Universitat Oberta de Catalunya uoc.edu

Desarrollo Aplicación Web de Control de Bebés

Alexandra Joana Parejo Marín
Grado de Ingeniería Informática
Desarrollo Web

Gregorio Robles Martínez
Santi Caballé Llobet
12 de Junio de 2019

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desarrollo Aplicación Web de Control de Bebés*
Nombre del autor:	*Alexandra Joana Parejo Marín*
Nombre del consultor/a:	*Gregorio Robles Martínez*
Nombre del PRA:	*Santi Caballé Llobet*
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	*Gando Ingeniería Informática*
Área del Trabajo Final:	*Desarrollo Web*
Idioma del trabajo:	*Español*
Palabras clave	*Desarrollo web, Aplicación web, Bebés*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La aplicación web de control de bebés, trata de dar un servicio de registro y control sobre los hábitos de los más pequeños, así como registrar los aspectos más básicos de su estado de salud, añadiendo además un listado de citas médicas.

El aplicativo intenta suplir de esta forma un vacío. Las aplicaciones que se ofertan actualmente suelen ser de dispositivo, lo que limita, en cierta manera, su uso. Además, la mayor parte de ellas suelen ser de pago. Por otra parte, se suelen centrar en aspectos concretos, como puede ser el desarrollo cognitivo o la alimentación, pero no existe un servicio en el que se trate de forma más centralizada los aspectos más destacados de las necesidades de los primeros meses de vida.

En cuanto a metodología, se sigue un esquema descendiente que consta de análisis, diseño, desarrollo y test. Se realiza un primer estudio sobre las aplicaciones ofertadas en el mercado y un pequeño estudio de requisitos. Seguidamente se enfoca la etapa de diseño, centrada tanto en la interfaz (*wireframes*, diagramas de interacción, etc.), como en la funcionalidad (diagrama UML). La fase de desarrollo tiene un componente iterativo según se van desarrollando las propuestas de la fase anterior. Para finalizar se realizan test de funcionalidad.

En definitiva, se ha intentado crear un producto que dé salida a una demanda, cada vez más creciente, utilizando la tecnología en favor de las necesidades de las personas de una forma útil.

Abstract (in English, 250 words or less):

The web application of baby control, tries to give a service of registration and control over the habits of the little ones, as well as record the most basic aspects of their well-being, also adding a list of medical appointments.

The application tries to fill a gap in this way. The applications currently offered, are usually of a device, which limits, in a certain way, their use. In addition, most of them are usually paid. On the other hand, they focus on specific aspects, such as cognitive development or nutrition, but there is no service that deals with the needs of their first months in a more centralised way.

In terms of methodology, a descendant scheme consisting of analysis, design, development and test must be followed. A first study is carried out on the applications offered in the market and a small study of requirements. Next, the focus relies on the design stage, centered on the interface (wireframes, interaction diagrams, etc.) and in the functionality (diagram UML). The development phase has an interactive component as the proposals of the previous phase develop. Finally, functionality test is carried out.

In short, an attempt has been made to create a product that on the basis of a growing demand, using technology in favor of the needs of the people in a useful way.

Índice

1. Introducción 1
1.1 Contexto y justificación del Trabajo 1
1.1.1 Contexto 1
1.1.2 Oferta Actual. 2
1.1.3 Carencias del mercado. 5
1.2 Objetivos del trabajo. 6
1.3 Enfoque y método seguido. 7
1.4 Planificación del trabajo. 8
1.4.1 Recursos, lenguajes, APIs y *Frameworks* 8
1.4.2 Tareas. 9
1.5 Breve sumario productos obtenidos. 13
1.6 Breve descripción de los otros capítulos de la memoria. 14
2. Fase de diseño 16
2.1 Interfaz. 16
2.1.1 Color. 16
2.1.2 Fuentes tipográficas. 17
2.1.3 Wireframes. 18
2.1.4 Prototipos. 21
2.1.5 Interacción. 23
2.2 Base de datos. 24
2.2.1 Requisitos. 24
2.2.1 Diagrama de clases. 25
3. Fase de creación. 26
3.1 Estructura. 26
3.1.1 *Head*. 26
3.1.2 Cabecera. 26
3.1.3 Navegador. 27
3.1.4 Zona central. 28
3.2 Base de datos. 28
4. Fase de desarrollo. 30
4.1 *Home*. 31
4.2 Formularios 32
4.2.1 *Login*. 32
4.2.2 Registros. 33
4.2.3 Específicos. 34
4.3 Graficas y tablas 35
4.3.1 Gráficas. 36
4.3.2 Tablas. 37
5. Fase de pruebas. 38
6. Conclusiones. 39
7. Glosario 40
8. Bibliografía 41
9. Recursos bajo licencia 43

Lista de figuras

Figura 1: Media de edad de maternidad según el INE (2017) [2] 1
Figura 2: Uso de Internet por rango de edad (Mujeres) según el INE (2017) [4] 1
Figura 3: Imágenes aplicación The Wonder Week [5]. 2
Figura 4: Imágenes aplicación Baby Tracker [6]. 2
Figura 5: Imágenes aplicación Baby Connet [7]. 3
Figura 6: . Imágenes aplicación Cry Translator [8]. 3
Figura 7: Imágenes aplicación LactMed [9]. 4
Figura 8: Imágenes aplicación LactMed [10]. 4
Figura 9: Penetración de dispositivos 2014, 2015, 2017 según Deloitte [11]. 5
Figura 10: Comparativo de uso de aplicaciones frente a navegador en móviles, Ditendria [12] 6
Figura 11: Muestra cromática. 16
Figura 12: Muestra de la fuente Kaushan Script [14] 17
Figura 13: Muestra de la fuente Meera Inimai [15] 18
Figura 14: Diseño de wireframes 19
Figura 15: Detalle cabecera diseño wireframes 19
Figura 16: Detalle navegador diseño wireframes 20
Figura 17: Detalle zona central diseño wireframes 20
Figura 18: Prototipo página inicio 21
Figura 19: Prototipo página de formulario 22
Figura 20: Prototipo página muestra de datos 22
Figura 21: Diagrama de interacción 23
Figura 22: Diagrama de clases UML de la base de datos 25
Figura 23: Diseño del banner de la aplicación 26
Figura 24: Detalle del código del header 27
Figura 25: Relación de clases de la base de datos 29
Figura 26: Imagen final de la página inicio en distintos dispositivos 31
Figura 27: Imagen final de formulario de login en diversos dispositivos 32
Figura 28: Imagen final del formulario de registro. 33
Figura 29: Formulario final de la categoría sueño 34
Figura 30: Detalle del resultado final de la página hábitos 35
Figura 31: Detalle del código de la gráfica temperatura 36
Figura 32: Detalle del Script de la gráfica temperatura 36
Figura 33: Detalle del código de la tabla medicación 37
Figura 34: Script de la tabla medicación 37

Lista de tablas

Tabla 1: Planificación inicial 11
Tabla 2: Planificación final. 12

1. Introducción

1.1 Contexto y justificación del Trabajo

1.1.1 Contexto

Según cifras del INE [1] (Instituto Nacional de Estadística) se produjeron en España durante el año 2017, un total de 393.181 nacimientos, llegando en términos estadísticos al 37.22 por mil, es decir, 37 nacimientos por cada 1000 mujeres. Siendo la media de edad de 32 años. Sin embargo, desglosando estos datos por edades se observa como en la mayoría de los casos la ratio de maternidad oscila entre los 25 y 40 años.

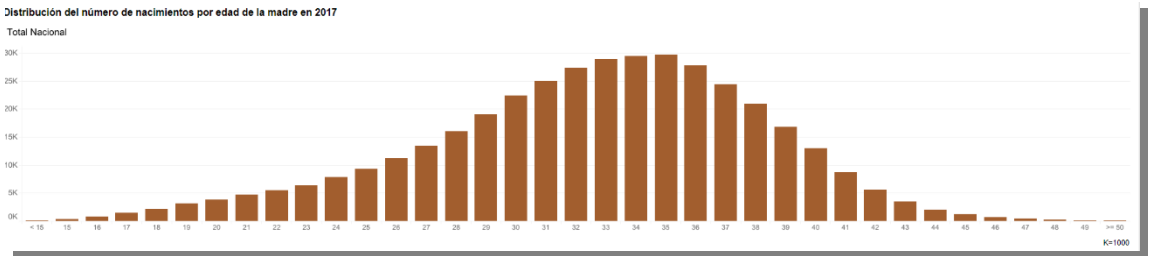

Figura 1: Media de edad de maternidad según el INE (2017) [2]

Atendiendo a las mismas fuentes [3], en este caso, sobre el uso de Internet entre la población, se puede observar, como el mayor uso de las nuevas tecnologías por parte de las mujeres recae en el intervalo comprendido entre 16 y 44 años.

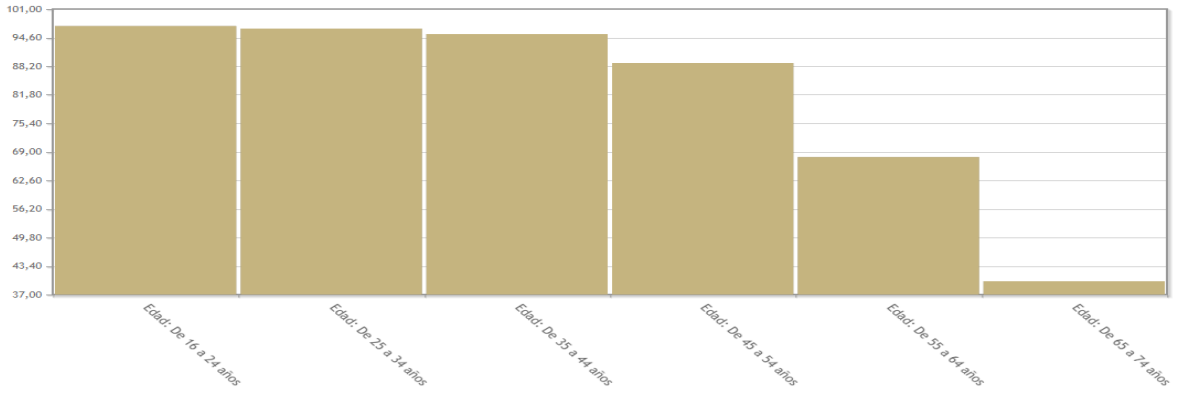

Figura 2: Uso de Internet por rango de edad (Mujeres) según el INE (2017) [4]

Combinando ambos datos objetivos, a los que habría que sumar los datos de los progenitores masculinos, se podría determinar que existe un amplio sector de la población que pasaría a considerarse público objetivo.

1.1.2 Oferta Actual.

A continuación, se lista una pequeña muestra de las aplicaciones ofertadas actualmente:

- The Wonder Week:

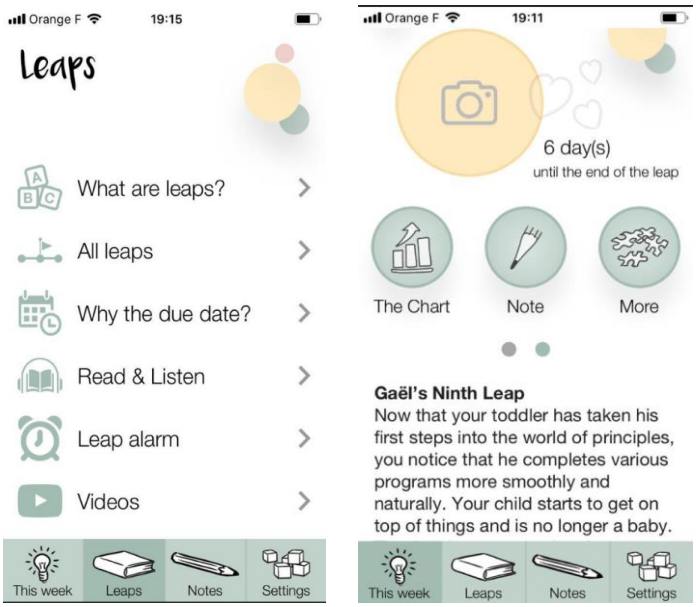

Figura 3: Imágenes aplicación The Wonder Week [5].

Se trata de una aplicación para sistemas iOS y Android que se centra en el desarrollo cognitivo del bebé, descubriendo a los progenitores en qué etapa de desarrollo se encuentra su hijo. Actualmente solo se oferta en inglés.

- Baby Tracker:

Figura 4: Imágenes aplicación Baby Tracker [6].

Ofrece el seguimiento diario del bebé en lo que se refiere a hábitos, tanto alimenticios como de patrones de sueño. Está disponible para Android y para iOS.

- Baby Connect:

Figura 5: Imágenes aplicación Baby Connet [7].

Aplicación de dispositivo, que cuenta con soporte para iOS, Android y Kindle Fire. El aplicativo hace un registro exhaustivo de la información diaria del niño, como puede ser alimentación, actividades, estados de ánimo, etc. Incluye soporte para fotos e intercambio de mensajes.

- Cry translator:

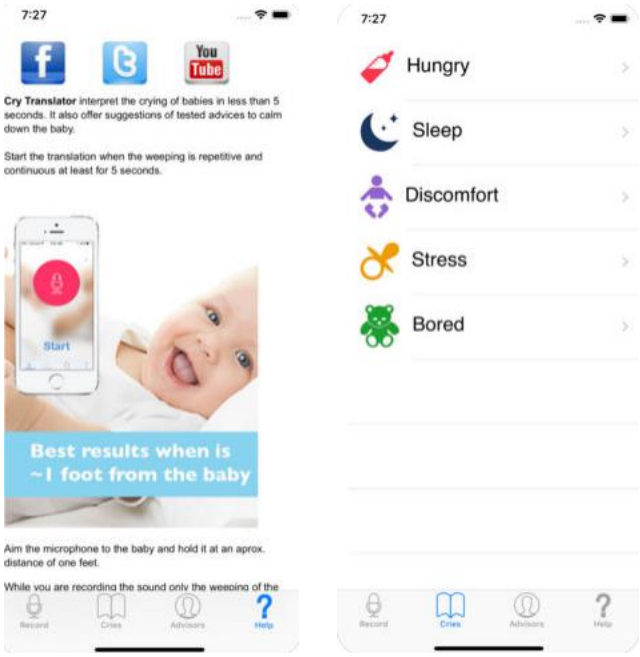

Figura 6: . Imágenes aplicación Cry Translator [8].

App exclusiva de sistemas iOS en la que se analizan los llantos del recién nacido y se aconsejan métodos para calmarlo.

- LactMed:

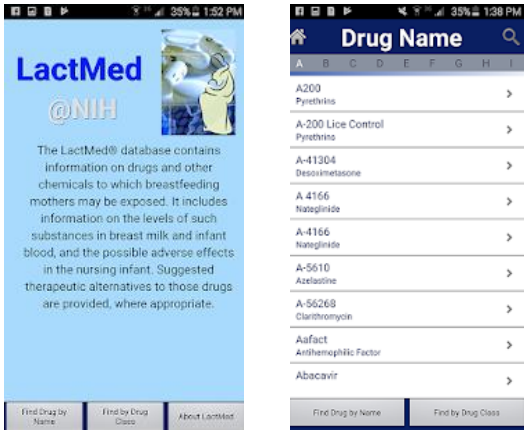

Figura 7: Imágenes aplicación LactMed [9].

Es una extensa base de datos sobre medicamentos y sustancias nocivas, que pueden ser transmitidas de la madre al lactante. Ofrece a su vez una serie de alternativas más seguras. Es ofertada tanto para sistemas iOS como Android.

- iPedriatic:

Figura 8: Imágenes aplicación LactMed [10].

Se trata de un aplicativo desarrollado por pediatras del hospital San Joan de Déu, en el que se ofrecen respuestas a las preguntas más frecuentes que surgen durante el primer año de vida. La app tiene soporte para iOS y Android.

1.1.3 Carencias del mercado.

Como se puede observar en el apartado anterior existen una gran variedad de productos ofertados en el mercado, en torno al desarrollo y cuidado de los recién nacidos. Sin embargo, ninguna de estas aplicaciones ofrece un soporte multiplataforma y/o multidispositivo, como el que puede llegar a brindar una aplicación web.
Bien es cierto que la mayoría de estos productos dan soporte a los sistemas operativos más populares. Pero qué pasa si el usuario no dispone de un smartphone o una Tablet con dichos sistemas, o si su dispositivo habitual se trata de un ordenador.

Según el *Global Mobile Consumer Survey*, 2017 (España) desarrollado por Deloitte [11], el 94% de los españoles afirman haber usado un dispositivo móvil en las últimas 24 horas. Así mismo, el 66% reconoce el uso de un ordenador en condiciones similares. Por último, el 52% confiesa el uso de una *Tablet*.

Siguiendo con el estudio de Deloitte, se puede observar la penetración de los distintos dispositivos electrónicos en los últimos años.

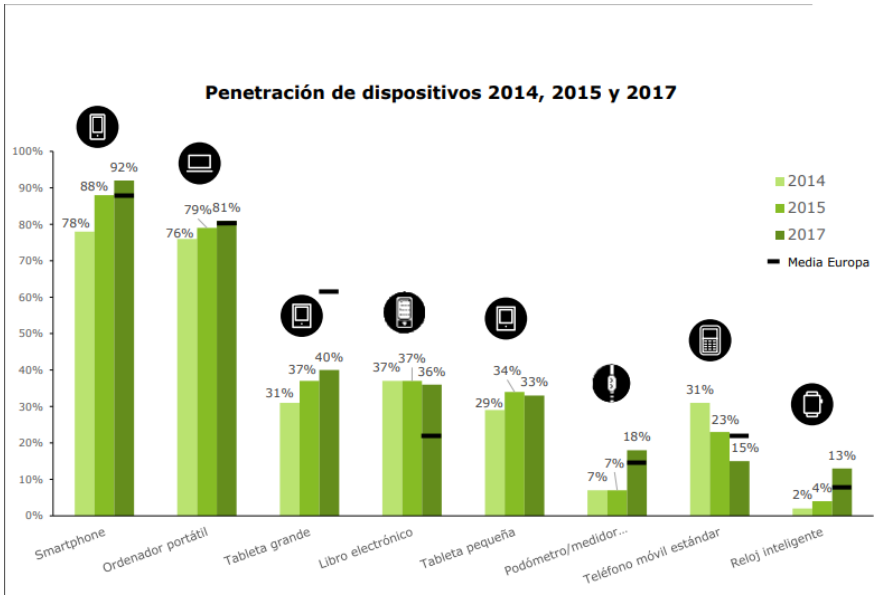

Figura 9: Penetración de dispositivos 2014, 2015, 2017 según Deloitte [11].

Por otro lado, y citando el Informe Ditendria: Mobile en España y en el mundo 2018 [12], el 90% del tiempo utilizado por los usuarios de dispositivos móviles, es dedicado al uso de aplicaciones, frente al 10% destinado al uso del navegador web.

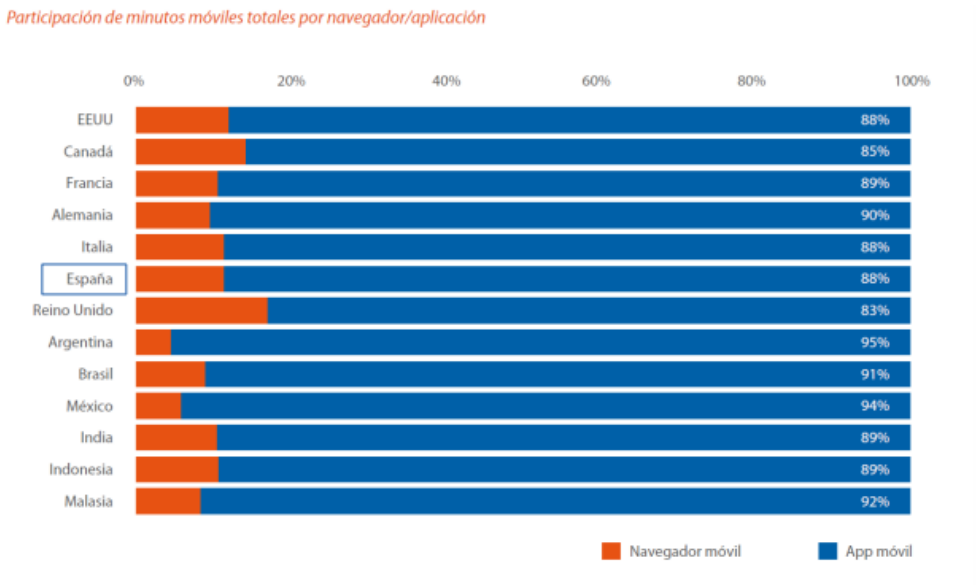

Figura 10: Comparativo de uso de aplicaciones frente a navegador en móviles, Ditendria [12]

En definitiva, después de que se hayan analizado los datos y tendencias del mercado actual y tras una reflexión, se llega a la conclusión de que el desarrollo de una aplicación web, que cubra las necesidades de los recién nacidos, así como de sus progenitores, puede resultar de utilidad para un gran *target* potencial. Así mismo, se considera que gracias a la multiplataforma y/o multidispositivo que ofrece este tipo de desarrollo, se espera llegar a cubrir un amplio sector que parece olvidado por los servicios que ofrece el mercado actual.

1.2 Objetivos del trabajo.

En un primer momento, se planteó la construcción del proyecto sobre la premisa, de utilizar la tecnología para ofrecer un servicio útil. Esto se tradujo, en buscar un tema que pudiera realmente cubrir una necesidad social. Bajo esta proposición y después de una investigación sobre varios posibles proyectos, se concluyó que el tema podía ser un candidato válido. Como se ha apuntado anteriormente la oferta en el mercado de aplicaciones sobre este tema, aunque amplia, no cubría las necesidades básicas de almacenar todos los datos relevantes, generados en el día a día, sobre un único aplicativo.

Por otro lado, estudiando las interfaces de las distintas aplicaciones se observó, salvo alguna excepción, que eran demasiado recargadas y en términos de usabilidad, quizás demasiado complicadas. En este sentido, aplicando la navaja de Ockham si es posible a este aspecto, se llegó al convencimiento de la importancia de desarrollar un aplicativo con una interfaz de usuario lo más clara y simple posible, en favor de dicha usabilidad.

En conclusión, en cuanto a objetivos, la finalidad principal del proyecto era la de dotar a la comunidad de un producto que fuera fácil de usar y que diera un servicio básico.

1.3 Enfoque y método seguido.

Existen distintas metodologías a la hora de enfrentar un proyecto de desarrollo de aplicaciones, o de software en general. Estas metodologías se dividen en dos grandes grupos.

Por un lado, se encuentran las metodologías tradicionales, que son aquellas basadas en las distintas disciplinas de ingeniería. Se centran en una planificación del trabajo en base a unos requisitos cerrados, que no son modificables, lo que hace de ellas que sean más rígidas. El desarrollo se lleva a cabo cuando todo está correctamente especificado. Conllevan un gran trabajo de documentación. Tanto es así, que Martin Fowler llego a denominarlo como "burocrático" en su informe del 2000, *The New Methology* (La nueva Metodología) [13].

En la década de los 90 aparece lo que se conoce como metodologías ágiles. Este tipo de métodos son más flexibles o como su nombre indica más ágiles. Partiendo de una pequeña planificación inicial se centran más en la experiencia del desarrollo y se van adaptando a las necesidades del proyecto. Otra característica de este tipo de métodos es, que, en lugar de estar orientados al proceso, están orientados a las personas.

Un punto que se analizó antes de tomar la decisión sobre la estrategia a seguir fue el factor temporal. Bien es cierto que existían ciertos tiempos de entrega concretos, pero la planificación temporal daba opción a una temporización más flexible en función de las necesidades de desarrollo.

Conociendo los distintos tipos de metodologías de desarrollo de software, extrapolándolo a la implementación de aplicaciones web y teniendo en cuenta el punto anterior, se decidió que la mejor estrategia era la que coincidía con el punto de vista de los desarrollos agiles. En este sentido, se optó, a partir de una propuesta inicial de requisitos, por el desarrollo de una primera fase de diseño, con diferentes elementos que aportaran una visión general del aplicativo y su funcionalidad.

Durante esta fase se tomaron decisiones especialmente concernientes a la interfaz de usuario de la aplicación. Concretamente, se buscó una rueda cromática que se adaptara al contexto. Se decidió el tipo de fuente más apto para dotar a la interfaz de un carácter más amigable, así como, una segunda tipografía en contraste con la primera. Se hizo una primera propuesta a nivel visual traducida en los diferentes *wireframes* y prototipos, que siguiera la premisa anteriormente mencionada de simplicidad para lograr una usabilidad lo más sencilla posible. Por último, en cuanto a funcionalidad se desarrolló tanto un diagrama UML en el que basar el desarrollo de la futura base de datos, como un diagrama de interacción entre los distintos elementos del aplicativo.

En la siguiente fase, la de desarrollo, se siguió un esquema mucho más iterativo, se siguió la técnica de prueba y error, es decir, se implementaban las funcionalidades y se realizaban pruebas sobre este desarrollo, con el objetivo de ir acoplando todas las partes del aplicativo. Durante este periodo también se tomaron decisiones en cuanto a los requisitos, que se detallaran más adelante, adaptando el primer diseño a los nuevos requerimientos.

Finalmente, se realizaron una serie de pruebas de funcionabilidad.

Una vez finalizado el trabajo, se llegó a la conclusión que, gracias a la estrategia elegida, el crecimiento del proyecto se vio favorecido, ya que se tuvo la oportunidad de realizar cambios que dieron lugar a nuevas funcionalidades que nacieron del conocimiento adquirido durante el

desarrollo de este. Del mismo modo, servicios que en un primer momento parecían esenciales pasaron a ser absolutamente prescindibles, bien porque no aportaban ningún beneficio sobre el objetivo, o bien, porque se alejaban del enfoque que se había decidido.

1.4 Planificación del trabajo.

1.4.1 Recursos, lenguajes, APIs y *Frameworks*.

Software

Para la realización del proyecto se ha requerido de una serie de recursos informáticos. Estos recursos se podrían clasificar en básicos, aquellos esenciales para el desarrollo del aplicativo, y secundarios, es decir, aquellos necesarios para realizar otro tipo de tareas complementarias, como podrían ser los diagramas UML, el tratamiento fotográfico, etc.

El segundo tipo de recursos asociados al desarrollo de ciertas tareas, se describirán en sus apartados correspondientes. En este espacio se analizarán, por tanto, los denominados recursos básicos.

- Adobe Dreamweaver CC, 2019: es el IDE (Entorno integrado de desarrollo) enfocado al diseño y desarrollo web por excelencia. Se estima que cubre el 90% del mercado de este tipo de aplicaciones. Tiene versiones para sistemas Windows o Mac. Actualmente forma parte de la *suite* Creative Cloud de Adobe, en la que podemos encontrar un gran número de programas de edición que abarcan diversos campos, como pueden ser el fotográfico, video, etc.
- XAMPP: es un paquete *open source* que nos permite recrear un servidor en el ordenador en modo local, gracias al cual, se puede ejecutar el código desarrollado sin necesidad de uno externo. Este paquete incluye diversas herramientas como Apache, MySql, Filezilla, Mercury, Tomcat, etc. En el caso de la aplicación desarrollada, es especialmente relevante puesto que este paquete es el que da soporte a la ejecución del código Php y la base de datos, que son esenciales.

Lenguajes

Para llevar a cabo el desarrollo de la aplicación se necesitará utilizar una serie de lenguajes de programación, tanto del lado del cliente, como del lado del servidor.

Pasamos a describir los utilizados:

- HTML, Lenguaje de Marcas Hipertextual (*Hyper Text Markup Language*): Como su nombre indica se trata de un lenguaje basado en marcas o etiquetas. Se utiliza para organizar el contenido de la página dotándolo de una estructura. Es el lenguaje base de cualquier desarrollo web. Ha ido evolucionando desde su primera versión en 1991 hasta la actual HTML 5 de 2014.
- CSS, Hoja de Estilos en Cascada (*Cascading Style Sheets*): es el lenguaje que se utiliza para dar formato o estilo a la página. Permite modificar los valores predeterminados de las etiquetas HTML a través de clases o identificadores modificando valores que

tienen que ver con el diseño, como pueden ser la anchura de los elementos, el color, la fuente, etc. La independencia entre el HTML y el CSS permite entre otras cosas modificar todo el diseño de la página sin alterar el contenido.

- JavaSript, es un leguaje que permite el uso de *Script* en el lado del cliente. En el caso de la aplicación se ha utilizado para la gestión de eventos principalmente en los formularios de registro y *login*, permitiendo validar la información introducida por el usuario. Hoy en día, es un lenguaje con muchas posibilidades utilizado generalmente para dotar de dinamismo a las páginas.
- PHP, es un lenguaje del lado del servidor desarrollado para generar contenido dinámico. En el caso de la aplicación, resulta especialmente interesante puesto que permite la interacción con MySql.
- SQL, es un lenguaje especifico utilizado por sistemas de gestión de base de datos relacionales. Se utiliza para realizar consultas o cambios en la BBDD.

BootStrap

Se trata de un framework de código abierto basado en HTML, CSS y JavaScript. Esta especialmente dirigido al diseño web. Sus principales características son que sigue los principios de diseño adaptativo (*Responsive Web Design*), así como el *mobile-first*.

GoogleChart

Es una API desarrollada por Google que permite la incorporación de gráficos dinámicos de una manera sencilla.

1.4.2 Tareas.

Como se comentó en el apartado dedicado a la metodología, el proyecto se dividió en varias fases. Cada una de ellas con unas tareas específicas. Que se exponen a continuación:

<u>Fase de Diseño.</u>

En esta etapa las tareas se centran, principalmente, en el diseño de la interfaz de usuario del aplicativo. En base a esto, se deciden las siguientes tareas:

- Búsqueda de una paleta cromática adecuada.
- Decisión sobre las fuentes tipográficas a utilizar.
- Diseño y creación de *Wireframes* y prototipos.
- Diagrama de interacciones.

Por otro lado, se hace necesario dar forma a la base de datos, que dará servicio a toda la aplicación. A este efecto, se diseñó un diagrama UML como sustento de ésta.

<u>Fase de creación.</u>

El objetivo de este periodo es la creación de la base de datos, que soporta el peso del aplicativo. Así como, la base estructural de la aplicación, es decir, la implementación de las estructuras comunes, como pueden ser la cabecera, el navegador, etc. Para ello se hará uso de

lenguajes como Html y Css. Además, se usará el *framework* Bootstrap, para dotar a la aplicación de un carácter adaptativo a diferentes plataformas.

Fase de desarrollo.

Durante este tiempo el proceso se centra en la implementación de las distintas funcionalidades. Estas funcionalidades se dividieron a su vez en tres secciones:

- Creación de formularios.
- Gestión de alarmas.
- Consulta de datos

Como se comentaba anteriormente, esta fase tiene un cierto carácter iterativo. Durante este periodo se van realizando pruebas de funcionalidad. Gracias a estas pruebas se llega a la conclusión que algunos de los requisitos no son válidos y se proponen alternativas. De este modo, las tareas principales pasan a ser:

- Creación de formularios.
- Exposición de datos.
- Consulta de datos específicos.

En los apartados dedicados a esta fase se explicarán detalladamente los motivos de estos cambios.

Fase de pruebas.

En este caso se analiza detalladamente el correcto funcionamiento de la aplicación. Poniendo especial énfasis en los enlaces que transportan de una página a otra del aplicativo.

1.4.3 Planificación.

En un primer momento se creó una primera propuesta de planificación:

	Objetivo	Inicio	Fin	Dias
DISEÑO	Flujos de iteracion	07/03/2019	09/03/2019	3
	wireframes	10/03/2019	12/03/2019	3
	BBDD (UML)	13/03/2019	16/03/2019	4
CREACIÓN	BBDD	17/03/2019	21/03/2019	5
	Estructura aplicación	22/03/2019	27/03/2019	5
DESARROLLO REGISTROS, LOGIN, LOGOUT	Registros	27/03/2019	31/03/2019	5
	Login	01/04/2019	05/04/2019	5
	Test/Solución errores	06/04/2019	09/04/2019	4
PEC 2	**Entrega Hito Intermedio**			
DESARROLLO FORMULARIOS	Habitos	11/04/2019	13/04/2019	3
	Medicos	14/04/2019	16/04/2019	3
	Comentarios	17/04/2019	18/04/2019	2
	Test/Solución errores	19/04/2019	20/04/2019	2
DESARROLLO GESTION ALARMAS	Habitos	21/04/2019	24/04/2019	4
	Medicos	25/04/2019	28/04/2019	4
	Test/Solución errores	29/04/2019	30/04/2019	2
DESARROLLO CONSULTA DATOS	Habitos	01/05/2019	03/05/2019	3
	Medicos	04/05/2019	06/05/2019	3
	Comentarios	07/05/2019	08/05/2019	2
	Test/Solución errores	09/05/2019	10/05/2019	2
PRUEBAS FINALES	Comprobacion del codigo final	11/05/2019	18/05/2019	8
MEMORIA FASE 1	Recopilacion de documentacion generada y comienzo de desarrollo primer borrador	19/05/2019	28/05/2019	10
PEC 3	**Entrega Hito Intermedio**			
MEMORIA FASE 2	Finalizar memoria	30/05/2019	04/06/2019	6
PRESENTACION	Preparacion	05/06/2019	07/06/2019	3
	Grabacion	08/06/2019	11/06/2019	4
PEC 4	**Entrega Final**			

Tabla 1: Planificación inicial

Como se comentaba anteriormente, durante la fase de desarrollo surgieron una serie de cuestiones que llevaron a la modificación de los requisitos iniciales. Esta decisión llevo a su vez, a un cambio en la planificación inicial y, por lo tanto, en la temporización. A continuación, se expone la planificación final.

	Objetivo	Inicio	Fin	Dias
DISEÑO	Flujos de iteracion	07/03/2019	09/03/2019	3
	wireframes	10/03/2019	12/03/2019	3
	BBDD (UML)	13/03/2019	16/03/2019	4
CREACIÓN	BBDD	17/03/2019	21/03/2019	5
	Estructura aplicación	22/03/2019	27/03/2019	5
DESARROLLO REGISTROS, LOGIN, LOGOUT	Registros	27/03/2019	31/03/2019	5
	Login	01/04/2019	05/04/2019	5
	Test/Solución errores	06/04/2019	09/04/2019	4
PEC 2	**Entrega Hito Intermedio**			
DESARROLLO FORMULARIOS	Habitos	11/04/2019	13/04/2019	3
	Medicos	14/04/2019	16/04/2019	3
	Comentarios	17/04/2019	18/04/2019	2
	Test/Solución errores	19/04/2019	20/04/2019	2
DESARROLLO EXPOSICION DE DATOS	Habitos	21/04/2019	24/04/2019	4
	Medicos	25/04/2019	28/04/2019	4
	Test/Solución errores	29/04/2019	30/04/2019	2
DESARROLLO CONSULTA DATOS CONCRETOS	Habitos	01/05/2019	03/05/2019	3
	Medicos	04/05/2019	06/05/2019	3
	Comentarios	07/05/2019	08/05/2019	2
	Test/Solución errores	09/05/2019	10/05/2019	2
PRUEBAS FINALES	Comprobacion del codigo final	11/05/2019	18/05/2019	8
MEMORIA FASE 1	Recopilacion de documentacion generada y comienzo de desarrollo primer borrador	19/05/2019	28/05/2019	10
PEC 3	**Entrega Hito Intermedio**			
MEMORIA FASE 2	Finalizar memoria	30/05/2019	04/06/2019	6
PRESENTACION	Preparacion	05/06/2019	07/06/2019	3
	Grabacion	08/06/2019	11/06/2019	4
PEC 4	**Entrega Final**			

Tabla 2: Planificación final.

1.4.4 Entregas.

Se enumeran a continuación, los diversos productos desarrollados dentro de su correspondiente entrega.

Hito I.

- ***Wireframe* y prototipos**: Primeros bocetos del diseño de la aplicación, enfocado tanto para web como para dispositivos móviles.

- **Diagrama de iteración**: Diagrama que muestra los flujos de iteración de la aplicación.
- **Diagrama UML**: Diseño de la base de datos que dará soporte a la aplicación.
- **Documento SQL**: Archivo con los comandos de creación de la base de datos. Se adjuntan algunas entradas de prueba.
- **Aplicación web**. Primera entrega de la página web.
- **Tema de color**
- **Fuentes tipográficas** de la aplicación web.

Hito II.

- **Aplicación web:** se entrega el aplicativo finalizado.
- **Base de datos:** se adjunta la base de datos finalizada, con algunas entradas de prueba**.**
- **Índice de la memoria:** primera propuesta de desarrollo de la memoria.

Entrega Final:

- Aplicación web.
- Memoria.
- Presentación.

1.5 Breve sumario productos obtenidos.

La aplicación web consta de las siguientes páginas:

- home.html: la correspondiente página de inicio de la aplicación tiene un carácter más publicitario que el resto.
- Login.php: identificación por parte del usuario
- Registro.php: registro de nuevo usuario
- registroBebe.php: registro de un nuevo bebé asociado a un usuario
- indiceBebe.php: permite elegir entre diversos bebés registrados bajo un mismo usuario
- datosMedicos.php: visualiza los datos médicos de un bebé concreto
- habitos.php: visualiza los datos sobre hábitos de un bebé
- conexión.php: inicia la conexión con la base de datos
- logout.php: finaliza la sesión de usuario
- formularios datos médicos: esta carpeta guarda todos los formularios relativos a datos médicos que son los siguientes:
    - formCitasMedicas.php
    - formGeneral.php
    - formMedicacion.php
    - formMedicamento.php
    - formMedico.php
    - formTemperatura.php

- formularios hábitos: carpeta que contiene todos los formularios referentes a hábitos:
    - formPecho.php
    - formBiberon.php
    - formComidas.php
    - formDeposiciones.php
    - formGeneral.php
    - formSueño.php
- gráficos: carpeta que contiene todas las páginas con gráficos específicos:
    - citasMedicas.php
    - comidas.php
    - deposiciones.php
    - estadoGeneral.php
    - general.php
    - medicación.php
    - suenio.php
    - tempCorporal.php
- CSS: contiene la hoja de estilos
    - style.css
- js: contiene el archivo JavaScript
    - código.js
- fotos: contiene las fotos y el banner
    - banner.jpg
    - churu2B.png
    - churu3B.png
    - churuB.png
    - ileB.png
    - ile2B.png

Además, también se cuenta con un archivo de la base de datos llamado webtfg.sql.

1.6 Breve descripción de los otros capítulos de la memoria.

En los siguientes capítulos se procederá a describir en profundidad, las tareas y decisiones que se irán desarrollando a lo largo del proceso de creación de la aplicación. Así mismo, se describirán las tecnologías, herramientas o lenguajes necesarios para llevar a cabo estas tareas.

Se comenzará por la fase de diseño, donde se explicarán los procesos y decisiones que conlleva el diseño de la interfaz de usuario y la definición de funcionalidades, que dará lugar al diagrama de clases UML.

Se pasará a la fase de creación, que estará dedicada a lo concerniente a la BBDD. Así como a la estructura base que se utilizará como armazón del aplicativo.

Se continuará, con la fase de desarrollo, donde tomará especial relevancia la funcionalidad de la aplicación. Esta fase estará compuesta por la creación de formularios, gráficos y demás elementos que conforman la aplicación.

Finalmente, se describirán los distintos procesos de pruebas a los que se someterá el producto final. Tomando especial protagonismo los estándares de usabilidad, gracias a los cuales, se garantiza el acceso a personas con diversidad funcional.

2. Fase de diseño

La fase de diseño se plantea, como hemos comentado anteriormente, como un esqueleto que vertebra todo el proceso de desarrollo de la aplicación web. Pese a esto, no se define como un planteamiento fijo, sino como una serie de pautas predefinidas que pueden ser modificadas en cualquier momento del proceso.
Bajo esta condición y la premisa inicial, de buscar la simplicidad y la mayor utilidad posible, durante esta fase se hace hincapié en el diseño de la interfaz de usuario. De igual forma, se hizo un estudio sobre los requerimientos del proyecto en busca de cual debía ser la finalidad de éste. En base a estas conclusiones, se diseñó un diagrama UML para la futura implementación de la BBDD.
A continuación, se desarrolla cada tarea de esta etapa de forma más específica.

2.1 Interfaz.

2.1.1 Color.

Durante el estudio de las aplicaciones similares se observó que muchas de ellas estaban basadas, a nivel cromático, en tonos rosas y azules. Como es sabido, este tipo de tonalidades están asociadas a lo relativo al sexo del individuo. El rosa a lo femenino y el azul a lo masculino. También se comprobó como otras de estas herramientas huían de este tipo de tópicos, para caer en otro semejante utilizando colores verdes más neutrales.

Se emprendió, por tanto, la búsqueda de un nuevo matiz que dotara a la aplicación de un aspecto singular, dentro de la oferta existente. Así pues, se llegó a la conclusión de que el espectro morado podría cubrir las necesidades del proyecto. Este tipo de color resulta de la mezcla del rojo y el azul, que lo dota de una cierta dualidad que en muchas ocasiones se asocia a lo relativo a la feminidad frente a la masculinidad. Del mismo modo, a nivel cognitivo se asocia a la armonía, la estabilidad o la paz, entre otras.

Una vez decidida la tonalidad base, se pasa a buscar una rueda monocromática. En este sentido, se decide utilizar únicamente dos tonos que contrasten lo suficiente. De este modo, se logra dar una cierta sensación de sencillez.

La rueda cromática elegida es:

Figura 11: Muestra cromática.

De los cuales se decide utilizar el #3E2C7F y el #B9A5FF, que van cambiando de rol entre principal y secundario según la parte de la estructura en la que se encuentran.

Adobe Color

Se trata de una herramienta online que permite la creación de ruedas cromáticas a partir de varias opciones. Entre estas opciones destacan las distintas reglas de armonía, como pueden ser análogas, monocromáticas, triada, etc. El modo de color, donde se puede elegir entre RGB, CMYK, etc. Además, se puede crear la rueda a partir de un color concreto.

2.1.2 Fuentes tipográficas.

Una vez decidido el tema de color, el siguiente paso lógico es la búsqueda de la tipografía. Existen diversos estilos tipográficos y al igual que ocurría con los colores, existe una infinidad de estudios sobre los efectos que éstas transmiten.

En este caso, se buscaba romper, relativamente, con la armonía que generaba el tono cromático, por tanto, se propuso buscar una fuente de tipo *Script*. Entre las propiedades de este estilo de fuentes se encuentra la elegancia, la creatividad o la seducción. Y especialmente, el afecto y la feminidad. Además, puesto que el público objetivo de la aplicación estaría formado mayoritariamente por mujeres en pleno proceso de maternidad, parecía a priori que se adaptaba al contexto.

Se consideraron algunas opciones sobre este estilo, llegando a la conclusión de que la Kaushan Script podría resultar idónea. Ya que, además, parecía aportar un cierto componente alegre o divertido que rompiera con la seriedad que en ocasiones pueden transmitir las composiciones simples.

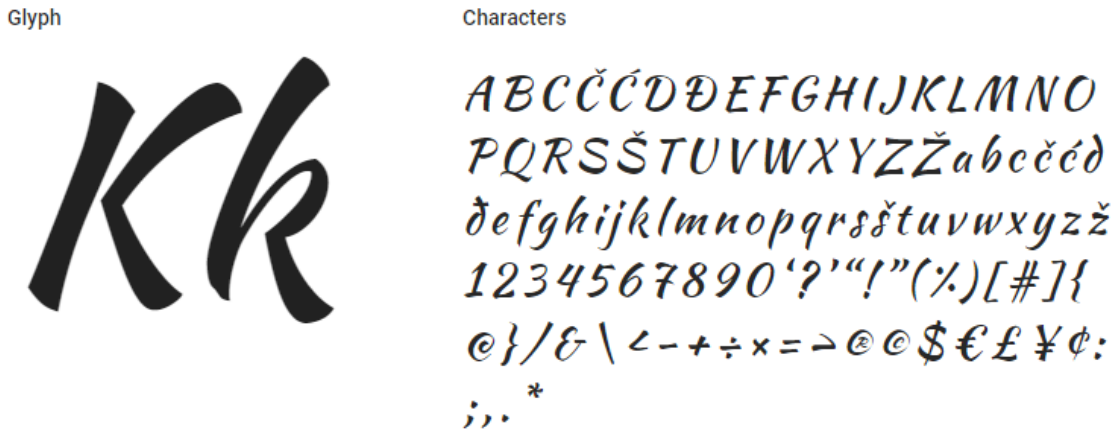

Figura 12: Muestra de la fuente Kaushan Script [14]

Seguidamente, se comenzó la búsqueda de una fuente secundaria que contrastara con la primera. La función de esta tipografía estaría básicamente limitada a la página principal donde se describiría la funcionalidad y las opciones que ofrecería la aplicación.

Por tanto, se necesitaba que transmitiera una cierta objetividad, estabilidad y seguridad. Con este propósito, se tomó la decisión de que el estilo que más se adaptaba a estos aspectos era el *san-serif*, en concreto la fuente Meera Inimai.

Figura 13: Muestra de la fuente Meera Inimai [15]

FontJoy

Es una herramienta web que permite buscar fuentes tipográficas que contrasten entre sí. Esto puede realizarse de manera aleatoria, o bien, introduciendo diversos parámetros. Las fuentes sugeridas se encuentran en el repositorio público de Google, GoogleFont.

2.1.3 *Wireframes*.

Según los autores Javier Cuello y Jose Vittone en su libro Diseñando apps para móviles, “Hacer *wireframes* en etapas tempranas supone un ahorro de costes y tiempo, permitiendo evaluar aspectos como la navegación e interacción para sentar unas bases sólidas que después den paso al nivel visual” (pag 66) [16].

En este sentido, se optó por realizar pequeños bocetos en papel con varias ideas sobre la composición de la página. Una vez estudiadas las distintas posibilidades, se decidió, que la mejor opción, en base al concepto de simplicidad, era la de optar por un diseño tradicional. Lo que supone, además, que al ser un diseño bastante aceptado reducirá notablemente la curva de aprendizaje que el uso de cualquier aplicación conlleva.

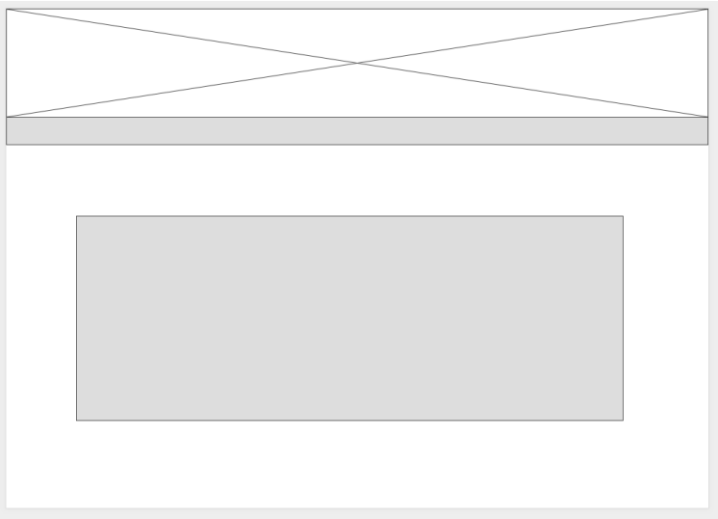

Figura 14: Diseño de *wireframes*

Este diseño se compondrá de un banner o cabecera, que tendrá dos funciones claras, la de ubicuidad, es decir, facilitar la localización al usuario del espacio (aplicación) en la que se encuentra. Por otro lado, dotar a todo el aplicativo de una cierta coherencia o hilo conductor en cuanto al diseño, ya que se trata de uno de los elementos fijos de las distintas pantallas de la aplicación.

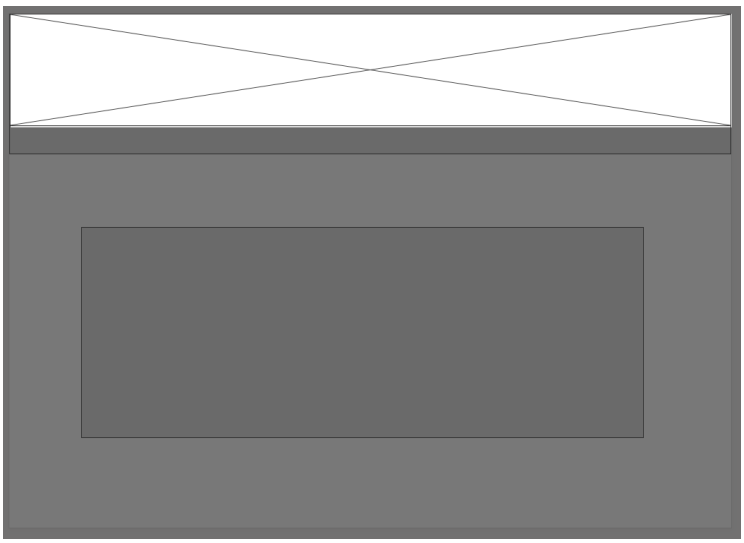

Figura 15: Detalle cabecera diseño *wireframes*

Por otro lado, tendríamos un navegador horizontal, destinado a enlazar las distintas páginas de la web. Este elemento también podría considerarse como un elemento fijo, pese a que su contenido (enlaces) ira variando dependiendo de la página en la que se encuentre y los enlaces que esta requiera.

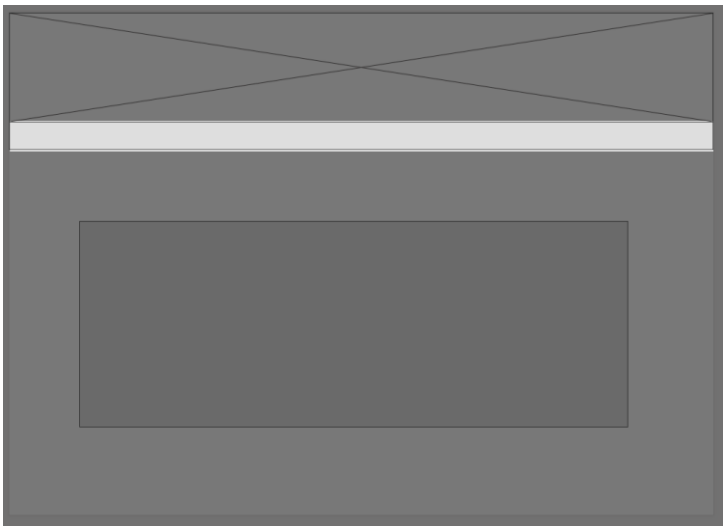

Figura 16: Detalle navegador diseño wireframes

Por último, una zona central, que será el espacio destinado, tanto a la recogida de datos, como a la exposición de estos. Por consiguiente, la zona donde cobra protagonismo realmente toda la funcionalidad de la aplicación.

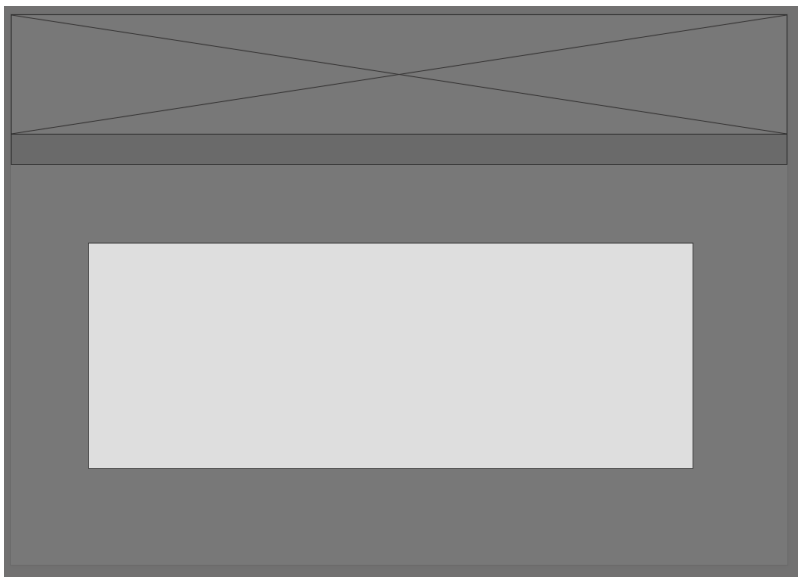

Figura 17: Detalle zona central diseño wireframes

Wireframe.cc

Es una aplicación web que permite el diseño de *wireframes* sencillos, en base a objetos geométricos. Ofrece la posibilidad de modificar la retícula, lo cual resulta de bastante utilidad para lograr una proporcionalidad y distribución de los elementos.

2.1.4 Prototipos.

Una vez definido el concepto de Interfaz a través del *wireframe*, y decidido el color y la tipografía, se paso a crear una serie de prototipos de lo que se esperaba fuera la imagen final del aplicativo.

En este caso, como la idea era desarrollar una aplicación que se fuera adaptando a los diversos dispositivos, se crearon propuestas tanto para pantallas de ordenador como para móviles.

Los diseños propuestos se pueden dividir en tres categorías.

En primer lugar, la página de inicio (*home*) cuya función principal es la de atraer a futuros usuarios. Por este motivo, la zona central, pasa a tener un rol más publicitario. Se debe tener en cuenta que es la única parte visible en caso de que el usuario no esté registrado.

La zona central se divide en dos secciones. La superior muestra imágenes con la intención de captar la atención del público. Mientras que la inferior ofrecerá información sobre las posibilidades y servicios que ofrecerá la app.

Figura 18: Prototipo página inicio

En segundo lugar, tenemos las paginas destinadas a la recogida de datos. Este tipo de diseño será el más común dentro de la aplicación, ya que gran parte, por no decir toda, la funcionalidad se basa en la recogida de estos. En este caso, la zona central será ocupada por los distintos formularios.

Figura 19: Prototipo página de formulario

Finalmente, la última propuesta es la que da soporte a la muestra de la información de una forma visual a través de gráficos. Quizás sea este diseño en donde se puede apreciar de manera más clara la adaptabilidad buscada. La intención es pasar de una visión horizontal, la cual es posible gracias a la anchura de los monitores convencionales, a un visionado vertical más apropiado de los dispositivos móviles.

Figura 20: Prototipo página muestra de datos

Adobe InDesing CC 2019.

Se trata de un programa perteneciente a la *suit* de Adobe, Creative Commons. Este está dedicado a la maquetación. Pese a no estar orientado al diseño de prototipos, y existan un sinfín de productos especializados en el mercado, mucho más específicos, se considero, que gracias a la gran cantidad de herramientas que ofrece en cuanto a estructuración, nos proporcionaría los recursos adecuados para lograr una organización de las secciones lo más proporcionada posible.

Adobe Illustrator CC 2019.

También se trata de un programa perteneciente a la *suit* de Adobe. En este caso está enfocado al trabajo y tratamiento de imágenes vectoriales. Se ha utilizado para montar los diseños sobre los dispositivos.

2.1.5 Interacción.

Llegados a este punto, el diseño de la interfaz estará casi completado. Se decide, realizar un pequeño diagrama de interacciones, a modo de visualizar los flujos o transiciones entre las distintas páginas. Por otro lado, también servirá para destacar de manera inicial los enlaces que deberán contener las barras de navegación en las distintas partes del aplicativo.

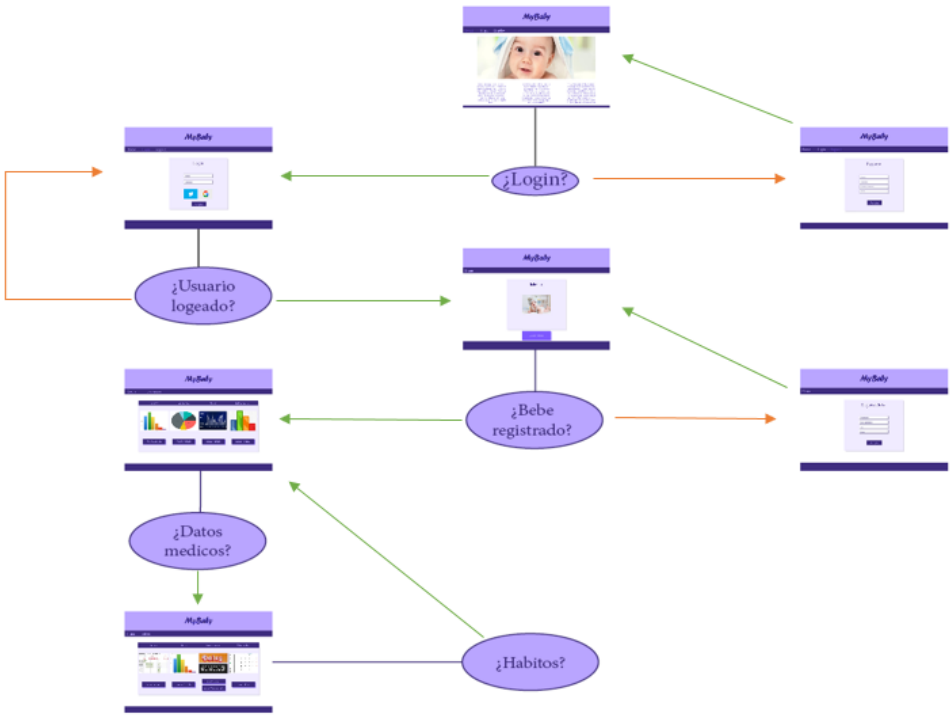

Figura 21: Diagrama de interacción

2.2 Base de datos.

La aplicación se basa en la recogida de datos y en su exposición, por tanto, es fundamental la creación de una base de datos que le dé soporte. El primer paso es el de identificar las diversas clases que la compondrán y sus respectivos atributos, para continuar con el diseño del diagrama de clases.

2.2.1 Requisitos.

Los usuarios de la aplicación deben registrar a sus hijos. De los usuarios se quiere registrar un identificador, su nombre, una contraseña y su dirección de correo electrónico. Cada usuario podrá registrar a uno o más niños, pero cada recién nacido tendrá que estar registrado por un único usuario. De los recién nacidos se quiere saber un identificador numérico, su nombre y su fecha de nacimiento.

Por cada bebe se quiere guardar una serie de datos.

En lo referente a las comidas se deberá saber la fecha y la hora. En el caso de que su alimentación se base en la leche materna interesara saber si se trata de la mama derecha o de la izquierda. En caso de que sea toma de biberón la cantidad ingerida.

Sobre los hábitos de sueño interesa el día, la hora de inicio y la de fin.

En cuanto a las deposiciones, almacenaremos la fecha y la hora, así como el estado del pañal.

También será necesario el control del estado general guardando registro temporal y su estado de ánimo que podría tomar los valores de bueno, regular y malo.

En cuanto a la información general se apuntarán referencias al peso, la altura o el perímetro cefálico.

En citas médicas se almacenará un registro temporal y los datos concernientes al doctor como son su nombre o su especialidad. Una cita estará asociada a un único médico, pero cada especialista podrá tener diversos pacientes.

De la medicación interesa la fecha, así como la hora de la última toma. De los medicamentos guardaremos su nombre, cada cuanto tiempo ha de tomarse y los días totales de tratamiento. Cada entidad de tipo medicación estará asociada a un medicamento. Así como un medicamento puede estar asociado a varias o ninguna medicación.

Por último, la temperatura corporal, se identificará con una marca temporal y guardará registro de la temperatura.

Cada recién nacido tendrá entre ninguno o varios registros sobre las distintas clases. Así como las clases estarán asociadas a un único bebe.

2.2.1 Diagrama de clases.

El diagrama se presenta como un referente visual entre las distintas clases con sus respectivos atributos. No se incluyen métodos puesto que el objetivo es simplemente tener una composición de las relaciones existentes entre las clases.

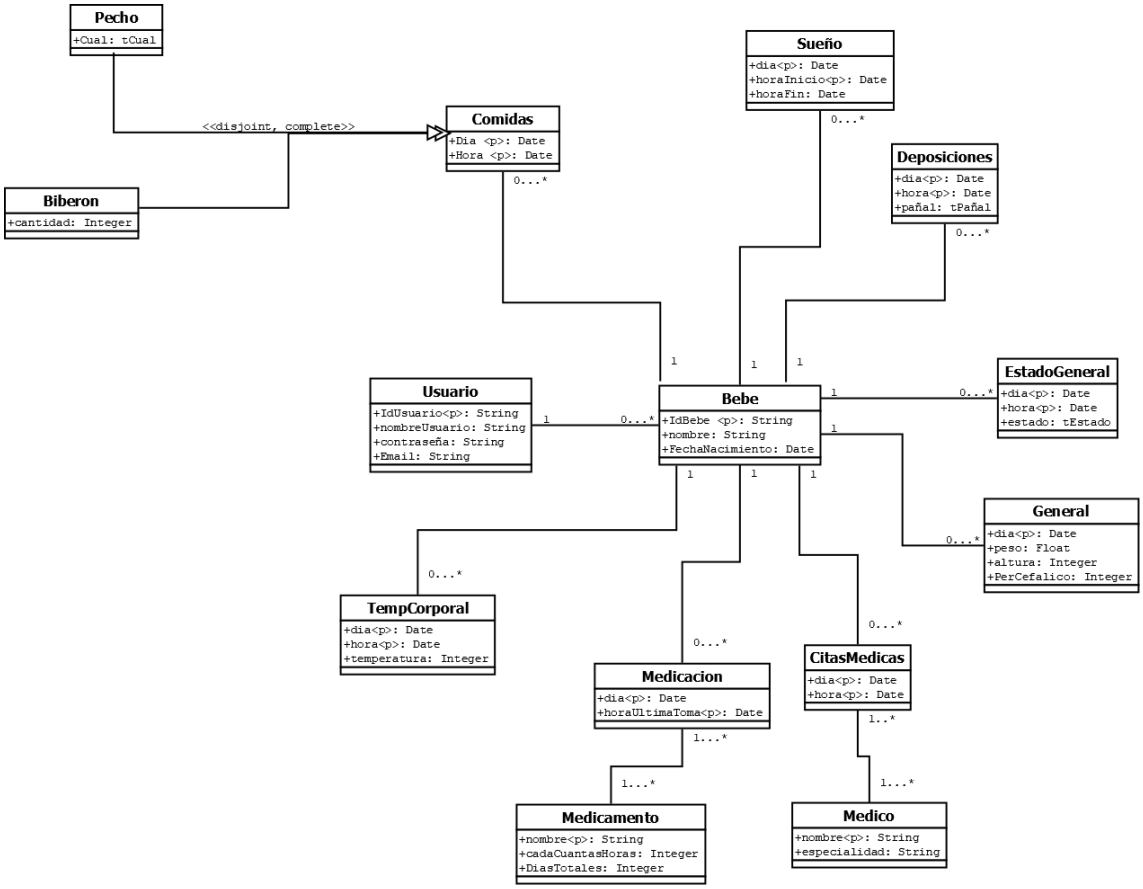

Figura 22: Diagrama de clases UML de la base de datos

3. Fase de creación.

Llegados a este punto se da comienzo al desarrollo propiamente dicho de la aplicación. La intención de esta etapa es la de crear una base que dé soporte a la funcionalidad. Se distinguen dos objetivos claros, la estructura de la página y la base de datos.

3.1 Estructura.

La base estructural de la aplicación se compone de una serie de elementos comunes a todas las páginas funcionando como nexo o hilo conductor, dotando al aplicativo de una cierta coherencia visual.

Así pues, el objetivo primordial es implementar en código aquellas decisiones tomadas en la fase de diseño. Esto se llevará a cabo mediante lenguajes Html, CSS o el *framework* BootStrap.

3.1.1 *Head*.

Esta parte del código no es mostrada por el navegador, sin embargo, contiene datos como el nombre de la página, metadatos y enlaces como, por ejemplo, a la hoja de estilos o recursos necesarios para la correcta carga de esta.

La información que se almacena aquí es relevante para buscadores, especialmente para técnicas de posicionamiento SEO.

En el código aparece el nombre de la página, la codificación, los recursos necesarios para la implementación del *framework* BootStrap, el enlace a código JavaScript (código.js), a la hoja de estilos (style.css) o referencias a las fuentes utilizadas.

Los atributos width y initial-scale, garantizan la adaptabilidad a la anchura del dispositivo y el zoom inicial respectivamente.

3.1.2 Cabecera.

Las páginas web se dividen generalmente en tres secciones. Estas son la cabecera (*header*), el cuerpo (*body*) y el pie (*footer*). La función del *header* es la de dotar de identidad a la página. Por ello, es el lugar donde se suele colocar los logotipos, imágenes de marca, etc.

Figura 23: Diseño del banner de la aplicación.

En este caso, se decidió la creación de un banner que contuviera el nombre del aplicativo y un par de imágenes vectoriales acordes al contenido de la página. Se utilizo para su creación la tipografía definida en las fases de diseño, así como los colores seleccionados en dicha etapa.

Como se comentaba se utilizaron imágenes vectoriales, por lo que se trabajó sobre Adobe Illustrator CC, 2019.

```
<header>
<!--CABECERA-->
<div class="jumbotron-fluid text-center" id="cabecera">
    <img src="fotos/banner.jpg" alt="Mi bebe" >
</div>
</header>
```

Figura 24: Detalle del código del *header*

En lo referente al código, se utilizó la clase `.junbotron-fluid` de BootStrap que proporciona un contenedor que ocupa toda la anchura del dispositivo, sin bordes redondeados.

En lo referente a estilo, se utilizó CSS para dar formato a los márgenes superior e inferior.

3.1.3 Navegador.

Lo siguiente que nos encontramos en la estructura, es la zona de navegación. Como se puede apreciar en los *wireframes* y prototipos se apostó por una barra horizontal en los dispositivos más anchos, que colapsara a medida que se fueran reduciendo las longitudes de las pantallas. Concretamente, esto ocurre cuando la distancia es menor de 768 px. La decisión de que el menú se comportara de esta manera obedece a criterios espaciales, mientras que en pantallas grandes la barra horizontal con todos los enlaces al descubierto facilita la usabilidad. Cuando se trata de dispositivos más pequeños, las opciones quedarán apiladas de forma vertical lo que reducirá notablemente la superficie central donde se ejecuta toda la funcionalidad de la aplicación.

En cuanto a programación, la barra de navegación se sustenta sobre una estructura desarrollada sobre el *framework* BootStrap, que es la que permite el colapso. Se explican las diferentes etiquetas a continuación:

- `. nabar:` define la creación de una barra de navegación estándar
- `. nabar-expande-md:` es la encargada de expandir el navegador para dispositivos de tamaño medio (md), es decir, 768px.
- `.navbar-dark:` define el estilo.

A continuación, tenemos el primer enlace, cuando la barra se encuentra en forma horizontal, que será el único visible junto con el botón (button), cuando colapse. Este será el encargado de mostrar el resto del menú cuando lo pulsemos. Seguidamente aparece una tabla con el resto de los enlaces.

Se modificaron algunos parámetros como la fuente y su color o el tono de fondo, a través de CSS para lograr una mayor integración con el resto del diseño.

3.1.4 Zona central.

Esta parte funcionará a nivel visual como un contenedor donde enmarcamos formularios, gráficos, etc.

A nivel de codificación se trata de un contenedor *div*, con una clase `.container` de BootStrap, que nos ayudará a mantener el *grid* en las sucesivas implementaciones sobre esta zona. En cuanto al estilo CSS modificaremos márgenes(*margin*) y relleno (*padding*) con valores porcentuales para facilitar la adaptabilidad a los distintos dispositivos.

3.2 Base de datos.

En lo referente a la BBDD, pasamos del modelo conceptual desarrollado en el diagrama de clases a un modelo lógico que finalmente se traduce en la creación de las distintas tablas relacionales.

Se pasa a la descripción de dicho proceso en los casos más relevantes.

En primer lugar, la relación de herencia de las clases Pecho y Biberón con respecto a Comida, se suprime dando lugar a sendas clases Pecho y Biberón que absorberán los atributos de la clase madre.

En cuanto a la clase Médico, se ha añadido un atributo relacional con la clase Bebé, con la intención de limitar el listado de doctores a únicamente los visitados por el recién nacido.

Por lo demás, el resto de las tablas, salvo Bebé, Usuario y Medicamento, se ha añadido un atributo bebé en concepto de clave foránea con respecto a idBebe de la tabla Bebé.

El resultado en forma de diagrama es el siguiente.

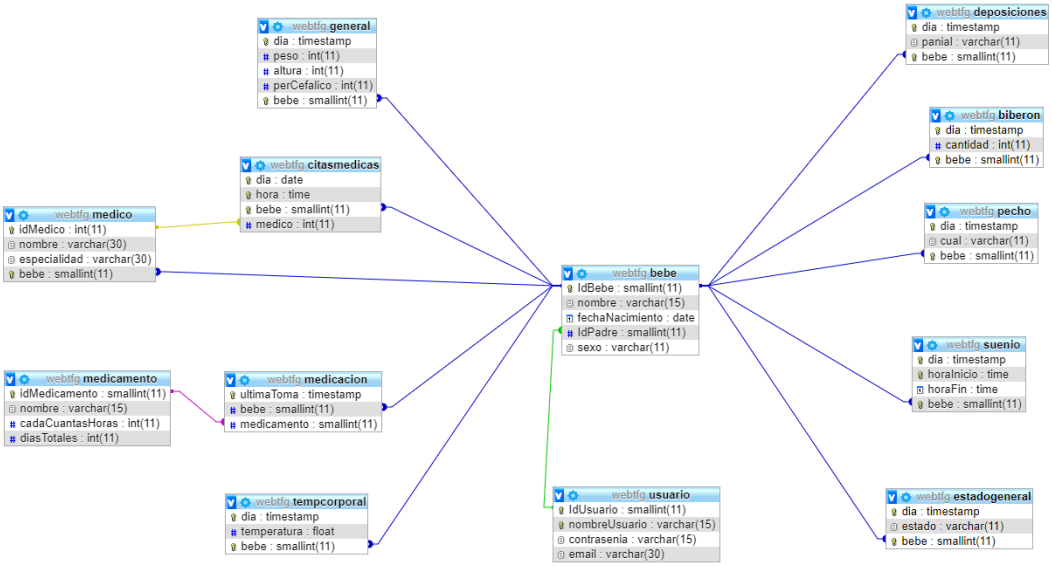

Figura 25: Relación de clases de la base de datos

4. Fase de desarrollo.

Ésta es, sin duda, la etapa más importante de todo el proyecto. En esta fase es donde se realizará toda la codificación de la funcionalidad, por tanto, es la fase más crítica.

En un primer momento se plantearon una serie de funcionalidades que fueron evolucionando a medida que el proyecto iba tomando forma. Concretamente, durante esta fase se tomaron dos decisiones.

La primera tenía que ver con el registro y *login*. Al principio se planteó que el usuario tuviera la opción de acceder a la aplicación a través de redes sociales. Finalmente, se descartó debido a que ambos formularios solo piden los datos más básicos (nombre y contraseña en el caso de *login* y usuario, contraseña y correo en el registro) y el tiempo que debe invertir el usuario en rellenarlos es mínimo. Así pues, no compensaba la implementación de este tipo de función y la consiguiente entrega de datos a terceros que conlleva.

Por otro lado, se pensaba en gestionar un sistema de alarmas a través del correo electrónico, que avisara con las horas a las que tocaba cambiar el pañal o administrar algún medicamento. Sin embargo, a medida que se iba concretando el sistema de gráficos y tablas se llegó a la conclusión de que era preferible dar más protagonismo a la visualización de datos dentro del aplicativo que al envío de emails. El envío masivo de correos por parte de una página web, suele saturar a los usuarios que acaban dándose de baja. Por no mencionar que posiblemente los gestores de correos tacharan estos envíos como spam y pasaran desapercibidos para el cliente.

Quizás en una futura ampliación de la aplicación se podría desarrollar un sistema de alarmas menos invasivo.

En conclusión, se sustituyó el sistema de alarmas por una muestra de los datos más detallada que la correspondiente a las pantallas principales (datos médicos y hábitos), pudiendo consultar los gráficos y tablas específicas de cada categoría.

Las tareas desarrolladas en esta fase se pueden dividir en las categorías que se exponen a continuación.

4.1 *Home*.

Figura 26: Imagen final de la página inicio en distintos dispositivos

La página de inicio es la carta de presentación de la aplicación, por lo que tiene un carácter más publicitario que de servicio como es el caso del resto de páginas. La zona central se divide en dos secciones.

La primera destinada a llamar la atención del usuario con un *slideshow* fotográfico. Éste está implementado con la clase carrusel de BootStrap, que tiene varios parámetros:

- `.carousel:` es la encargada de crear el carrusel.
- `.slide:` añade las transiciones y efectos mediante CSS.
- `.carousel-inner:` es necesaria para indicar la lista de fotografías.
- `.carousel-item:` especifica los ítem que formaran parte del *slideshow*
- `.carousel-control:` añade controles para pasar o retroceder.

En la parte inferior se encuentran tres columnas donde se describen los servicios ofertados. Están desarrolladas sobre un sistema de rejilla (grid). Para entender este sistema se debe dividir la pantalla en doce columnas de igual valor, en este caso, tenemos .col-sm-4, es decir, las franjas individuales se aglomeran en grupos de 4, dividiendo la pantalla en tres super columnas que pasaran de posición horizontal a una vertical cuando el ancho del dispositivo sea menor del tamaño *small-device* (sm, 576px).

En cuanto al CSS, lo más destacable es el cambio de letra (class="letra2"), ya que el cuerpo de las columnas esta implementado en esta tipografía.

4.2 Formularios

Los formularios son la manera que tiene la aplicación de recoger información de los usuarios. En el caso de la página web se pueden distinguir tres tipos, de identificación (*login*), de registro (usuario, bebe) y el resto cuya función es la de recolectar datos específicos de alguna actividad.

En cuanto a BootStrap, tenemos las clases:

- `.form-control:` adapta la longitud de la caja al ancho del contenedor.
- `.form-group:` se adapta al ancho de la pantalla

También se utilizan distintos tipos de input para adecuarse a los datos.

Todos comparten su funcionamiento. Se recogen datos del formulario y mediante el método POST se auto reenvían a la misma página gracias a:

```
<?php $_SERVER['PHP_SELF']; ?>
```

Una vez que los datos llegan se gestiona con el código Php que conecta con la BBDD.

El Php también tiene parte de su estructura común a todos los formularios:

1. Se establece conexión con la BBDD (conexión.php)
2. Se inicia la sesión (sesión_start())

La principal diferencia entre los distintos tipos de formularios es la gestión de los datos que se comentará a continuación.

4.2.1 *Login*.

Figura 27: Imagen final de formulario de login en diversos dispositivos

La función de esta página como su nombre indica es que el usuario se identifique y acceda de este modo a sus datos.

En este caso la gestión de datos transcurre de la siguiente manera:

1. Se comprueba si la variable $_SESSION["ID"] esta inicializada. Si es así, se carga la página indiceBebe. Si por el contrario no lo está, se continúa.
2. Se comprueba que la variable $_POST contenga datos. Si los contiene se asignan a variables.
3. Se pasa a la parte de interacción con la BBDD.
    I. Se inserta el código Sql en una variable. Este código devolverá si encuentra alguna coincidencia con el nombre y la contraseña en la tabla usuario.
    II. Se ejecuta la variable con el código Sql en la base de datos.
    III. La base de datos devuelve los datos obtenidos
    IV. Se comprueba que haya resultados y se inicializa la variable $_SESSIN["ID"] con el nombre de usuario devuelto por la BBDD.
    V. Se carga la página indiceBebe
4. Si todo ha ido bien se cierra la conexión.

4.2.2 Registros.

Figura 28: Imagen final del formulario de registro.

Estas páginas, como su nombre indican, son las que permiten registrar a nuevos usuarios o en caso de que éste ya esté, registrado añadir nuevos bebés.

Se sigue un esquema semejante al caso anterior. Sin embargo, se pueden apreciar algunas diferencias.

En primer lugar, es un formulario de registro, por lo tanto, no se comprueba que el usuario este “logeado”. Seguidamente se comprueba que la base de datos no contenga ningún usuario registrado con ese nombre. Si no existe se pasa a insertar los datos del nuevo usuario.

En lo referente a JS, se usa la gestión de eventos para comprobar lo siguiente:

- Nombre de usuario (comprobarNombreUsuario()): Esta función comprueba que el nombre de usuario no supere los 15 caracteres.
- Contraseña valida (comprobarContraseña()): Comprueba que la contraseña tenga un mínimo de 8 caracteres, entre los que se debe incluir una letra mayúscula y un número.
- Contraseñas iguales (contrasenaIguiales()): Comprueba que ambas contraseñas coincidan.

El formulario de registro de bebés sigue el mismo patrón, a diferencia de que sí es necesario que el usuario este autentificado.

Con respecto a JavaScript se comprueba que el nombre del bebé no supere los 15 caracteres.

4.2.3 Específicos.

Figura 29: Formulario final de la categoría sueño

El resto de los formularios son los encargados de recoger los datos específicos para almacenar en cada tabla de la BBDD. Siguen todos los mismos esquemas. Así que usaremos el formulario de la categoría sueño como ejemplo.

La única diferencia que podemos encontrar es el comando Sql que pasamos a la BBDD. En estos casos simplemente se trata de una inserción con los datos recogidos en el formulario.

4.3 Graficas y tablas

Figura 30: Detalle del resultado final de la página hábitos

Uno de los puntos fuertes de la aplicación es la muestra de los datos de una manera gráfica, a través de diagramas y tablas. Para este cometido se utilizó la API Google Chart. Es una potente herramienta que permite incluir este tipo de contenido en la página de una forma relativamente sencilla.

Este recurso se utiliza dentro de la web de dos formas. En las páginas habitos.php y datosMedicos.php, donde se ofrecen varias imágenes resumen, y en las páginas dedicadas donde se amplían los resultados para conseguir graficas más completas.

Todas las gráficas y tablas funcionan de manera similar, por lo tanto, solo veremos un ejemplo de cada categoría.

4.3.1 Gráficas.

Se ha elegido la gráfica temperatura corporal a modo de ejemplo.

```
<!--------------------------------TEMPERATURA------------------------------->
  <div class="col-sm" id="Temperatura">
   <a class='tituloColumnas2' href="graficos/tempCorporal.php" title="ir a temperatura corporal">
      <h3 class='tituloColumnas2'>Temperatura corporal</h3>

     <?php
        $sql="SELECT dia, temperatura FROM tempcorporal WHERE bebe=$bebe ORDER BY dia DESC LIMIT 5";
        $comprobarBebe=$conn->query($sql);
        $rows = $comprobarBebe->num_rows;
        if ($rows>0) {
        echo "<div class='col-sm' id='tempCorporal_Grafico'></div>";
          }
        else{
            echo("<div class='col-sm' id='sinDatos' >No hay datos</div>");
        }
    ?>
    </a>
    <a id="enlace" href="formulariosDatosMedicos/formTemperatura.php"><p class='tituloColumnas'>Añadir datos</p> </a>
</div>
```

Figura 31: Detalle del código de la gráfica temperatura

Como se aprecia en el código, la parte de Php hace una llamada a la base de datos (esta ha sido conectada anteriormente) para comprobar que existen datos asociados a esta categoría, de ser así, se crea el contenedor donde ira encajada la gráfica:

De no ser así, se crea otro contenedor que avisa de la falta de datos.

```
<!-------------------------TEMPERATURA CORPORAL--------------------------------->
<script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(tempCorporalFuncion);
    function tempCorporalFuncion() {
      var tempCorporalData = google.visualization.arrayToDataTable([
        [ 'dia','temperatura'],
        <?php
        $sql="SELECT DAY(dia) AS dia, temperatura FROM tempcorporal WHERE bebe=$bebe ORDER BY dia DESC LIMIT 5";
        $comprobarBebe=$conn->query($sql);
        $rows = $comprobarBebe->num_rows;
        if ($rows>0) {
        while($row = $comprobarBebe->fetch_assoc()) {
          echo "[ '".$row['dia']."',".$row['temperatura'].],";
          }
           }
        ?>
]);
    var tempCorporalOption = {
      title: 'Temperatura corporal',
      hAxis: {title: 'Temperatura(grados)',  titleTextStyle: {color: '#3e2c7f'}},
      vAxis: {minValue: 0},
      width: '100%',
      height: '100%',
      backgroundColor: '#b9a5ff',
      titleTextStyle: {color: '#3e2c7f'},
      colors: ['#3e2c7f'],
    };
    var tempCorporalGrafico = new google.visualization.BarChart(document.getElementById('tempCorporal_Grafico'));
    tempCorporalGrafico.draw(tempCorporalData, tempCorporalOption);
  }
</script>
<!-------------------------------------------------------------------------->
```

Figura 32: Detalle del Script de la gráfica temperatura

La funcionalidad propiamente dicha se realiza en el Script anterior. Se crea una función en este caso tempCorporalFuncion(), que se encarga de crear un array con los datos que se quieren mostrar. En nuestro caso rellenamos el array a través del código Php que va extrayendo los

datos de la BBDD. A continuación, crea una variable tempCorporalOption donde se configuran los parámetros de estilo. Finalmente, las dos últimas líneas de la función indican donde se encajará la gráfica, y llaman a su ejecución con los parámetros anteriores.

4.3.2 Tablas.

En este caso desarrollaremos la tabla medicación.

```
    <!-------------------------------MEDICACION----------------------------------->
    <div class="col-sm" id="Medicamentos">
    <a class='tituloColumnas2'  href="graficos/medicacion.php" title="ir a medicacion">
        <h3 class='tituloColumnas2' >Medicacion</h3>

      <?php
          $sql="SELECT  CONVERT(medicacion.ultimaToma, CHAR) AS ultimaToma, medicacion.medicamento AS idM, medicamento.idMedicamento AS idMedicamento,
          medicamento.nombre AS nombre FROM medicacion, medicamento WHERE medicacion.bebe=$bebe AND medicacion.medicamento=medicamento.idMedicamento ORDER BY
          ultimaToma DESC LIMIT 8";
          $comprobarBebe=$conn->query($sql);
          $rows = $comprobarBebe->num_rows;
          if ($rows>0) {
          echo "<div class='col-sm' id='tabla_medicacion' ></div>";
            }
          else{
              echo("<div class='col-sm' id='sinDatos' >No hay datos</div>");
          }
      ?>
      </a>
      <a id="enlace" href="formulariosDatosMedicos/formMedicacion.php"><p class='tituloColumnas'>Añadir datos</p> </a>
  </div>

    <!-------------------------------GENERAL-------------------------------------->
```

Figura 33: Detalle del código de la tabla medicación

Como se puede apreciar en el código, se sigue el mismo patrón de las gráficas.

```
  <!-------------------------------------------------------------------------->
  <!-----------------------------MEDICACION----------------------------------->
  <script type="text/javascript">
    google.charts.load('current', {'packages':['table']});
    google.charts.setOnLoadCallback(medicacionFuncion);

    function medicacionFuncion() {
      var medicacionData = new google.visualization.DataTable();
      medicacionData.addColumn('string', 'Ultima Toma');
      medicacionData.addColumn('string', 'medicamento');

      medicacionData.addRows([
        <?php
          $sql="SELECT  CONVERT(medicacion.ultimaToma, CHAR) AS ultimaToma, medicacion.medicamento AS idM, medicamento.idMedicamento AS idMedicamento,
          medicamento.nombre AS nombre FROM medicacion, medicamento WHERE medicacion.bebe=$bebe AND medicacion.medicamento=medicamento.idMedicamento ORDER BY
          ultimaToma DESC LIMIT 8";
          $comprobarBebe=$conn->query($sql);
          $rows = $comprobarBebe->num_rows;
          if ($rows>0) {
          while($row = $comprobarBebe->fetch_assoc()) {
            echo "[ '".$row['ultimaToma']."','".$row['nombre']."'],";
            }
             }
          ?>
      ]);

      var medicacionTabla = new google.visualization.Table(document.getElementById('tabla_medicacion'));

      medicacionTabla.draw(medicacionData, {showRowNumber: false, width: '100%', height: '100%',backgroundColor: '#b9a5ff', background:'#b9a5ff'});
    }
  </script>
```

Figura 34: Script de la tabla medicación

Sin embargo, como podemos ver en el Script si existen algunas diferencias. En este caso la función crea una tabla, el primer paso es definir el título de cada columna, así como el tipo de contenido, en este caso *String*. Se vuelve a ejecutar el código Php para extraer la información de la base de datos. Hay que destacar que en este caso debemos convertir los distintos tipos de la base a carácter para que coincidan con los definidos anteriormente. A partir de aquí, se sigue el mismo procedimiento que en el caso de las gráficas, a excepción de los parámetros de estilo, se incorporan cuando se llama a la función *draw* que es la encargada de montar la tabla esperada.

5. Fase de pruebas.

A medida que se va desarrollando el proyecto, se comprueba de forma recurrente todo aquello que se va implementando. Comprobando que todo funcionara correctamente. Sin embargo, durante la fase final se comprueban de una manera más formal una serie de aspectos.

Home

- Enlaces
- Correcto funcionamiento del *SlideShow*.

Registro

- Nombre de usuario ya existente
- Las contraseñas deben coincidir
- Las contraseñas deben contener 8 caracteres, incluyendo una mayúscula y un digito como mínimo.
- El email debe tener el formato adecuado (xxxxx@xxxxxx.xxxx)

Login

- El nombre de usuario y la contraseña deben estar registradas en la BBD para poder acceder
- Si se concede acceso se pasa a indiceBebe.php

Registro bebé

- Nombre menor de 15 caracteres
- Formato fecha
- El bebé queda registrado bajo el usuario correspondiente
- Si el registro es positivo se pasa a indiceBebe.php

Índice bebé

- Los bebés que se muestran corresponden al usuario “logeado”
- Si no hay bebés registrados se informa
- Los enlaces funcionan correctamente

Gráficas

- Los datos se muestran correctamente
- Si no hay datos se informa
- Se revisan los enlaces

Una vez comprobado que todo funciona correctamente se da por superada esta fase.

6. Conclusiones.

En primer lugar, pese a ser la primera vez que se hacía frente a un proyecto de estas características, los resultados obtenidos se asemejan bastante a los esperados cuando se planteó en un primer momento. Si bien es cierto, que se hubieron de realizar un par de modificaciones sobre las expectativas iniciales.

La primera, el *login* y registro a través de redes sociales, pese a que en su concepción parecía una buena solución para facilitar esta gestión. En la práctica real, una vez desarrollados sendos formularios el tiempo que se podía ahorrar usando este sistema era insignificante.

La segunda, el sistema de gestión de alarmas sobre el papel resultaba útil, pero la forma de implementación, a través de emails quizás no fuera la más efectiva. A medida que se fue investigando sobre lo que podría conllevar un envío masivo de este tipo de correos, se concluyó que quizás fuera contraproducente. Por lo tanto, como iniciativa parecía acertada, sin embargo, se carecía de los medios adecuados para poder llevarla a cabo.

Pese a estos cambios el enfoque metodológico y la planificación temporal respondieron bien. Al dividir el proyecto en distintas fases, realmente la única que se vio afectada por estos cambios, fue la etapa de desarrollo. Afortunadamente se había previsto que fuera la fase más complicada y por tanto a la que más tiempo se dedicaría.

Además, estos cambios dieron la oportunidad de enfocar el desarrollo alrededor de la exposición de los datos, lo que finalmente resulto positivo.

En cuanto a una futura ampliación sería interesante contar con más recursos, así como un equipo médico que la respaldara. Una cosa que se ha echado en falta es poder recurrir a un equipo especializado en la materia, sobre todo al principio cuando se decidía cual era la información realmente relevante.

En lo referente a posibles ampliaciones habría que buscar la forma en la que poder volcar todos los datos generados por el cliente a algún formato, por ejemplo, un PDF. También sería útil que el usuario pudiera alternar los formatos de visualización a su elección. De este modo, podría optar por una gráfica si quiere una imagen más visual o una tabla si lo que requiere es la búsqueda de un dato concreto. También se podrían añadir calendarios de vacunación. En definitiva, la aplicación tiene diversas posibilidades.

7. Glosario

API: Interfaz de programación de aplicaciones.
Framework: Es un marco de trabajo que ofrece un conjunto de herramientas estandarizadas.
IDE: Entorno de desarrollo integrado.
Open source: se refiere a aplicaciones de código libre.
Suite: hace referencia a un conjunto de programas.
UML: Lenguaje estandarizado de modelado.

8. Bibliografía

[1] Estadística de nacimientos. Movimientos de población INE (Instituto Nacional Estadística)
https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177007&menu=ultiDatos&idp=1254735573002 (junio 2019)

[2] Fenómenos demográficos INE (Instituto Nacional Estadística)
https://public.tableau.com/views/Fenmenosdemogrficos/Fenomenosdemograficos?%3AshowVizHome=no&%3Aembed=true (junio 2019)

[3] Resumen de datos de Personas por sexo, características demográficas y tipo de uso de TIC (tabla)
https://www.ine.es/jaxi/Datos.htm?path=/t25/p450/base_2011/a2018/l0/&file=01002.px (junio 2019)

[4] Resumen de datos de Personas por sexo, características demográficas y tipo de uso de TIC (grafica)
https://www.ine.es/jaxi/Datos.htm?path=/t25/p450/base_2011/a2018/l0/&file=01002.px (junio 2019)

[5] Aplicación The Wonder Week
https://www.thewonderweeks.com (junio 2019)

[6] Aplicación Baby Tracker en Google Play
https://play.google.com/store/apps/details?id=com.nighp.babytracker_android&hl=es (junio 2019)

[7] Aplicación Baby Connect
https://www.babyconnect.com (junio 2019)

[8] Aplicación Cry Traslator en iTunes
https://itunes.apple.com/es/app/cry-translator/id332636983?mt=8 (junio 2019)

[9] Aplicación LactMed en Google Play
https://play.google.com/store/apps/details?id=gov.nih.nlm.sis.lactmed&hl=es_419 (junio 2019)

[10] Aplicación iPediatric
https://appsmedicina.com/es/apps/detalle_app.php?id=20 (junio 2019)

[11] Informe *Global Mobile Consumer Survey*, 2017 de Deloitte
https://www2.deloitte.com/content/dam/Deloitte/es/Documents/tecnologia-media-telecomunicaciones/Deloitte-ES-TMT-Consumo-Movil-2017.pdf (junio 2019)

[12] Informe *Mobile* en España y en el mundo 2018 de Ditrendia
https://mktefa.ditrendia.es/hubfs/Ditrendia-Informe%20Mobile%202018.pdf?fbclid=IwAR13B249fMuQB5O_6UcYdjNIh_uSX567Jj3GZRPFXdS2tzqkEfCx6kzugkI (junio 2019)

[13] Artículo The New Methodoly, 2000 de Martin Fowler
https://www.martinfowler.com/articles/newMethodologyOriginal.html (junio 2019)

[14] Kaushan Script en Google Font
https://fonts.google.com/specimen/Kaushan+Script?selection.family=Kaushan+Script (junio 2019)

[15] Meera Inimai en Google Font
https://fonts.google.com/specimen/Meera+Inimai?selection.family=Meera+Inimai (junio 2019)

[16] Diseñando apps para móviles
Javier Cuello, José Vittone. ISBN: 978-84-616-5070-5
Google Books:
https://books.google.es/books?id=ATiqsjH1rvwC&printsec=frontcover&dq=Hacer+wireframes+en+etapas+tempranas+supone+un+ahorro+de+costes+y+tiempo,+permitiendo+evaluar+aspectos+como+la+navegación+e+interacción+para+sentar+unas+bases+sólidas+que+después+den+paso+al+nivel+visual&hl=es&sa=X&ved=0ahUKEwj0lsqTi9_iAhXO34UKHWT2DT0Q6AEIMTAB#v=snippet&q=wireframes&f=false (junio 2019)

9. Recursos bajo licencia

Vectores:

- Banner:
  - Cigüeña:

https://pixabay.com/es/vectors/bebé-ave-entrega-mujeres-niña-1299514/
(licencia *Pixabay Licence*: https://pixabay.com/es/service/license/)

  - Biberón:

https://www.vexels.com/vectors/preview/117945/cute-baby-theme-vector
(licencia: Attribution Non-Commercial Share Alike), Autor: qvectors

- Dispositivos:

https://www.freepik.es/vector-gratis/diferentes-dispositivos-maqueta_1075504.htm
(licencia: atribución) autor: Freepik.com