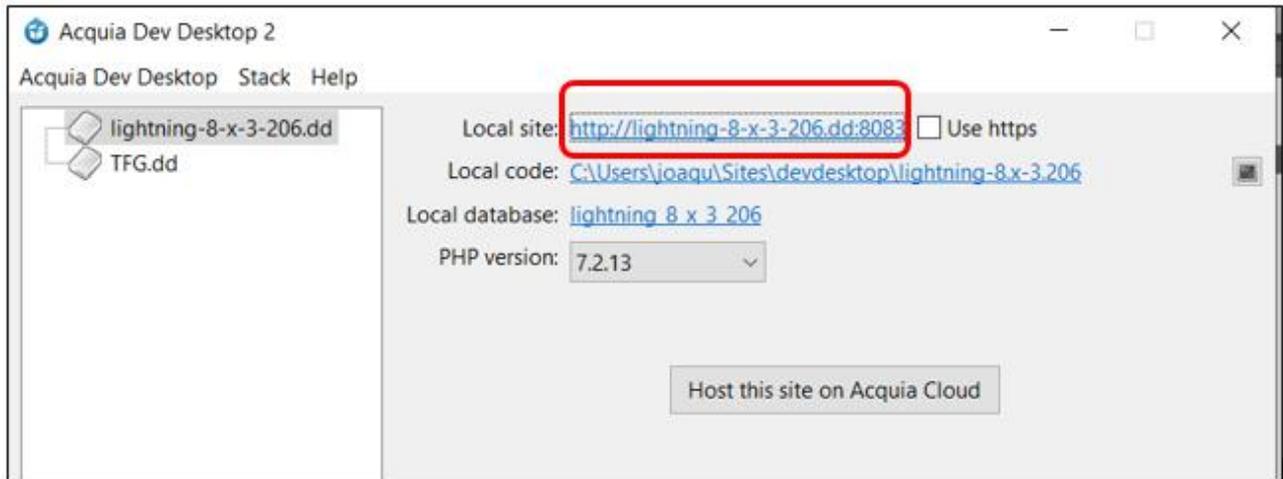


Figura 62 : Importación con Dev Desktop-selección de base de datos

18. Instrucciones de uso

Credenciales de usuario

Rol Administrador

Nombre: jgallegobe

Contraseña: qwertyuiop123@

Para facilitar las pruebas todos los demás usuarios utilizan la misma contraseña: **cacahue1**

Rol Organizer

Organizer1

Rol Participant

Participant2

Participant4

Rol Participant_plus

Participant1

Rol Participant_minus

Participant3

Rol The_Couple

Groom Test

Bride Test

Anexos

Annex 1. Entregables del proyecto

Anexo 2- Librerías y código externo utilizado

Instalación base

Lightning Drupal con Acquia Dev Desktop 2: <https://www.acquia.com/drupal/acquia-dev-desktop>

Módulos

Lightning Media https://www.drupal.org/project/lightning_media

Pathauto: <https://www.drupal.org/project/pathauto>

Devel: <https://www.drupal.org/project/devel>

AddToAny: <https://www.drupal.org/project/addtoany>

Like Button: <https://www.drupal.org/project/likebtn>

Dependencia: Voting API: <https://www.drupal.org/project/votingapi>

Easy Breadcrumb: https://www.drupal.org/project/easy_breadcrumb

Dropdown Language: https://www.drupal.org/project/dropdown_language

Simple Google Maps: https://www.drupal.org/project/simple_gmap

jQuery Countdown timer: https://www.drupal.org/project/jquery_countdown_timer

Field Permissions: https://www.drupal.org/project/field_permissions

Google Translate: <https://www.drupal.org/project/gtranslate>

Content Access: https://www.drupal.org/project/content_access

DrupalChat: <https://www.drupal.org/project/drupalchat>

Paragraphs: <https://www.drupal.org/project/paragraphs>

Entity Reference Revisions: https://www.drupal.org/project/entity_reference_revisions

Entity Usage: https://www.drupal.org/project/entity_usage

Admin Toolbar: https://www.drupal.org/project/admin_toolbar

Colorbox: <https://www.drupal.org/project/colorbox>

Webform: <https://www.drupal.org/project/webform>

Video: <https://www.drupal.org/project/video>

Google Analytics: https://www.drupal.org/project/google_analytics

Imagefield SlideShow: https://www.drupal.org/project/imagefield_slideshow

PhotoSwipe: <https://www.drupal.org/project/photoswipe>

Views Photo Grid: https://www.drupal.org/project/views_photo_grid

Image Effects: https://www.drupal.org/project/image_effects

Bootstrap Paragraphs: https://www.drupal.org/project/bootstrap_paragraphs
Views Reference Field: <https://www.drupal.org/project/viewsreference>
Field Timer: https://www.drupal.org/project/field_timer
Date Popup: https://www.drupal.org/project/date_popup
Geofield: <https://www.drupal.org/project/Geofield>
Styled Google Map: https://www.drupal.org/project/styled_google_map
Libraries: <https://www.drupal.org/project/libraries>
Extrafield views integration: https://www.drupal.org/project/Extrafield_views_integration
Like It: <https://www.drupal.org/project/likeit>
User protect: <https://www.drupal.org/project/userprotect>
Role delegation: https://www.drupal.org/project/role_delegation

CKEditor Plugins

CKEditor Upload Image: https://www.drupal.org/project/ckeditor_uploadimage
CKEditor Panel Button: <https://www.drupal.org/project/panelbutton>
CKEditor Color Button: <https://www.drupal.org/project/colorbutton>
CKEditor Ballon Panel: https://www.drupal.org/project/ckeditor_balloonpanel
CKEditor Accessibility Checker: https://www.drupal.org/project/ckeditor_a11ychecker
CKEditor Emojione: https://www.drupal.org/project/ckeditor_emojione
CKEditor List Style: https://www.drupal.org/project/ckeditor_liststyle
CKEditor FakeObjects: <https://www.drupal.org/project/fakeobjects>
CKEditor Anchor Link: https://www.drupal.org/project/anchor_link

Tema Base

Bootstrap : <https://www.drupal.org/project/bootstrap>

Anexo 3. Bibliografía

(1) Why do people think Drupal is better than WordPress for scalability? - Quora. [en línea], [sin fecha]. [Consulta: 3 mayo 2019]. Disponible en: <https://www.quora.com/Why-do-people-think-Drupal-is-better-than-WordPress-for-scalability>.
BEM and SMACSS: Advice From Developers Who've Been There. *SitePoint* [en línea], 2015. [Consulta: 28 abril 2019]. Disponible en: <https://www.sitepoint.com/bem-smacss-advice-from-developers/>.
BEM y SMACSS en Drupal 8. *Softwin Perú* [en línea], [sin fecha]. [Consulta: 28 abril 2019]. Disponible en: <http://softwinperu.com/blog/agustin-rojas-silva/bem-y-smacss-en-drupal-8>.
CSS architecture (for Drupal 8) | Develop guide on Drupal.org. [en línea], [sin fecha]. [Consulta: 4 mayo 2019]. Disponible en: <https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8>.
Drupal - Open Source CMS. *Drupal.org* [en línea], 2018. [Consulta: 2 mayo 2019]. Disponible en: <https://www.drupal.org/home>.

DRUPAL ASSOCIATION, [sin fecha]. *DrupalCon Baltimore 2017: Bootstrap Paragraphs* [en línea]. [Consulta: 30 abril 2019]. Disponible en: <https://www.youtube.com/watch?v=kqfvH3OstSE&feature=youtu.be&t=1726>.

DRUPAL UP - DRUPAL 8 VIDEO TUTORIALS, [sin fecha]. *Theming the first paragraph*  *Real Life Drupal 8 Theming Project Episode 12* [en línea]. [Consulta: 28 abril 2019]. Disponible en: <https://www.youtube.com/watch?v=MeVFrt7BUyw>.

Home - Scalable and Modular Architecture for CSS. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <http://smacss.com/>.

Joomla Content Management System (CMS) - try it for free! *Joomla!* [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <https://www.joomla.org/>.

Methodology / BEM. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <https://en.bem.info/methodology/>.

MORENO, B., 2019. Drupal 8, el CMS más escalable. *Blog de Hiberus Tecnología* [en línea]. [Consulta: 3 mayo 2019]. Disponible en: <https://www.hiberus.com/crecemos-contigo/drupal-8-cms-mas-escalable/>.

Naming convention / Methodology / BEM. [en línea], [sin fecha]. [Consulta: 5 mayo 2019]. Disponible en: <https://en.bem.info/methodology/naming-convention/#two-dashes-style>.

Pico - A stupidly simple, blazing fast, flat file CMS. [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <http://picocms.org/>.

WEBWASH, [sin fecha]. *Introduction to the Paragraphs Module in Drupal 8* [en línea]. [Consulta: 28 abril 2019]. Disponible en: <https://www.youtube.com/watch?v=Fy9s01T30uo>.

WordPress vs. Drupal? Which is better. [en línea], [sin fecha]. [Consulta: 3 mayo 2019]. Disponible en: <https://www.commonplaces.com/blog/wordpress-drupal>.

WordPress.com: crea un sitio web o blog gratuito. *WordPress.com* [en línea], [sin fecha]. [Consulta: 2 mayo 2019]. Disponible en: <https://es.wordpress.com/>.

(1) Introduction to the Paragraphs Module in Drupal 8 - YouTube. [en línea], [sin fecha]. [Consulta: 19 junio 2019]. Disponible en: <https://www.youtube.com/watch?v=Fy9s01T30uo>.

Bootstrap 3 Tutorial. [en línea], [sin fecha]. [Consulta: 19 junio 2019]. Disponible en: <https://www.w3schools.com/bootstrap/default.asp>.

Burgundy Meaning, Combinations and Hex Code - Canva Colors. *Canva's Design Wiki* [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://www.canva.com/colors/color-meanings/burgundy/>.

Clippy — CSS clip-path maker. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://bennettfeely.com/clippy/>.

CONTRIBUTORS, M.O., Jacob Thornton, and Bootstrap, [sin fecha]. Buttons. [en línea]. [Consulta: 18 junio 2019]. Disponible en: <https://getbootstrap.com/docs/4.0/components/buttons/>.

CSS object-fit Property. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: https://www.w3schools.com/css/css3_object-fit.asp.

Generador Lorem Ipsum. *Generador Lorem Ipsum* [en línea], [sin fecha]. [Consulta: 19 junio 2019]. Disponible en: <https://getlorem.com/es>.

Google Fonts. *Google Fonts* [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://fonts.google.com/>.

Installing Acquia Dev Desktop — Acquia Product Documentation. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://docs.acquia.com/dev-desktop/install/>.

Like iOS Glyph Icons. [en línea], [sin fecha]. [Consulta: 17 junio 2019]. Disponible en: <https://icons8.com/icon/set/like/ios-glyphs>.

MOLICH, R. y NIELSEN, J., 1990a. Improving a H&man- Computer Dialogue. *Communications of the ACM*, vol. 33, no. 3, pp. 11.

NIELSEN, J., 2006. Ten Usability Heuristics. . S.l.: s.n.,

Stack Overflow - Where Developers Learn, Share, & Build Careers. *Stack Overflow* [en línea], [sin fecha]. [Consulta: 19 junio 2019]. Disponible en: <https://stackoverflow.com/>.

Teal Meaning, Combinations and Hex Code - Canva Colors. *Canva's Design Wiki* [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://www.canva.com/colors/color-meanings/teal/>.

The Ultimate Tutorial for Drupal's Paragraphs Module. [en línea], [sin fecha]. [Consulta: 19 junio 2019]. Disponible en: <https://www.ostraining.com/blog/drupal/paragraphs-module/>.

uiGradients - Beautiful colored gradients. [en línea], [sin fecha]. [Consulta: 18 junio 2019]. Disponible en: <https://uigradients.com/#Limeade>.

Creación de Página Web para Eventos - Tu Fábrica de Eventos. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <http://www.tufabricadeeventos.com/es/soluciones/crea-tu-web/>.

Plantillas Web HTML de Bodas y Fiestas | Wix. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <https://es.wix.com/website/templates/html/events/weddings-celebrations>.

Plantillas Web HTML de Eventos | Wix. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <https://es.wix.com/website/templates/html/events>.

Web de Boda Gratis. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <https://www.zankyou.es/web-de-bodas>.

Web De Mi Boda. *Web De Mi Boda* [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <http://www.webdemiboda.com/>.

Wedding Websites - Free Wedding Websites - The Knot. [en línea], [sin fecha]. [Consulta: 11 marzo 2019]. Disponible en: <https://www.theknot.com/gs/wedding-websites>.