

BOLITA

**desarrollo de una aplicación web para la
gestión de un diario privado de la infancia
de un niño**

Autor: Patricia Gende Rial

Tutor: Kenneth Capseta Nieto

Profesor: Carlos Casado Martínez

Grado de Multimedia

Desarrollo de aplicaciones interactivas

07/01/2019

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento – NoComercial - SinObraDerivada

[3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>BOLITA: desarrollo de una aplicación web para la gestión de un diario privado de la infancia de un niño</i>
Nombre del autor:	<i>Patricia Gende Rial</i>
Nombre del colaborador/a docente:	<i>Kenneth Capseta Nieto</i>
Nombre del PRA:	<i>Carlos Casado Martinez</i>
Fecha de entrega (mm/aaaa):	<i>Enero/2019</i>
Titulación o programa:	<i>Grado de Multimedia</i>
Área del Trabajo Final:	<i>Desarrollo de aplicaciones interactivas</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Aplicación web, Drupal, CMS</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Creación de una aplicación web en la cual usuarios anónimos podrán crear una cuenta de usuario que les permitirá gestionar un diario de la infancia de un niño/a, a través del cual tendrán la posibilidad de generar una lista de entradas en orden cronológico (timeline) o un álbum de presentación.</p> <p>Así un usuario, los padres de un crío, crearán una cuenta privada en el portal y una vez validada dicha cuenta, tendrán permisos para crear ciertos contenidos dentro del portal. Estos contenidos serán privados y sólo con su cuenta serán visibles. De esta forma generarán entradas en el diario con texto enriquecido y a las cuales podrán adjuntar vídeos y fotos, pudiendo además categorizarlas por temática (primeras veces, revisiones, vacunas, viajes, momentos con familia, etc.).</p> <p>La aplicación permitirá realizar una búsqueda de entradas por una serie de filtros. Así el usuario podrá localizar las entradas de un período de tiempo determinado o categorizadas por una determinada temática (como puede ser por ejemplo todas las entradas asociadas a las vacunas o a viajes). Así como generar a partir de estas búsquedas un timeline gráfico o también un álbum tipo presentación, de manera visual, con los vídeos y fotos adjuntos a las entradas asociadas.</p>	

Abstract (in English, 250 words or less):

Creation of a web application in which anonymous users will be able to create a user account that will allow them to manage a diary of a child's infancy, through which they will have the possibility of generating a list of entries in chronological order (timeline) or a presentation album.

Then a user, the parents of a child, will create a private account on the portal and once validated this account, will have permission to create certain content inside the portal. These contents will be private and only with your account will be visible. In this way they will generate entries in the diary with enriched text and to which they will be able to attach videos and photos, being able also to categorize them by thematic (first times, revisions, vaccines, trips, moments with family, etc.).

The application will allow a search of entries by a series of filters. This way the user will be able to locate the entries of a determined period or categorized by a determined thematic (as it can be for example all the entries associated to the vaccines or to trips). As well as to generate from these searches a graphic timeline or also a presentation-type album, in a visual way, with the videos and photos attached to the associated entries.

Dedicatoria/Cita

Dedico este trabajo a la hija de un gran amigo, a “mi Bolita”, que así le llamábamos mientras era un garbancito en la barriga de su mami y no tenía nombre. La idea de este proyecto salió gracias a ti Julia.

Agradecimientos

Mi más sincero agradecimiento a mi familia y amigos por la paciencia que han tenido durante la realización de este trabajo final de grado. En especial a mi gran amigo Juan Pablo, que gracias a él obtuve la idea y por el apoyo que me ofreció en el día a día, dándome ánimos a continuar cada vez que me encontraba con un problema. Y sobre todo a mi novio, que fue el que vivió más el día a día mis desesperaciones en la realización de este Trabajo Final de Grado (TFG).

Abstract

Creation of a web application in which anonymous users will be able to create a user account that will allow them to manage a diary of a child's infancy, through which they will have the possibility of generating a list of entries in chronological order (timeline) or a presentation album.

Then a user, the parents of a child, will create a private account on the portal and once validated this account, will have permission to create certain content inside the portal. These contents will be private and only with your account will be visible. In this way they will generate entries in the diary with enriched text and to which they will be able to attach videos and photos, being able also to categorize them by thematic (first times, revisions, vaccines, trips, moments with family, etc.).

The application will allow a search of entries by a series of filters. This way the user will be able to locate the entries of a determined period or categorized by a determined thematic (as it can be for example all the entries associated to the vaccines or to trips). As well as to generate from these searches a graphic timeline or also a presentation-type album, in a visual way, with the videos and photos attached to the associated entries.

Resumen

Creación de una aplicación web en la cual usuarios anónimos podrán crear una cuenta de usuario que les permitirá gestionar un diario de la infancia de un niño/a, a través del cual tendrán la posibilidad de generar una lista de entradas en orden cronológico (timeline) o un álbum de presentación.

Así un usuario, los padres de un crío, crearán una cuenta privada en el portal y una vez validada dicha cuenta, tendrán permisos para crear ciertos contenidos dentro del portal. Estos contenidos serán privados y sólo con su cuenta serán visibles. De esta forma generarán entradas en el diario con texto enriquecido y a las cuales podrán adjuntar vídeos y fotos, pudiendo además categorizarlas por temática (primeras veces, revisiones, vacunas, viajes, momentos con familia, etc.).

La aplicación permitirá realizar una búsqueda de entradas por una serie de filtros. Así el usuario podrá localizar las entradas de un período de tiempo determinado o categorizadas por una determinada temática (como puede ser por ejemplo todas las entradas asociadas a las vacunas o a viajes). Así como generar a partir de estas búsquedas un timeline gráfico o también un álbum tipo presentación, de manera visual, con los vídeos y fotos adjuntos a las entradas asociadas.

Palabras clave

Aplicación web, Drupal, CMS, Trabajo Final de Grado, Aplicaciones interactivas, Desarrollo aplicación

Notaciones y Convenciones

Los estilos tipográficos usados en esta memoria son los siguientes:

	Estilo	Características
Título	Título	Fuente: Arial, 24 puntos, negrita Alineación: justificada Estilo siguiente: Normal
Subtítulo	---	Fuente: Arial, 20 puntos, negrita Alineación: justificada Estilo siguiente: Normal
Sección de subtítulo	Non-chapter section heading	Fuente: Arial, 12 puntos, negrita Alineación: justificada Estilo siguiente: Normal
Capítulo	Título 1	Fuente: Arial, 20 puntos, negrita Esquema numerado nivel 1 Alineación: izquierda Estilo siguiente: Normal
Sección	Título 2	Fuente: Arial, 13 puntos, negrita Esquema numerado nivel 2 Alineación: izquierda Estilo siguiente: Normal
Sub-sección	Título 3	Fuente: Arial, 13 puntos, negrita Esquema numerado nivel 3 Alineación: izquierda Estilo siguiente: Normal
Texto general	Normal	Fuente: Arial, 10 puntos Alineación: justificada Estilo siguiente: Normal
Título figura / tabla	Descripción	Fuente: Arial, 8 puntos Alineación: centrada

		Estilo siguiente: Normal
Listado	Párrafo de lista	Fuente: Arial, 10 puntos Alineación: justificada Estilo siguiente: Normal
Nota al pie	---	Fuente: Arial, 9 puntos Alineación: izquierda

Índice

1. Introducción.....	13
1.1. Introducción.....	13
1.2. Descripción.....	14
1.3. Objetivos generales.....	16
1.3.1. Objetivos principales.....	16
1.3.2. Objetivos secundarios.....	16
1.4. Metodología y proceso de trabajo.....	17
1.5. Planificación.....	18
1.6. Presupuesto.....	22
1.7. Estructura del resto del documento.....	24
2. Análisis de mercado.....	25
2.1. Público objetivo y perfiles de usuario.....	25
2.2. Competencia / Antecedentes.....	25
2.3. Análisis DAFO.....	26
3. Propuesta.....	27
3.1. Definición de objetivos / especificaciones del producto.....	27
4. Diseño.....	28
4.1. Arquitectura general de la aplicación.....	28
4.2. Arquitectura de la información y diagramas de navegación.....	28
4.3. Diseño gráfico e interfaces.....	29
4.3.1. Estilos.....	29
4.3.2. Usabilidad /UX.....	29
4.4. Lenguajes de programación y APIs utilizados.....	30
4.4.1. Módulos y temas contribuidos.....	31
4.4.2. Módulos y temas custom.....	33
4.5. Requisitos de instalación.....	40
4.6. Instrucciones de instalación.....	40

4.6.1. Código fuente.....	40
4.6.2. Base de datos	41
4.6.3. Configuraciones.....	41
5. Demostración	42
5.1. Prototipos.....	42
5.1.1. Prototipos Lo-Fi.....	42
5.2. Tests.....	42
5.3. Ejemplos de uso del producto (o guía de usuario).....	42
6. Conclusiones y líneas de futuro	44
6.1. Conclusiones	44
6.2. Líneas de futuro.....	45
Bibliografía.....	47
Anexos.....	48

Figuras y tablas

Índice de figuras

Figura 1: Diagrama de Gantt	21
Figura 2: código del archivo “paragraph--section-images--default.html.twig”	33
Figura 3: código del archivo “bolita_common.module”	34
Figura 4: código del archivo “bolita_common.module”	34
Figura 5: código del archivo “bolita_controlpanel.module”	35
Figura 6: código del archivo “ <i>bolita_controlpanel.links.menu.yml</i> ”	36
Figura 7: código del archivo “ <i>global.styles.css</i> ”	37
Figura 8: código del archivo “ <i>html.html.twig</i> ”	38
Figura 9: logo creado para la aplicación web	38
Figura 10: icono con el favicon para la aplicación web	39
Figura 11: página de configuración de la apariencia del portal	39
Figura 12: Propuesta de diseño para la portada	49
Figura 13: Propuesta de diseño para una ficha de página básica	50
Figura 14: Propuesta de diseño para una ficha de entrada del diario	51

Índice de tablas

Tabla 1: Definición de sprint.....	18
Tabla 2: Tareas del sprint 1	18
Tabla 3: Tareas del sprint 2	19
Tabla 4: Tareas del sprint 3	20
Tabla 5: Tareas del sprint final.....	21
Tabla 6: Estimación del esfuerzo total en horas	22
Tabla 7: Análisis DAFO.....	26

1.Introducción

1.1. Introducción

La aplicación web BOLITA nace de una propuesta funcional aportada por futuros usuarios que ven en el mercado una necesidad. Actualmente para los padres es importante guardar todos los recuerdos posibles del nacimiento y crecimiento de sus hijos. BOLITA pretende que de una forma sencilla puedan almacenar en un diario interactivo todos los recuerdos asociados a la vida de su hijo y poder consultarlos fácilmente. Aunque parece que estos recuerdos nunca se van a olvidar, con el paso del tiempo y en la sociedad estresada en la que vivimos, terminan por desaparecer de la memoria.

Con este diario lo que se pretende es que los padres, a partir de ahora usuarios de la aplicación, tengan un diario de la infancia de sus hijos con una serie de funcionalidades: localizar todas las visitas médicas o las de un período concreto, localizar las primeras veces que el niño hizo algo, localizar entradas relacionadas con viajes... además de crear de forma interactiva presentaciones de fotos aleatorias de estas entradas y generar una lista de entradas en orden cronológico.

Así un usuario se dará de alta en la aplicación web y de esta forma tendrá acceso a un panel de control privado en el cual podrá gestionar todas las entradas al diario de su hijo. En un primer momento, esta gestión será gratuita, aunque se pretende que en futuras versiones tenga un pequeño coste mensual, que podrá ser pagado en cuotas anuales.

Los usuarios generarán entradas en el diario de su niño a las cuales adjuntarán contenido multimedia: fotos, vídeos y audios. Con este contenido multimedia se generarán presentaciones o timelines.

1.2. Descripción

La propuesta funcional de este proyecto parte de una necesidad detectada en el mercado por un grupo de usuarios. Se busca la creación de un diario de la infancia de un niño, que sea gestionado por un usuario registrado y cuyo contenido sea privado.

Así un usuario anónimo sólo tendrá acceso a los contenidos públicos del portal, principalmente informativos. Mientras que un usuario registrado, visualizará un panel de control a través del cual podrá gestionar entradas en el diario, creando nuevas o editando las existentes. Además, tendrá permisos para visualizar listados de sus entradas y generar timelines y presentaciones con los contenidos multimedia de dichas entradas.

La idea principal de este proyecto es en una primera versión facilitar al usuario la creación de una cuenta privada y la gestión de un único diario asociado a la misma. En futuras versiones, se añadirá el cobro mensual del uso de la cuenta y se analizará la posibilidad de crear más de un diario por usuario.

La propuesta técnica se basa en el desarrollo de una nueva aplicación web partiendo de Drupal. Drupal es un CMS (*Content Management System*) utilizado en la creación de aplicaciones web y sitios web dinámicos, altamente configurable, escalable y versátil, y que además aporta una alta disponibilidad para su personalización.

Para la creación de esta nueva aplicación se opta por la última versión de este CMS, la versión 8, por tratarse de un producto ya estable y que ofrece numerosas ventajas frente a su versión anterior:

- Un gran número de funcionalidades se han integrado completamente en el CORE de esta versión 8. Así como un editor WYSIWYG.
- Las pantallas de creación y edición de contenidos poseen un nuevo diseño adaptado a mejorar la usabilidad.
- Drupal 8 ha sido diseñado pensando primero en la versión móvil. Así, por ejemplo, la barra de herramientas está pensada para que se adapte y sea usable en dispositivos móviles. De la misma forma que los formularios de creación y edición de contenidos.
- Esta versión está basada en Symfony 2, así Drupal puede hacer uso de muchos de los componentes: Routing, Yaml, Twig, DependencyInjection, Serializer, Validator...

Como características fundamentales de DRUPAL, como plataforma, cabe destacar:

- Se trata de un CMS cuyo desarrollo está basado en **software libre**.
- La arquitectura del sistema, y la existencia de un grupo de desarrollo específico para controlar la seguridad, dotan a esta plataforma de un **alto nivel de seguridad**.
- Todos los **contenidos y configuraciones** se almacenan y editan de forma **dinámica** desde la aplicación, almacenándolos en una base de datos, en este caso MySQL.

- El código propio de Drupal cumple los **estándares web** XHTML y CSS para garantizar su correcta visualización en todos los navegadores, en sus últimas versiones.
- Drupal es el gestor de contenidos que más reconocimiento tiene por parte de los programadores por su **estabilidad** y por la estructuración interna de sus componentes.
- El **diseño modular** de Drupal permite trabajar en el desarrollo de nuevas funcionalidades sin afectar al funcionamiento básico del sistema.
- Mucha de la funcionalidad que se necesita en la aplicación puede obtenerse mediante la **integración de módulos** y la customización de pequeñas adaptaciones o nuevas funcionalidades.
- Permite la **configuración** específica de múltiples parámetros e incluso la activación y desactivación de módulos y funcionalidades **en caliente**, consiguiendo de esta forma una adaptación a las necesidades de cada momento.
- Drupal está diseñado para ser **multiplataforma**. Pudiendo funcionar con Apache o Microsoft IIS como servidor web, y en sistemas como Linux, Windows, MAC OS... Además, al estar implementado en **PHP**, es totalmente portable. Además, por su diseño, es posible la ampliación de la plataforma sobre la que se ejecuta de un modo simple, con lo cual el sistema puede **escalarsse** para adaptarse fácilmente a nuevas necesidades.
- La plataforma está perfeccionada para garantizar un **alto rendimiento**.
- Cuenta con una gran comunidad. De esta manera, se puede asegurar tanto mejoras de la herramienta como **soporte** para problemas que puedan surgir.

Para el diseño se hará uso de las tecnologías más modernas de desarrollo web, que proporcionan una apariencia visual actual, muy atractiva e intuitiva (HTML 5, CSS3, ...). Asimismo, se contemplarán las características propias de la web 2.0 para generar un portal amigable y funcional.

Se implementará un framework desarrollado en código abierto, Bootstrap, que tiene como objetivo facilitar el diseño web. Así, permite una maquetación sencilla de diseño adaptable y responsivo.

1.3. Objetivos generales

1.3.1. Objetivos principales

El objetivo principal de este proyecto es el desarrollo de una aplicación web que sirva como herramienta de gestión del diario de un niño a los padres. Para ello se hará uso de un gestor de contenidos que permite implementar esta herramienta de una forma más robusta y sencilla. Por lo tanto, la base de la aplicación será un DRUPAL 8.

Los objetivos a los que se pretende dar cabida para los usuarios son:

- Visualizar una serie de información pública para todos los usuarios.
- Registrarse en el gestor de contenidos, creando una cuenta de usuario.
- Acceder con su cuenta de usuario a una parte interna.
- Permitir al usuario gestionar las entradas del diario de su niño (creación, modificación y eliminación).
- Permitir al usuario crear una lista de entradas en orden cronológico (timeline).
- Permitir al usuario crear un álbum de presentación.

Como objetivos personales puedo mencionar los siguientes:

- Cubrir una necesidad detectada para un público adaptado a las nuevas tecnologías.
- Añadir conocimiento de Drupal 8, así como de algunos módulos contribuidos y aventurarse en el desarrollo de funcionalidad customizada: creación de plantillas de visualización propias para ciertas entidades y alteración en el envío de los datos a las mismas, creación de un panel de control personalizado para la parte privada que es incluido dentro de la barra de herramientas administrativa.
- Gestionar el proyecto, así como realizar trabajos tanto de desarrollo como de maquetación y diseño.

1.3.2. Objetivos secundarios

El objetivo secundario principal es alcanzar un gran número de usuarios para sacar una segunda versión de pago. Permitiendo así en esta primera versión a los usuarios conocer la aplicación y probarla de forma gratuita. Añadiendo en versiones posteriores tanto el pago de la cuota mensual, como añadir nuevas funcionalidades a la aplicación como puede ser la creación de más de un diario por cuenta, presentaciones del contenido multimedia más complejas, permitir compartir contenidos o presentaciones, ...

1.4. Metodología y proceso de trabajo

La metodología de desarrollo de aplicaciones propuesta para este proyecto es una metodología ágil basada en Scrum que permite abordar todas las fases de desarrollo de la aplicación web.

Así se define una serie de entregas parciales que permiten un seguimiento idóneo del proyecto. El proceso parte de una lista de tareas repartidas en iteraciones y entregas.

Para llevar a cabo esta metodología Scrum se seguirán una serie de actividades:

- Planificación de cada iteración (sprint):
 - A partir de los requisitos de la aplicación se elabora una lista de tareas necesarias para poder abordarlos. Se realiza una estimación del trabajo a realizar y se dividen las tareas en cada sprint.
- Ejecución de la iteración:
 - Se abordará cada una de las tareas definidas resolviendo problemáticas y dudas con el equipo docente (Scrum manager).
- Revisión:
 - El último día de la iteración se realiza la entrega prevista, asumiendo en algunas de ellas una demostración del trabajo realizado.
 - El cliente (equipo docente) analiza que se cumplieran todos los requisitos esperados.

Como herramienta de gestión de las tareas de desarrollo utilizo Redmine en su versión easyREDMINE.

1.5. Planificación

La planificación del proyecto se divide en 4 sprint, cada uno de los cuales lleva asociada una entrega. A continuación, se detalla la fecha de inicio y fin de cada uno, junto a las horas estimadas de las tareas incluidas en cada sprint.

		Fecha inicio	Fecha fin	Horas estimadas
Sprint	Sprint 1	19/09/2018	02/10/2018	41
	Sprint 2	03/10/2018	31/10/2018	81
	Sprint 3	01/11/2018	10/12/2018	124
	Sprint Final	11/12/2018	07/01/2019	66
TOTAL				312

Tabla 1: Definición de sprint

Para la primera versión de la aplicación se definen una serie de tareas asignadas cada una de ellas a un sprint, estimando la fecha de inicio y fin y el tiempo estimado de realización:

Sprint 1			
Tareas	Fecha inicio	Fecha fin	Horas estimadas
Análisis funcional	19/09/2018	27/09/2018	24
Documentación: Memoria I	27/09/2018	02/10/2018	16
Entrega: PEC 1	02/10/2018	02/10/2018	1
TOTAL			41

Tabla 2: Tareas del sprint 1

Sprint 2			
Tareas	Fecha inicio	Fecha fin	Horas estimadas

Montaje de servidor	03/10/2018	04/10/2018	4
Página básica: Crear el tipo de contenido	08/10/2018	08/10/2018	1
Instalación de DRUPAL	05/10/2018	08/10/2018	4
Página básica: Crear contenido	08/10/2018	09/10/2018	1
Creación de la taxonomía	08/10/2018	09/10/2018	1
Entrada del diario: Crear el tipo de contenido	10/10/2018	10/10/2018	1
Creación de roles	10/10/2018	12/10/2018	1
Creación de perfiles de usuario	10/10/2018	12/10/2018	2
Menú principal	12/10/2018	13/10/2018	1
Menú del pie de página	12/10/2018	13/10/2018	1
Portada: Elementos comunes	15/10/2018	16/10/2018	2
Buscador de entradas	29/10/2018	30/10/2018	1
Timeline	29/10/2018	30/10/2018	4
Documentación: Memoria II	03/10/2018	31/10/2018	16
Configuraciones de DRUPAL	07/10/2018	31/10/2018	8
Creación del diseño de la portada	08/10/2018	31/10/2018	8
Creación del diseño de una página básica	08/10/2018	31/10/2018	4
Creación del diseño de una entrada	08/10/2018	31/10/2018	4
Creación del tema base	15/10/2018	31/10/2018	16
Entrega: PEC 2	31/10/2018	31/10/2018	1
TOTAL			81

Tabla 3: Tareas del sprint 2

Sprint 3			
Tareas	Fecha inicio	Fecha fin	Horas estimadas

Redirección de inicio de sesión	01/11/2018	03/11/2018	1
Asignación de permisos	01/11/2018	10/11/2018	2
Menú del panel de control	01/11/2018	10/11/2018	1
Menú de búsquedas predefinidas	01/11/2018	10/11/2018	1
Elementos de la portada	01/11/2018	10/11/2018	4
Desarrollo del modo presentación	01/11/2018	17/11/2018	16
Página básica: Maquetar	12/11/2018	08/12/2018	8
Entrada del diario: Maquetar	12/11/2018	08/12/2018	16
Formulario de inicio de sesión / registro: Maquetar	12/11/2018	08/12/2018	10
Portada: Maquetar	12/11/2018	08/12/2018	12
Tema base: Maquetar	12/11/2018	08/12/2018	8
Maquetar buscador de entradas	12/11/2018	08/12/2018	4
Maquetar timeline	12/11/2018	08/12/2018	16
Documentación: Memoria III	01/11/2018	10/12/2018	16
Vídeo presentación I	06/12/2018	10/12/2018	8
Entrega: PEC 3	10/12/2018	10/12/2018	1
TOTAL			124

Tabla 4: Tareas del sprint 3

Sprint Final			
Tareas	Fecha inicio	Fecha fin	Horas estimadas
Maquetar presentación	11/12/2018	22/12/2018	24
Maquetar página inicial del panel de control	11/12/2018	22/12/2018	4
Documentación: Memoria final	11/12/2018	07/01/2019	16
Vídeo presentación II	24/12/2018	07/01/2019	12

Presentación formato libre	24/12/2018	07/01/2019	8
Entrega: Entrega final	07/01/2019	07/01/2019	2
TOTAL			66

Tabla 5: Tareas del sprint final

Adjunto el diagrama de Gantt con la estimación de las tareas realizadas:

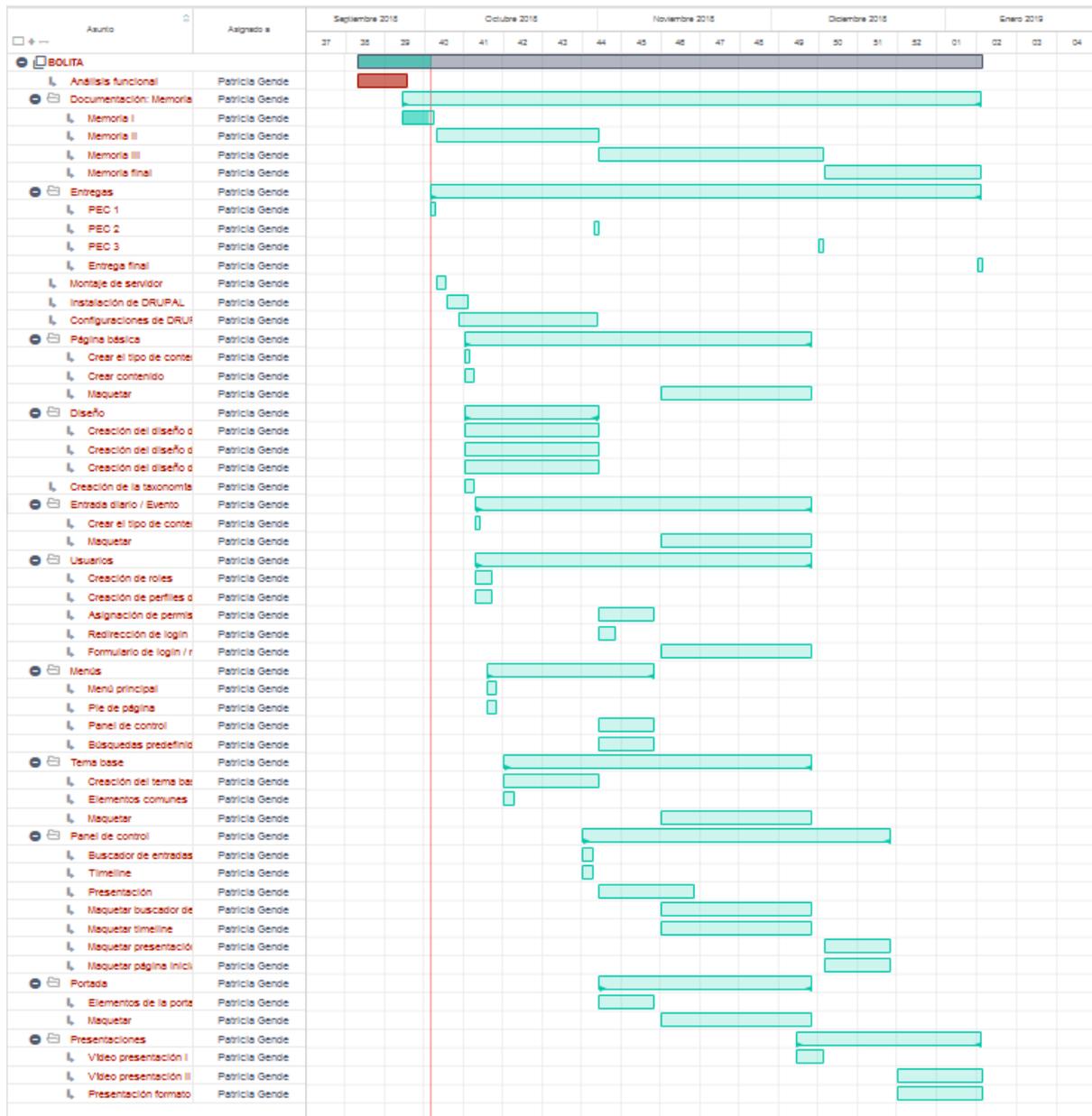


Figura 1: Diagrama de Gantt

1.6. Presupuesto

Una vez elaborada la planificación de este proyecto, se realiza una estimación total del esfuerzo para cada perfil identificado para las fases de desarrollo del proyecto.:

		Perfil	N.º horas	TOTAL
Equipos	Equipo de control y gestión	Gestor del proyecto	64 horas	64 horas
	Equipo funcional y de arquitectura	Analista funcional	24 horas	76 horas
		Diseñador gráfico	16 horas (diseño) 28 horas (vídeos)	
		Administrador de sistemas	4 horas (servidor) 4 horas (Drupal)	
	Equipo de desarrollo, implantación y pruebas	Analista programador	80 horas	172 horas
		Maquetador	92 horas	
TOTAL				312 horas

Tabla 6: Estimación del esfuerzo total en horas

Aunque este proyecto va a ser desarrollado por una sola persona que será la encargada de todos los perfiles, se realiza un presupuesto en función de diferentes precios por perfil.

		Perfil	N.º horas	Precio hora	Precio total
Equipos	Equipo de control y gestión	Gestor del proyecto	64 horas	50 €/hora	3200 €
	Equipo funcional y de arquitectura	Analista funcional	24 horas	40 €/hora	960 €

		Diseñador gráfico	44 horas	25 €/hora	1100 €
		Administrador de sistemas	8 horas	35 €/hora	280 €
	Equipo de desarrollo, implantación y pruebas	Analista programador	80 horas	30 €/hora	2400 €
		Maquetador	92 horas	30 €/hora	2760 €
TOTAL					10700 €

1.7. Estructura del resto del documento

En los siguientes capítulos se detallará todos los pasos realizados durante la ejecución de este proyecto.

En el capítulo 2 se presentará la fase de **análisis de la situación actual del mercado**. Para ello se considerará el público objetivo, así como la competencia que existe actualmente.

En el capítulo 3 se presentará la fase de **análisis funcional**. Se definirán los objetivos y especificaciones que requerirá la aplicación web.

En el capítulo 4 se presentará la fase de **análisis de arquitectura y diseño**. Se especificará la arquitectura requerida para la aplicación, así como el análisis realizado sobre la plataforma de desarrollo sobre la que se llevará a cabo del proyecto. También se especificará la fase de diseño gráfico, en la cual se incluirán las propuestas de diseño, teniendo en cuenta el manual de estilos que se definirá también en esta fase, y se incluirá una propuesta del mapa web.

En el capítulo 5 se presentará la fase de **implementación**. Se especificarán todos los requisitos mínimos de instalación, así como las instrucciones para su realización, es decir, todos los detalles técnicos.

En el capítulo 6 se presentará la fase de **implantación**. Se especificarán los prototipos generados y todos los informes de pruebas realizados.

En el capítulo 7 se presentarán todas las **conclusiones** obtenidas de las reuniones de retrospectiva que se realizan al final de cada sprint.

Por último, se detallarán las **referencias bibliográficas** empleadas durante la generación de esta memoria, así como una serie de **anexos** que complementarán diferentes apartados de esta.

2. Análisis de mercado

2.1. Público objetivo y perfiles de usuario

BOLITA está pensada principalmente para padres de familia que quieren gestionar un diario interactivo con el cual poder guardar todos los momentos de su hijo y adjuntarle fotos, vídeos y audios.

En esta aplicación convivirán cuatro perfiles de usuario diferentes:

- **Usuario anónimo:** usuario que navega por la aplicación sin iniciar sesión en la misma.
- **Usuario editor:** usuario registrado que dispone de una cuenta con rol de editor y con la cual ha iniciado sesión en la aplicación. Este tipo de cuenta puede ser creada por un usuario anónimo, simplemente cubriendo una serie de datos en un formulario de registro.
- **Usuario administrador:** usuario registrado que dispone de una cuenta con rol de administrador y con la cual ha iniciado sesión en la aplicación. Este tipo de cuenta sólo podrá ser creada por el súper administrador, que será el único que pueda gestionar este tipo de cuentas de usuario.
- **Súper administrador:** usuario registrado. Esta cuenta de usuario se crea durante la instalación de la aplicación. Se trata del único usuario que se saltará por defecto todos los permisos y podrá realizar todas las configuraciones y gestiones desde el panel de control.

2.2. Competencia / Antecedentes

Realizando un análisis de la competencia actual se ha detectado que durante un tiempo existió en el mercado una aplicación que cubría parte de las funcionalidades que aporta el proyecto BOLITA, aunque la idea no era exactamente la misma. La aplicación **suspasitos.com** permitía crear un diario del bebé al que adjuntar fotos, permitiendo además el acceso a este diario a amigos y familiares cercanos, los cuales podían realizar comentarios sobre las fotos de este. Actualmente esta aplicación ha cerrado.

Durante este análisis de competencia se ha descubierto que actualmente en el mercado existen muchas aplicaciones dirigidas a los padres de familia, sobre todo APP para móvil. Pero ninguna de ellas aporta la idea de negocio de BOLITA. La gran mayoría están más dirigidas a lo que es el cuidado del bebé y en muchos casos sólo están pensadas para el primer año de vida. Se mencionan a continuación una serie de este tipo de aplicaciones existentes en el mercado:

- La aplicación **Baby Connect**, más específica para bebés, que permite realizar un registro diario del bebé. Pero se trata más de un registro de actividades y cambios que realiza el bebé cada día: horas de sueño, cambio de pañal, visitas al médico, alimentos ingeridos, medicinas,

vacunas, estado anímico. Es más, como un registro de eventos que van sucediendo en el día a día, pensada como intercambio de información entre los diferentes cuidadores del bebé.

- La aplicación **Diario de mi bebé**, para iPhone, o la aplicación **Nibaneba bebé diario**, para Android, son muy parecidas a la aplicación del punto anterior. Más pensadas como un registro de la evolución diaria de los eventos, incluyendo incluso un recordatorio de eventos.
- También la marca de nutrición infantil Hero ha desarrollado una aplicación **Hero Baby** que permite realizar un registro de su evolución, así como guardar fotos o un pequeño diario de anécdotas graciosas del niño. Aunque esta aplicación está más dirigida hacia el cuidado del bebé que realmente la creación del diario.
- La aplicación **Mi embarazo y mi bebé día a día** está más pensada para la evolución de todo el embarazo, así como del primer año del bebé. Permite realizar un seguimiento diario del bebé a través de un calendario personalizado.

2.3. Análisis DAFO

DAFO	Debilidades	Mercado pequeño.
	Fortalezas	Orientada a un público muy específico que convive diariamente con las nuevas tecnologías.
	Amenazas	La existencia de otras aplicaciones con funcionalidades parecidas.
	Oportunidades	Las familias suelen tener uno o dos hijos y cada vez los padres intentan guardar más recuerdos del crecimiento de estos, sobre todo del primero. Es una aplicación web, no es necesaria la instalación de una nueva APP en el móvil, con lo cual no ocupa espacio.

Tabla 7: Análisis DAFO

3.Propuesta

3.1. Definición de objetivos / especificaciones del producto

Se necesita una aplicación web que permita crear una cuenta de usuario privada con la cual crear las diferentes entradas de un diario. El usuario con los datos de estas entradas podrá crear un timeline o una presentación.

Teniendo en cuenta esto se define a continuación los objetivos y especificaciones de la aplicación web a desarrollar.

Para los usuarios anónimos:

- Permitir crear una cuenta de usuario con rol de editor.
- Permitir la navegación por la parte pública del portal.
- Permitir enviar una sugerencia o duda a través del formulario de contacto.

Para los usuarios editores (padres):

- Permitir iniciar sesión con la cuenta de usuario creada con rol de editor.
- Gestionar entradas en el diario: crear nuevas entradas o modificar las existentes.
- Buscar entradas a través de una serie de filtros específicos.
- Crear un timeline con todas las entradas o con los resultados de una búsqueda.
- Crear una presentación de imágenes aleatorias de todas las entradas o de las filtradas como resultado de una búsqueda.

Para los usuarios administradores:

- Permitir la gestión de elementos genéricos como menús, taxonomías, estructuras de contenidos.
- Permitir la gestión de los contenidos de la parte pública.
- Permitir la activación / desactivación de funcionalidades.
- Permitir la configuración genérica de la aplicación.
- Permitir la gestión de usuarios y permisos.

4. Diseño

4.1. Arquitectura general de la aplicación

Dadas las características de la aplicación, donde la cantidad de usuarios no debería alcanzar altos picos de visitas simultáneas, se considera partir de una arquitectura inicialmente simple. En el caso de que se produzca un incremento considerable de usuarios concurrentes, se podrá ampliar horizontalmente el sistema sin repercutir negativamente en el funcionamiento de este y consiguiendo solucionar esta carga de usuarios.

Se propone la siguiente arquitectura de sistemas para alojar la aplicación BOLITA:

- Un servidor web: Apache 2.4+ con PHP 7.2
- Un servidor de base de datos: MySQL 5.7.18+

4.2. Arquitectura de la información y diagramas de navegación

En la aplicación web existirá una navegación principal, tanto como usuario anónimo como con usuario identificado:

- **Inicio:** enlaza con la portada de la aplicación.
- **Presentación:** página que hace una presentación de la aplicación.
- **Cómo acceder:** página que explica a los usuarios como crear una cuenta e iniciar sesión.

En este menú los usuarios anónimos además visualizarán una cuarta opción:

- **Iniciar sesión:** formulario de inicio de sesión. En esta página además el usuario tiene la opción de crear una nueva cuenta o reinicializar su contraseña.

Además, en el pie de página existirá un menú común a todas las páginas de la aplicación, que será accesible por todos los usuarios:

- **Mapa web:** muestra el mapa web del portal: menú navegación principal y pie de página.
- **Aviso legal:** página en la cual se detalla el aviso legal del portal.
- **Política de cookies:** página en la cual se detalla la política de cookies del portal.
- **Contacto:** formulario de contacto, que permite enviar un mensaje al buzón de correo de la aplicación.

Todos los usuarios que inicien sesión con una cuenta con perfil de editor tendrán acceso a un panel de control con un menú específico, que constará de los siguientes elementos:

- **Ver perfil:** permite al usuario acceder a los datos de su cuenta de usuario y acceder al formulario de edición.

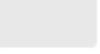
- **Crear nueva entrada diario:** enlaza con el formulario de creación de una nueva entrada en el diario.
- **Diario:** enlaza con el buscador de las entradas del diario específicas del usuario, permite visualizar todas las entradas del diario creadas por el usuario y acceder a la ficha con los datos completos de cada una de las entradas.
- **Línea de tiempo:** enlaza con el buscador que permitirá generar un timeline con las entradas filtradas.
- **Presentación:** enlaza con el buscador que permitirá generar una presentación con imágenes aleatorias de las entradas filtradas.

4.3. Diseño gráfico e interfaces

4.3.1. Estilos

La aplicación seguirá un estilo sencillo, con un diseño limpio y plano, adaptado a la tendencia actual.

Se define una gama de colores llamativos, intentando escapar del clásico azul y rosa que caracteriza a los bebés y niños, siguiendo hacia tonalidades verdes y violetas.

	HEX	#BA2CFF	RGB	186, 44, 255
	HEX	#8FB21A	RGB	143, 178, 26
	HEX	#000000	RGB	0, 0, 0
	HEX	#333333	RGB	51, 51, 51
	HEX	#74797B	RGB	116, 121, 123
	HEX	#E8E8E8	RGB	232, 232, 232
	HEX	#FFFFFF	RGB	255, 255, 255

4.3.2. Usabilidad /UX

Para asegurar la usabilidad de la aplicación web se seguirán una serie de pautas:

- Diseño sencillo para evitar una carga innecesaria que prolongaría la espera del usuario.

- Aplicación de estándares de accesibilidad que permitirán el acceso a un mayor número de usuarios.
- Estructuración sencilla de la navegación, que permitirá al usuario localizar de forma fácil la información y sin necesidad de realizar una multitud de clics.
- Independencia del contenido, estructura y presentación, aportada por DRUPAL.
- Solicitud requerida de textos alternativos para los elementos de tipo imagen.
- Se establece una política de privacidad para los contenidos dados de alta por los usuarios editores.
- Se intenta seguir una estructura de diseño en donde todos los elementos comunes se ubiquen siempre en la misma posición.
- Se proporciona al usuario un mapa web del sitio.
- Se añadirán funcionalidades de búsqueda con filtros específicos.
- La presentación de los contenidos será siguiendo un diseño común.

4.4. Lenguajes de programación y APIs utilizados

La propuesta técnica se basa en el desarrollo de una nueva aplicación web en **Drupal 8**, por ser un software libre y dotado de un control de nivel alto de seguridad, así como de un alto rendimiento y un gran equipo de soporte. Además, se trata de un CMS que nos permite la gestión de contenidos dinámicos de forma fácil y sencilla. Así como de ampliar nuevas funcionalidades ya desarrolladas por la comunidad con módulos contribuidos o, gracias a su diseño modular, desarrollar nuevas funcionalidades sin afectar al funcionamiento básico del sistema.

Se ha elegido la última versión de este CMS, por tratarse de un producto ya estable y que ofrece numerosas ventajas frente a su versión anterior.

Hacer uso de esta versión de Drupal permitirá:

- Aplicar nuevas actualizaciones, mejoras y parches de seguridad del núcleo de Drupal, así como de sus módulos contribuidos, de una forma sencilla y rápida.
- Hacer uso de los nuevos componentes que aporta Drupal 8, ya que está basado en Symfony2, como son Routing, Yamp, Twig ...
- Uso de herramientas de desarrollo y administración como pueden ser composer, drupal console, drush... donde la integración es mayor que en versiones anteriores.
- Sincronizar configuraciones de forma rápida y sencilla desde el entorno de desarrollo al entorno de producción en futuras versiones.

Se instala la versión 8.6.2 de Drupal.

4.4.1. Módulos y temas contribuidos

Además de las funcionalidades básicas que vienen integradas en el núcleo de Drupal, es posible añadir nuevas funcionalidades a través de módulos. Los módulos son componentes software que aportan funcionalidad al núcleo de Drupal 8.

Módulos contribuidos

Los **módulos contribuidos (contrib)** son los módulos creados por desarrolladores independientes que aportan funcionalidades genéricas. Estos, por lo general, se pueden descargar desde el repositorio de módulos de Drupal.

En este proyecto se usan los siguientes módulos contribuidos:

- **Address** (*address*): Proporciona funcionalidades para gestionar direcciones postales.
- **Adminimal Admin Toolbar** (*adminimal_admin_toolbar*): Le aplica a la interfaz del Admin Toolbar los estilos propios del tema de administración contribuido Adminimal. Requiere de:
 - **Admin Toolbar** (*admin_toolbar*): Proporciona una interfaz más intuitiva al menú de administración.
- **CAPTCHA** (*captcha*): Añade retos usando la API de CAPTCHA para evitar envío fraudulento de formularios.
- **Context UI** (*context_ui*): Proporciona una sencilla interfaz de usuario para la configuración de una estructura del sitio usando el contexto.
 - **Context** (*context*): Permite definir condiciones y configurar reacciones que deben ser implementadas cuando se cumpla una condición.
- **Colorbox** (*colorbox*): Un plugin ligero y personalizable para jQuery.
- **Date popup** (*date_popup*): Añade selectores de fecha HTML5 a los filtros de las vistas.
- **Eu Cookie Compliance** (*eu_cookie_compliance*): Este módulo tiene como objetivo hacer que el sitio web cumpla con la nueva regulación de cookies de la UE.
- **Field Group** (*field_group*): Proporciona la capacidad de agrupar los campos tanto en el formulario de creación / edición como en la plantilla de presentación.
- **Paragraphs** (*paragraphs*): Permite añadir entidades de tipo Paragraphs, estas entidades son usadas para crear contenido de forma cómoda y ordenada y facilita la visualización de este. Requiere de:
 - **Entity Reference Revisions** (*entity_reference_revisions*): Añade un tipo de campo que permite referenciar entidades con soporte para revisiones.
- **Pathauto** (*pathauto*): Proporciona la opción a otros módulos de generar alias de URL de contenido. Requiere de:
 - **Chaos tool suite** (*ctools*): Aporta utilidades que facilitan el desarrollo.

- **Token** (*token*): Proporciona una interfaz de usuario para el API de comodines, además de algunos comodines que faltan en el núcleo de Drupal.
- **Registration Role** (*registration_role*): Asigna un role a todos los nuevos usuarios que se registran en el portal.
- **Redirect After login** (*redirect_after_login*): Permite redirigir al usuario después de identificarse a una URL configurable.
- **Simple timeline** (*simple_timeline*): Proporcionar un estilo de vistas para mostrar el contenido en una línea de tiempo vertical.
- **Slick** (*slick*): Proporciona un estilo para mostrar el contenido en un carrusel.
 - **Slick UI** (*slick_ui*): Proporciona una interfaz de usuario de configuración al carrusel Slick.
 - **Slick views** (*slick_views*): Permite integrar la funcionalidad anterior del carrusel en vistas.
 - **Blazy** (*blazy*): Proporciona integración básica Blazy para imágenes de carga perezosa y multi-servicio.
- **SMTP Authentication Support** (*smtp*): Permitir que los correos electrónicos del sitio sean enviados a través de un servidor SMTP de su elección.
- **Sitemap** (*sitemap*): Facilita la visualización del sitemap del portal.
- **Simple XML Sitemap** (*simple_sitemap*): Crea un sitemap XML del contenido del portal conforme a hreflang.

Los temas son los componentes del portal que hacen que se visualice de la de la forma en que lo hace, cada tema está formado por archivos que definen como será la presentación del sitio. Un tema puede ser realizado desde la nada o puede estar basado en otro tema base. Todo lo relacionado con la presentación del portal está en los archivos del tema.

Temas contribuidos

El tema de administración usado en el portal es el tema contribuido *Adminimal*, este tema aporta un diseño moderno y minimalista al panel de configuración haciéndolo más intuitivo para el usuario:

- **Adminimal - Responsive Administration Theme** (*adminimal_theme*): Tema de administración de Drupal con diseño minimalista moderno.

El tema customizado está basado en un tema contribuido:

- **Bootstrap** (*bootstrap*): tema construido para utilizar Bootstrap, un marco de trabajo elegante, intuitivo y potente para un desarrollo web más rápido y sencillo.

4.4.2. Módulos y temas custom

Módulos custom

Los **módulos customizados (custom)** son los módulos creados específicamente para esta aplicación y que permiten alterar funcionalidades existentes o añadir nuevas.

En este proyecto se usan los siguientes módulos custom:

- **BOLITA common:** módulo que permite añadir funcionalidades específicas de este portal. Se crearon una serie de plantillas twig para la visualización de algunas entidades paragraph y campos asociados con estas. Además de implementar estos templates en el tema. También se crearon una serie de funciones que permiten sobrescribir o modificar campos en el preprocess de la visualización de campos asociados a las entidades paragraph anteriores. Y también asociados a una entidad de tipo contenido, landing page, para tratar los campos de visualización del título y la miga de pan en los contenidos creados de este tipo de contenido.

A continuación, se explican algunas de las implementaciones realizadas dentro de este módulo:

- En el código que se muestra a continuación, en la figura 2, se implementa el código HTML que tendrá salida para visualizar el contenido de un párrafo del tipo *section-images*. En esta plantilla twig se añaden las clases CSS específicas, como se puede observar se añade la *class_num_column* que indica una clase diferente según el valor indicado en el campo *Número de imágenes por fila* del párrafo en cuestión y viene definida en la variable *class_num_column*. Esta plantilla devuelve un elemento div que tiene como atributos dichas clases y dentro del mismo se pintan todo el contenido del párrafo exceptuando el valor del campo *Número de imágenes por fila*, que solo era usado para indicar la clase CSS específica.

```

{% set classes = [
  'paragraph',
  'paragraph--type--' ~ paragraph.bundle|clean_class,
  view_mode ? 'paragraph--view-mode--' ~ view_mode|clean_class,
  class_num_column ? class_num_column|clean_class,
] %}

{% block paragraph %}
  <div{{ attributes.addClass(classes) }}>
    {% block content %}
      {{ content|without('field_column_number') }}
    {% endblock %}
  </div>
{% endblock paragraph %}

```

Figura 2: código del archivo "paragraph--section-images--default.html.twig"

- En el siguiente código, que se muestra en la figura 3, se indica la existencia de nuevas plantillas twig customizadas. Se hace uso del `hook_theme()` propio de Drupal para hacer dicha implementación.

```

/**
 * Implements hook_theme().
 */
function bolita_common_theme() {
  return [
    'field_paragraph_field_section_name' => [
      'template' => 'field--paragraph--field-section-name',
      'base hook' => 'field'
    ],
    'paragraph_section_images_default' => [
      'template' => 'paragraph--section-images--default',
      'base hook' => 'paragraph'
    ],
    'paragraph_section_text_image_default' => [
      'template' => 'paragraph--section-text-image--default',
      'base hook' => 'paragraph'
    ],
  ];
}

```

Figura 3: código del archivo "bolita_common.module"

- En el código que se muestra a continuación, en la figura 4, se implementa dentro de cada función de tipo `hook_preprocess_paragraph()` la creación de nuevas variables que van a llegar a la plantilla twig con valores asociados a los de los campos del propio párrafo. Así en el primer caso (`section_images`) se envía el nombre de la clase CSS `column-num-X`, donde X es el valor indicado por el usuario en el campo *Número de imágenes por fila* del párrafo. En el segundo caso (`section_text_image`) se envía la clase CSS `image-float-X` donde X es el valor indicado por el usuario en el campo *Posición de la imagen* del párrafo (left o right).

```

/**
 * Implements hook_preprocess_paragraph().
 */
function bolita_common_preprocess_paragraph_section_images($variables) {
  $paragraph = $variables['paragraph'];
  if (!$paragraph->field_column_number->isEmpty()) {
    $number_value = $paragraph->field_column_number->getValue();
    $column_num = isset($number_value[0]['value']) ? $number_value[0]['value'] : 0;
    if (!empty($column_num)) {
      $variables['class_num_column'] = "column-num-" . $column_num;
    }
  }
}

function bolita_common_preprocess_paragraph_section_text_image($variables) {
  $paragraph = $variables['paragraph'];
  $float = 'left';
  if (!$paragraph->field_image_position->isEmpty()) {
    $float_value = $paragraph->field_image_position->getValue();
    $float = isset($float_value[0]['value']) ? $float_value[0]['value'] : '';
  }
  $variables['class_float'] = "image-float-" . $float;
}

```

Figura 4: código del archivo "bolita_common.module"

- **BOLITA Panel de control:** Módulo que crea un menú personalizado para los usuarios con role de editor. Este menú es añadido a la barra de administración superior (*toolbar*). Se creó un permiso para indicar que usuario tienen acceso y pueden visualizar este menú. Se creó una ruta para incluir el menú del panel de control, a la cual se le aplicó el permiso creado anteriormente. Esta ruta se creó dentro de la parte de administración (*/admin*), así la ruta del panel estará ubicada en (*/admin/panel-control*). Además, se alteró la visualización de los elementos de la barra de administración para que el editor solo tenga acceso al de Inicio, al de Usuario y a este nuevo elemento del panel.

A continuación, se explican algunas de las implementaciones realizadas dentro de este módulo:

- En el código que se muestra a continuación, en la figura 5, se altera la visualización de los elementos de la barra de administración para los usuarios con el role de editor. Así estos sólo visualizarán los elementos: Inicio (*home*), Usuario (*user*) y el nuevo elemento customizado Panel de Control (*panel*).

```

/**
 * Implements hook_toolbar_alter().
 */
function bolita_controlpanel_toolbar_alter($items) {
  $current_user = \Drupal::currentUser();
  if (in_array('editor', $current_user->getRoles())) {
    $items_editor = array('home', 'user', 'panel');
    foreach ($items as $key => $item) {
      if (!in_array($key, $items_editor)) {
        unset($items[$key]);
      }
    }
  }
}

```

Figura 5: código del archivo "bolita_controlpanel.module"

- En el archivo "*bolita_controlpanel.links.menu.yml*" se añadieron todos los elementos de menú que se visualizarán en este menú del panel de control. Para cada uno de estos elementos se indica la ruta interna de Drupal, la descripción, el título que será el que se mostrará en el menú, el peso que indicará la posición del elemento dentro del menú, el elemento padre que indica de que elemento del menú cuelga y el nombre del menú al que pertenecen, en este caso todos pertenecen al menú *panel*.

```
bolita_controlpanel.panel:
  route_name: bolita_controlpanel.panel
  description: 'Panel de control'
  title: Panel de control
  weight: 0
  menu_name: panel

bolita_controlpanel.view_user:
  route_name: user.page
  description: 'Acceso a la ficha de usuario'
  title: Ver perfil
  weight: 1
  parent: bolita_controlpanel.panel
  menu_name: panel

bolita_controlpanel.create_dailyentry:
  route_name: node.add
  route_parameters: { node_type: 'dailyentry' }
  description: 'Crear entrada en el diario'
  title: Crear nueva entrada diario
  weight: 2
  parent: bolita_controlpanel.panel
  menu_name: panel

bolita_controlpanel.search_daily:
  route_name: view.dailyentry.page_daily
  description: 'Buscador de entradas del diario que se visualizan en un listado'
  title: Diario
  weight: 3
  parent: bolita_controlpanel.panel
  menu_name: panel

bolita_controlpanel.search_timeline:
  route_name: view.dailyentry.page_timeline
  description: 'Buscador de entradas del diario que se visualizan en un timeline'
  title: Timeline
  weight: 4
  parent: bolita_controlpanel.panel
  menu_name: panel

bolita_controlpanel.search_book:
  route_name: view.dailyentry.page_book
  description: 'Buscador de entradas del diario que se visualizan en una presentación tipo libro'
  title: Presentación
  weight: 5
  parent: bolita_controlpanel.panel
  menu_name: panel
```

Figura 6: código del archivo "bolita_controlpanel.links.menu.yml"

Tema custom

Tema específico con el estilo de la aplicación:

- **Bolita** (*bolita*): tema creado exclusivamente para el portal, basado en Bootstrap. Dentro del directorio **css** están todos los archivos css específicos de este tema. Así dentro de “*global.styles.css*” están los estilos más específicos referentes a cada una de las secciones y contenidos de este portal.
 - A continuación, en la figura 7, se muestra un ejemplo de un archivo con los estilos CSS propios del tema Bolita:

```
/* Main header menu END */

/* Footer menu BEGIN */
} footer.site-footer {
    flex: 0 0 auto;
    background-color: rgba(238, 238, 238, 1);
    color: #fff;
}

} footer.site-footer #block-bolita-footer > ul li {
    text-align: center;
}

} footer.site-footer #block-bolita-footer > ul a {
    text-transform: uppercase;
    font-family: 'Raleway', sans-serif;
    color: #000000;
    font-weight: 700;
}

} footer.site-footer #block-bolita-footer > ul li a:hover {
    color: #fff;
    background-color: #333;
}

} footer.site-footer #block-bolita-footer > ul li.active a {
    color: #8fb21a;
}

@media (min-width: 768px) {
} footer.site-footer {
    min-height: 80px;
}

} footer.site-footer .container {
    top: 15px;
    position: relative;
}
}
```

Figura 7: código del archivo “*global.styles.css*”

Dentro del directorio **templates** están todos los archivos twig que sobrescriben templates propias del core o de los módulos contribuidas. Algunas de ellas ya están sobrescritas también en el tema Bootstrap. Por ejemplo, el archivo “*html.html.twig*” tiene la estructura genérica del html de las páginas del portal.

- A continuación, en la figura 8, se muestra la plantilla twig que devuelve el código HTML de la estructura básica de una página:

```

[{% set body_classes = [
  logged_in ? 'user-logged-in',
  not root_path ? 'path-frontpage' : 'path-' ~ root_path|clean_class,
  node_type ? 'page-node-type-' ~ node_type|clean_class,
  db_offline ? 'db-offline',
  theme.settings.navbar_position ? 'navbar-is-' ~ theme.settings.navbar_position,
  theme.has_glyphicons ? 'has-glyphicons',
  theme.name ? 'theme-' ~ theme.name|clean_class
] %}
<!DOCTYPE html>
<html {{ html_attributes }}>
<head>
  <head-placeholder token="{{ placeholder_token|raw }}">
    <title>{{ head_title|safe_join(' | ') }}</title>
    <css-placeholder token="{{ placeholder_token|raw }}">
      <js-placeholder token="{{ placeholder_token|raw }}">
    </head>
<body{{ attributes.addClass(body_classes) }}>
  <h1 class="sr-only">{{ head_title_array['name'] }}</h1>
  <a href="#main-content" class="visually-hidden focusable skip-link">
    {{ 'Skip to main content'|t }}
  </a>
  {{ page_top }}
  {{ page }}
  {{ page_bottom }}
  <js-bottom-placeholder token="{{ placeholder_token|raw }}">
</body>
</html>
```

Figura 8: código del archivo “*html.html.twig*”

Además de incorporar la imagen png con el logo del portal y el archivo con el favicon:



Figura 9: logo creado para la aplicación web



Figura 10: icono con el favicon para la aplicación web

La imagen *screenshot.png* es la imagen que se muestra como imagen representativa del tema. Por lo tanto, es la imagen que se muestra dentro del apartado Apariencia de la configuración del Drupal:

Temas instalados



Bolita 8.x-1.0 (tema predeterminado)
Drupal 8 and Bootstrap 3 theme for BOLITA website.

[Configuración](#)



Adminimal 8.x-1.3 (tema de administración)
Drupal administration theme with modern minimalist design.

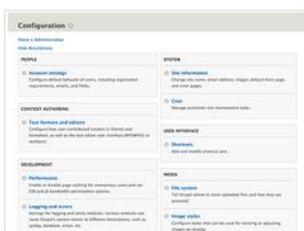
[Configuración](#) | [Configurar como predeterminado](#)



Bootstrap 8.x-3.16

Construido para usar Bootstrap, un entorno de desarrollo de front-end limpio, intuitivo y potente para acelerar y facilitar el desarrollo web.

[Configuración](#) | [Configurar como predeterminado](#)



Seven 8,6,4

El tema de administración predeterminado para Drupal 8 fue diseñado con líneas limpias, bloques simples, y fuente sans-serif para enfatizar las herramientas y tareas a mano.

[Configuración](#) | [Configurar como predeterminado](#)

Figura 11: página de configuración de la apariencia del portal

Implementación

4.5. Requisitos de instalación

Se requiere un servidor web con:

- Apache 2.4 o superior,
- con versión de PHP 7.2, recomendado por la última versión estable de Drupal (8.6),
- con MySQL en su versión 5.5.3 o superior, para la gestión de la base de datos.

Además, se tendrán en cuenta los siguientes requisitos del servidor:

- Memoria RAM asignada a PHP en la máquina mínimo de 256MB,
- PHP debe haber sido compilado con la extensión *CURL* y debe tener activado *PDO* y *OPcache*,
- la extensión de Apache *mod_rewrite* instalada y activada,
- además, son necesarias otras extensiones de PHP como son:
 - *mbstring*
 - *xml*
 - *mysql*
 - *json*
 - *gd*

Para la gestión e instalación de Drupal es recomendado instalar los paquetes **Drush** y **Composer**.

4.6. Instrucciones de instalación

4.6.1. Código fuente

1. Lo primero que hay que hacer para realizar la instalación de Drupal es copiar los archivos de la entrega en el directorio web del servidor en el que se vaya a realizar la instalación.
2. Hay una serie de directorios que Apache (la aplicación Drupal) necesita escribir en ellos, por lo que es necesario que tenga permisos. Lo ideal es que el usuario Apache y el usuario que copia los directorios en el servidor pertenezcan al mismo grupo. De esta forma se le dan permisos de escritura al propietario y a los usuarios del mismo grupo (con un *chmod 775*). Si no, es necesario cambiarle el propietario al usuario Apache. En esta aplicación los directorios son:
 - a. *web/sites/default/files* (incluidos todos sus directorios)
 - b. *private*

4.6.2. Base de datos

3. El siguiente paso consiste en crear una base de datos e importar el *dump* entregado.
4. Una vez tenemos la base de datos creada, es necesario modificar los archivos de configuración de Drupal para que este se conecte a la misma. En concreto tenemos que modificar el archivo “*web/sites/default/settings.php*” y buscar el siguiente código:

```
$databases['default']['default'] = array (  
  'database' => [database],  
  'username' => [username],  
  'password' => [password],  
  'prefix' => '',  
  'host' => [host],  
  'port' => '3306',  
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',  
  'driver' => 'mysql',  
);
```

En el fragmento superior hay que modificar los parámetros que están entre corchetes para que coincidan con los de nuestra base de datos.

5. Una vez modificado el archivo de configuración se recomienda sacarle permisos de escritura (*chmod 444*).

NOTA: en la base de datos de instalación el usuario administrador es “*admin.tfg*” y la contraseña es “*admin*”. Esta contraseña será necesario resetearla cuando se ponga el portal en PRODUCCIÓN.

4.6.3. Configuraciones

6. Para el correcto funcionamiento del envío de correos es necesario configurar el módulo SMTP. Para ello es necesario iniciar sesión en el portal con la cuenta de usuario superadministrador y desde la barra de menú superior de administración ir a *Configuración >> Sistema >> Soporte de autenticación SMTP*.
 - a. Activar el módulo.
 - b. Configurar el servidor SMTP e introducir los datos de autenticación SMTP.

5. Demostración

5.1. Prototipos

Prototipos creados a lo largo del proceso de desarrollo.

5.1.1. Prototipos Lo-Fi

Se adjuntan en el anexo tres de las primeras propuestas de diseño:

- Portada
- Ficha de una página básica
- Ficha de una entrada del diario

El diseño del portal se basó en estas propuestas iniciales, aunque según se fue evolucionando el desarrollo de este también se fue evolucionando el diseño, adaptándolo más a los contenidos del portal y al público hacia el que va dirigido.

5.2. Tests

Se revisaron todas las páginas de la parte pública comprobando su correcta visualización en un dispositivo móvil, en un Tablet y en dos monitores de tamaño diferente. Se creó una cuenta de usuario para poder acceder a la parte privada y crear un diario. Se realizaron pruebas de creación, modificación y eliminación de entradas en el diario. Así como búsquedas a través de los filtros del diario, comprobando que los resultados mostrados eran los correctos. También se realizaron pruebas con la generación del timeline y de la presentación.

Se ha comprobado que el usuario no puede acceder a entradas creadas por otro usuario, ni visualizarlas de ninguna forma.

5.3. Ejemplos de uso del producto (o guía de usuario)

La navegación como usuario anónimo permite acceder a las páginas informativas, legales y de contacto de la aplicación, así como crear una nueva cuenta como usuario editor.

Desde el menú de navegación principal, el usuario puede acceder al formulario de creación de una nueva cuenta de usuario, desde el elemento de *Iniciar sesión*. Al visualizar el formulario de inicio de sesión, en la parte superior de dicho formulario tiene acceso a una pestaña de *Crear nueva cuenta de usuario*.

Al crear una nueva cuenta el usuario recibirá un correo en la cuenta de correo indicada al darse de alta, en el cual se le indicará como puede terminar de activar su cuenta. Automáticamente se le agregará el role de editor con el cual tendrá permisos para gestionar el diario.

Al iniciar sesión con su usuario y contraseña, el usuario editor tendrá acceso en la barra superior de administración a un menú de *Panel de Control* en el cual tendrá acceso a:

- los datos de su perfil,
- al formulario de creación de una nueva cuenta de usuario,
- al diario completo mostrado en formato listado y en el cual tendrá un buscador para filtrar las entradas,
- a un buscador que le permitirá generar una línea gráfica (timeline) con los resultados generados con los filtros proporcionados en la búsqueda,
- así como a un buscador que le permitirá generar una presentación de imágenes aleatorias de los resultados generados con los filtros proporcionados en la búsqueda.

6. Conclusiones y líneas de futuro

En este capítulo se detallarán ciertas conclusiones sobre el trabajo realizado y lo aprendido durante la realización del este trabajo final de grado, así como nuevas líneas de futuro que se podrían tener en cuenta en próximas versiones.

6.1. Conclusiones

Durante la ejecución de este trabajo he aplicado muchas de las cosas aprendidas durante el estudio de este grado. La línea de mayor aprendizaje, en el desarrollo de la aplicación, fue sobre el CMS Drupal 8, la implementación de nuevas funcionalidades y el uso de sus módulos contribuidos. Así como la maquetación responsiva de la aplicación.

Con la idea ya cerrada y realizado el análisis funcional comencé con su materialización.

Una vez instalada la máquina virtual y todas las herramientas necesarias comencé con la implementación de los hitos propuestos. Aquí fue donde comencé a tropezar con más de una problemática, las cuales fui resolviendo poco a poco. Además, de resolver algunas de las funcionalidades de forma diferente a como estaba previsto en un primer momento, lo más complejo con lo que me encontré fue con la idea inicial de crear una presentación personalizada con una librería jQuery. Adaptar esta librería a Drupal 8 no resultaba una tarea tan sencilla como se preveía a primera vista. Al final, ya fuera de la estimación de esta tarea, decidí solventar esta funcionalidad de otra forma y apoyándome con un módulo contribuido. Otra de las tareas que en un principio tenía pensado realizar de otra forma, fue el panel de control de la parte privada de un editor. Una vez inmersa en la nueva estructura de Drupal 8, analicé la posibilidad de crear este panel dentro de la barra de herramientas propia de la administración, asemejando este a la funcionalidad que tiene el panel de control propio de la gestión del Drupal. De esta forma, cree un módulo customizado para esta tarea en el cual añadí toda la funcionalidad necesaria para la implementación de dicho panel.

Uno de mis mayores retos, era la maquetación del portal. Quería que en todo momento fuera responsivo, ya que es una aplicación pensada para un público que la va a usar tanto desde un dispositivo móvil, como de una Tablet o desde la pantalla de un ordenador. Cree un tema específico para mi aplicación, basado en Bootstrap. En la primera fase de maquetación, realicé la misma sobre la pantalla del ordenador y una vez hecha empecé a implementar la maquetación para un dispositivo móvil. Fue en este momento, cuando me di cuenta de que no lo había hecho correctamente, ya que Bootstrap está pensado para maquetar sobre un dispositivo móvil y luego adaptarlo al resto de dispositivos. Llegado a este punto, decidí hacerlo de esta forma, intentando aprovechar el diseño ya definido y los estilos CSS del mismo.

Como se puede intuir de mis comentarios, han sido unos meses de trabajo en los cuales además de realizar un análisis funcional y la planificación de desarrollo, he tenido que implementar las funcionalidades, tomar decisiones con las problemáticas encontradas y buscar alternativas a

alguna de las tareas iniciales. Además, también fue necesario formarme en el desarrollo de Drupal 8, ya que solo tenía conocimientos previos de desarrollo sobre Drupal 7. Y también adquirir más conocimientos sobre maquetación, ya que los tenía eran bastante básicos.

Con la realización de este TFG he realizado un trabajo de varios perfiles, pues yo misma era la gestora del proyecto, el analista funcional, el coordinador, el desarrollador, el diseñador, el maquetador y al mismo tiempo la persona que realizaba el seguimiento y las pruebas de la aplicación.

Cabe mencionar también, que a la hora de publicar mi aplicación en el servidor de la UOC me he encontrado con un problema, por temas de usuarios y permisos. El cual he tenido en cuenta para especificar en el apartado de instrucciones de instalación de esta memoria.

Finalmente, debo destacar que han sido unos meses de mucho trabajo, investigación y desarrollo al final de los cuales he obtenido una primera versión de los objetivos iniciales de la aplicación, pero a medida que iba desarrollando, surgían en mi cabeza nuevas funcionalidades y mejoras que dejan una puerta abierta a la continuación del proyecto en nuevas versiones futuras.

6.2. Líneas de futuro

La aplicación desarrollada para este trabajo final de grado sería la primera versión desarrollada para poner en un entorno de producción, pero se podría evolucionar en futuras versiones con otras nuevas funcionalidades como pueden ser:

- Que los usuarios con role de “editor” puedan crear con su cuenta de usuario más de un diario, ya que esta primera versión está pensada para que cada usuario solo pueda crear un único diario con múltiples entradas.
- Que los usuarios con role de “editor” puedan crear cuentas de usuario con role “invitado”. Estas cuentas permitirán a estos usuarios acceder al diario al que están invitados a visualizar. Pero no tendrán permisos para realizar ninguna gestión sobre las entradas de este. Además, el editor podrá dar permisos a los usuarios invitados a su cuenta, para que estos puedan comentar sus entradas.
- Los usuarios con role de “invitado” podrán acceder a la parte privada del portal, pudiendo visualizar las entradas del diario al que han sido invitados, así como al timeline y a la presentación. Además, si tienen permiso, podrán añadir comentarios sobre las entradas creadas.
- Que los usuarios con role de “editor” puedan crear un póster con las fotos de una serie de entradas en el diario y bajo una temática, además de poder guardarlo en formato imagen o imprimirlo.
- Que los usuarios con role de “editor” puedan crear como un diario maquetado en formato libro y exportarlo a PDF.

- Generar una presentación en la cual además de imágenes también se incluyan los vídeos adjuntados a las entradas.
- Añadir nuevos campos al formulario de gestión de las entradas, como puede ser un campo de ubicación para los casos de vacaciones, viajes, ... Pudiendo añadir estos campos como filtros en los buscadores del diario, etc.

Bibliografía

En esta sección se realiza una recopilación de las fuentes bibliográficas empleadas para la redacción de esta memoria y de la ejecución de la aplicación web.

Páginas web visitadas durante el análisis del mercado actual:

- <https://www.elmundo.es/vida-sana/familia-y-co/2016/12/27/585a5c01ca4741242d8b4590.html>
- <https://www.hero.es/baby/descargar-app-hero-baby-babyayuda-android-iphone-ipad-tablet>
- <https://www.bebesymas.com/recursos-en-la-web/sus-pasitos-aplicacion-para-crear-el-diario-del-bebe>
- <http://suspasitos.com/>
- <https://play.google.com/store/apps/details?id=com.babycenter.pregnancytracker&hl=es>

Imágenes:

- Todas las imágenes usadas en el portal son descargadas de <https://pixabay.com/es/> bajo la licencia [CC0 1.0 Universal](https://creativecommons.org/licenses/by/4.0/) (CC0 1.0). Es decir, todas ellas son libres y dedicadas al dominio público.

Información para contenido:

- El texto de la página de cookies se obtuvo a partir del modelo de la página: <https://www.modelosycontratos.com/2016/04/aviso-cookies-pagina-web/>
- El texto del aviso legal se obtuvo a partir del modelo de la página: <https://www.nireleku.com/gtoys/legal/aviso-legal.html>

Anexos

Anexo A: Glosario

- **CMS:** sistema de gestión de contenidos (*Content Management System*, abreviado CMS), software que permite crear, editar, gestionar y publicar contenido digital.
- **Módulos:** componentes software que aportan funcionalidad al núcleo de Drupal.
- **Temas:** componentes del portal que hace que se visualice de la de la forma en que lo hace, cada tema está formado por archivos que definen como será la presentación del sitio. Un tema puede ser realizado desde la nada o puede estar basado en otro tema base. Todo lo relacionado con la presentación del portal está en los archivos del tema.
- **Entidades de Drupal:** Las entidades son objetos que se utilizan para el almacenamiento persistente de información de contenido y configuración, son elementos a los que se le pueden añadir diferentes campos, como por ejemplo texto, fecha, archivo, etc. Su finalidad es gestionar una unidad de información estructurada.
- **Role:** conjunto de permisos. Los roles se asignan a los usuarios para asignarle el conjunto de permisos que representan. Por ejemplo, un usuario con el rol editor podría contener los permisos para poder crear, editar y borrar contenidos de tipo entrada del diario.

Anexo B: Entregables del proyecto

- Memoria del Trabajo Final de Grado.
- Directorio con todas las fuentes y ficheros del portal.
- Backup de la base de datos.
- Vídeo de presentación.
- Presentación escrita.

Anexo C: Propuestas de diseño

Propuestas iniciales

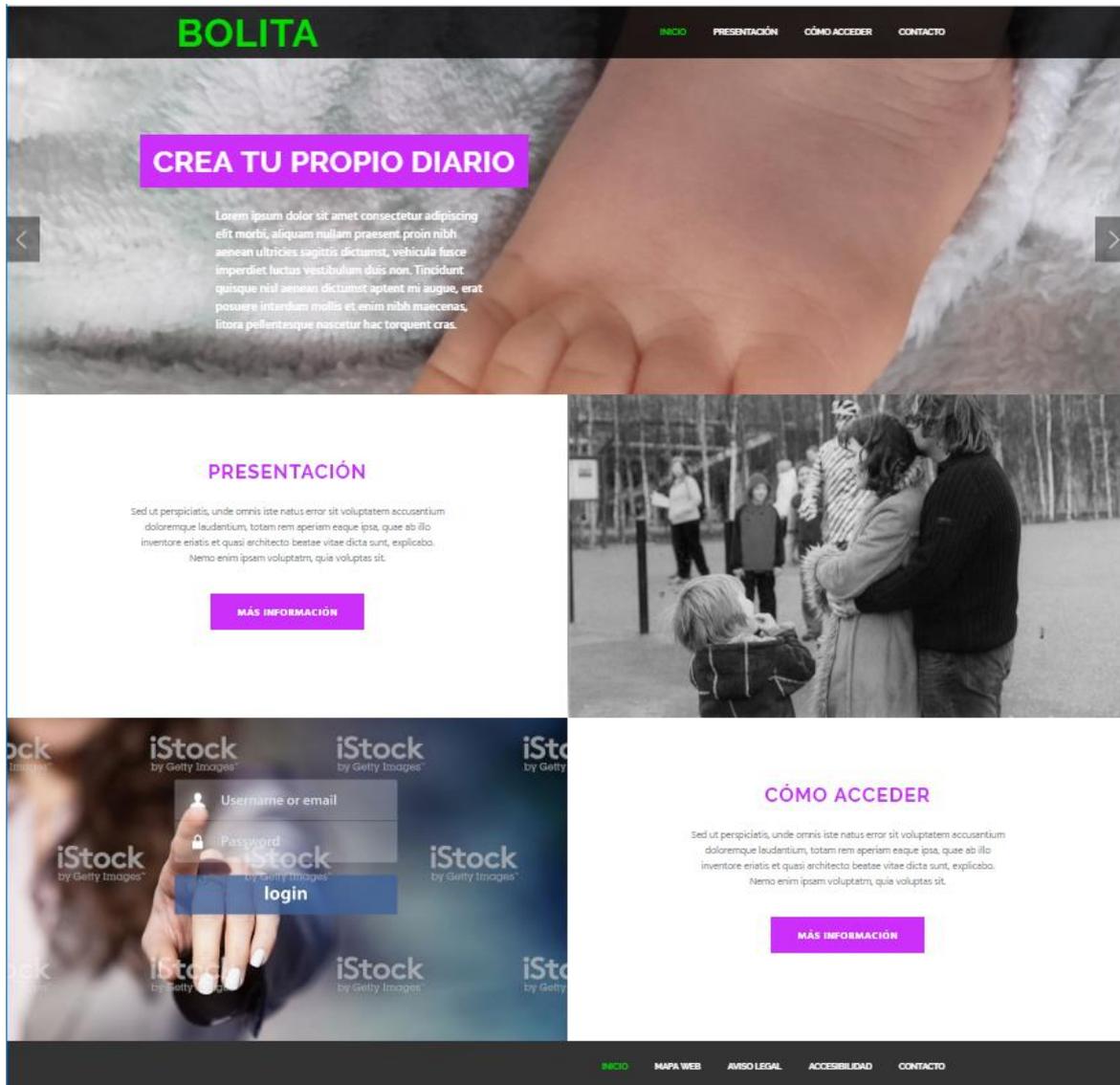


Figura 12: Propuesta de diseño para la portada



Figura 13: Propuesta de diseño para una ficha de página básica



Figura 14: Propuesta de diseño para una ficha de entrada del diario