

MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

Autor: Brais Oliveira Sanjurjo

Tutor: Sergio Malo Molina

Tintorería Avenida

2S2021

Enlace Google Drive:

https://drive.google.com/drive/folders/1_qohCfbUualoA3sSJDmPGad6IzW4uVnN?usp=sharing



INDICE:

Contenido

1. INTRODUCCIÓN	3
2. MOTIVACIÓN	4
3. ABSTRACT.....	4
4. OBJETIVOS PROPUESTOS.....	5
5. METODOLOGÍA UTILIZADA.....	5
6. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO	6
Arquitectura cliente – servidor y entorno de desarrollo	6
Sistemas.....	6
Editores.....	6
Base de datos.....	7
UML.....	8
Control de versiones	8
Prototipado Frontend.....	8
Imagen.....	9
Responsividad.....	9
Google Developers	9
PayPal Developer	9
HTML5 (HyperText Markup Language)	9
CSS3 (Cascading Style Sheets).....	10
SQL (Structured Query Language)	10
Java Script.....	10
PHP (Hypertext Preprocessor)	10

7. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN.....	12
8. ANÁLISIS.....	16
Persona	16
Casos de uso.....	17
Estructura de datos	22
9. DISEÑO.....	26
Accesibilidad.....	26
Wireframes.....	28
Header	28
Index.....	29
Nosotros	31
Servicios.....	31
Contacto	32
Contratar	32
10. DESPLIEGUE Y PRUEBAS	33
Despliegue.....	33
Pruebas	33
11. CONCLUSIONES.....	34
12. VÍAS FUTURAS	35
13. GLOSARIO Y ACRÓNIMOS.....	36
14. BIBLIOGRAFÍA / WEBGRAFÍA	37
Accesibilidad.....	37

Ciclo de vida de software	37
CSS.....	37
Iconos.....	37
Mapas	37
Paletas	37
Pasarela de pago	37
PHP	38

1. Introducción

Hay una necesidad enorme de desarrollos sencillos pero poderosos que apliquen todo el potencial de los lenguajes clásicos de la web para dar soluciones a negocios que actualmente se ven bombardeados de ofertas de CMS que prometen la mayor de las facilidades y un servicio funcional en menos de una semana. Esto puede ser verdad hasta cierto punto, pero cualquier programador que se haya tenido que pelear con un CMS acaba llegando tarde o temprano a la conclusión de que “todo este código, esta maraña de directorios y dependencias de plugins mantenidos por terceros, ¿para qué?”

Deseando poder tomar un control real de su proyecto.

No solo eso. Tampoco la supuesta facilidad de gestión de la web por parte de la pequeña empresa a la que se ofrece este servicio es tal, y si quien acaba al mando del panel de administración del CMS no es un experto en la materia lo habitual es que en el mejor de los casos la página se estanque y no se toque por miedo, creando tremendos problemas de seguridad (actualizaciones de PHP...) hasta que simplemente empieza a fallar porque los distintos plugins usados empiezan a generar conflictos entre ellos o dejan de tener soporte. Esto en el mejor de los casos, para el resto de los casos el dueño del negocio (de oficio mecánico, frutero...) se anima con el diseño y el desastre se produce mucho antes.

Hay que tener en cuenta que casi siempre un CMS es una solución sobredimensionada y menos optimizada. Recordemos que quizás no todo el público objetivo al que apunta el

negocio son usuarios con buenos dispositivos y una conexión FTTH, y un tiempo de carga poco ágil hace que los potenciales clientes se marchen antes de que tengamos siquiera métricas del porcentaje de rebote.

En resumen, quien va a mantener una web creada en un CMS debe ser un experto en esa materia que realmente es amplia y compleja, y si lo va a mantener un desarrollador al final va a desear mantenerse pegado al código manejándolo simple pero eficaz.

En su defensa, hay espacio para CMS en proyectos que serán mantenidos por Administradores de Sistemas, equipos de diseño gráfico, y otros casos similares de trabajadores con una base técnica adecuada y en los que el core de su negocio puede estar muy ligado al mantenimiento web. Pero este no es el caso que nos atañe en este trabajo.

En el orden siguiente trataremos la metodología, tecnologías necesarias, planificación, análisis, diseño y pruebas realizadas al software hasta llegar a un producto presentable y preparado para ser puesto en producción.

2. Motivación

Existe una página previa realizada por el autor del presente desarrollo en WordPress y en producción desde 2018 que debía ser actualizada por quedarse claramente desfasada.

Tras consultarlo con la empresa se lleva a cabo la renovación, aprovechando únicamente algunos textos y sin utilizar CMS alguno.

3. Abstract

The present project deals with the necessity of facing web development for small companies from a point of view which is as independent as possible of heavy tools and falsely considered as friendly for users without much technical knowledge.

Furthermore, this framework is positive in order to free our clients from companies that make their tools limiting for the developer from a creative point of view and also limiting with regards to future movements. This happens because the owner of the web would not be the owner of the code that forms it, being therefore unable to migrate their web or app to a different hosting without having to begin all the programming process from scratch.

Due to this philosophy, no framework has been used and the use of libraries has tried to be limited to those reasonably essential.

The document goes with diagrams and illustrative images, all of them necessary to properly understand it, in addition to a glossary and a bibliography.

We go all the way from the initial contact with the client until the deployment phase, including all the intermediate points of analysis, technology selection, design, implementation and testing, leaving software maintenance out of this report because it is developed out of the timeframe that can be dealt with in a document presented before being this application in production.

4. Objetivos propuestos

Crear una plataforma desde la que se dé servicio informativo y publicitario actualizado como una web estática tradicional y compra de los servicios ofertados por el negocio usando para esto las ventajas de una web dinámica.

El usuario debe tener varios cauces de contacto con la empresa que seleccionará según su conveniencia. El cauce principal será el correo electrónico, por lo que se habilitará una herramienta de envío desde la propia web.

Todo respetando la imagen de marca tradicional previa.

El proyecto debe presentarse servido desde un hosting:

<http://braisoliveira.eu>

5. Metodología utilizada

Este proyecto se ha realizado bajo el paradigma de desarrollo en cascada con retroalimentación.

De tal modo, se da comienzo a la fase de análisis escuchando las necesidades que deben ser cubiertas e intentando comprender el desafío a resolver.

En esta fase se genera la documentación con las demandas iniciales.

Con esta documentación lista, en un segundo paso se conviene cual va a ser la arquitectura y los lenguajes de programación adecuados. Se deciden también posibles frameworks y bibliotecas.

Se inicia después la fase de codificación, traduciendo las acciones definidas en los algoritmos a instrucciones en los diferentes lenguajes seleccionados anteriormente.

Aunque el ciclo de vida de un software termina en la parte de mantenimiento, para el caso que nos ocupa se termina con la fase de pruebas.

En este paso se usa el método de caja negra, en el cual se lleva a la parte interactiva de la web a todos los escenarios posibles, para asegurar que la respuesta es la esperada en todos los casos.

6. Tecnologías y herramientas utilizadas en el proyecto

Arquitectura cliente – servidor y entorno de desarrollo

Sistemas

Se desarrolla desde los sistemas operativos **Ubuntu 18.04.5 LTS** (laptop) y **Windows 10 pro** virtualizado en hipervisor Oracle VM **Virtual Box** 6.0 (PC)

Se decide de este modo para tener una solución de movilidad y un equipo de escritorio.

La utilización de máquinas virtuales se debe a la importancia de poder portar todo el entorno de desarrollo independientemente del hardware usado en cada momento con facilidad, llegado el caso.

Para la selección de los sistemas operativos no hay más motivo que la comodidad del desarrollador.

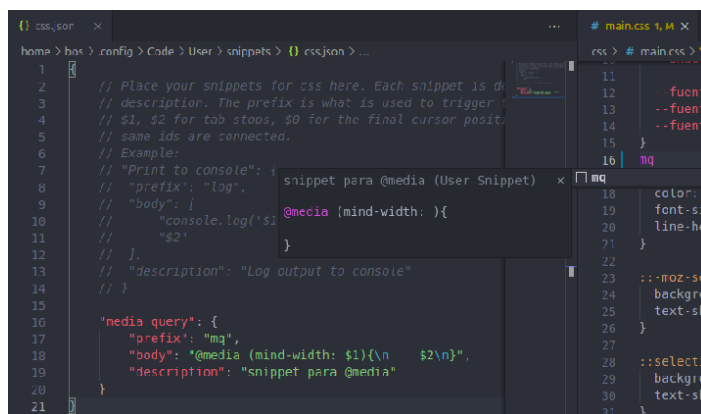
Finalmente se suben los archivos de desarrollo a un hosting mediante FTP para que la página pueda ser accedida desde Internet.

Editores

En Ubuntu se utilizan el editor **VSCode**. Tras comenzar con VSCodium, se aprecian integraciones menos óptimas con el programa **GitHub Desktop** (solo permite abrir el código

directamente desde la aplicación con VSCode o Atom). Siendo un proyecto al que se le dedican numerosas horas estas pequeñas cosas se antojan importantes.

VSCode tiene un soporte amplio para todos los lenguajes utilizados habitualmente en desarrollo web, Emmet para el autocompletado y posibilidad de crear snippets personalizados que aligeran la tarea.



The screenshot shows the VS Code interface. On the left, the 'css.json' file is open, displaying a snippet configuration for a media query. The configuration includes a prefix 'mq', a body with a console log, and a description. On the right, the 'main.css' file is open, showing the application of the snippet, which has generated a media query rule for 'mq' with a color property.

```
1 // Place your snippets for css here. Each snippet is <
2 // description. The prefix is what is used to trigger
3 // $1, $2 for tab stops, $0 for the final cursor posit
4 // same ids are connected.
5 // Example:
6 // "Print to console": {
7 //   "prefix": "log",
8 //   "body": [
9 //     "console.log('$1', '$2')",
10 //   ],
11 //   "description": "Log output to console"
12 // },
13 // "media query": {
14 //   "prefix": "mq",
15 //   "body": "@media (mind-width: $1){\n  $2\n}",
16 //   "description": "snippet para @media"
17 // }
18
19 "media query": {
20   "prefix": "mq",
21   "body": "@media (mind-width: $1){\n  $2\n}",
22   "description": "snippet para @media"
23 }
24
25 "media query": {
26   "prefix": "mq",
27   "body": "@media (mind-width: $1){\n  $2\n}",
28   "description": "snippet para @media"
29 }
```

```
11 color:
12 --fuent
13 --fuent
14 }
15 }
16 mq
17
18 color:
19 font-si
20 line-he
21 }
22
23 ::-moz-se
24 backgro
25 text-sh
26 }
27
28 ::selecti
29 backgro
30 text-sh
31 }
```

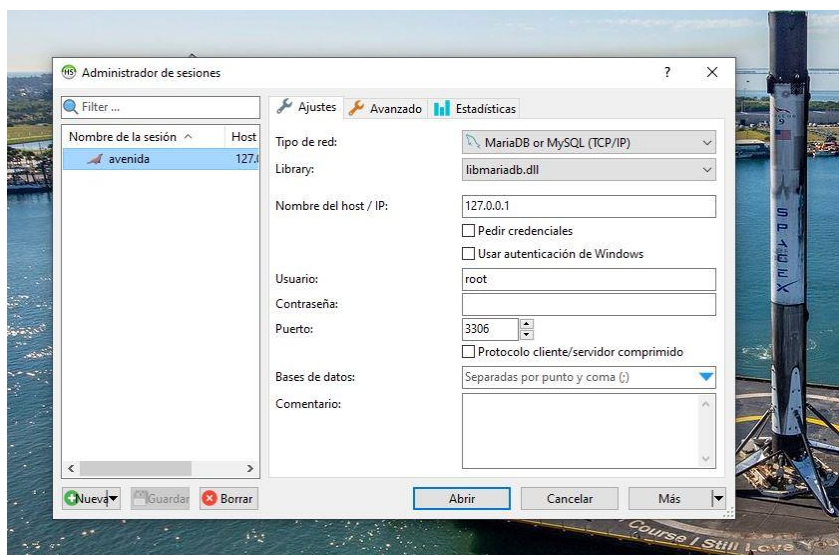
1 Ejemplo de snippet para sacar estructura de Media Query escribiendo mq

En Windows se utiliza el **IDE Atom**, porque es ampliamente personalizable e igual que VS Code tiene una integración sencilla y completa con Git para el control de versiones desde su consola. En caso de Atom la consola se añade con un plugin y se selecciona platformio-ide-terminal con resultados satisfactorios.

Base de datos

La base de datos es **MariaDB** manejada desde **PHPMyAdmin** en Ubuntu y desde **HeidiSQL** en Windows.

Se decide usar estos manejadores debido principalmente a que el proyecto no está especialmente dimensionado en el aspecto de base de datos, y se privilegian manejadores ligeros. PHPMyAdmin viene integrado en el entorno **XAMPP** y se puede acceder desde cualquier navegador, mientras que HeidiSQL (que no cuenta con versión para sistemas operativos con kernel Linux) puede ser utilizado desde su versión portable, esto es, sin necesidad de instalación. En caso de crecimiento de la aplicación sería interesante decantarse por un programa como MySQL Workbench.



2 conexión HeidiSQL portable en Windows 10 en fase de desarrollo

El resto de las tecnologías no difieren apenas del sistema operativo usado y son las que siguen:

UML

Aunque el proyecto no destaca por su dificultad conceptual, al tratar de ofrecerse unos servicios ampliamente estandarizados, bien documentados y conocidos por la práctica totalidad de la población con acceso a Internet, se utiliza UML (*Unidified Modeling Language*) para hacer los diagramas que facilitarán los casos de pruebas. El software utilizado para este cometido es **Draw.io**

Control de versiones

Git, para el control de versiones. Se usa la consola **Git Bash** para el manejo, así como las consolas propias del IDE.

GitHub Desktop, que otorga una capa GUI cómoda para hacer los push y los pull hacia y desde GitHub mientras se ven los cambios.

Prototipado Frontend

Balsamic Wireframes y **Marvel** son los dos programas seleccionados para hacer los mockups del frontend que fueron presentados al cliente.

En una primera aproximación también se utiliza **Note** en sistema operativo iPadOS para hacer bocetos directamente a lápiz sobre iPad. Este es un gran recurso para que el propio cliente pueda dibujar un esquema de lo que realmente desea y tiene en la cabeza, pero algunas veces es complicado verbalizar.

Imagen

Gimp, para edición y reescalado de imágenes mapa de bits.

Responsividad

Responsively App para monitorizar en cada momento el comportamiento de los cambios introducidos en el estilo en diferentes dispositivos (solo para Windows). Este trabajo en Ubuntu se hizo con la herramienta que Firefox integra en su consola.

Para el proyecto no se ha utilizado ningún tipo de framework, pero sí diferentes APIs y librerías, así como HTML5 Boilerplate. A continuación, un repaso a estas herramientas y a los lenguajes que se han utilizado:

Google Developers

Plataforma para desarrolladores de Google que da acceso a todas sus herramientas. Se utiliza para métricas y colocar el mapa (*Google Maps Platform*) en la `contacto.php`

PayPal Developer

Plataforma de pago utilizada para el cobro de los servicios desde la página de compras. Tiene soporte para las principales monedas del mundo, y es un servicio sin coste.

HTML5 (HyperText Markup Language)

Para la estructura de datos. Utilizar las novedades de esta revisión del lenguaje en cuanto a formularios usando los nuevos tipos fue una ventaja decisiva para decidir usar este estándar sobre XHTML.

HTML5 Boilerplate, es una estructura de código muy bien documentada y comentada al extremo que facilita la optimización para dispositivos móviles, la estandarización de estilos entre diversos navegadores, incluido Internet Explorer. Es la base de muchos otros proyectos en producción a lo largo del mundo y es un proyecto que vale la pena conocer.

CSS3 (Cascading Style Sheets)

Para dar el estilo a las páginas, no hay en todo el código un estilo inline, todos los estilos se dan desde una página externa enlazada. Se han aprovechado novedades como display flex y grid, que cuentan ya con soporte amplio para todos los navegadores (especial mención a Firefox Developer Edition para el manejo de grid). **normalize.css**, es usado junto a HTML Boilerplate para hacer la renderización de los estilos similar entre distintos navegadores.

SQL (Structured Query Language)

El proyecto incorpora líneas de código en este lenguaje al estar la parte dinámica del proyecto sustentada en una base de datos relacional. Se utiliza para crear la base de datos, pero también para el **CRUD** (*Create, Read, Update and Delete*).

Java Script

Utilizado en el proyecto para manejar los eventos y lanzar las funciones necesarias en la interacción con el DOM (*Document Object Model*) así como ciertas funciones aritméticas.

Se usa también la biblioteca **jQuery**, para simplificar el acceso al DOM y sobre todo para usar los plugins lettering.js y colorbox.

PHP (Hypertext Preprocessor)

Siendo una página dinámica se utiliza PHP para generar todo el HTML y crear la lógica detrás de la conexión a la base de datos y las consultas. Se utilizan tanto funciones clásicas como el paradigma **POO** (*programación orientada a objetos*) en las partes que tiene más sentido, como la creación de items comercializables.

También se hace uso de **PHPMailer** para el envío de correos electrónicos desde la zona de contacto, ya que la función nativa `mail()`; tiene limitaciones a la hora de utilizar servidores SMTP de determinados dominios.

Las clases de toda esta estructura están bien comentadas, pero son suficientemente complejas como para que intentar modificar algo con éxito sea un reto para alguien que lo tenga muy claro que quiere sufrir.

```

98  if ($plus > 0){
99  $articulo1 = new Item();
100  $articulo1->setName('Plus')
101          ->setCurrency('EUR')
102          ->setQuantity((int) $plus)
103          ->setPrice((float)$precioPlus);
104  array_push($matrizPedido, $articulo1);
105  }

```

3 ejemplo de seteo de una instancia de Item

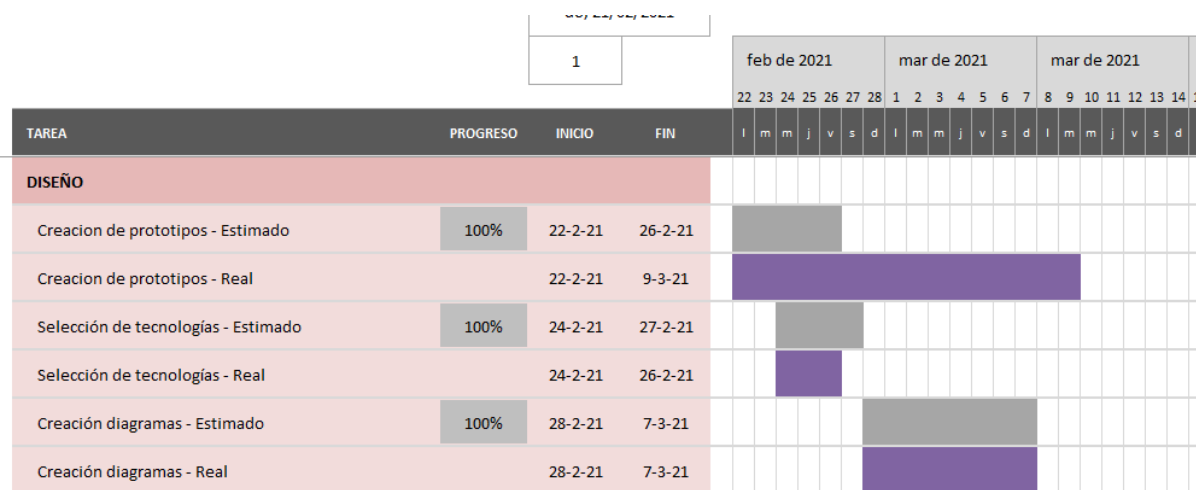
7. Estimación de recursos y planificación

Se realiza un diagrama de Gantt para representar el tiempo planificado y el tiempo real otorgado a cada tarea. Debido al tamaño del diagrama, se copia fragmentado por fases para su correcta visualización.

TINTORERÍA AVENIDA

Desarrollo de Aplicaciones Web - Ilerna Online

Brais Oliveira Sanjurjo



Tintorería Avenida – Brais Oliveira Sanjurjo

1				feb de 2021							mar de 2021							mar de 2021							mar de 2021							
				22	23	24	25	26	27	28	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	2
TAREA	PROGRESO	INICIO	FIN	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	
Implementación																																
Creación entonrno de desarrollo - Estimado	100%	8-3-21	9-3-21																													
Creación entonrno de desarrollo - Real		8-3-21	11-3-21																													
Descarga de H5BP - Estimado	100%	11-3-21	11-3-21																													
Descarga de H5BP - Real		11-3-21	11-3-21																													
HTML Head & Header - Estimado	100%	11-3-21	11-3-21																													
HTML Head & Header - Real		11-3-21	16-3-21																													

3				mar de 2021							mar de 2021							mar de 2021							mar de 2021							abr de 2021							abr de 2021						
				8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
TAREA	PROGRESO	INICIO	FIN	l m m j v s d							l m m j v s d							l m m j v s d							l m m j v s d							l m m j v s d							l m m j v s d						
Agreación Fontawesome & Google Fonts - Estimado	100%	11-3-21	11-3-21																																										
Agreación Fontawesome & Google Fonts - Real		11-3-21	11-3-21																																										
CSS globals -Estimado	100%	12-3-21	12-3-21																																										
CSS globals - Real		13-3-21	20-4-21																																										
CSS Header - Estimado	100%	12-3-21	13-3-21																																										
CSS Header - Real		13-3-21	14-3-21																																										
HTML navegacion principal - Estimado	100%	14-3-21	14-3-21																																										
HTML navegacion principal - Real		14-3-21	14-3-21																																										
CSS navegacion principal - Estimado	100%	14-3-21	15-3-21																																										
CSS navegacion principal - Real		14-3-21	17-3-21																																										

				mar de 2021							mar de 2021							mar de 2021							mar de 2021						
				8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4
TAREA	PROGRESO	INICIO	FIN	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d
1ª Revisión responsividad (media queries) - Estimado	100%	15-3-21	16-3-21																												
1ª Revisión responsividad (media queries) - Real		15-3-21	17-3-21																												
HTML body index - Estimado	100%	18-3-21	18-3-21																												
HTML body index - Real		19-3-21	31-3-21																												
Selección video e imágenes -Estimado	100%	18-3-21	19-3-21																												
Selección vídeo e imágenes - Real		19-3-21	31-3-21																												
CSS body index sections -Estimado	100%	18-3-21	30-3-21																												
CSS body index sections -Real		20-3-21	31-3-21																												
HTML comprar forms -Estimado	100%	20-3-21	21-3-21																												
HTML comprar forms - Real		21-3-21	21-3-21																												

Tintorería Avenida – Brais Oliveira Sanjurjo

				mar de 2021							mar de 2021							mar de 2021							mar de 2021							abr de 2021							abr de 2021																																			
				3																																																																						
				8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18																													
TAREA	PROGRESO	INICIO	FIN																																																																							
2ª revisión responsividad - Estimado	100%	23-3-21	23-3-21																																																																							
2ª revisión responsividad - Real		28-3-21	28-3-21																																																																							
JS manejo eventos form compra - Estimado	100%	23-3-21	26-3-21																																																																							
JS manejo eventos form compra - Real		29-3-21	29-3-21																																																																							
JS aritmetica compras - Estimado	100%	26-3-21	28-3-21																																																																							
JS aritmetica compras - Real		30-3-21	1-4-21																																																																							
Añadir mapa API Google - Estimado	100%	29-3-21	29-3-21																																																																							
Añadir mapa API Google - Real		2-4-21	2-4-21																																																																							
jQuery añadir letterings.js (plugin) - Estimado	100%	29-3-21	29-3-21																																																																							
jQuery añadir lettrings.js (plugin) - Real		16-4-21	16-4-21																																																																							

			4	mar de 2021							mar de 2021							mar de 2021							abr de 2021							abr de 2021						
				15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
TAREA	PROGRESO	INICIO	FIN	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v		
HTML galeria staff instalaciones - Estimado	100%	31-3-21	31-3-21																																			
HTML galeria staff instalaciones - Real		3-4-21	3-4-21																																			
GIMP modificacion img - Estimado	100%	2-4-21	6-4-21																																			
GIMP modificacion img - Real		4-4-21	4-4-21																																			
jQuery añadir lightbox.js (plugin) - Estimado	100%	7-4-21	8-4-21																																			
jQuery añadir lightbox.js (plugin) - Real		5-4-21	5-4-21																																			
jQuery - CSS menu navegacion responsivo - Estimado	100%	10-4-21	11-4-21																																			
jQuery - CSS menu navegacion responsivo - Real		6-4-21	9-4-21																																			
SQL creación de base de datos - Estimado	100%	12-4-21	12-4-21																																			
SQL creación de base de datos - Real		12-4-21	14-4-21																																			

				8							abr de 2021							abr de 2021							a
											12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
TAREA	PROGRESO	INICIO	FIN								l	m	m	j	v	s	d	l	m	m	j	v	s	d	l
PHP creación de includes (modularización) - Real		13-4-21	13-4-21																						
PHP creación archivo de conexión BD - Estimado	100%	14-4-21	14-4-21																						
PHP creación archivo de conexión BD - Real		14-4-21	14-4-21																						
PHP - SQL creación de consultas - Estimado	100%	15-4-21	18-4-21																						
PHP - SQL creación de consultas - Real		15-4-21	17-4-21																						
jQuery envíos email contacto.php - Estimado	100%	18-4-21	23-4-21																						
jQuery envíos email contacto.php - Real		15-4-21	21-4-21																						
PHP pasarela de pago API PayPal - Estimado	100%	18-4-21	23-4-21																						
PHP pasarela de pago API PayPal - Real		15-4-21	18-4-21																						
Refactoring - Comentarios código - Estimado	100%	24-4-21	25-4-21																						
Refactoring - Comentarios código - Real		???	???																						

4 El refactorizado y los comentarios se extienden hasta el último momento

				9							abr de 2021														a
											19	20	21	22	23	24	25	26							
TAREA	PROGRESO	INICIO	FIN								l	m	m	j	v	s	d	l							
Pasarela de pago - Real		25-4-21	25-4-21																						
Validación de campos - Estimado	100%	25-4-21	25-4-21																						
Validación de campos - Real		25-4-21	25-4-21																						
Envío de correo - Estimado	100%	25-4-21	25-4-21																						
Envío de correo - Real		25-4-21	25-4-21																						
Inserción correcta BBDD - Estimado	100%	25-4-21	25-4-21																						
Inserción correcta BBDD - Real		25-4-21	25-4-21																						
Responsividad - Estimado	100%	25-4-21	25-4-21																						
Responsividad - Real		25-4-21	25-4-21																						

8. Análisis

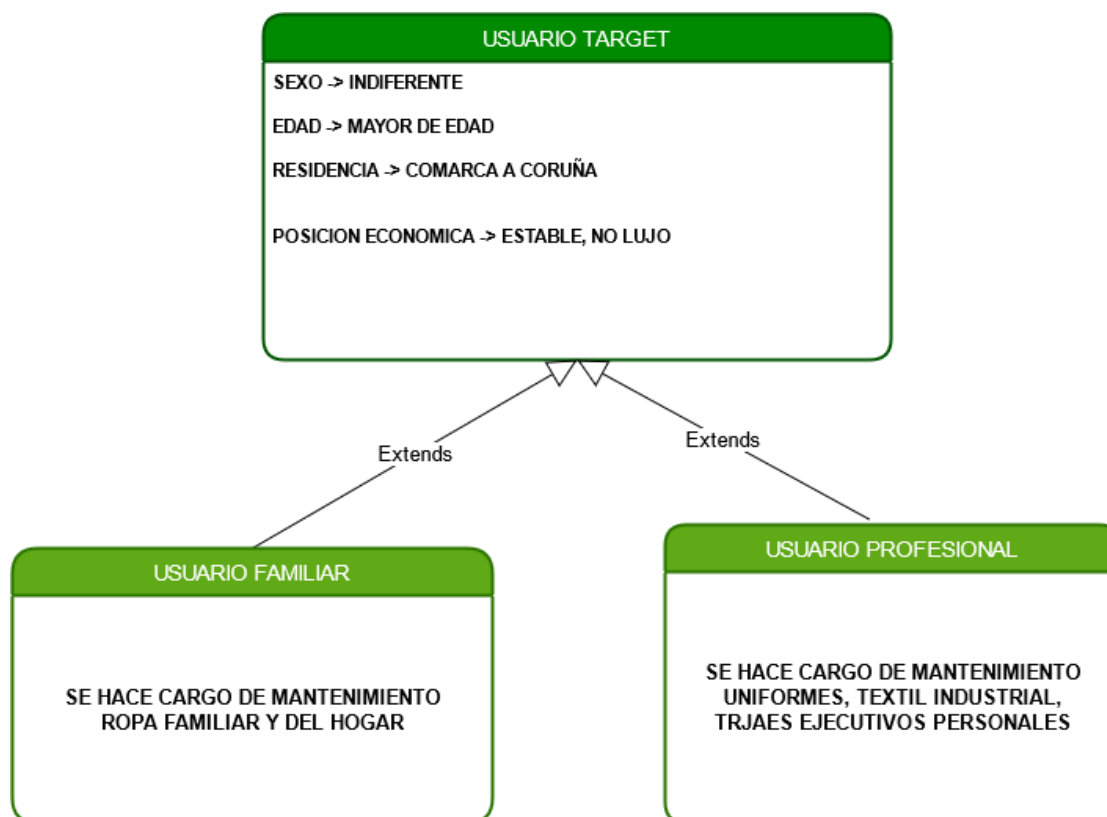
En esta parte empezaremos definiendo una persona idealizada que actuará como “persona objetivo” del negocio, para luego trasladar a UML los casos de uso de la página web, bajo un enfoque de funcionalidades posibles.

Los diagramas se acompañan de tablas donde se añaden datos y textos aclaratorios. Se espera que estas tablas y diagramas sean de ayuda en el proceso de pruebas.

Persona

Con el objetivo de acertar lo más posible en las decisiones se crea un usuario idealizado que representa el target natural del negocio. Esto habrá que tenerlo en cuenta posteriormente en la toma de decisiones de imagen y textos (ver Diseño) pero también en la parte de casos de uso.

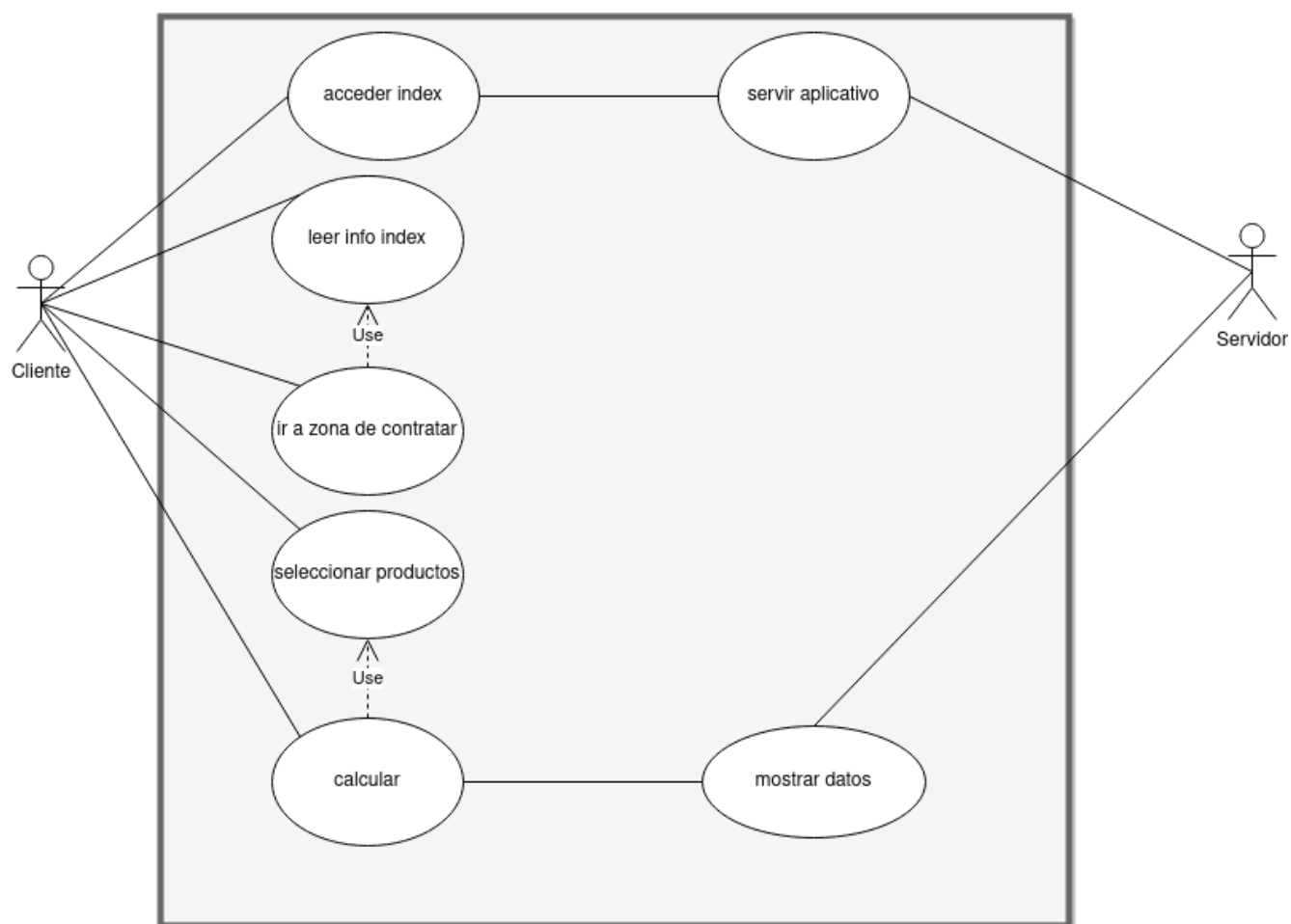
Para ilustrar el público objetivo se hace el siguiente diagrama.



5 Público objetivo de la aplicación

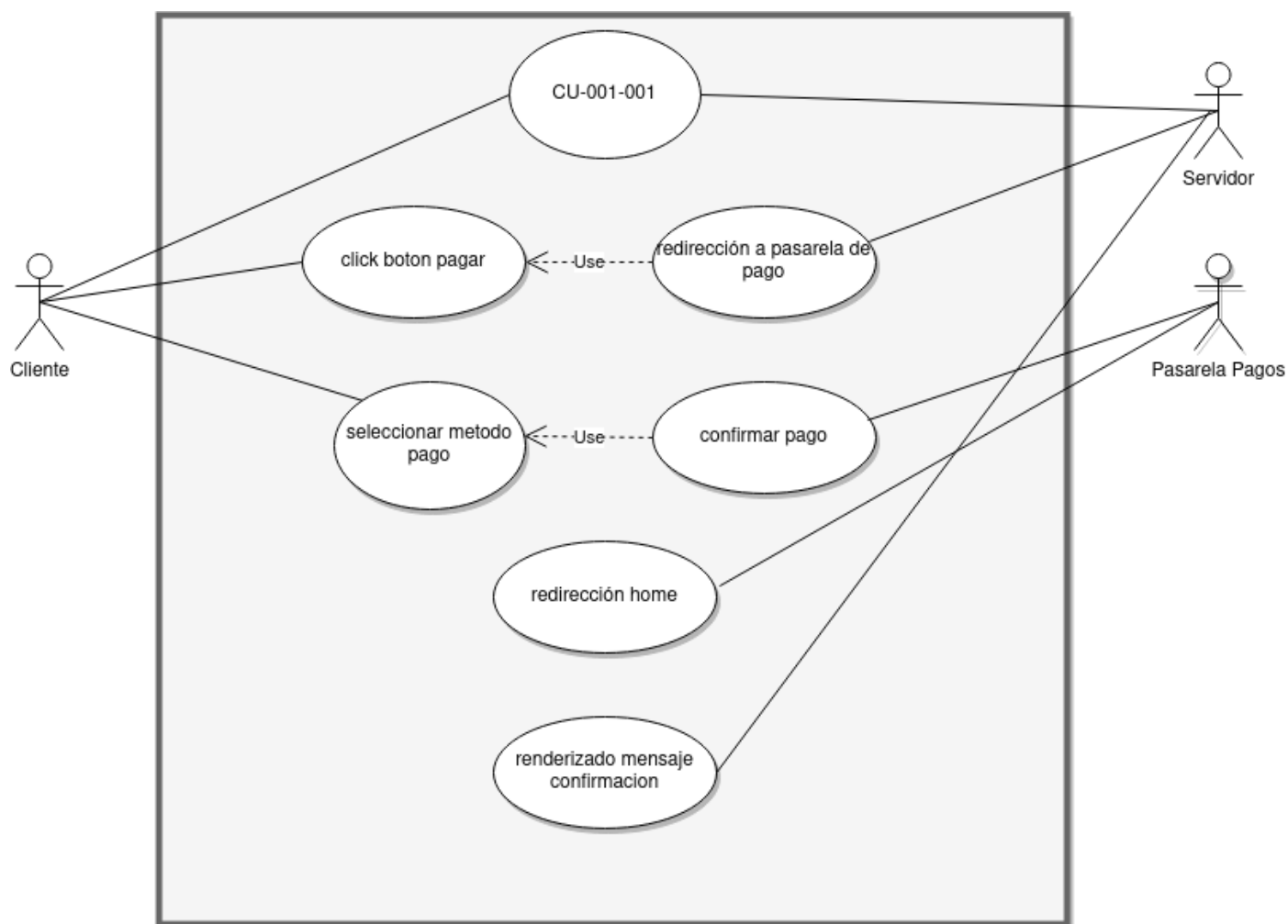
Casos de uso

Nombre del caso de uso		Consulta precios y servicios
<i>ID del caso de uso</i>		CU-001-001
<i>Descripción</i>		Caso de un cliente desear información acerca de lo ofrecido por la empresa
<i>Actores implicados</i>		<ul style="list-style-type: none"> • Usuario ajeno al sistema • Sistema
<i>Precondiciones</i>		Necesario poder acceder a Internet (dispositivo con conexión de red, navegador instalado). Java Script activado en el navegador
<i>Curso normal</i>		<ol style="list-style-type: none"> 1. Ingresar al sitio mediante un navegador 2. Scroll en Home para ver la información 3. Click en barra de navegación para ir a “contratar” donde se encuentra la misma información y una calculadora de montos 4. Seleccionar productos y cantidades 5. Click en botón “calcular”
<i>Postcondiciones</i>		Ninguna, se mantienen las condiciones previas
<i>Alternativas</i>		Si Java Script no está activado el propio navegador emite una alerta y permite activarlo



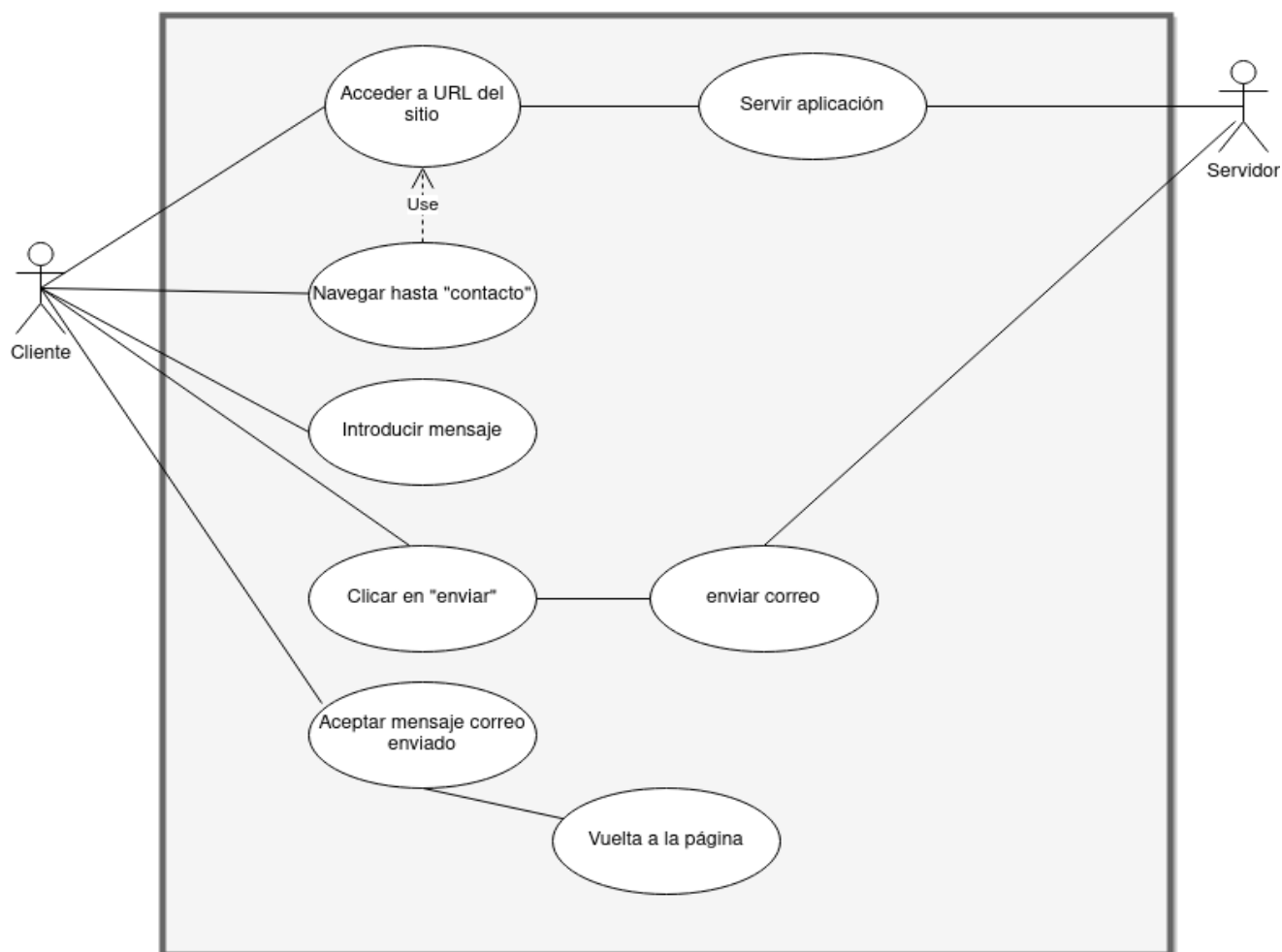
6 Consulta de precios CU-001-001

Nombre del caso de uso	Compra producto
<i>ID del caso de uso</i>	CU-002-001
<i>Descripción</i>	Caso en el que un cliente compra y paga uno o varios productos de los ofertados
<i>Actores implicados</i>	<ul style="list-style-type: none"> • Usuario ajeno al sistema • Sistema • Pasarela de Pago
<i>Precondiciones</i>	Necesario poder acceder a Internet (dispositivo con conexión de red, navegador instalado). Java Script activado en el navegador. Caso CU-001-001 como subordinado.
<i>Curso normal</i>	<ol style="list-style-type: none"> 1. Se realiza sin errores el CU-001-001 2. Click en botón pagar 3. Ir a pasarela de pago 4. Seleccionar método 5. Confirmar pago 6. Mensaje éxito o error 7. Volver al home
<i>Postcondiciones</i>	Ninguna, se mantienen las condiciones previas
<i>Alternativas</i>	Mismas que CO-001-001 + pueden producirse errores en la pasarela de pago ajenos al cliente o al servidor de la aplicación, dador que es un servicio externo



7 Compra producto CU-002-001

Nombre del caso de uso	Contacto email
<i>ID del caso de uso</i>	CU-003-001
<i>Descripción</i>	Caso de uso de la herramienta de correo
<i>Actores implicados</i>	<ul style="list-style-type: none"> • Usuario ajeno al sistema • Sistema
<i>Precondiciones</i>	Necesario poder acceder a Internet (dispositivo con conexión de red, navegador instalado). Java Script activado en el navegador. No necesario que el usuario posea cuenta de correo electrónico.
<i>Curso normal</i>	<ol style="list-style-type: none"> 1. Ingresar al sitio mediante un navegador 2. Navegar desde la barra superior hasta “contacto” 3. Introducir datos de contacto y mensaje 4. Clicar en botón “enviar” 5. Mensaje de correo enviado 6. Clicar en “aceptar” para volver a la página
<i>Postcondiciones</i>	Ninguna, se mantienen las condiciones previas

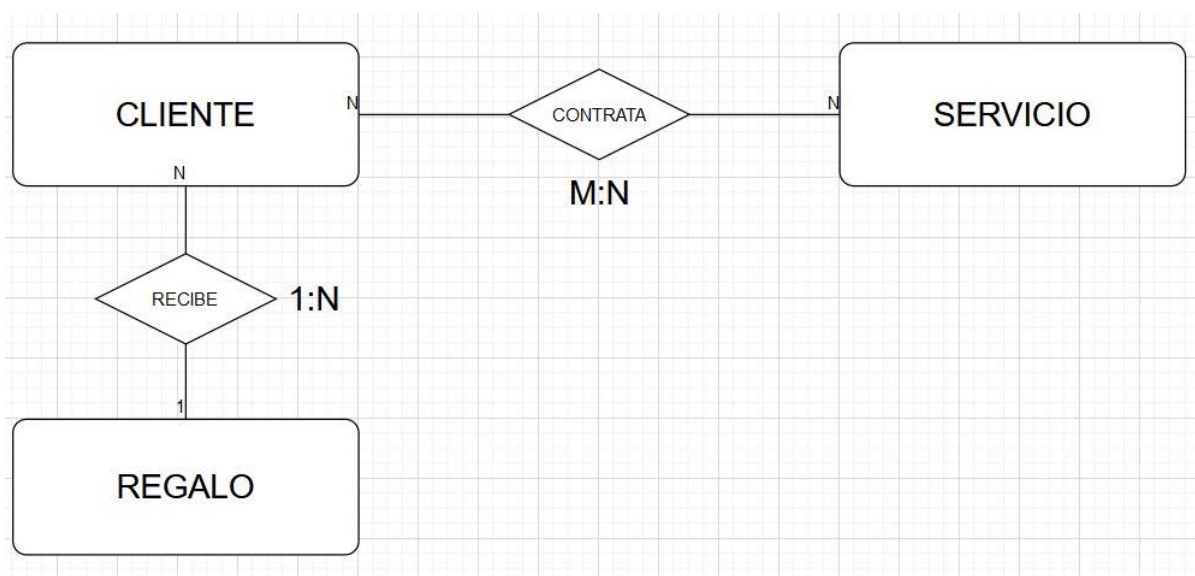
*8 Contacto email CU-003-001*

Estructura de datos

En esta fase se analiza también la estructura de datos, para lo que se hacen diagramas de entidad – relación.

El programa necesita registrar las compras. Para esto hace falta una tabla de CLIENTE, una tabla de producto susceptible de ser comprado que en este caso se denomina SERVICIO y cada cliente recibirá un detalle seleccionable por el propio usuario. Se crea una tabla para registrar estos productos con el nombre REGALO.

En una primera aproximación la base de datos se pudo estructurar así:



9 Primera aproximación objeto – relacional, se obvian las elipsis de los atributos

Un cliente puede contratar N servicios y estos a su vez pueden ser solicitados por N clientes creando una cardinalidad de muchos a muchos.

A su vez el cliente recibe un regalo. Este tipo de regalo puede ser solicitado por N clientes creando la cardinalidad 1:N que se resolvería sin tabla intermedia propagando la clave foránea de la tabla REGALO a la tabla CLIENTE.

Finalmente se llega a la conclusión de que es mejor que el regalo no se asocie al cliente, si no a la tabla intermedia SERVICIO_CLIENTE que se genera fruto de la cardinalidad entre CLIENTE y SERVICIO. Esto es debido a que según la lógica de negocio el cliente tiene un regalo cada vez que paga por un servicio.

En el proceso de desarrollo se han realizado también diagramas de grafos para sintetizar esquemáticamente el resultado final a implementar. A continuación, se deja un ejemplo ilustrativo de la base de datos ya en la tercera forma normal.

ANÁLISIS BBDD TINTORERÍA AVENIDA

DIAGRAMA DE GRAFOS

SERVICIO (PK id-servicio, nombre-servicio, descripcion-servicio, precio-servicio, img-servicio)

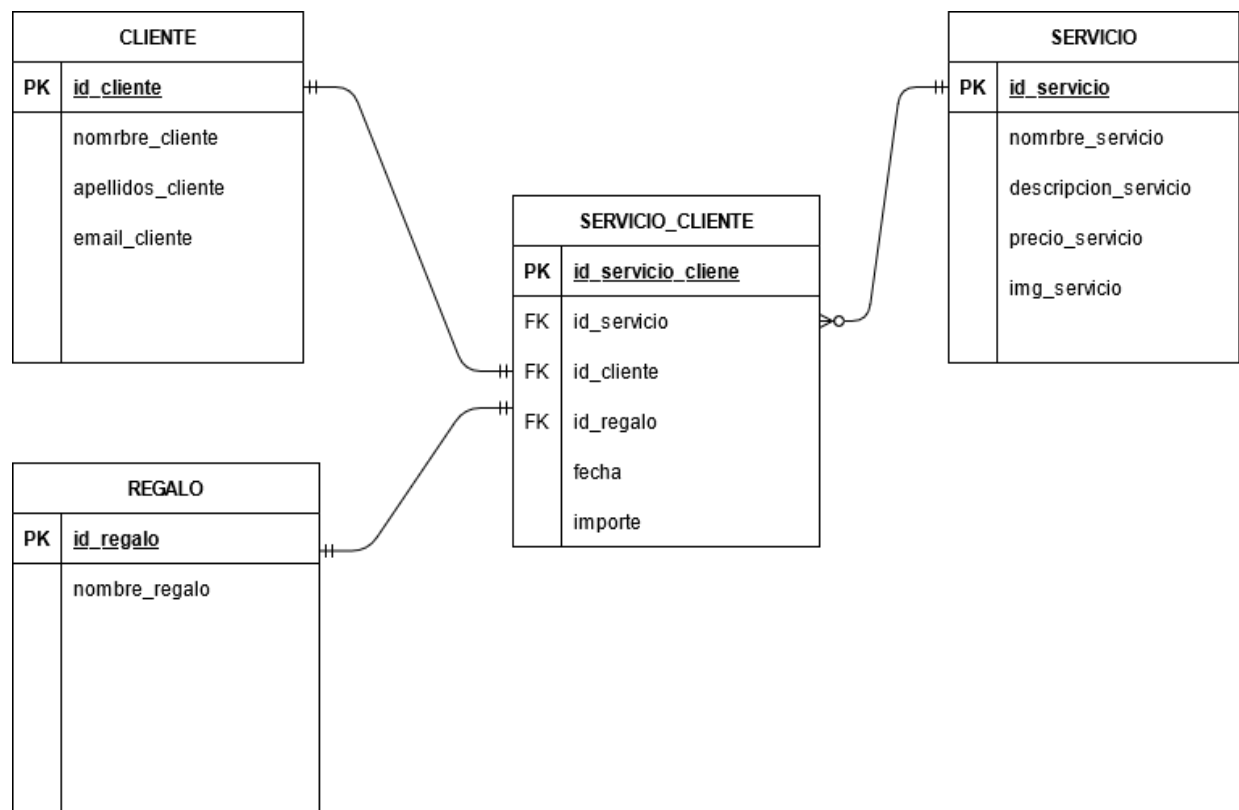
CLIENTE (PK id-cliente, nombre-cliente, apellidos-cliente, email-cliente)

REGALO (PK id-regalo, nombre-regalo)

SERVICIO-CLIENTE (PK id-servicio-cliente, FK id-servicio, FK id-cliente, FK id-regalo, fecha-servicio-cliente, precio-total)

Brais Oliveira Sanjurjo

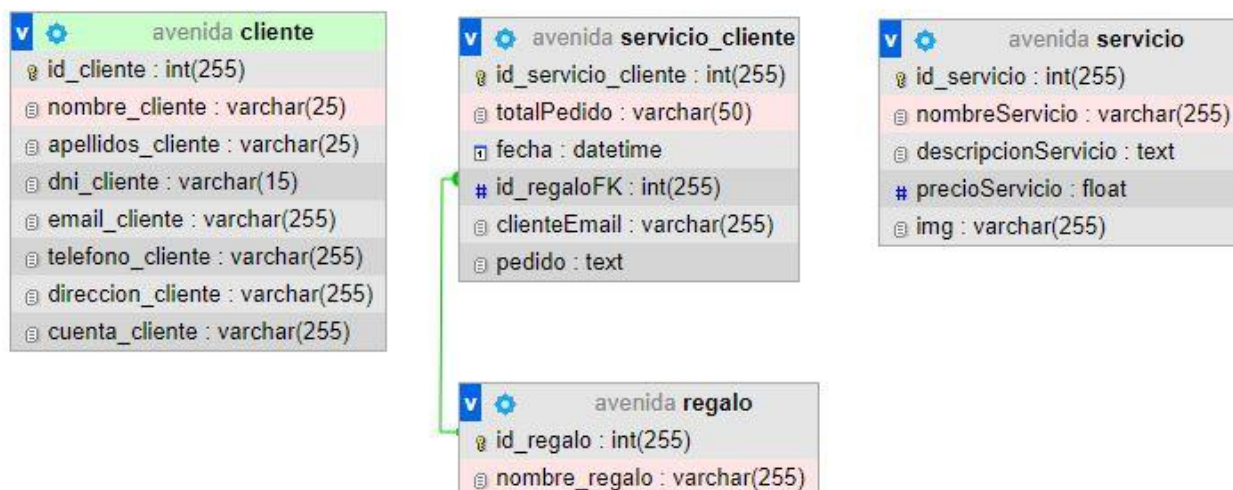
Podemos ver el diagrama de la siguiente manera:



11 diagrama base de datos Avenida normalizada

En la aplicación se mantienen las tablas, pero se añaden campos no obligatorios (*mandatory*) pensando en futuras ampliaciones del proyecto. Estos campos no se están cumplimentando desde la web en estos momentos, ya que no hacen falta para el correcto funcionamiento, y por evitar problemas con la ley de protección de datos (LOPD) mientras no se tenga acceso a un aval jurídico especializado.

Se ilustra también lo comentado anteriormente en la siguiente captura sacada de PHPMyAdmin.



12 Muestra base de datos desde PHPMysqlAdmin

9. Diseño

En la parte de diseño se privilegia la usabilidad sobre las pretensiones artísticas, sin renunciar a una correcta selección de paletas, fuentes y disposición espacial de los elementos, respetando los espacios entre estos.

Todas las páginas están divididas en 4 zonas:

Cabecera y pie idénticos, y en el cuerpo dos secciones diferenciadas. En algún caso hay `<main>` y `<section>` o si las dos partes tienen la misma importancia dos `<section>`

Accesibilidad

La web sigue el estándar de **HTML semántico** implementado desde la versión 5. Esto se hace siguiendo las recomendaciones **WCAG 2.0** (*Web Content Accessibility Guidelines*) promovidas por W3C (*World Wide Web Consortium*).

De este modo se diseña teniendo en cuenta que ni el color ni la fuente hagan la web difícil de percibir.

No se usan colores que impliquen un significado funcional que no estén acompañados del texto correspondiente, de hecho, siendo el logotipo de la empresa de color rojo (color asociado tradicionalmente a la funcionalidad de borrado, eliminar, delete o drop) se usa un

tipo de rojo y una variante secundaria a lo largo de todo el proyecto sin connotación funcional alguna.

Los textos están redactados en un registro formal pero cercano al habla habitual en un entorno familiar con el objetivo de que sean fácilmente comprensibles por la mayor cantidad de gente posible. Se entiende que el target de la web es el de un usuario adulto que se encarga de la limpieza y mantenimiento de prendas de ropa y dispone de la posibilidad de hacer un pedido de manera autónoma (ver Diseño – Persona).

Siguiendo una estructura habitual en la navegación de la web, con una cabecera que se repite en todas las páginas e incluye una barra de navegación con el nombre de cada destino, una indicación de cuál es la página que está activa y una navegación intra-página con scroll vertical, se consigue una gran facilidad de operabilidad, ya que este modo de usar la web es un estándar de facto.

Se incluyen animaciones, pero son siempre activadas por la interacción del usuario con la interfaz (IPO), no hay en la web ninguna animación automática.

A si mismo, las imágenes contienen el pertinente atributo *alt* para los lectores de pantalla y no se incluyen tablas desordenadas para estos lectores.

```
<div class="galeria">  
  <a href="img/mujer-sonriendo.jpg" data-lightbox="galeria">  
      
  </a>
```

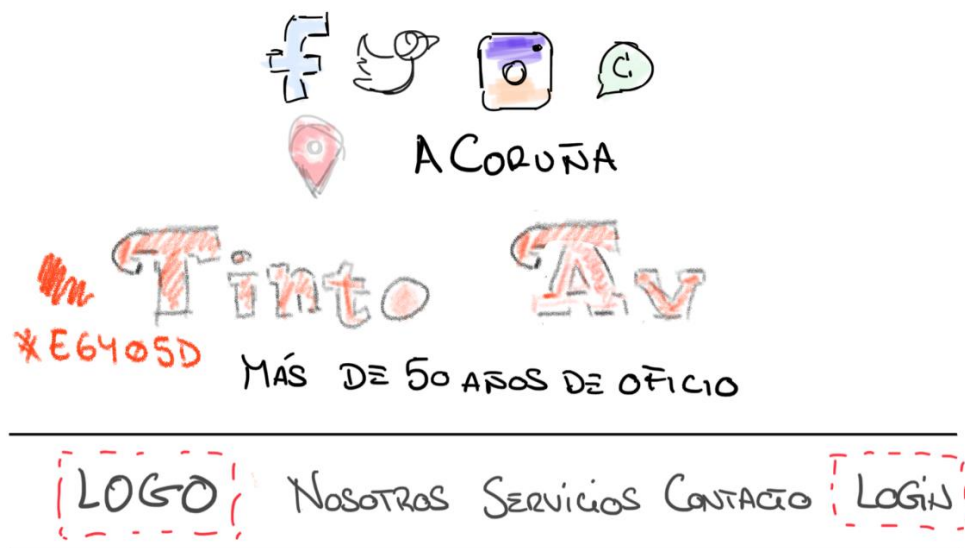
13 Ejemplo de atributo alt

Wireframes

Los bocetos se presentan en primera instancia a mano para luego hacer un prototipo en alguna de las aplicaciones seleccionadas.

Header

La cabecera se repite mediante un `include_once` en todas las páginas de la web.



14 Primer boceto de cabecera



15 Boceto maquetado desde MarvelApp para la cabecera



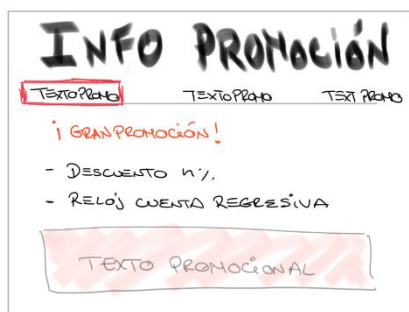
16 Resultado final de cabecera desde máquina virtual Windows utilizada en entorno de desarrollo

Index

La parte del index se divide en dos zonas. Una en la que se da información sobre las promociones temporales, y otra donde se informa de los planes fijos que se pueden contratar, que en principio solo varían en el precio con el paso del tiempo, y esto de manera muy espaciada.

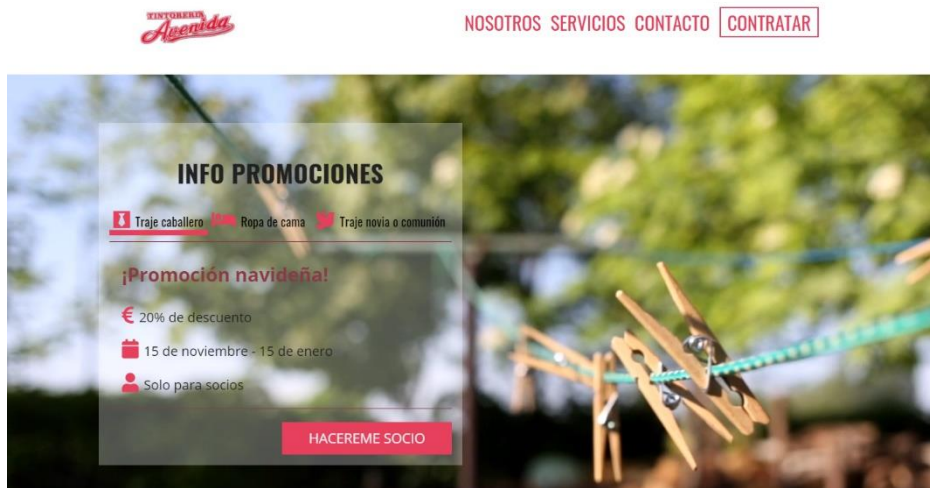
Zona de novedades

La zona para la información temporal se resuelve con un frame navegable con su propia barra interna y un background semi transparente sobre un gif en movimiento.



17 De esta manera fue imaginada en primera instancia la zona de novedades

Los bocetos de esa zona del index acabaron en el resultado que se muestra en la siguiente imagen.



18 Resultado zona de novedades

Información permanente

Para este apartado, no se hace una barra navegable, ya que son tres productos similares que conviene comparar para que el usuario decida cual es más conveniente para él.


Se realiza un solo boceto con varias fuentes para comparar que queda de la siguiente manera:

TARIFA PLANA



19 Boceto de segunda zona de index

Y una vez pasado a código queda de la siguiente manera:



[NOSOTROS](#)
[SERVICIOS](#)
[CONTACTO](#)
[CONTRATAR](#)

TARIFA PLANA

Basic	Plus	Premium
44.00€	77.00€	99.00€
<ul style="list-style-type: none"> ✓ UNA RECOGIDA SEMANAL ✓ UNA DE NUESTRAS BOLSAS LLENA ✓ EN 24 HORAS TENDRÁS TU ROPA LIMPIA, SECA Y DOBLADA 	<ul style="list-style-type: none"> ✓ UNA RECOGIDA SEMANAL ✓ DOS DE NUESTRAS BOLSAS LLENAS ✓ SELECCIONA 5 PRENDAS QUE QUIERES QUE TE PLANCHAMOS ✓ EN 48 HORAS TENDRÁS TU ROPA LIMPIA, SECA, DOBLADA Y PLANCHADA 	<ul style="list-style-type: none"> ✓ UNA RECOGIDA SEMANAL ✓ DOS DE NUESTRAS BOLSAS LLENAS ✓ SELECCIONA 12 PRENDAS QUE QUIERES QUE TE PLANCHAMOS ✓ EN 48 HORAS TENDRÁS TU ROPA LIMPIA, SECA, DOBLADA Y PLANCHADA
SUBSCRIBIRME	SUBSCRIBIRME	SUBSCRIBIRME

20 Resultado final segunda zona index

Nosotros

En la zona destinada a la información sobre la propia empresa se incluye un texto y una galería de imágenes en la que se ponen las instalaciones y empleados.

Esta galería se realiza con la ayuda de la biblioteca de CSS lightbox.css para dar los efectos.

Servicios

En este apartado pueden consultarse en profundidad los diferentes trabajos en los que el negocio está especializado.

Se realiza con texto, imágenes de stock y propias del establecimiento. Se da animación a la página con CSS, ampliando las cajas de contenido en :hover, esto amplía la fuente en a 1.1 y aumenta el contraste de la caja de texto seleccionada, siendo un buen ejemplo de estilo que puede hacer más interesante la página y al mismo tiempo más accesible.

Contacto

En contacto.php encontramos una primera zona con una herramienta de envío de email y datos de contacto tradicionales como la dirección postal, el número de teléfono y el horario de apertura.

Debajo de esta zona se sitúa el mapa traído desde la API de Google.

ZONA DE CONTACTO

B **I** **U** ABC

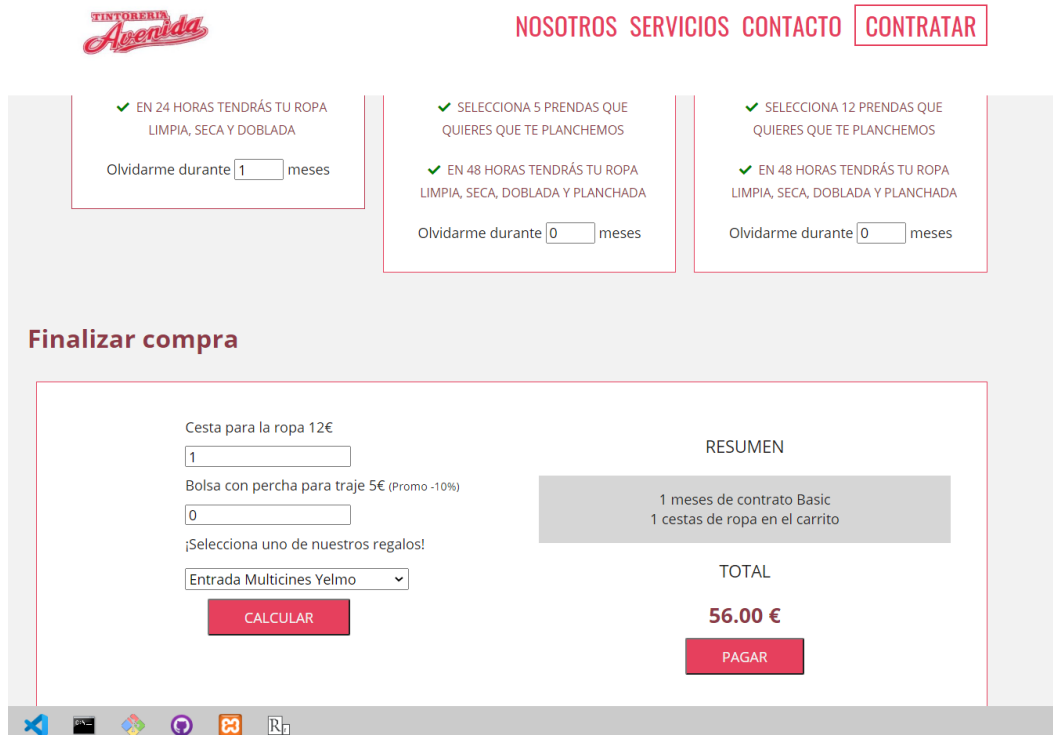
Type here your message

Button

21 Boceto para la zona de contacto

Contratar

Contiene un input para ingresar datos personales, una información sobre los planes disponibles que es la misma que se encuentra en el index pero sin animación CSS con selectores para poder hacer la compra del servicio. Hay además una herramienta calculadora útil para ver el montante de la operación y los items seleccionados, y un botón de submit que envía los datos a la base de datos y lanza la pasarela de pago de PayPal.



Tintorería Avenida

NOSOTROS SERVICIOS CONTACTO **CONTRATAR**

✓ EN 24 HORAS TENDRÁS TU ROPA LIMPIA, SECA Y DOBLADA
Olvidarme durante meses

✓ SELECCIONA 5 PRENDAS QUE QUIERES QUE TE PLANCHEMOS
✓ EN 48 HORAS TENDRÁS TU ROPA LIMPIA, SECA, DOBLADA Y PLANCHADA
Olvidarme durante meses

✓ SELECCIONA 12 PRENDAS QUE QUIERES QUE TE PLANCHEMOS
✓ EN 48 HORAS TENDRÁS TU ROPA LIMPIA, SECA, DOBLADA Y PLANCHADA
Olvidarme durante meses

Finalizar compra

Cesta para la ropa 12€

Bolsa con percha para traje 5€ (Promo -10%)

¡Selecciona uno de nuestros regalos!
Entrada Multicines Yelmo

CALCULAR

RESUMEN
1 meses de contrato Basic
1 cestas de ropa en el carrito

TOTAL
56.00 €
PAGAR

22 Aspecto definitivo de la calculadora de pagos

10. Despliegue y pruebas

Despliegue

Para el despliegue de la página se utiliza un dominio comprado a Dinahosting y se despliega en un host de la misma compañía.

Es accesible desde la dirección:

<http://braisoliveira.eu>

Pruebas

En la práctica se realizan pruebas informales durante todo el proceso de codificado, de todas maneras se reserva un día para hacer una última sesión de testeo pormenorizado con los siguientes resultados.

VERDE = SATISFACTORIO

NARANJA = MEJORA A IMPLEMENTAR

ROJO = REPARACION URGENTE

Nº	ESPECIFICACIÓN	RESULTADO
1	Funcionamiento correcto de enlaces entre páginas	
2	Aritmética correcta en herramienta de compra	
3	Funcionamiento correcto pasarela de pago	
4*	Validación de campos en formularios	
5	Envío de correo desde herramienta de contacto	
8	Inserción correcta en base de datos	
7	Responsividad imágenes y menús	

- En el caso de pruebas nº4 se comprueba que las validaciones deberían ser realizadas desde el lado del servidor y que en el lado cliente se deberían utilizar expresiones regulares robustas y no simples validaciones HTML5.

11. Conclusiones

Se presentó este trabajo por un lado como oportunidad de estudio empírico sobre la conveniencia de realizar desarrollos web personalizados para pequeñas empresas y por otro como oportunidad de demostración de las habilidades aprendidas durante el Ciclo Formativo y la capacidad de desenvolverse profundizando en el temario por iniciativa propia o directamente tomar caminos que no se habían explorado en las clases, todo sin soporte más allá del que cualquier profesional pueda encontrar en su día a día.

Finalmente, se han alcanzado casi todas las metas grandes propuestas en la propuesta de TFC. La página está funcionando según los parámetros previamente establecidos y pasada del entorno de desarrollo a un hosting externo accesible desde Internet.

Por el camino han quedado sin embargo en el tintero algunas cosas por falta de tiempo.

Por ejemplo, debe optimizarse el tamaño de las imágenes, ya que en algún caso estos están determinados con líneas de CSS en lugar de crear la imagen adecuada con una herramienta de edición de imagen. Puede valer con imágenes vectoriales, pero la mayor parte de las imágenes en el proyecto son mapa de bits.

El desempeño de la página en pantallas reducidas como la de un smartphone es correcto en cuanto a la funcionalidad, se puede acceder a todos los apartados y realizar las operaciones, pero haría falta ampliar la batería de media queries para optimizar los estilos en estos

dispositivos. El tamaño de los inputs puede ser un problema para personas con problemas de motricidad.

El sistema de seguimiento de prendas no se ha podido hacer debido a que hace falta un nuevo módulo que haga de dashboard para alguien de la tintorería y pueda ir dando información al cliente por vía de algún sistema de mensajería. No se encuentra una solución automatizada a ese proceso, debería hacerlo un ser humano.

La creación de ese panel de administración implicaría más de una semana de dedicación. Pasa a vías futuras.

En cuanto a la conveniencia de los desarrollos personalizados y a la no utilización de CMS queda demostrado ser muy beneficioso, ya que lo que se ha podido hacer con los recursos de un solo estudiante en 9 semanas con tiempo limitado (empleado a tiempo completo como administrador de sistemas en una compañía) es bastante remarcable dadas las circunstancias. ¿Qué podría haber hecho un desarrollador senior?, ¿y un equipo de programadores coordinados?

La conclusión es que si se están utilizando sistemas administradores de contenido más allá de la casuística recomendada es porque hay mucha gente ofertando servicios web que simplemente no quiere tomarse el tiempo que lleva aprender a programar. Y los que lo acaban sufriendo son los clientes.

Está claro que el proceso para llegar a ser un desarrollador senior es un camino arduo y largo, como casi siempre lo es el camino correcto.

12. Vías futuras

La primera acción será afinar el tamaño en el tipo de datos de los atributos en la base de datos. Se ha dejado como poco prioritario porque en una base de datos tan pequeña la diferencia de tamaño y rendimiento no se va a notar, pero es algo que hay que hacer.

En futuras iteraciones podría añadirse una zona de registro de usuarios que otorgue algún tipo de privilegio o acceso a recursos de la web solo para usuarios registrados.

Esto es sencillo de hacer con una cookie que almacene en la variable `$_SESSION` el valor `true` o `false`. En la versión actual del programa no se añade esta funcionalidad porque el propietario no desea dificultar en ningún caso el acceso ni la compra de productos. Solo hace

falta añadir un nombre y apellidos (nada comprueba que sean reales) y un correo de contacto en caso de realizar alguna compra.

Otra ampliación que se podría hacer a futuro es la de tracking de prendas, de manera que los clientes tengan en todo momento un feedback de en qué proceso se encuentran y cuando serán entregadas.

13. Glosario y acrónimos

API	Application Programming Interface
CMS	Content Management System
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DNS	Domain Name System
DOM	Document Object Model
HTML	HyperText Markup Language
FTP	File Transfer Protocol
FTTH	Fiber To The Home
FK	Foreign Key
ID	Identificador
IDE	Integrated Development Environment
LOPD	Ley Orgánica de Protección de Datos
PHP	Hypertext Preprocessor
PK	Primary Key
Script	Conjunto de órdenes en un lenguaje de programación
SQL	Structured Query Language
SMTP	Simple Mail Transfer Protocol
Snippet	Pequeño fragmento reusable de código fuente

UML Unified Modeling Language

14. Bibliografía / Webgrafía

Accesibilidad

Revilla Muñoz, Olga y Carreras Montoro, Olga (2018). *Accesibilidad Web: WCAG 2.1 de forma sencilla* Madrid: Itákora Press

Inclusite <https://www.inclusite.com/es/>

Ciclo de vida de software

Roger S. Pressman, Ph.D. (2010). *Ingeniería del software: Un enfoque práctico*

México DF:McGraw-Hill

CSS

CSSMatic <https://www.cssmatic.com>

Can I Use <https://caniuse.com>

Iconos

Fontawesome <https://fontawesome.com/>

Tablericons <https://tablericons.com/>

Mapas

Leaflet <https://leafletjs.com/>

Google Maps <https://console.cloud.google.com/google/maps-apis>

Paletas

Coolors <https://coolors.co/>

Colourlovers <https://www.colourlovers.com/>

Pasarela de pago

PayPal <https://paypal.github.io/PayPal-PHP-SDK>

PHP

PHP docs <https://www.php.net/docs.php>