

# RENITE

## *Template de React para páginas web de producto nicho*

Autor: Francisco Castillo Martínez

Tutor: Mikel Zorrilla Berasategui

Profesor: Ferran Adell Español

Grado de Multimedia

Ingeniería Web

# Créditos/Copyright



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

# FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>RENITE, Template de React para páginas web de producto nicho</i>
<b>Nombre del autor:</b>	<i>Francisco Castillo Martínez</i>
<b>Nombre del colaborador/a docente :</b>	<i>Mikel Zorrilla Berasategui</i>
<b>Nombre del PRA:</b>	<i>Ferran Adell Español</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2022</i>
<b>Titulación o programa:</b>	<i>Grado de Multimedia</i>
<b>Área del Trabajo Final:</b>	<i>76.570 TFG-Ingeniería Web</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>desarrollo, web, nicho, plantilla, accesible</i>
<b>Resumen del Trabajo:</b>	
<p>Identificar un nicho de mercado abre muchas posibilidades en internet pero muchas veces se ven frustradas por la falta de infraestructura, capacidad de negocio, tiempo o riesgo del proyecto. Tecnologías que permiten un desarrollo rápido y reutilizable y modelos de datos sencillos, junto con la posibilidad de retribución económica que presentan los modelos de enlace de afiliado, consiguen la oportunidad de crear páginas de venta sin la necesidad de tener stock y gestionar el proceso de compra, sumándose al proceso de venta de terceros.</p> <p>Este tipo de proyectos carecen de escalabilidad sin asumir la venta directa del producto; por este motivo la mejor estrategia de negocio es generar una red de páginas nicho en torno a diferentes productos.</p> <p>Esta inquietud se ha traducido en la generación de una plantilla React para páginas web con diseño y desarrollo estructurado en componentes reutilizables, librería propia de estilos CSS y un sencillo modelo de datos mediante documento JSON que permite reutilizar y/o modificar la web y el contenido de manera rápida. A su vez, se hace especial hincapié en la accesibilidad y performance web para llegar a todo el público y obtener un buen posicionamiento en buscadores. De esta manera se aprovecha todo el tiempo invertido en la creación de la primera web nicho para poder generar nuevas, sin tener que repetir todo el proceso y esfuerzo. Para su diseño, desarrollo, testeo y presentación se ha creado un proyecto de página nicho focalizada en la venta de cafeteras AeroPress.</p>	

**Abstract:**

Identify a market niche provides opportunities on the internet but many times can find themselves fruitless due to a lack of infrastructure, ability to manage business, time, or the project's risks themselves. Technologies that allow a fast and reusable development, simple data model, in conjunction with the possibility of an economical reward given by the affiliate link offer, gets the opportunity of create new sale sites without the need of having stock, managing the buying funnel, joining to the third party sales.

Such projects lack scalability without accept responsibility of direct selling of the product, for this reason the best business strategy is to generate a net of niche websites around different products.

This inquisitiveness has turn into the generation of a React template for web pages with a design and development structured in reusable components developed by React, an own CSS style library and a simple data model by a JSON document allowing reuse and to modify the website and its content quickly.

In turn, special emphasis is placed on accessibility and web performance to reach all consumers and get a good search engine optimization as well. In this way, takes advantage of all the time spent in the first niche web creation to allow the generation of new ones, without repeating all the process and effort. For its design, development, testing and presentation, has been created a niche web's project focused on Aeropress coffee maker selling.

## **Dedicatoria/Cita**

Este proyecto pone el broche final a 4 años de esfuerzo y tiempo sacrificado que le he robado a todos aquellos que me rodean, amigos y familia, por este motivo se los dedico a todos ellos.

# Agradecimientos

Al maravilloso grupo de amigos que he hecho en la UOC que me han ayudado a nivel universitario, tecnológico y moral.

# Notaciones y Convenciones

- Título principal de sección: Arial, 20, negrita, **Título 1**
- Título de subsección: Arial, 16, **Título subsección**
- Título secundario de subsección: Arial, 14, **Título secundario**
- Contenido: Arial, 10, Contenido.
- Código: Consolas, 10,

Ejemplo:

```
export const BtnDropdown = ( { onClick } ) => {
```

# Índice

<b>1. Introducción</b>	<b>10</b>
1.1 Introducción/Prefacio	10
1.2 Descripción	11
1.3 Objetivos generales	13
1.3.1 Objetivos principales	13
1.3.2 Objetivos Secundarios	13
1.4 Metodología y proceso de trabajo	14
1.5 Planificación	14
1.5.1. Diagrama de Gantt y Diagrama de Tareas	15
1.6 Presupuesto	16
1.7 Estructura del documento	17
<b>2. Análisis de mercado</b>	<b>18</b>
2.1. Análisis tecnológico	18
2.1.1 Público objetivo y perfiles de usuario	18
2.1.2 Competencia/Antecedentes	18
2.1.3 Tabla comparativa	19
2.1.3 Posición frente la a competencia	20
2.2. Análisis producto nicho	20
2.2.1 Análisis de nicho	20
2.2.2 Público objetivo y perfiles de usuario de AeroPress®	20
2.2.3 Competencia/Antecedentes	21
2.3 Análisis DAFO	22
<b>3. Propuesta</b>	<b>23</b>
3.1. Definición de objetivos/especificaciones del producto	23
3.1.1 Objetivos desde el punto de vista de negocio	23
3.1.2 Objetivos desde el punto de vista del usuario	23
3.1.3 Objetivos desde el punto de vista del diseñador y desarrollador	23
3.2. Modelo de negocio.	24
3.2.1 Modelo de negocio basado en red de páginas webs nicho	24
3.2.1 Modelo de negocio basado en venta de plantillas React	24
3.3. Estrategía de marketing.	25
3.3.1 Estrategia de marketing para página web de producto nicho	25
<b>4. Diseño</b>	<b>27</b>
4.1. Arquitectura general de la aplicación	27
4.2. Arquitectura de la información	27
4.3 Diseño UX/UI (experiencia de usuario e interfaces)	28
4.3.1 Sistema de diseño	28
4.3.2 Diseño responsive	29
4.3.3 Diseño Accesible	31
4.4. Stack tecnológico y Desarrollo	32



4.4.1 Stack tecnológico	32
4.4.2 Arquitectura	34
4.4.3 Front-End y lógica de la aplicación.	35
4.4.3.1 Anatomía de un componente	36
4.4.3.2 Componentes	40
4.4.4 Helpers	42
4.4.5 Styles CSS	43
4.4.5.1 Anatomía de una hoja de estilos.	43
4.4.5.2 Hojas de estilos	45
4.4.6 Modelo de datos	48
4.4.7 Preparación y optimización de imágenes.	50
4.4.7.1 ImageMagick.	50
4.4.7.2 Uso de Webp	51
<b>5. Implementación</b>	<b>53</b>
5.1. Requisitos de instalación	53
5.1.1 Software necesarios	53
5.1.2 Hardware	53
5.2 Instrucciones de instalación en local	53
5.3 Instrucciones en entorno de producción	54
<b>6. Demostración</b>	<b>55</b>
6.1 Instrucciones de uso	55
6.2. Prototipos Lo-FI	55
6.2. Prototipos Hi-FI	56
6.3 Test	56
6.3.1 PropTypes	56
6.3.2 Test accesibilidad	57
6.3.3 Test usabilidad y negocio	57
6.4 Ejemplos de uso del producto (o guía de usuario).	58
6.4.1 Desde el punto de vista del propietario.	58
6.4.2 Desde el punto de vista del desarrollador.	64
6.4.3 Desde el punto de vista del usuario.	65
<b>7. Conclusiones y líneas de futuro</b>	<b>66</b>
7.1. Conclusiones	66
7.1.1 Aprendizaje	66
7.1.2 Asunción de objetivos	66
7.2. Líneas de futuro	66
<b>Bibliografía</b>	<b>69</b>

# 1. Introducción

## 1.1 Introducción/Prefacio

Internet se ha convertido en un lugar donde muchos buscan un santo grial que les permita crear negocios digitales ya que, en este entorno no tienen todos los condicionantes que un negocio convencional en entorno físico aplica a los negocios: espacio comercial, suministros, varios pagos fijos... El *offline* supone una apuesta muy fuerte para un negocio, por buena que sea la idea o el modelo de negocio, nadie va a garantizar que pueda salir bien. Internet permite probar proyectos con inversiones mínimas, tomarle el pulso, obtener *feedback* y tomar decisiones. Fruto de esta posibilidad nacen metodologías como *Lean Startup*<sup>1</sup> centradas en obtener rápidamente el mínimo producto viable, sacarlo al mercado ver su viabilidad y actuar en base al feedback obtenido, caminando sobre seguro y perdiendo poco cuando se falla. Tim Ferris en su *best seller* “*La semana laboral de 4 horas*”<sup>2</sup> también habla de la importancia de probar la idea bien rápido y con el mínimo presupuesto posible. Su estrategia consiste en crear una web con el producto que se desea vender, invertir 300 euros en, por entonces, Google AdWords, y ver cuanta gente completaba el proceso de venta. Al llegar al final se le mostraba un mensaje diciendo que lamentablemente el producto estaba fuera de stock. Una vez obtenidos los datos se trata de hacer los cálculos; de completarse las ventas, ¿el ingreso hubiera compensado los anuncios y gastos? Si la respuesta es afirmativa estamos ante un posible negocio online. Si la respuesta es negativa, se debe probar con otro producto.

La verdad es que la propuesta de Tim Ferris parece sensata, mejor arriesgarse a perder 300 euros, que invertir en un stock mínimo, tratar de venderlo y ver si funciona el negocio perdiendo algunos miles de dólares si sale mal. Ahora bien, la primera edición de este libro fue en 2007 e internet ha evolucionado mucho desde entonces. Encontramos un nuevo modelo de ingresos que conocemos como enlaces de afiliados. Un enlace de afiliados es un enlace a un producto o servicio de un tercero que de completarse la compra o contratación, en la web del tercero, se obtendrá una comisión. Muchas compañías ofrecen este tipo de servicio pero sin duda la popularización de este modelo de ingresos la trajo Amazon con su extenso catálogo que nos permite enlazar mediante afiliados prácticamente cualquier tipo de producto.

Esto aporta una capa más de testeo de proyecto al modelo que presentaba Tim Ferris, pues en lugar de mostrar una web de “mentira”, llegando a un proceso de venta con final fallido, en este caso si se genera la venta, y con cada visita de usuario y venta se generan datos. Estos datos son la clave para saber cuándo se vende más el producto, bajo qué precios, condiciones..., todo un banco de pruebas que no sólo no cuesta muy poco, sino que además genera ingresos. Llegados a este punto uno puede decidir si lanzarse a la piscina y pasar a comercializar él mismo el producto o no, o simplemente mantener el proyecto en este punto y beneficiarse de los ingresos que genera.

En los párrafos anteriores se ha resuelto el “cómo” pero queda algo igual de importante, sino más: el “qué” vender. Encontrar un nicho y un producto dentro de éste no es fácil, la competencia en internet es abrumadora. Proyectos como los que se han descrito deben ser lo menos generalistas posible o habrá que competir con grandes empresas de internet con mucha presencia y recursos. Para elegir el nicho es interesante prestar atención al termino japonés “otaku” que describe el gurú del marketing Seth Godin en su libro “*La vaca púrpura*”<sup>3</sup> como “*deseo irresistible*” y pone de ejemplo el deseo que hace que alguien coja el coche y conduzca un montón de kilómetros solo por cenar en un restaurante o adquirir ciertos productos. En este sentido hay que buscar un nicho y producto con un valor añadido muy fuerte, algunos ejemplos serían:

- Cinturones de caucho para veganos
- Hotel rural donde se admiten mascotas
- Gorros estampados para cirujanos

Como puede verse, en los 3 ejemplos anteriores hay un valor añadido intangible que hace al usuario salirse del paradigma caro-barato y sólo plantearse si lo puede adquirir o no. ¿Qué precio tiene el respeto animal? ¿Cuál es el precio de poder disfrutar tus vacaciones con tu perro? ¿Qué precio tiene la originalidad y diferenciación con otros médicos? No se puede poner precio a las emociones.

Por último, es importante tener presente que un nicho demasiado específico puede no llegar a tener suficiente movimiento, seguramente habrá en el mundo apasionados de la mítica serie “Cosas de casa” dispuestos a pagar por una réplica de *Steve Urkel* a tamaño real, pero ¿hay suficientes interesados como para poder montar un negocio entorno a ello? Seguramente no.

## 1.2 Descripción

En este documento se presenta el diseño y desarrollo de una plantilla para la librería Javascript React<sup>4</sup> para página web centrada en un producto nicho y en cómo ésta se formula de manera que sea completamente reutilizable para generar otras páginas web centradas en un producto nicho de manera rápida. Para su creación se ha seleccionado un producto: *AeroPress*®, de un nicho concreto: *Café de especialidad*, que se desarrollarán más adelante, que servirán como modelo y ejemplo de proyecto web basado en producto nicho.

### **La web cubre los siguientes aspectos comunicativos y de negocio:**

- Presentación del producto
- Compra de producto mediante enlaces de afiliados de terceros
- Valoraciones de producto de usuarios
- Descripción detallada del producto

### **El diseño cubre los siguientes aspectos:**

- Diseño realizado con la herramienta Figma
- Librería que facilita paletas de colores, espaciados, tipografías, estilos de texto
- Diseño basado en componentes
- Cumplir los estándares de accesibilidad WCAG 2.2<sup>5</sup> relevantes a diseño

### **El desarrollo cubre los siguientes aspectos:**

- Desarrollo en React
- Paradigma de programación funcional basado en componentes
- Estilos con sistema de variables en base a diseño
- Cumplir los estándares de accesibilidad WCAG 2.2 relevantes a desarrollo web
- Contenido completamente separado del código
- Modelo de datos mediante archivo JSON con textos e imágenes asociados a variables
- Especial cuidado en el *performance*
- Uso de formatos de imágenes de última generación: Webp y SVG

La web resultante, realizada con la plantilla Renite, cumple con los estándares de accesibilidad WCAG 2.2 para asegurar el acceso a cualquier usuario independientemente de sus características personales o su dispositivo/navegador para navegar por internet. Las personas con discapacidad representan el 6,12% de la población Española, no contar con ellas cierra la puerta a un gran número

de usuarios que por un lado, tiene derecho a la información y por otro lado, pueden ser consumidores potenciales. Además, hay que tener presente que la accesibilidad afecta a todos, una web accesible permite interactuar mejor con ella en situaciones con visibilidad reducida, difícil interacción táctil, complejidad de los controladores periféricos (reflejos en pantalla, poco contraste, nivel de lúmenes del dispositivo, dispositivo demasiado pequeño...). Se considera pues que la accesibilidad tiene un impacto enorme en una buena experiencia del usuario independientemente de sus condiciones.

A la hora de consumir la información la velocidad es vital para poder ofrecer una buena experiencia de usuario, y más en un medio con consumidores tan exigentes como es internet. Este factor puede eclipsar la propuesta de valor, por buena que sea, si el usuario tiene que esperar más tiempo del que considera necesario para acceder a la información. Otro aspecto, vinculado a la velocidad de carga, es el peso de los diferentes elementos que componen la web. El modelo actual de consumo de datos en dispositivos móviles se basa en precio por volumen de datos, en cierta manera se cobra al usuario por cada kilobyte que descarga para acceder a el contenido. Por respeto al usuario y en pos de una buena experiencia de usuario, todos los recursos y peticiones necesarias para consumir la web están optimizadas por tal de no hacer gastar más datos de los necesarios al usuario. Acciones como usar formatos de imagen Webp de alta compresión, uso de gráficos vectoriales SVG, cargar sólo los recursos necesarios para esa vista, unificar peticiones..., ayudan a optimizar el peso y rendimiento de la página. Por los motivos anteriormente expuestos la web cumple con los criterios de *web performance* que Google *PageSpeed*<sup>6</sup> pone a disposición de los desarrolladores.

El ranking de *PageSpeed* no sólo ayuda a tomar el pulso al *performance* de la web: es una métrica vital para saber cuán atractiva es la web para el algoritmo de posicionamiento de Google. Puesto que una web nicho con enlace de afiliados se nutre completamente de la ratio visitas/conversión, aumentar el número de visitas es de extrema necesidad. Por este motivo, uno de los pilares de la estrategia de negocio es potenciar el posicionamiento natural de la página. Se habla mucho sobre el posicionamiento web, pero lo que se puede tener por seguro es que una buena puntuación en Google *page speed* y cumplir con los criterios de accesibilidad WCAG A.A. influyen de manera decisiva.

Todo el código generado está escrito bajo los principios de *Clean Code* para que cualquier desarrollador pueda comprenderlo, haciéndolo escalable, y está correctamente estructurado y cobijado en repositorio GIT. El proceso de diseño dota de unas guías de estilo y sistema de diseño escalable y reutilizable. Tanto diseño como desarrollo están enfocados en sistemas que permiten la evolución continua del producto y su escalabilidad. Por último, el diseño y desarrollo del *site* se aborda desde una perspectiva 100% *responsive*, ningún dispositivo o resolución se ve discriminada o favorecida, por este motivo el diseño cuenta con una filosofía "líquida", rellenando los componentes el entorno que se le ofrece para reproducirse.

Se identifica una estructura web clara y reutilizable para diferentes nichos. Los modelos de negocio alrededor de páginas nichos suelen comprender diferentes páginas nicho, ya que como se ha visto, al ser una temática/producto muy determinado, su público es limitado y su escalabilidad es relativa si no se da el salto a la venta directa. Cuando se considera que se ha asumido una cuota de usuarios que responde a la capacidad del nicho, se centran esfuerzos en identificar otros nichos y crear páginas para estos replicando el modelo. Por ese motivo, se ha tenido especial cuidado en que el sistema de diseño utilizado y la tecnología/código generado durante el desarrollo sean reutilizables por tal de poder implementarlos en proyectos y/o evolutivos posteriores reduciendo así tiempo y esfuerzo en el futuro.

## 1.3 Objetivos generales

### 1.3.1 Objetivos principales

Objetivos de la aplicación web:

- Web basada en componentes reutilizables
- Contenido separado del código para una rápida modificación y no comprometer el código
- Poder generar diferentes páginas de producto nicho de manera rápida gracias a la plantilla React creada.
- Dar a conocer AeroPress®
- Obtener un *site* que genere ingresos pasivos mediante enlace de afiliados

Objetivos para el cliente/usuario:

- Encontrar información clara y unificada sobre AeroPress®
- Poder comprar AeroPress®

Objetivos personales del autor del TFG:

- Formarme en el uso React a nivel avanzado
- Desarrollar una web que cumpla los criterios de accesibilidad WCAG 2.2
- Desarrollar una web con buena puntuación en los test de rendimiento de *PageSpeedInsight*

### 1.3.2 Objetivos Secundarios

Objetivos adicionales que enriquecen el TF.

- Crear un sistema de diseño reciclable para otros proyectos
- Crear una librería de componentes React reciclable para futuros proyectos

## 1.4 Metodología y proceso de trabajo

El proyecto gira en torno a la creación de una plantilla para la librería de Javascript React con la que generar una página de producto nicho 100% reutilizable escogiendo AeroPress® como primer producto a divulgar información y dar acceso a la compra aprovechando su escasa representación en el país y ninguna de manera oficial a través de la marca. Por lo tanto, la información de referencia se obtendrá de [aeropress.com](https://aeropress.com). Dicha información será asimilada y procesada para enfocarla y divulgarla a las características propias del *target* español.

A pesar de que la incertidumbre es baja y la solución conocida, el desarrollo del proyecto se planteará como si de un *start up* se tratase, con la necesidad de sacar un producto al mercado en el menor tiempo posible para después ir iterando sobre él y tomarle el pulso constantemente a su producto/servicio y su relación con el público así como las necesidades de éste y del mercado en cada interacción. Se elige por consiguiente trabajar con metodología Agile con *framework* SCRUM asumiendo sprints de diseño y desarrollo de 2 semanas centrados en completar las diferentes historias de usuario.

La tecnología utilizada para diseño será principalmente Figma aunque se utilizarán illustrator, Photoshop e ImageMagick cuando sean necesarios para generar y/o optimizar recursos gráficos. A nivel de desarrollo el proyecto se centrará en HTML5, SCSS y JavaScript ES6 desarrollado con la librería REACT, se trabajarán algunas funcionalidades con NodeJS para la gestión de paquetes de npm.

Contaremos con herramientas de analítica web de Google para obtener información del comportamiento de los usuarios

Se utilizará Git como gestor de versiones utilizando github.com como repositorio del proyecto.

Por último, siguiendo la línea de agilidad y velocidad de desarrollo se utilizará Vercel para albergar la web ya que permite vincular un repositorio de GitHub con el servidor, cada vez que la rama master del repositorio sea actualizada la web también será actualizada.

## 1.5 Planificación

El primer mes debe cubrir 2 sprints, uno centrado en arquitectura de la información y diseño y otro centrado en desarrollo. Tras estos 2 sprints iniciales una primera versión de la web debe estar publicada. Tras los 2 sprints iniciales, cada sprint abarcará diseño y desarrollo del MVP seleccionado para tal sprint hasta llegar al 16/05, momento en el que se dará por finalizado el desarrollo y se procederá a cerrar el proyecto y preparar la información y presentación.

Fechas clave:

- 07/03/2022 Inicio del proyecto
  - 07/03/2022 Inicio Sprint Arquitectura de la información y Diseño
  - 21/03/2022 Inicio Sprint Desarrollo en base a diseño.
- 03/04/2022 Publicación de Producto mínimo viable
- 04/04/2022 Sprint 3
- 18/04/2022 Sprint 4
- 16/05/2022 Publicación final del producto

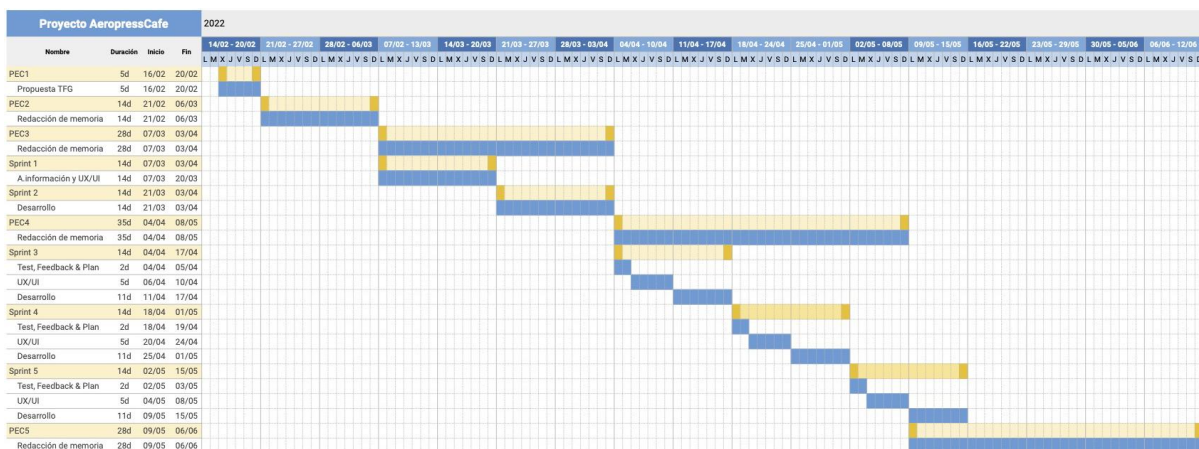
	Dedicación		
	Duración	Inicio	Final
PEC1	5 días	16/02/2022	20/02/2022
PEC2	14 días	21/02/2022	06/03/2022
PEC3	28 días	07/03/2022	03/04/2022
PEC4	35 días	04/04/2022	08/05/2022
PEC5	28 días	09/05/2022	06/06/2022

Entrega PECS

	Dedicación		
	Duración	Inicio	Final
Sprint 1 <i>A. Información y Diseño</i>	14 días	07/03/2022	20/03/2022
Sprint 2 <i>Desarrollo en base a diseño</i>	14 días	21/03/2022	03/04/2022
Sprint 3	14 días	04/04/2022	17/04/2022
Sprint 4	14 días	18/04/2022	01/05/2022
Sprint 5	14 días	02/05/2022	15/05/2022

Planificación de ciclos Agile

### 1.5.1. Diagrama de Gantt y Diagrama de Tareas



[ver Diagrama Gantt en detalle](#)

## 1.6 Presupuesto

Aunque diseño y desarrollo recaen en la misma persona, quedan identificados **2 roles, el de diseño UX/UI y el del desarrollador Front-End**. El precio reflejado es el precio promedio en España, para estipular el precio se ha tenido en cuenta el precio promedio que ofrece Malt.es (plataforma de contratación de Freelance) y la propia experiencia en el sector.

### Presupuesto recursos humanos

	Dedicación (horas)	Precio/hora	Subtotal
<b>Análisis</b>			
Recopilación de requisitos	12	36€	432€
<b>Diseño UX/UI</b>			
Estudio, conceptualización y wireframe	20	36€	720€
Arquitectura de la información	5		180€
Diseño de interfaces	20		720€
Preparar recursos gráficos para desarrollo.	2		72€
<b>Desarrollo</b>			
Set up entorno, dominio y servidor	1	48€	48€
Desarrollo de componentes en react y lógica de Front	40		1.920€
Modelo de datos	2		96€
Test	8		384€
Despliegue	1		48€
<b>Precio Total</b>			<b>4.620€</b>



## Presupuesto recursos técnicos

Servicio	Tipo contratación	Proveedor	Precio	Total
Dominio web	Anual	DonDominio	15,67€	15,67€
Servidor web	Indefinido	Vercel (plan Hobbie)	Gratis	0,00€
Ilustraciones	Pago único	Laura España	20,00€	100,00€
<b>Total</b>				<b>115,67€</b>

### 1.7 Estructura del documento

A continuación se exponen de manera resumida los puntos que componen el resto del documento.

**Análisis:** Estudio del mercado actual en relación a las opciones que se ofrecen para la creación de páginas web de producto nicho, viendo sus principales ventajas e inconvenientes. Definición del público objetivo. Análisis DAFO con tal de conocer las debilidades y fortalezas de la aplicación web.

A su vez se realizará también un análisis referente a la temática propia de la web: Aeropress, realizando un estudio del mercado actual en relación a la venta, y sitios enfocados, de AeroPress® y tendencias del mundo del café. Definición del público objetivo

**Propuesta:** A partir del análisis anterior, se definen los objetivos de la aplicación.

**Diseño:** Presentación del diseño de la aplicación web, mostrando su arquitectura de la información, el diseño UX/UI y sistema de diseño utilizado por tal de agilizar los procesos de diseño, teniendo presente su capacidad *responsive* y accesibilidad. Seguido de la presentación del stack tecnológico seleccionado, la arquitectura de la aplicación y el planteamiento para las diferentes capas de desarrollo: front, lógica y modelo de datos.

**Implementación:** Se detallan los requisitos de instalación y las instrucciones de instalación de la aplicación tanto en local como en entorno de producción.

**Demostración:** Se explican las instrucciones de uso de la aplicación, desde el punto de vista de los diferentes roles implicados, y el trabajo de prototipado.

**Conclusiones:** Se presentan las conclusiones en base al trabajo realizado y posibles líneas de mejora y escalabilidad del producto así como su modelo de negocio.

## 2. Análisis de mercado

### 2.1. Análisis tecnológico

#### 2.1.1 Público objetivo y perfiles de usuario

El público potencial de la plantilla React como herramienta de desarrollo se divide en 2 perfiles:

Por un lado, emprendedores que desean crear una página web en base a un producto nicho, con un mínimo nivel técnico, suficiente para entender cómo funciona una página web, identificar los diferentes lenguajes que implica, pero no tienen la habilidad suficiente como para poder desarrollar un sitio web entero de inicio a fin o cuanto menos en un tiempo o con una calidad aceptable. De esta manera pueden utilizar la plantilla ayudados con la pequeña guía de publicación cambiando únicamente el contenido así como la paleta de colores y tipografía si lo desean.

Por otro lado, desarrolladores con conocimientos profundos de tecnología web y React que utilizarán la plantilla como base para su desarrollo, añadiendo y reorganizando contenido y secciones en base a los componentes que ofrece o incluso desarrollando nuevas funcionalidades gracias a que el proyecto está estructurado y desarrollado de manera que pueda entenderse y reciclarse con facilidad. Este segundo perfil puede adquirir la plantilla por inquietudes laborales ya que el proyecto en el que trabajen responde bien a la plantilla y les puede acelerar el desarrollo, o también con inquietudes de negocio.

#### 2.1.2 Competencia/Antecedentes

Para entender los antecedentes y competencia actual es necesario entender cómo ha cambiado el paradigma web en los últimos años con la popularización de JavaScript como lenguaje de desarrollo general más allá del uso que ostentaba como una tecnología que permite dotar de cierta funcionalidad a una interfaz web. No hay un momento exacto para este cambio pero sí algunos hitos clave:

- Lanzamiento de JavaScript EcmaScript 6<sup>7</sup>, 2015: Una de las mayores revisiones a las que se ha sometido el lenguaje en su historia dotándolo de robustez como lenguaje y añadiendo muchas funcionalidades.
- Creación de la Node.js Foundation<sup>8</sup>, 2015: Esta fundación trabaja activamente por la popularización de JavaScript y Node como herramientas para crear soluciones tecnológicas y aglutina la información y comunidad entorno al lenguaje ayudando a su popularización y aprendizaje.
- Popularización de React<sup>9</sup>, 2015: Grandes compañías como Netflix y Airbnb empiezan a desarrollar con React a raíz de la última versión más estable y robusta que las anteriores. Este hecho populariza mucho la librería.
- Lanzamiento de Angular 4<sup>10</sup>, Diciembre 2016: Esta versión de Angular que extiende y compatibiliza con Angular 2, se populariza rápidamente debido a su robustez y la visión de Angular como framework/librería con continuidad versión tras versión.
- JSON recibe estándar ISO<sup>11</sup>, Noviembre 2017 ISO/IEC 21778:2017

Estos hechos hacen que el desarrollo en JavaScript se popularice cada vez más y empiece a rivalizar con lenguajes muy asentados en el desarrollo web como Java, ASP.NET, PHP, desafiando el patrón modelo vista controlador, modelos de datos relacionales y otros paradigmas presentes en internet.

Conociendo el escenario se puede analizar la competencia directa.

### **WordPress**

En realidad este apartado se dedicaría a los CMS para webs pequeñas y medianas, pero la imposición de WordPress sobre otros CMS como Joomla o Drupal es tan grande que se identifica a WordPress como el rival a batir si se refiere a CMS. WordPress aporta un desarrollo y publicación relativamente rápido. Basta con instalar WordPress en el servidor (es imprescindible que el servidor soporte PHP y SQL) en el que previamente se ha creado una base de datos SQL, asociar la base de datos al WordPress, seleccionar una plantilla y empezar a añadir contenido y estructura con una metodología de arrastrar y soltar elementos.

#### **Fortalezas:**

- Sencillez de creación de contenido
- Diversidad de plantillas y plugins gratuitos
- Gran comunidad

#### **Debilidades:**

- Necesidad de Bases de datos
- Si bien es sencillo de usar se necesita una formación continua
- Las plantillas basadas en arrastrar y soltar generan mucho código lo que perjudica la ratio contenido/código siendo muy penalizado por Google
- Seguridad, al ser un CMS popular está sujeto a muchos ciberataques por lo que tener un sistema de seguridad y respaldo es imprescindible

### **Plantilla HTML+CSS+JS**

Otra fuente de competencia son las webs de venta de plantillas basadas en HTML+CSS+JS como son <https://www.templatemonster.com> o <https://webflow.com>. El presente documento se centra en plantillas que, se supone, tienen unos criterios de accesibilidad y performance similares a la propuesta que este proyecto recoge. Se hace notar este aspecto ya que, uno de los principales problemas es encontrar una plantilla con suficiente calidad. Este tipo de páginas de compra de plantillas se basan más en cantidad que en calidad por lo que hasta dar con la plantilla adecuada se pueden haber visto varias decenas y comprado algunas en balde. Suponiendo que se encuentre una plantilla que cumpla estas expectativas se han de considerar:

#### **Fortalezas:**

- Código ligero
- Sencillez de archivos
- Rapidez de desarrollo si se ciñe al diseño propuesto

#### **Debilidades:**

- El contenido y el código no están separados
- Difícil escalabilidad

### **Plantilla React genérica**

Siguiendo en la línea de plantillas se encuentran de la misma manera páginas dedicadas a la venta de plantillas React como <https://themeforest.net/tags/react>. En este caso las Fortalezas y Debilidades se mantienen. Mayoritariamente son plantillas pensadas en ofrecer una web con un diseño concreto, a la que dedicar un tiempo y mimo para crear una página web pequeña o mediana. No se está teniendo presente el separar el código del contenido y la componentización de elementos para poder generar varias páginas de manera rápida cambiando un par de archivos que albergan la información.

## **2.1.3 Tabla comparativa**

	Renite	WordPress	Plantilla HTML+CSS+JS	Plantilla React genérica
Necesidad de bases de datos	No	Si	No	No
Contenido separado del código	Si	Sujeto al tipo de plantilla y no usar editores visuales	No	No
Escalabilidad	Si	Si	No	No
Tiempo de desarrollo	Extremadamente Bajo	Alto	Bajo	Medio

### 2.1.3 Posición frente la a competencia

Como se ha podido ver tras el análisis de la competencia, Renite ofrece un tiempo de desarrollo extremadamente rápido, factor diferenciador principal frente a su competencia ya que su motivación principal es que las empresas o emprendedores puedan probar sus productos en el mercado cuanto antes y en base a eso tomar decisiones.

Por otro lado, asumiendo que el proyecto va a evolucionar y se va a ir iterando sobre la web respondiendo al feedback de los usuarios, se ha estructurado de tal manera que la edición y/o desarrollo sobre la plantilla sea sencilla y una vez más, rápida. Por estos motivos, otro punto diferenciador es su escalabilidad y sencillez de edición.

## 2.2. Análisis producto nicho

### 2.2.1 Análisis de nicho

El producto nicho elegido para la página es AeroPress®, una cafetera muy popularizada en el nicho del café de especialidad. Se entiende por café barista aquel que se ha elaborado buscando el mejor resultado posible en aroma, sabor, cuerpo y personalidad del café. También puede referirse a esta modalidad como café barista, si bien barista es el título que un experto profesional en preparar y servir café adquiere, los consumidores de café de especialidad y practicantes de diferentes técnicas que permiten maximizar la calidad y la experiencia, son en cierta manera aspirantes a baristas o más coloquialmente: baristas de estar por casa.

AeroPress® es una nueva variante de las conocidas cafeteras de émbolo. La principal ventaja frente a otros tipos de cafetera para la elaboración de café de especialidad es su velocidad, de hecho fue la motivación principal del creador.

### 2.2.2 Público objetivo y perfiles de usuario de AeroPress®

Nuevamente se encuentran 2 perfiles de para el producto. Por un lado, el usuario gourmet que busca una experiencia superior en el consumo de café y una cafetera con la que poder manejar todas las variables (selección del tipo de café, temperatura y cantidad del agua, tiempo de infusión) en la elaboración del café. Y por el otro, se ha identificado un segundo tipo de usuario que utiliza AeroPress® como alternativa ecológica al café de cápsulas tan popularizado hoy en día, ya que le permite elaborar café sin generar más desperdicio que el propio café y en unos años, la propia cafetera aeropress cuando se deteriore que a su vez es reciclable.

### 2.2.3 Competencia/Antecedentes

Por su función de ser, cualquier cafetera que permita elaborar café de especialidad es competencia directa, como producto de AeroPress®. Encontramos los siguientes tipos:

- cafetera de émbolo o francesa.



*fuelle: [www.alambique.com](http://www.alambique.com)*

- cafetera de vacío



*fuelle: [cafesminana.es](http://cafesminana.es)*

- cafetera de filtro



*fuelle: [cocinaconestilo.com](http://cocinaconestilo.com)*

Dentro de las cafeteras de émbolo se encuentra AeroPress® como una nueva variante más moderna de este tipo. Como se ha mencionado en el punto anterior la principal ventaja de AeroPress® frente a los diferentes tipos de cafetera mencionados es su velocidad.

En lo relativo a sociedad de la información, todas las webs de divulgación, blogs y foros sobre café de especialidad y/o Aeropress® son competencia directa a la hora de luchar por el posicionamiento en los motores de búsqueda. Se puede considerar competencia directa del proyecto en sí toda página que permita al usuario comprar AeroPress por internet.

## 2.3 Análisis DAFO

### Debilidades

- El desarrollador aprovecha el proyecto para aprender Figma y React lo que impacta enormemente en el tiempo de desarrollo.
- La página web resultante no cuenta con antigüedad en internet lo que la penaliza en las búsquedas frente a otras webs que llevan más tiempo.
- Hasta obtener un posicionamiento natural la web necesita de campañas de Google Ads para atraer tráfico lo que hace que el proyecto dependa en gran medida de los algoritmos de Google Ads.

### Amenazas

- Proyecto humilde ante posibles campañas comerciales de grandes compañías.
- Baja demanda. Posibilidad de que el interés de los usuarios por el producto que muestran los datos no tenga correlación con la compra.
- Que el beneficio que genere el proyecto no supere los costes.

### Fortalezas

- Modelo de negocio flexible, iterable y escalable.
- Tecnología web flexible, escalable y adaptable a los cambios.
- Web accesible y optimizada para buscadores.

### Oportunidades

- Plantilla React replicable para otros productos nichos en cuestión de horas.
- Posibilidad de vender la plantilla en webs de venta de plantillas.
- No se han encontrado páginas web focalizadas especialmente en AeroPress a nivel Nacional.

## 3. Propuesta

La plantilla react pretende ser una herramienta de generación de páginas web, en torno a un producto nicho, rápida, permitiendo generar páginas nicho en pocas horas, sin conocimientos avanzados de programación y desarrollo, con costes mínimos. Hay que valorar que todo el proceso de diseño UX/UI es aprovechado para crear un sistema de diseño que permite ser reciclado para futuros proyectos y establece las bases para el desarrollo de estilos CSS.

A su vez, se publica una página de producto nicho tras identificar AeroPress® como producto que cumple con las características necesarias para poder considerarse producto nicho.

De esta manera obtenemos:

- Plantilla web que puede modificarse y publicarse más rápidamente que las soluciones de la competencia anteriormente comentada, con un modelo de datos separado del contenido bajo documento JSON que permite separar el contenido del código, pero sin la necesidad de crear y gestionar una base de datos.
- Sistema de diseño para futuros diseños y facilitar el desarrollos de interfaces.
- Página web nicho en pleno funcionamiento con enlace de afiliados a Amazon.

### 3.1. Definición de objetivos/especificaciones del producto

A continuación se describen los objetivos principales del proyecto en función del actor.

#### 3.1.1 Objetivos desde el punto de vista de negocio

1. Atraer usuarios interesados en AeroPress y obtener una comisión de la compra del usuario en Amazon.
2. Contar con un sistema de diseño accesible, componentizado y reutilizable que acelere los tiempos de diseño futuros, ya sea para evolutivos del site o nuevos proyectos de web nicho en torno a producto.
3. Contar con una plantilla React en base a componentes y librería propia que permita acelerar los tiempos de desarrollo futuros, ya sea para evolutivos del site o nuevos proyectos de web nicho en torno a producto.

#### 3.1.2 Objetivos desde el punto de vista del usuario

##### **Desarrollador**

1. Contar con un plantilla que permita un desarrollo rápido
2. Contar con un sistema de componentes y librerías de estilos que soporte evolutivos

##### **Consumidor**

1. Obtener información sobre AeroPress y su funcionamiento
2. Poder comprar AeroPress

#### 3.1.3 Objetivos desde el punto de vista del diseñador y desarrollador

- 1) Aprender a crear sistemas de diseño y componentes con Figma
- 2) Aprender a desarrollar página web con React

- 3) Aprender a integrar y usar Google Analytics
- 4) Familiarizarse con sistemas de desarrollo Agile

## 3.2. Modelo de negocio.

### 3.2.1 Modelo de negocio basado en red de páginas webs nicho

Como se ha visto, el modelo de negocio de las páginas web nicho propuesta gira en torno al producto y a la comisión percibida beneficiándose de la compra en terceros: Enlaces de afiliados. En el presente caso enlaces de afiliados de Amazon, pero muchas compañías cuentan con planes de afiliados y en función de la tipología de producto interesará decantarse por uno u otro.

Por la característica propia de un nicho, recordando que un nicho es un mercado concreto y específico dirigido a un grupo de personas específico así pues, el crecimiento no siempre será exponencial y su escalabilidad tiene ciertos límites. A medida que se optimice la captación mediante las diferentes estrategias de marketing, las visitas y clicks en los enlaces de afiliados aumentarán, pero después de un tiempo se asumirá el punto álgido del nicho y se entrará en un valle.

Llegados a este valle se debe considerar si habiendo obtenido la saturación máxima del nicho la ratio de conversión y el beneficio justifica la viabilidad del proyecto. Si es así, se deberá abarcar el proyecto desde un punto de vista de mantenimiento. Si no lo es, se debe probar otro nicho.

Que no se pueda escalar una página web nicho llegado a un cierto punto de tráfico no significa que no se pueda escalar el negocio. La escalabilidad viene por la réplica del modelo en una red de diferentes páginas web nicho. Habiendo lanzado y consolidado aeropresscafe.com puede plantearse publicar nuevas páginas identificando nuevos nichos. Dada la estructura tecnológica y la sencillez de edición una sola persona puede administrar 10 páginas de estas características. La escalabilidad viene de optimizar cada una de las 10 páginas, descartar las que no tienen el rendimiento esperado, crear de nuevas... en un bucle constante manteniendo siempre 10 páginas nicho que generen entre todas suficientes ingresos.

En todas estas interacciones y pruebas de nichos puede darse el caso de que la respuesta por parte de los usuarios sea extremadamente positiva. En este caso puede ser interesante plantearse vender directamente el producto encontrando un buen proveedor. Este último evolutivo de negocio podría considerarse un proyecto nuevo en sí mismo que generaría otro modelo de negocio, pero es importante tenerlo presente porque esta red viva de páginas en torno a productos nicho es un buen termómetro de mercados y se pueden identificar múltiples oportunidades de negocio.

### 3.2.1 Modelo de negocio basado en venta de plantillas React

Por otro lado, se encuentra una plantilla react pensada para su fácil y rápida edición, con diferentes componentes definidos cubriendo un sector muy específico de tipología web. Esta plantilla puede ser muy útil para emprendedores con un mínimo conocimiento tecnológico o desarrolladores que buscan una buena base para sus proyectos como se expone en el punto 2.1.1 *Público objetivo y perfiles de usuario*. Por este motivo, sería viable publicar la plantilla en diferentes webs de venta de plantillas como es themeforest.com que permite publicar la plantilla, decidir el precio deseado y aporta varias herramientas de análisis de tráfico que ayudan a posicionarla en el *marketplace*. El servicio es completamente gratuito, pero la comisión es del 50%. Estos números son suficientemente atractivos pues publicando la plantilla a 30€, precio dentro de la media, se obtendrían 1500€ (antes de impuestos) por cada 100 descargas. Además, se puede generar cierto *engagement* con los compradores publicando cada determinado tiempo nuevos componentes y funcionalidades que se adquieran por un determinado precio.



Dado que se trata de un proyecto universitario que genera conocimiento y defendiendo la universidad como fuente de información y conocimiento la plantilla será pública y gratuita mediante descarga en Github.com, al menos por el presente y futuro inmediato.

## 3.3.Estrategía de marketing.

### 3.3.1 Estrategia de marketing para página web de producto nicho

La estrategia de marketing de aeropresscafe.com, como la de otras posibles páginas nicho girará en torno a dos pilares.

#### **SEM (Search Engine Marketing)**

El marketing de buscadores (Search Engine Marketing) se basa en campañas de anuncios de pago en motores de búsqueda, aunque en la actualidad también engloba a los anuncios en redes sociales. Principalmente se publicarán anuncios en Google Ads para atraer usuarios interesados en el café de especialidad y/o AeroPress al site. La estrategia de construcción de anuncios se centrará por un lado, en identificar aquellas franjas horarias, días de la semana y zonas geográficas que responden mejor y centrar ahí los esfuerzos económicos.

Por otro lado, se centrará en la iteración constante de mejora de anuncio, estas iteraciones se harán alrededor de 3 anuncios diferentes entre sí, una vez identificado el anuncio que responde mejor se eliminarán los otros dos y se crearán dos variantes del anuncio que ha dado mejores resultados manteniendo éste. En la siguiente iteración se volverá a identificar el anuncio con mejores resultados descartando los otros dos, creando 2 nuevos anuncios en base al óptimo para acompañarlo. De esta manera se mejora iteración a iteración la ratio de conversión del anuncio.

Es importante señalar que para tomar estas decisiones anteriormente mencionadas se debe tener suficiente volumen de datos, por encima de 300 *clicks*. Una muestra de 25 o 100 *clicks* puede no ser significativa o verse afectada por espontaneidades (festividades, rebajas...).

#### **SEO (Search Engine Optimization)**

La optimización en buscadores (Search Engine Optimization) se basa en la obtención de usuarios a la página a través de los resultados en buscadores web. Si bien existen diferentes buscadores web, el principal aglutinador de las búsquedas en internet es Google, por este motivo todos los esfuerzos se centrarán en contentar a los robots de este gigante de internet. Por un lado tenemos el SEO técnico trabajando adecuadamente el correcto etiquetado HTML, el uso de imágenes optimizadas, uso de formato Webp y otros aspectos técnicos. Por el otro, habrá que trabajar bien la redacción de textos seleccionando adecuadamente las palabras clave utilizadas. Esta estrategia, habiendo trabajado bien los aspectos hasta ahora mencionados, también trabaja por volumen, por lo que sería interesante complementar este proyecto con un blog en el mismo dominio que hable sobre temas relacionados con AeroPress y el café de especialidad, que apunte constantemente a la web o apuntando directamente a Amazon mediante los enlaces de afiliados.

Las dos estrategias presentadas son complementarias y enriquecen la una a la otra, ambas aportan datos sobre el comportamiento de los usuarios que ayudan a optimizar las campañas de marketing. Y también ambas generan tráfico a la web, mejorando el posicionamiento de la web y con ello el rendimiento de las campañas. Dado que el SEO es una estrategia a medio/largo plazo, inicialmente los esfuerzos se centrarán en las campañas de SEM para conseguir esas primeras visitas, pero posteriormente el SEO debería ir supliendo o complementando las estrategias de SEM por tal de maximizar el beneficio.

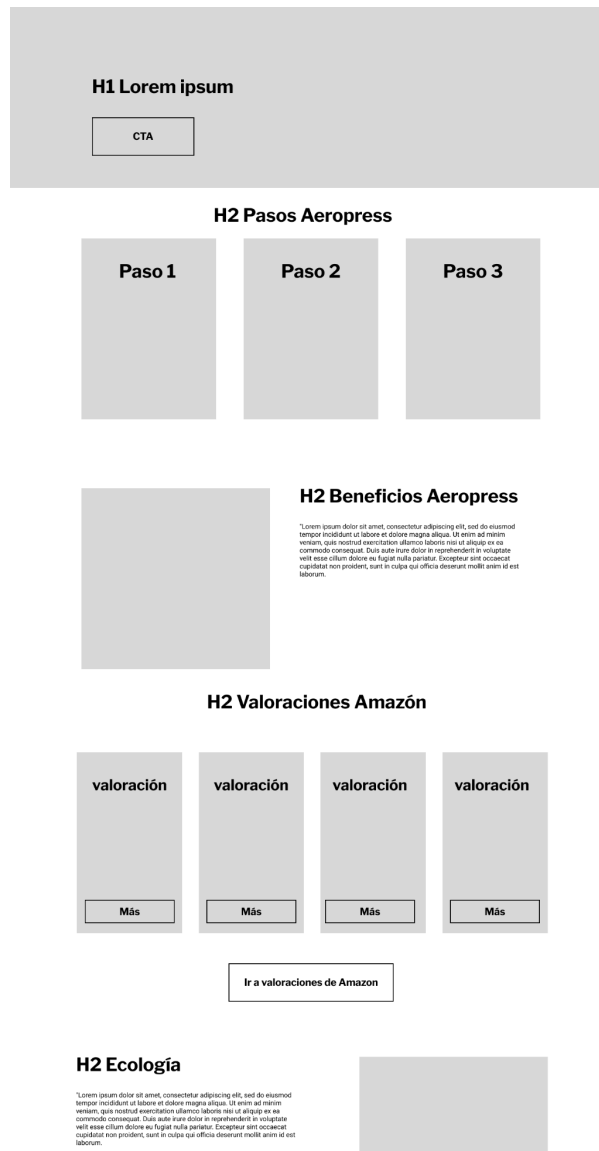
# 4. Diseño

## 4.1. Arquitectura general de la aplicación

La aplicación, una vez hecho el build de producción, se compone únicamente de una capa de front-end. Previo a este paso, encontramos un desarrollo Front-End desarrollado en HTML para la estructura, SCSS para los estilos y JavaScript para aplicar la lógica. Los datos son servidos desde un documento JSON donde se referencian mediante *keys*. Para facilitar el desarrollo en base a componentes se ha utilizado React como librería de *front-end*.

## 4.2. Arquitectura de la información

*Wireframe* que recoge las bases para poder trabajar sobre el proyecto de manera gráfica, conteniendo todo lo necesario para poder pasar al diseño gráfico de la interfaz.



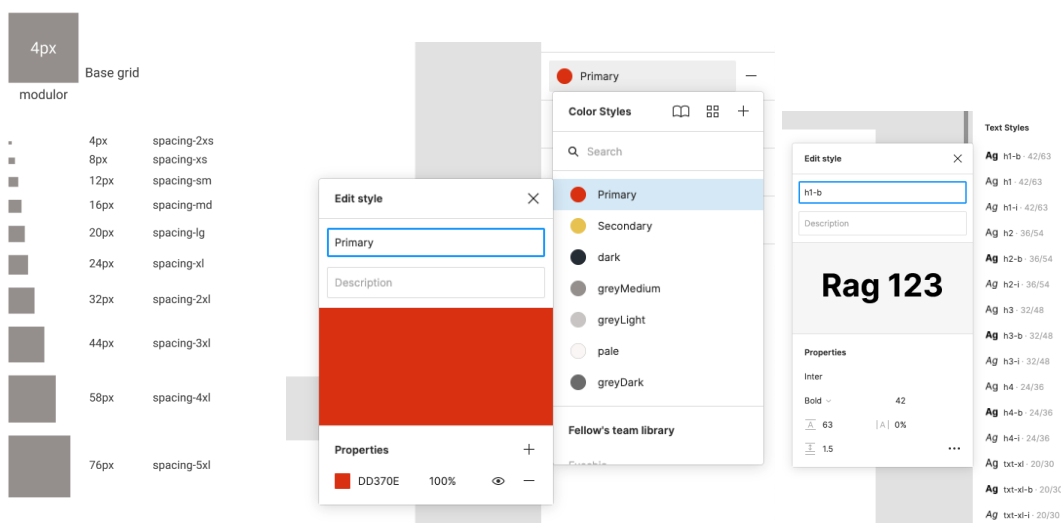
## 4.3 Diseño UX/UI (experiencia de usuario e interfaces)

### 4.3.1 Sistema de diseño

Para el diseño de la página web se utiliza Figma como principal herramienta ya que esta permite diseñar y crear elementos reutilizables así como la creación de una librería de estilos propia.

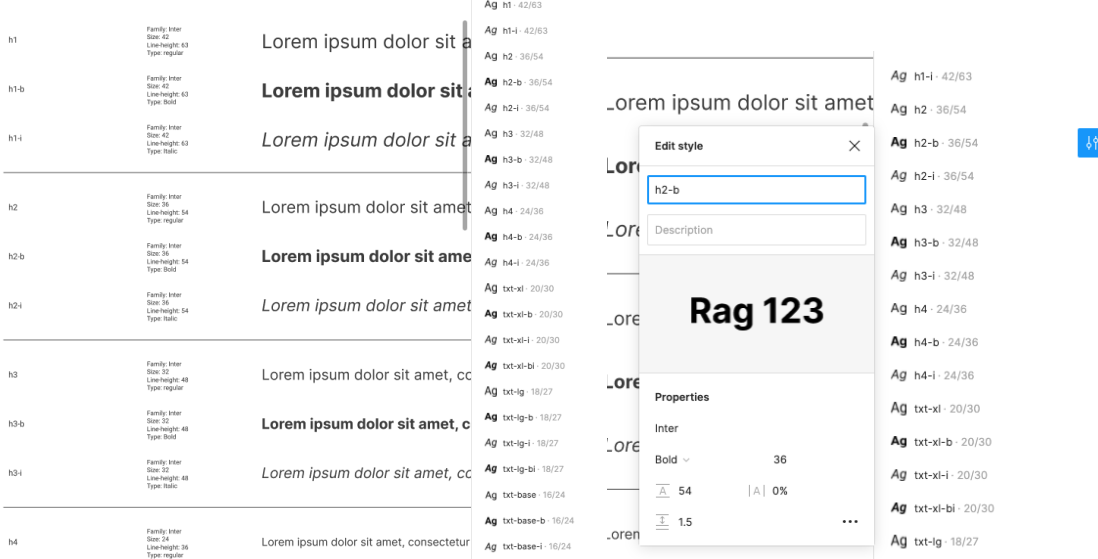
Los elementos referentes a la librería de estilos permiten crear un conjunto de estilos que ayudarán a homogeneizar la web y establecer las bases sobre las que construir el diseño. La creación de los diferentes componentes ayuda también a poder propagar cambios en todo el diseño modificando únicamente el componente padre.

#### Spacing



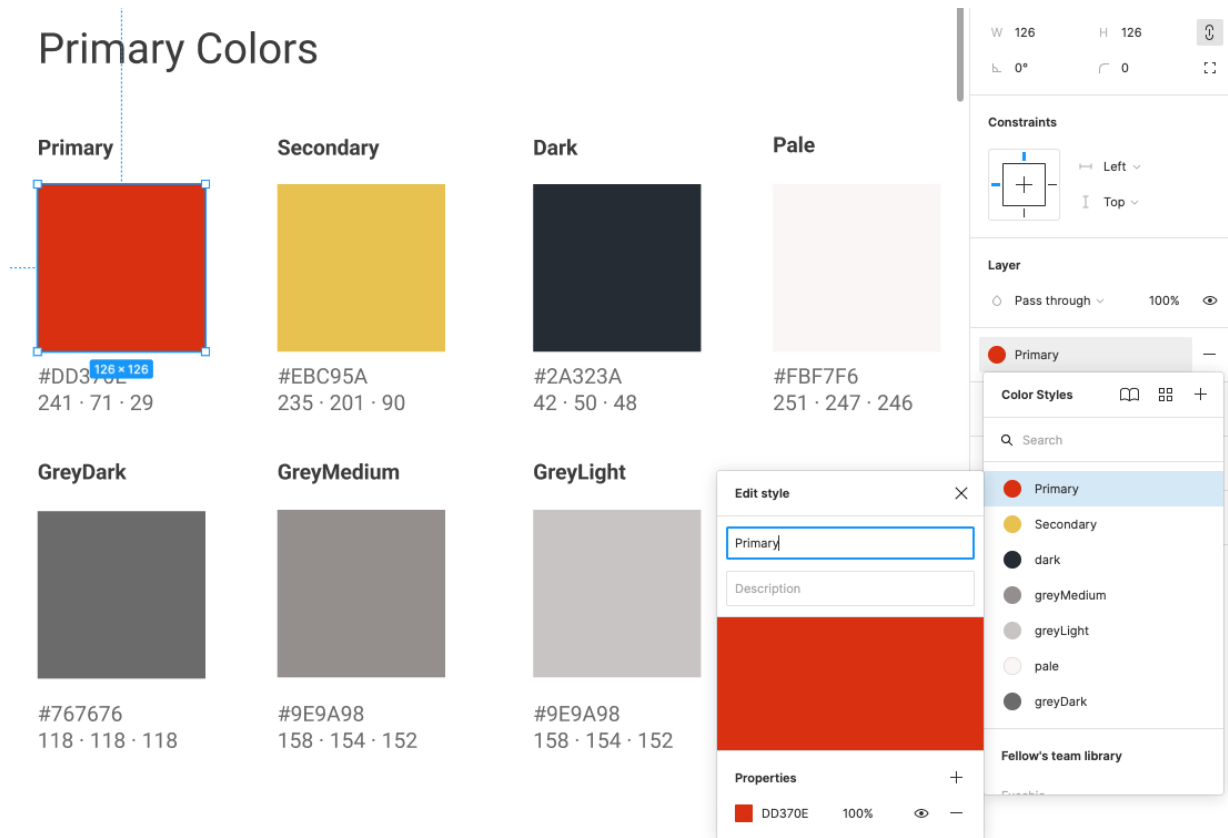
*Creación de opciones de espaciado, librería de colores, librería de estilos de texto*

#### Typography

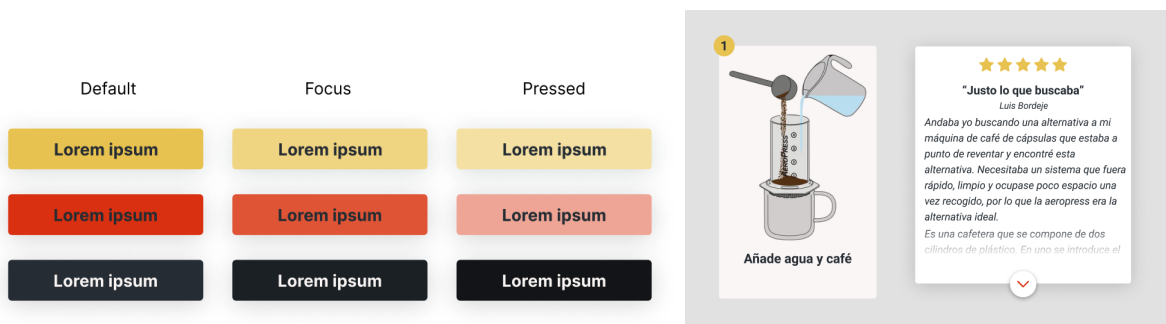


*Captura de pantalla de Figma donde podemos ver los diferentes estilos de texto creados para poder aplicar a un texto con tan sólo seleccionarlo.*

## Primary Colors



*Paleta de color del diseño y su correspondientes entradas en la librería*



*Diferentes componentes reutilizables*

Tanto la librería de estilos como los componentes se alinean con el objetivo de tener un sistema de diseño con el que diseñar evolutivos o nuevos productos, modificarlos y generar diferentes versiones de manera rápida.

### 4.3.2 Diseño responsive

Uno de los requisitos estipulados es tener una web que pueda visualizarse en cualquier dispositivo, la filosofía es trabajar con elementos flotantes que se van reordenando en función del espacio. No se pueden trabajar todas las resoluciones a las que una web puede ser visualizada, pero bajo esta filosofía es suficiente con diseñar el escenario de mayor tamaño y el de menor y si algún elemento cuenta con una casuística especial trabajar su propiedad responsive de manera aislada.

*Diferentes vistas de la aplicación en función del dispositivo*



Disfruta de café barista en casa respetando el medio ambiente

[Comprar en Amazon](#)

### Cómo preparar café con AeroPress



### Prepárate el café que más te guste

AeroPress te abre un mundo de posibilidades, al no depender de cápsulas compatibles, puedes preparar todo tipo de café y lo más importante: **prepara café a medida como más te guste.**

Elige el tipo que más te guste: arábica, ibérica, robusta... Decide el origen: Colombia, Brasil, Etiopía, si es ecológico, de comercio justo... muelo tú mismo o compralo molido, añade la cantidad de agua que sea de tu agrado.

AeroPress es el aliado perfecto para poder disfrutar de **café de máxima calidad, sin condiciones y 100% personalizado.**

### Los usuarios opinan de AeroPress

<p>★★★★★ "Justo lo que buscaba" Luis Borja</p> <p>Andaba yo buscando una alternativa a mi máquina de café de cápsulas que estaba a punto de reventar y encontré esta alternativa. Necesitaba un sistema que fuera rápido, limpio y ocupase poco espacio una vez recogido por lo que la AeroPress era la alternativa ideal. Es una cafetera que se compone de dos... <small>...de aluminio. En caso de artículos de...</small></p>	<p>★★★★★ "Super Contenta" Marta</p> <p>Me encanta esta cafetera, si eres amante del café de filtro, hazte con una, es fácil de usar y transportar. Si tienes poca paciencia esta no es tu cafetera, esta cafetera es para disfrutar del ritual completo del café. Tiene una capacidad para dos cafés, el café... <small>...se recomienda en un café en filtro, etc.</small></p>	<p>★★★★★ "Café excelente, ligero y práctico" Isaí Pérez García</p> <p>Perfecto para usar en casa o cuando te vas de camping. Aconsejable tener una herramienta de agua a mano para cada vez que quieras un café. Yo uso 2 o 3 filtros para aplicar más presión y que saque aún más interés el café. Y luego limpio los filtros y los reuso durante 1 mes o más. Los filtros que trae te duran para...</p>	<p>★★★★★ "Genial cafetera" Paloma</p> <p>Está muy bien. Yo usaba hasta ahora una de esas cafeteras italianas de toda la vida, y desde que recibí la AeroPress se ha convertido en mi cafetera del día a día. Ahora me queda trastear un poco con los "justos" (sigamos, el tiempo de preparación, la cantidad de agua/café... pero si sigues las instrucciones del manual, sin modificaciones, se vale un café. Si cualquier...</p>
---	---	---	---

[Ver valoraciones de Amazon](#)

### Ayuda a mantener limpio el planeta

Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo y hasta 29 000 acaban en vertederos\*. Aunque se empiezan a hacer esfuerzos en el reciclado y segunda vida de las cápsulas la gran mayoría acaban alimentando montañas en vertederos, cápsulas desechadas prevalecerán durante décadas e incluso cientos de años algunas, hasta desaparecer.

AeroPress dura años y no genera residuos adicionales. Ahórrale al planeta más de 1000 cápsulas cada año. Además, no estar obligado a tener que comprar café de un tipo o marca determinado te da la oportunidad de elegir **café ecológico y de comercio justo.**

Disfruta del café mientras le haces un favor al medio ambiente.

Estudio realizado por halo coffee en 2018

Contacto: [halo@aeropresscafe.com](mailto:halo@aeropresscafe.com)



Disfruta de café barista en casa respetando el medio ambiente

[Comprar en Amazon](#)



### Prepárate el café que más te guste

AeroPress te abre un mundo de posibilidades, al no depender de cápsulas compatibles, puedes preparar todo tipo de café y lo más importante: **prepara café a medida como más te guste.**

Elige el tipo que más te guste: arábica, ibérica, robusta... Decide el origen: Colombia, Brasil, Etiopía, si es ecológico, de comercio justo... muelo tú mismo o compralo molido, añade la cantidad de agua que sea de tu agrado.

AeroPress es el aliado perfecto para poder disfrutar de **café de máxima calidad, sin condiciones y 100% personalizado.**



### Los usuarios opinan de AeroPress

<p>★★★★★ "Justo lo que buscaba" Luis Borja</p> <p>Andaba yo buscando una alternativa a mi máquina de café de cápsulas que estaba a punto de reventar y encontré esta alternativa. Necesitaba un sistema que fuera rápido, limpio y ocupase poco espacio una vez recogido por lo que la AeroPress era la alternativa ideal. Es una cafetera que se compone de...</p>	<p>★★★★★ "Super Contenta" Marta</p> <p>Me encanta esta cafetera, si eres amante del café de filtro, hazte con una, es fácil de usar y transportar. Si tienes poca paciencia esta no es tu cafetera, esta cafetera es para disfrutar del ritual completo del café. Tiene una capacidad para dos cafés, el café... <small>...se recomienda en un café en filtro, etc.</small></p>	<p>★★★★★ "Café excelente, ligero y práctico" Isaí Pérez García</p> <p>Perfecto para usar en casa o cuando te vas de camping. Aconsejable tener una herramienta de agua a mano para cada vez que quieras un café. Yo uso 2 o 3 filtros para aplicar más presión y que saque aún más interés el café. Y luego limpio los filtros y los reuso durante 1 mes o más. Los filtros que trae te duran para...</p>	<p>★★★★★ "Genial cafetera" Paloma</p> <p>Está muy bien. Yo usaba hasta ahora una de esas cafeteras italianas de toda la vida, y desde que recibí la AeroPress se ha convertido en mi cafetera del día a día. Ahora me queda trastear un poco con los "justos" (sigamos, el tiempo de preparación, la cantidad de agua/café... pero si sigues las instrucciones del manual, sin modificaciones, se vale un café. Si cualquier...</p>
---	---	---	---

[Ver valoraciones de Amazon](#)

### Ayuda a mantener limpio el planeta

Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo y hasta 29 000 acaban en vertederos\*. Aunque se empiezan a hacer esfuerzos en el reciclado y segunda vida de las cápsulas la gran mayoría acaban alimentando montañas en vertederos, cápsulas desechadas prevalecerán durante décadas e incluso cientos de años algunas, hasta desaparecer.

AeroPress dura años y no genera residuos adicionales. Ahórrale al planeta más de 1000 cápsulas cada año. Además, no estar obligado a tener que comprar café de un tipo o marca determinado te da la oportunidad de elegir **café ecológico y de comercio justo.**

Disfruta del café mientras le haces un favor al medio ambiente.

Estudio realizado por halo coffee en 2018



Contacto: [halo@aeropresscafe.com](mailto:halo@aeropresscafe.com)

### 4.3.3 Diseño Accesible

Con tal de satisfacer las necesidades de accesibilidad WCAG A.A. de la W3C<sup>12</sup> requeridas se han trabajado diferentes aspectos:

1. **Contraste de color:** Siguiendo las indicaciones de la W3C<sup>12</sup> y haciendo uso de la herramienta online que proporciona WebAIM(Web Accessibility In Mind)<sup>13</sup> se ha elaborado la paleta de colores y establecido su uso para asegurar un contraste óptimo.









	WCAG AA		WCAG AAA	
	Normal Text	Normal Text	Normal Text	Normal Text
	✓	✓	×	✓
	×	✓	✓	×
	✓	×	✓	×
	×	✓	×	✓
	✓	✓	×	×
	×	✓	×	✓
	×	✓	×	✓

Tabla de contraste y usos de la paleta creada.

### Contrast Checker


[Home](#) > [Resources](#) > Contrast Checker

**Foreground Color**

#000000 

Lightness

**Background Color**

#EBC95A 

Lightness

Contrast Ratio

13.03:1

[permalink](#)

**Normal Text**

WCAG AA: Pass

WCAG AAA: Pass

The five boxing wizards jump quickly.

**Large Text**


WCAG AA: Pass

WCAG AAA: Pass

The five boxing wizards jump quickly.

**Graphical Objects and User Interface Components**

WCAG AA: Pass



Text Input

Ejemplo de test de contraste con la herramienta de webAIM

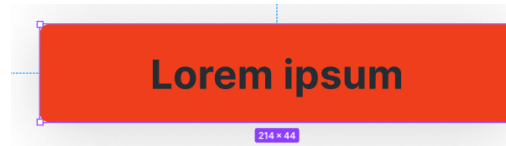
2. **Tamaño de texto, interlineado e interletrado:** La W3C en su guía de accesibilidad WCAG A.A. estipula que los textos deben ser escalables para que los usuarios puedan ampliarlos a su gusto, el interlineado debe ser de al menos 1.5 veces el tamaño de la fuente, el interletrado de al menos 0.12 veces el tamaño de la fuente y el espacio entre palabras de al

menos 0.16 veces el de la fuente. Estas directrices se han creado para presentar textos accesibles en los diseños y así se refleja en los estilos de la librería.

Typography		
h1	Family: Inter Size: 42 Line-height: 63 Type: regular	Lorem ipsum dolor sit amet, consectetur adipiscing elit
h1-b	Family: Inter Size: 42 Line-height: 63 Type: Bold	<b>Lorem ipsum dolor sit amet, consectetur adipiscing elit</b>
h1-i	Family: Inter Size: 42 Line-height: 63 Type: Italic	<i>Lorem ipsum dolor sit amet, consectetur adipiscing elit</i>
<hr/>		
h2	Family: Inter Size: 36 Line-height: 54 Type: regular	Lorem ipsum dolor sit amet, consectetur adipiscing elit
h2-b	Family: Inter Size: 36 Line-height: 54 Type: Bold	<b>Lorem ipsum dolor sit amet, consectetur adipiscing elit</b>
h2-i	Family: Inter Size: 36 Line-height: 54 Type: Italic	<i>Lorem ipsum dolor sit amet, consectetur adipiscing elit</i>
<hr/>		
h3	Family: Inter Size: 32 Line-height: 48 Type: regular	Lorem ipsum dolor sit amet, consectetur adipiscing elit
h3-b	Family: Inter Size: 32 Line-height: 48 Type: Bold	<b>Lorem ipsum dolor sit amet, consectetur adipiscing elit</b>
h3-i	Family: Inter Size: 32 Line-height: 48 Type: Italic	<i>Lorem ipsum dolor sit amet, consectetur adipiscing elit</i>
<hr/>		
h4	Family: Inter Size: 24 Line-height: 36 Type: regular	Lorem ipsum dolor sit amet, consectetur adipiscing elit
h4-b	Family: Inter Size: 24 Line-height: 36 Type: Bold	<b>Lorem ipsum dolor sit amet, consectetur adipiscing elit</b>

*Muestra de algunos de los estilos de texto de la librería creada.*

- 3. Área sensible:** La guía de accesibilidad de la W3C también estipula que la zona sensible de los elementos interactivos debe ser de al menos 44x44 píxeles, por este motivo todos los elementos interactivos son superiores o iguales a ese tamaño.



*Ejemplo del componente botón a 214x44 píxeles en este caso*

## 4.4. Stack tecnológico y Desarrollo

### 4.4.1 Stack tecnológico

El desarrollo de la web se realizará principalmente utilizando la librería React, para ello se necesitarán conocimientos de los siguientes lenguajes de programación y/o tecnologías:

#### JavaScript ES6

Dado que toda la lógica que se realice con React y Node se basa en JS ES6.

Site oficial: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

#### NodeJS

No serán necesarios grandes conocimientos de Node pero sí los imprescindibles para poder utilizar y gestionar correctamente NPM (Node Package Management), plataforma que permite instalar o desinstalar software de dependencias a través de la consola y archivos de configuración de dependencias.

Site oficial: <https://nodejs.org>

## Sass

Metalenguaje de hojas de estilo en cascada (CSS) que mediante *scripts* es traducido a CSS. Sass permite dos modelos diferentes de sintaxis `.sass` y `.scss`, para este proyecto se utilizará `.scss`. Sass provee de selectores y pseudo-selectores que agrupan las reglas que son aplicadas y aporta opciones de lógica y reutilización comunes en los lenguajes de programación y de los que CSS carece.

Site oficial: <https://sass-lang.com>

## HTML

Dado que el proyecto tiene como uno de sus objetivos la accesibilidad, serán necesarios conocimientos avanzados de HTML con tal de dotar a las etiquetas de valor semántico, orientar las diferentes etiquetas a lectores de pantalla, ayudas a la navegación con tabulador o navegación por voz y aportar suficiente información al navegador y/o dispositivo tecnológico que interpretará el lenguaje y la web en general.

Site oficial: <https://www.w3.org/html>

## ImageMagick

Software por línea de comandos que permite editar, convertir y optimizar imágenes (entre otras opciones). Este software es especialmente potente en cuanto a optimización de imágenes se refiere. Por otro lado, permite convertir formatos de imagen a Webp; formato que utilizará la web, de soportarlo el navegador. Google lo recomienda en su guía de optimización de imágenes<sup>12</sup> gracias a él también se pueden modificar el factor de muestreo, entrelazado y otros aspectos de la manera en que le gusta a Google.

Site oficial: <https://imagemagick.org/>

El anterior stack tecnológico será utilizado en el entorno de React. React es una librería de JavaScript de código abierto creada por Meta Open Source (anteriormente Facebook) en 2013, centrada en obtener mejoras en performance, modularidad, promoviendo un flujo muy claro de datos y eventos, facilitando el desarrollo de aplicaciones complejas.

Características principales de React:

- Presenta lo que denominan como ReactDOM, un DOM virtual en base al DOM real que permite modificar solamente las partes del DOM que han sido cambiadas sin necesidad de tener que actualizarlo todo, motivo por el que mejora el web performance.
- Utiliza JSX, un combinado de JavaScript y XML que permite incorporar en JS HTML y otra información a través de código JavaScript.
- Ofrece constructores y bases para que el desarrollador pueda crear sus propios componentes de manera rápida y cómoda.
- Cuenta con hooks, una funcionalidad que permite tener estado y otras características, en los componentes creados con una función. Haciendo muy fácil gestionar los estados y la lógica en base a estos de los componentes creados.
- Paradigma de programación funcional destinado a evitar cambios de estado y mutabilidad de los datos. Dado un mismo argumento una función debería devolver siempre lo mismo. Este paradigma procura una ejecución del código predecible y funciones reutilizables.

## Git

Con tal de asegurar que no pueda perderse la información, así como poder deshacer cambios, todo el proyecto está respaldado con GIT. La metodología de trabajo parte de la creación de una rama por cada tarea (crear un componente, crear una función genérica, unos estilos para X elemento...) Una

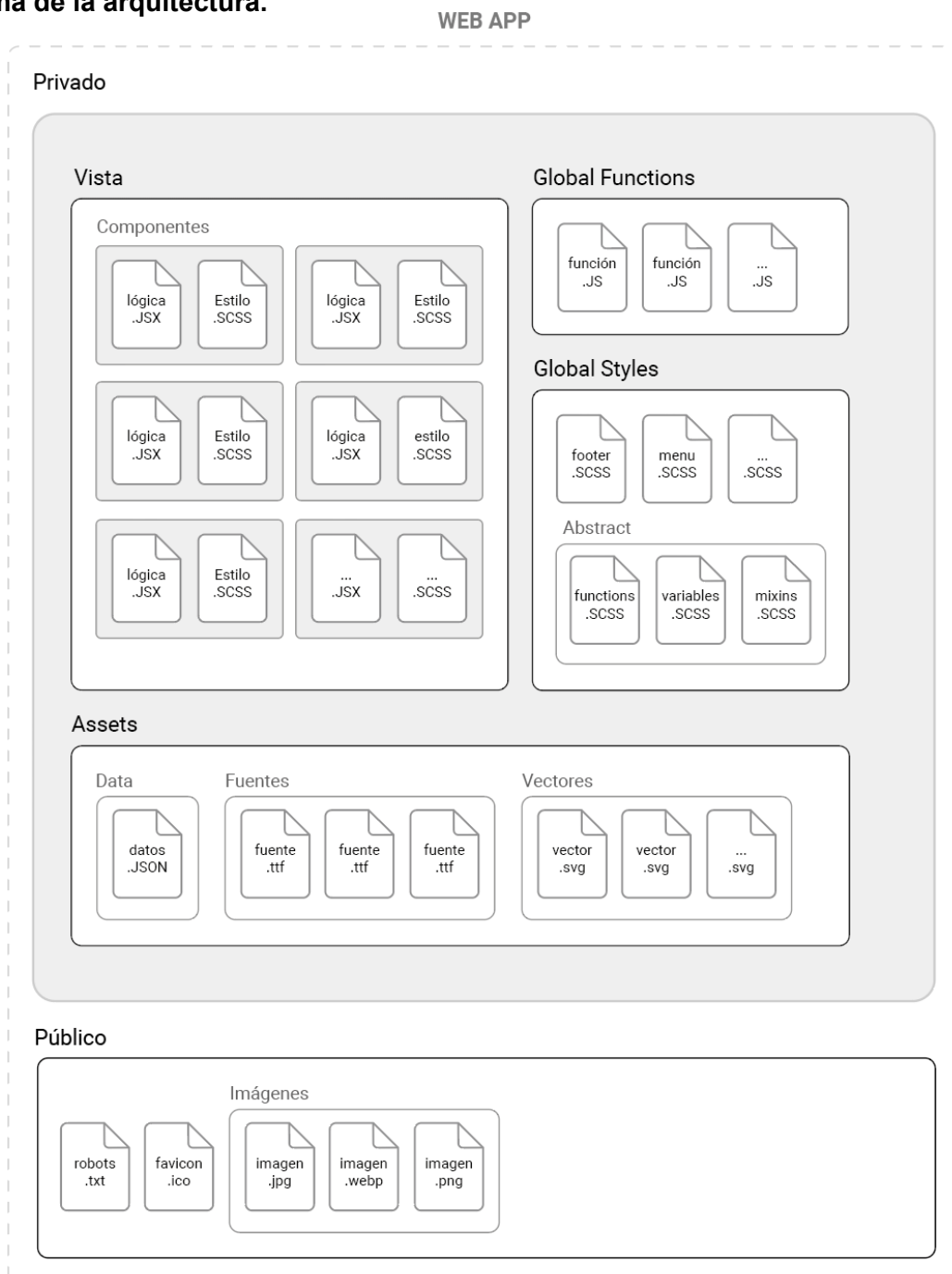


vez finalizada la tarea se hace *merge* a la rama *Develop*. Por último, si el *merge* funciona correctamente en *Develop* se hará *merge* a la rama *Master* y pasará a entorno de producción. Enlace al repositorio en Github: <https://github.com/francastillods/renite>

## 4.4.2 Arquitectura

Dado que la página web se desarrolla con React, la arquitectura se adapta al paradigma de aplicación web que ofrece. Este paradigma se caracteriza por encontrar vistas (en realidad son componentes), que son compuestas por componentes. Cada componente cuenta con su propia lógica y estilos y a su vez, puede interactuar con otros componentes. De manera general y como base, se encuentran diferentes funciones que pueden ser llamadas desde cualquier componente, un sistema de librería de estilos CSS mediante SCSS que aplicará por defecto de no haber ningún estilo en el componente y un espacio con *assets* privado y otro público. Es importante remarcar que los componentes en react son en realidad funciones que recibiendo X parámetros devolverán siempre X resultados.

### Esquema de la arquitectura.



### 4.4.3 Front-End y lógica de la aplicación.

Es importante destacar que React es una librería, no propone ninguna arquitectura o sistema de trabajo per se, pero el modelo planteado en el punto anterior y el que ahora se desarrolla, es ampliamente utilizado y común en la comunidad React.

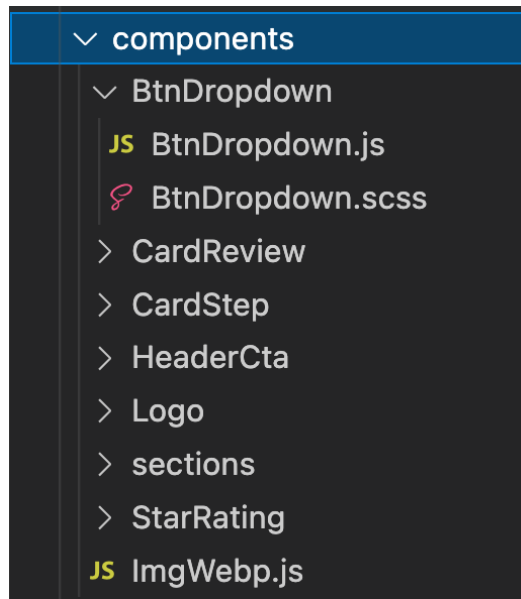
El planteamiento general, dado que es una *single page web* y no hay que hacer gestión de vistas, es el de una APP. Por lo tanto, se encuentra el primer componente, que es la APP en sí, *AeroPressCafeApp.js* donde se llama a los diferentes componentes y se les pasan los diferentes parámetros.

Aunque se hace uso del convencionalismo “llamar a los componentes” es importante tener presente que React se basa en programación funcional, en realidad lo que se está haciendo es importar funciones y ejecutarlas, esas funciones acostumbran a devolver elementos HTML que construyen y gestionan esa parte concreta de la página. No se trata de instancias de clases ni otro tipo de paradigma, son funciones que tienen como objetivo “pintar” o gestionar una pieza de interfaz.

```
JS AeroPressCafeApp.js
src > JS AeroPressCafeApp.js > ...
1 import React from 'react';
2 import { SectionDiagonal } from './components/sections/SectionDiagonal/SectionDiagonal';
3 import { CardReview } from './components/CardReview/CardReview';
4 import { CardStep } from './components/CardStep/CardStep';
5 import { HeaderCta } from './components/HeaderCta/HeaderCta';
6 import { SectionTwoCol } from './components/sections/SectionTwoCol/SectionTwoCol';
7
8 const data = require('./assets/data/dataEs.json');
9
10 export const AeroPressCafeApp = () => {
11   return (
12     <>
13       <HeaderCta
14         srcSet = { data.HOME.HEADER.IMG_SRC }
15         imgAlt = { data.HOME.HEADER.IMG_ALT }
16         claim = { data.HOME.HEADER.TITLE }
17         cta = { data.HOME.HEADER.CTA_BTN_TXT }
18         href = { data.HOME.HEADER.CTA_BTN_HREF }
19         title = { data.HOME.HEADER.CTA_BTN_TITLE }
20         rel= "noopener noreferrer"
21         target = '_blank'
22       />
23
24       <h2>Cómo preparar café con AeroPress</h2>
25       <div className= 'container card-wrapper'>
26         <CardStep
27           stepNumber = {data.HOME.SECTION_STEPS.STEP1.NUMBER}
28           imgSrc = {data.HOME.SECTION_STEPS.STEP1.IMG_SRC}
29           imgAlt = {data.HOME.SECTION_STEPS.STEP1.IMG_ALT}
30           txt = {data.HOME.SECTION_STEPS.STEP1.TXT}
31         />
32         <CardStep
33           stepNumber = {data.HOME.SECTION_STEPS.STEP2.NUMBER}
34           imgSrc = {data.HOME.SECTION_STEPS.STEP2.IMG_SRC}
35           imgAlt = {data.HOME.SECTION_STEPS.STEP2.IMG_ALT}
36           txt = {data.HOME.SECTION_STEPS.STEP2.TXT}
37         />
38         <CardStep
39           stepNumber = {data.HOME.SECTION_STEPS.STEP3.NUMBER}
40           imgSrc = {data.HOME.SECTION_STEPS.STEP3.IMG_SRC}
41           imgAlt = {data.HOME.SECTION_STEPS.STEP3.IMG_ALT}
42           txt = {data.HOME.SECTION_STEPS.STEP3.TXT}
43         />
44       </div>
45     </>
46   );
47 }
```

Vista del componente principal donde se llama a los diferentes componentes y se les pasan los parámetros.

Cada componente nace en su propio archivo JS y si es necesario utilizar estilos más allá de los generales (los estilos generales se explicarán en el siguiente punto) se crea un archivo SCSS y se empaqueta en una carpeta.

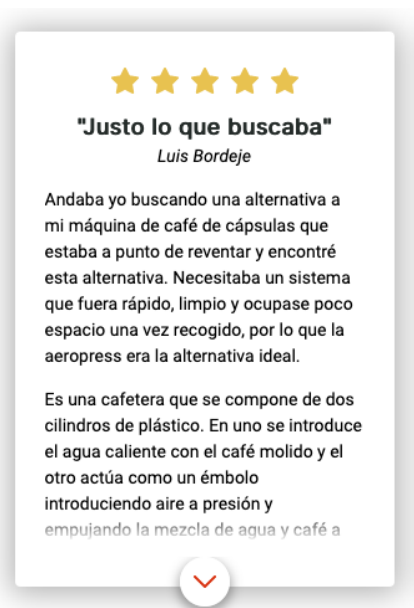


Sistema de organización de los diferentes componentes.

#### 4.4.3.1 Anatomía de un componente

Se entiende un componente como un elemento web reutilizable con su propio código que puede ser reproducido tantas veces como se desee y siempre se comportará y visualizará de una forma esperada, modificando únicamente los parámetros que así se especifiquen como susceptibles de ser cambiados. A continuación se muestra como ejemplo el componente *CardReview*, una tarjeta que muestra una valoración de producto de un usuario y su puntuación. Este componente siempre se renderizará como una tarjeta con información desplegable con los parámetros:

- rating: number (valoración en número)
- title: string (título de la tarjeta)
- name: string (Nombre del valorador)
- review: string[ ] (Texto de la valoración)



A su vez, *CardReview* hace uso del componente *StarRating* que dado un número devuelve una lista con las diferentes estrellas. Y también hace uso de *BtnDropdown* un componente que renderiza un botón de despliegue.

```

Js CardReview.js M ●
src > components > CardReview > JS CardReview.js > [🔍] CardReview > 📦 review.map() callback
1  import React from 'react'
2  import { randomKeyGen } from '../helpers/randomKeyGen'
3  import { BtnDropdown } from '../BtnDropdown/BtnDropdown'
4  import PropTypes from 'prop-types';
5  import { useState } from 'react'
6  import './CardReview.scss'
7  import StarRating from '../StarRating/StarRating'
8
9  export const CardReview = ({ rating, title, name, review }) => {
10
11     const [ switchBtn, setSwitchBtn ] = useState(false);
12     const toggleSwitch = () => {
13         switchBtn ? setSwitchBtn(false) : setSwitchBtn(true)
14     }
15
16     if (rating > 5 || rating < 0) {
17         rating = 0;
18         console.error ('Error! Rating has to be a number between 0 - 5');
19     }
20
21     return (
22         <div className='column'>
23             <div className='card-review'>
24
25                 <StarRating rating={ rating } />
26
27                 <div className= 'card-review_info ' + (switchBtn ? "card-review_info_show" : "") >
28                     <h3>{ title }</h3>
29                     <p className="card-review_info_reviewer">{ name }</p>
30                     {
31                         review.map( (p, i) => {
32                             return <p key={ randomKeyGen('rw', i) } >{ p }</p>
33                         })
34                     }
35                 </div>
36                 <BtnDropdown
37                 onClick={ toggleSwitch }
38                 />
39             </div>
40         </div>
41     )
42 }
43
44 CardReview.propTypes = {
45     rating: PropTypes.number.isRequired,
46     title: PropTypes.string.isRequired,
47     name: PropTypes.string.isRequired,
48     review: PropTypes.array.isRequired
49 }
50

```

### Dependencias, línea 1 a 7:

Las primeras líneas del documento se destinan a las diferentes dependencias que el componente pueda tener, como son la propia librería de React, los diferentes componentes, los estilos de componentes, y funciones o hooks propias o de las que React proporciona.

### Declaración del componente, línea 9:

Esta línea estará presente siempre que se cree un componente. Exactamente se crea una función de flecha de ES6 exportable, declarada como una constante y los parámetros que dicha función necesita.

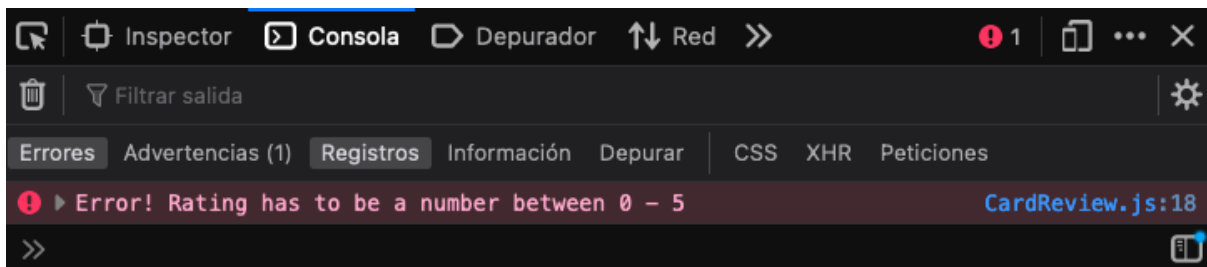
### Variables y funciones que requerirá la función. líneas 11 - 14:

Como recomiendan las buenas prácticas de desarrollo se dedica el primer apartado del cuerpo de la función para crear las variables y funciones propias de la función que quedarán ceñidas únicamente a su *scope*.

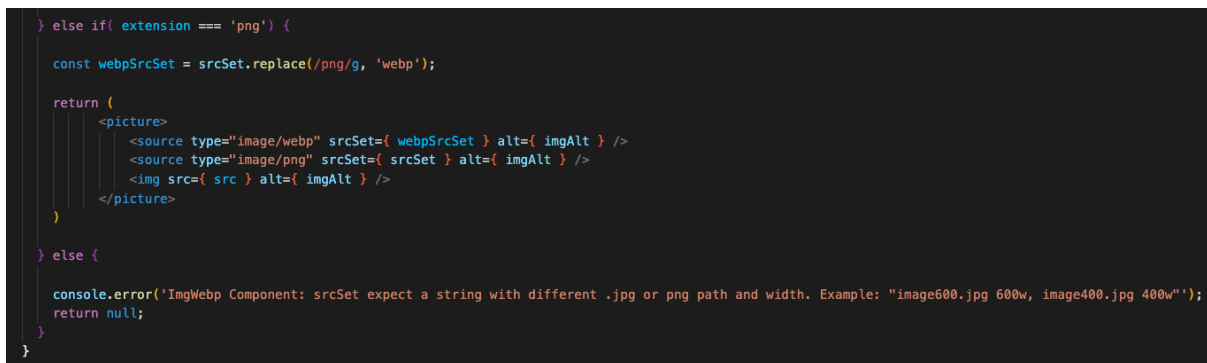
En la línea 11 aparece un buen ejemplo de *hook* de react, en concreto uno muy utilizado: *useState*. *useState()* es una función que crea internamente una variable donde almacenar el estado del componente, en este caso se quiere almacenar el estado del botón que despliega más contenido. Acepta un valor inicial para esa variable y devuelve un array con dos elementos, el valor de la variable y la función para modificarla. Como el valor devuelto por la función es un array, se hace uso de desestructuración de arrays para acceder a sus elementos de manera individual y almacenarlo en variables. El uso de este *hook* junto con la desestructuración ahorra muchísimas líneas de código.

### Verificación del parámetro rating, línea 16 - 19:

En el caso de que el parámetro *rating* sea mayor de 5 o un número negativo, se le asigna un valor de 0, por lo que todas las estrellas serán grises y se presenta un error por consola.



En este caso no se interrumpe la renderización del componente, pero el componente podría retornar *Null* y no renderizarse. Un ejemplo de ello lo es el componente *ImgWebp* que si detecta un archivo de entrada que no es *jpg* o *png* devuelve *Null* ya que no tiene garantías de que sea una imagen.



### Retorno de la función, líneas 21 - 41:

En este caso dado que la lógica del componente no es muy extensa se integra en el mismo retorno del componente. Los componentes de react siempre han de ser devueltos en una tag de tipo *wrapper* (*<div>*, *<header>*,...) o haciendo uso de *fragment* (*<></>*) En este caso se devuelve el componente en una *<div>*.

### Llamada al componente StarRating, línea 25:

Se hace referencia al componente *StarRating* para que devuelva la valoración en estrellas en base al parámetro *rating* que se le facilita.

### Contenido de la tarjeta, líneas 27 - 37:

Se pasan los parámetros *título* y *nombre* como encabezado (*<H3>*) y párrafo (*<p>*).

### Texto en párrafos, líneas 29-34:

Para mostrar los diferentes párrafos del texto se recorre el array que se le ha pasado como parámetro, pero es importante señalar el uso de la función `randomKeyGen()`;

### randomKeyGen()

Cuando react encuentra una serie de elementos iguales, como serían las entradas de una lista, varias imágenes seguidas o como es este caso varios párrafos (`<p>`) seguidos, necesita la asignación de una key única a cada elemento para poder diferenciarlo del resto, esta key no se visualiza una vez renderizado el elemento. Por este motivo, se ha creado la función `randomKeyGen()` para agilizar este proceso, pues se encuentra esta casuística varias veces a lo largo del desarrollo. Esta función en la carpeta helpers (src/helpers/randomKeyGen.js) recibe como parámetros una `string` que definirá un prefijo y una número que define su posición. La función crea una key que empieza con el prefijo facilitado, seguido de tres letras aleatorias, un número random entre 0 y 100 y por último el `counter`. De esta manera se consigue una key en modo matrícula como por ejemplo: `img-rtd-08-3`, donde se reconoce por el prefijo que se trata de una lista de imágenes, se tiene una combinación alfanumérica aleatoria y por último el número que ocupa en la lista.

Código de la función

```
JS randomKeyGen.js X
src > helpers > JS randomKeyGen.js > randomKeyGen
1  const randomString = (length) => {
2
3    let result = '';
4    const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
5    let charactersLength = characters.length;
6
7    for ( let i = 0; i < length; i++ ) {
8      result += characters.charAt(Math.floor(Math.random() * charactersLength));
9    }
10
11   return result;
12 }
13
14 const randomNumber = (maxNumber) => {
15   return( Math.floor((Math.random() * maxNumber) + 1) );
16 }
17
18 export const randomKeyGen = (prefix, counter) => {
19   return (
20     prefix + '-' + randomString(3) + '-' + randomNumber(100) + '-' + counter
21   )
22 }
```

Ejemplo haciendo un `console.log` de las keys asignadas a cada una de las valoraciones con prefijo 'rw' haciendo referencia a `review`

```
console.log(randomKeyGen('rw', i));
```

Errores	Advertencias (1)	Registros	Información	Depurar	CSS	XHR	Peticiones
							CardReview.js:32
							CardReview.js:32
							CardReview.js:32
							CardReview.js:32
							CardReview.js:32

### Componente BtnDropDown, líneas 36 - 39:

Llamada al componente *BtnDropDown* que renderiza un botón de tipo desplegable, en el atributo *onClick* le pasamos la función *toggleSwitch* definida en la línea 12 como observador de si hay que desplegar o no el contenido.

### PropTypes, líneas 44 - 49

Por último se definen las *PropTypes* de los diferentes parámetros. JavaScript es un lenguaje muy poco restrictivo lo que en ocasiones puede dar pie a una pérdida de control sobre el código o a resultados no deseados. Para poder suplir este tipo de compromisos que tiene el lenguaje React presenta *PropTypes*, un mecanismo que asegura que se están pasando los parámetros con el tipo adecuado y de la manera adecuada. En este caso se fuerza que el parámetro *rating* sea de tipo numérico, *title* y *name* de tipo cadena y por último la *review* ha de ser un arreglo. A su vez, declara que todos los parámetros son requeridos. Este mecanismo se encargará de lanzar un error por consola de no cumplirse las características especificadas.

```
44 CardReview.propTypes = {
45   rating: PropTypes.number.isRequired,
46   title: PropTypes.string.isRequired,
47   name: PropTypes.string.isRequired,
48   review: PropTypes.array.isRequired
49 }
```

## 4.4.3.2 Componentes

### BtnDropdodown

Genera un botón circular con sombra y flecha *chevron* apuntando hacia abajo en su interior. Al hacer click la flecha apunta hacia arriba mediante animación.

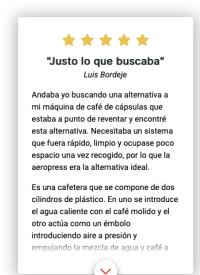


#### Parámetros:

- **onClick:** Variable que almacena la función que ha de ser llamada con el evento click del botón

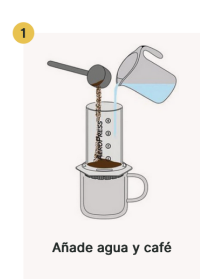
### CardReview

Genera una tarjeta de valoración con una puntuación en estrellas, título, autor y valoración en texto.



#### Parámetros:

- **rating:** Número de estrellas para la puntuación
- **title:** Título de la tarjeta
- **name:** Nombre del autor de la valoración
- **review:** Array con los diferentes párrafos



### CardStep

Genera una tarjeta de tipo "paso" para procesos de varios pasos.

#### Parámetros:

- **stepNumber:** Número y/o letra del paso

- **imgSrc:** Ruta para la imagen
- **imgAlt:** Descripción alternativa de la imagen
- **txt:** Texto del paso

### HeaderCta

Genera una cabecera de tipo llamada a la acción con imagen de bodegón, slogan y botón con llamada a la acción.



### Parámetros:

- **srcSet:** Set de imágenes
- **imgAlt:** Descripción alternativa de la imagen
- **claim:** Reclamo para el título
- **cta:** Texto del botón
- **href:** Enlace del botón
- **title:** Título del botón

### Logo

Genera el logo.



### Parámetros:

- **logoSrc:** Fuente del logo

### StarRating

Dado un número de puntuación pinta una puntuación en estrellas.



### Parámetros:

- **rating:** Número de puntos

### SectionDiagonal

Genera una sección de dos columnas, la primera con imagen y la segunda con texto. El fondo es un rectángulo en diagonal con color degradado.



### Parámetros:



- **imgSrcSet:** Set de imágenes
- **imgAlt:** Descripción alternativa de la imagen
- **title:** Título del contenido (<H2>)
- **txt:** Array con los diferentes párrafos

## SectionTwoCol

Genera una sección de 2 columnas, texto en la primera e imagen en la segunda.

### Ayuda a mantener limpio el planeta

Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo y hasta 29 000 acaban en vertederos. Aunque se empieza a hacer esfuerzo en el reciclado y segunda vida de las cápsulas la gran mayoría acaban alimentando montañas en vertederos, cápsulas desechadas que convierten durante décadas e incluso cientos de años algunos bienes desparecidos.

AeroPress dura años y no genera residuos adicionales. Ahorrate al planeta más de 1000 cápsulas cada año. Además, no eres obligado a tener que comprar café de un tipo o marca determinado te da la oportunidad de elegir café ecológico y de comercio justo.

Defensa del café mientras le haces un favor al medio ambiente.



### Parámetros:

- **title:** Título del contenido (<H2>)
- **txt:** Array con los diferentes párrafos
- **imgSrcSet:** Set de imágenes
- **imgAlt:** Descripción alternativa de la imagen

## ImgWebp

Dada una ruta de imagen JPG o PNG, busca las imágenes de mismo nombre en formato Webp y crea un elemento picture de HTML con las diferentes versiones de imagen Webp para cada resolución y otra en etiqueta img con la imagen jpg o png proporcionada en el caso de que el navegador/dispositivo no soporte el formato Webp.

Todas las imágenes no vectoriales de la web utilizan este componente.

```
<picture>
  <source type="image/webp" srcset="/img/aeropress-preparar-cafe-2000x512.webp 2000w, /img/aeropress-preparar-cafe-360x250.webp 376w" alt="Preparar café con AeroPress" />
  <source type="image/jpg" srcset="/img/aeropress-preparar-cafe-2000x512.jpg 2000w, /img/aeropress-preparar-cafe-360x250.jpg 376w" alt="Preparar café con AeroPress" />
  
</picture>
```

### Parámetros:

- **srcSet:** Ruta de la imagen en JPG o PNG
- **imgAlt:** Descripción alternativa de la imagen

## 4.4.4 Helpers

### randomKeyGen

React requiere una *key* única a cada elemento cuando este forma parte de una serie del mismo elemento (varios párrafos seguidos, listas...). *RandomKeyGen* genera una clave única.

### Parámetros:

- **prefix:** Prefijo previo a la *key*
- **counter:** Número del elemento en la lista

```
review.map( (p, i) => {
  return <p key={ randomKeyGen('rw', i) } >{ p }</p>
})
```

En este caso por cada elemento <p> del *array* que se le facilita generaría una *key*:

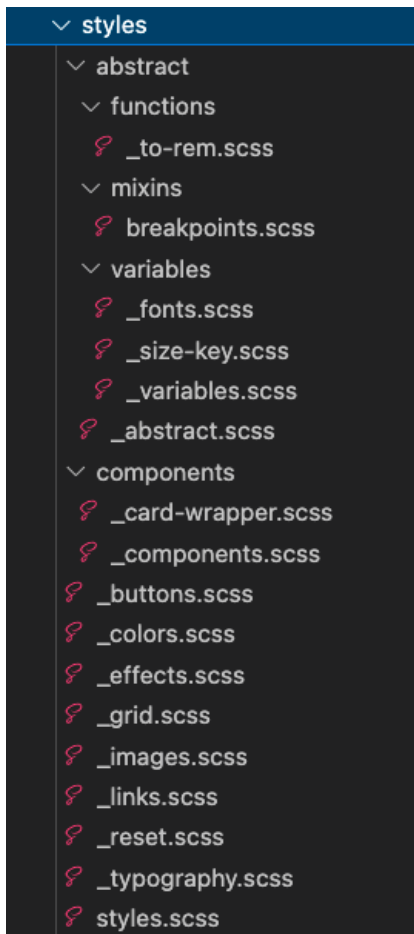
[rw] - [string alfanumérico aleatorios] - [posición]

Ejemplo: rw-xen376-1

## 4.4.5 Styles CSS

Todos los estilos del desarrollo han sido escritos en SCSS y compilados en CSS. Utilizar SCSS permite reducir el tiempo de desarrollo y futuro mantenimiento, una organización modular de los estilos, aplicar ciertas estructuras de programación (condicionales, operadores matemáticos...), se puede automatizar la minificación y compresión del código.

La estructura de documentos está pensada para ser específica y generar archivos pequeños que ayuden a la identificación y organización.



Se detectan diferentes niveles en la organización de las hojas de estilos. En la raíz se halla todo aquello relevante a efectos generales de la librería. Paletas de colores, *grid*, botones, efectos...

En el directorio “*abstract*” se encuentra todo aquello que no tiene un valor gráfico en sí mismo y sirve de apoyo al desarrollo de estilo tales como *mixins*, funciones y grupos de variables.

La carpeta “componentes” recoge estilos de elementos gráficos que funcionan como componentes pero sólo a nivel de estilos, no tienen un componente React que les acompañe, como es el caso de un contenedor de tarjetas.

Es importante mencionar que esta no es una estructura restrictiva. A medida que los evolutivos vayan generando más archivos y se encuentren nuevos posibles grupos, es recomendado crear una carpeta para agruparlos y así poder ser identificado rápidamente. Cada grupo cuenta con un archivo con el mismo nombre que el directorio donde se recogen todos los estilos del directorio, de esta manera el *styles.scss*, archivo encargado de seleccionar todos los estilos, sólo ha de mencionar dicho archivo.

### 4.4.5.1 Anatomía de una hoja de estilos.

El siguiente código hace referencia a los estilos para botones de la aplicación. Es importante apreciar que gran parte de los valores de las propiedades se definen con valores relativos que se recogen de *abstrasc.scss* donde se hallan variables y valores relativos. Serán explicados en adelante.

Otro aspecto a destacar es que todas las propiedades se ordenan alfabéticamente siempre que no sea estrictamente necesario que una propiedad superponga a otra, de esta manera el desarrollador puede encontrar la propiedad rápidamente y asegurarse de que no esté duplicada, un error muy común.

```
_buttons.scss M X
src > styles > _buttons.scss > .btn-secondary
1  @use 'abstract/abstract' as *;
2
3  button,
4  .btn {
5      background-color: rgba( var(--color-white), 1 );
6      border-radius: to-rem($size-key);
7      border: solid 2px rgba( var(--color-primary), 1 );
8      color: rgba( var(--color-primary), 1 );
9      cursor: pointer;
10     font-size: to-rem($font-body-lg);
11     font-weight: 700;
12     padding: to-rem($size-key * 2) to-rem($size-key * 4);
13     text-decoration: none;
14     width: auto;
15     width: fit-content;
16
17 }
18 .btn-primary {
19     background-color: rgba( var(--color-primary), 1 );
20     color: rgba( var(--color-white), 1 );
21
22     &:hover {
23         background-color: rgba(var(--color-primary-sh80), 1);
24         border: solid 2px rgba( var(--color-primary-sh80), 1 );
25         color: rgba( var(--color-primary), 1 );
26     }
27 }
28 .btn-secondary {
29     background-color: rgba( var(--color-secondary), 1 );
30     border: solid 2px rgba( var(--color-secondary), 1 );
31     color: rgba( var(--color-dark), 1 );
32
33     &:hover {
34         background-color: rgba(var(--color-secondary-sh80), 1);
35         border: solid 2px rgba( var(--color-secondary-sh80), 1 );
36         color: rgba( var(--color-secondary), 1 );
37     }
38 }
```

### Dependencias, línea 1:

Como se ha visto con JavaScript, el apartado superior es para las dependencias. En este caso *abstract.scss* del que sacar los valores relativos, *mixins* y funciones. Es importante observar que use *@use* en este caso o *@forward* en otros pero no *@import* que aunque funciona se considera obsoleto y se desaconseja su uso.

### Selectores de botones y clase *.btn*, líneas 3 - 17

Se selecciona el elemento *button* de html y a su vez la clase *.btn*. El objetivo es que las siguientes propiedades afecten a cualquier botón html y a aquellos elementos que quieran simular su apariencia

mediante el uso de la clase `.btn`. Esto acostumbra a ser muy común con los enlaces entre páginas bajo etiquetas `<a>` pero que se representan con un estética de botón.

### Tipos de botones, 18 a 38

En las líneas anteriores se define el estilo general de un botón en la aplicación, en las siguientes se crean clases para variaciones sobre este. En este caso se crean dos variaciones de color en función de si es un botón primario o secundario.

#### 4.4.5.2 Hojas de estilos

##### Generales

###### `_buttons.scss:`

Estilos para los botones y elementos con apariencia de botón de la aplicación.

###### `_colors.scss:`

Paleta de colores usados en la aplicación definidos en variables `root` para poder ser modificados en el cliente. Se encuentra la paleta principal, las tintas (colores de la paleta principal en gradaciones a blanco) y las sombras (colores de la paleta principal en gradaciones a negro).

Todos los colores han sido definidos describiendo su rol (*primary, secondary, pale, dark...*) y no su descripción cromática, de esta manera al reciclar el código para otro proyecto de web nicho sólo hay que reemplazar los valores dado que la variable `–color-primary` describe el color primario y no un color en sí. Por este motivo se ha evitado el uso de nombres como ‘bermellón’, ‘turquesa-claro’... ya que generarán un trabajo adicional. Además, a un desarrollador no le interesa, ni tiene conocimientos para saber algo en base a su valor cromático.

Ejemplo de paleta principal

```
--color-primary: 241, 71, 29;
```

Ejemplo de tinta de sombra al 20%

```
--color-secondary-sh20: 188, 161, 72;
```

###### `_effects.scss:`

Efectos que pueden aplicar a los estilos, en este caso sombras.

###### `_grid.scss:`

Estilos que componen la retícula y sistemas de contenedores y columnas del diseño.

###### `_images.scss:`

Estilos relevantes a imágenes.

###### `_reset.scss:`

Estilos destinados a quitar los estilos por defecto de ciertos elementos.

###### `_typography.scss:`

Estilos que aplican a los textos del proyecto: párrafos y encabezados.

##### Abstract

###### `functions > to-rem.scss:`

Función que dado un valor en píxeles lo divide entre 16 (valor *rem* por defecto) y devuelve el número en *rem*. `to-rem(24)` devolvería `1.5rem`.

```
@function to-rem($value) {
  $rem-value: math.div($value, 16) + rem;
  @return $rem-value;
}
```

#### mixins > breakpoints.scss:

Mixin con las *media queries* en base a los *breakpoints* para las diferentes resoluciones más comunes, mobile, tablet, desktop. Se ha utilizado una nomenclatura descriptiva para los nombres ya que el valor de corte en pixel puede variar en un futuro o en función del diseño, pero los dispositivos no cambiarán a medio plazo.

```
@mixin for-phone-only {
  @media (max-width: 599px) { @content; }
}
@mixin for-tablet-portrait-up {
  @media (min-width: 600px) { @content; }
}
@mixin for-tablet-landscape-up {
  @media (min-width: 900px) { @content; }
}
@mixin for-desktop-up {
  @media (min-width: 1200px) { @content; }
}
@mixin for-big-desktop-up {
  @media (min-width: 1800px) { @content; }
}
```

#### variables > \_size\_key.scss:

Como figura en el apartado 4.3.1 *Sistema de diseño* el diseño, el proyecto está basado en un *base grid* de 4 píxeles, tamaños de fuentes, espaciados y diferentes medidas son en base 4 para homogeneizar el diseño y percibirse cohesionado.

*Size-key* hace referencia al *base grid* del diseño. En este caso es 4.

```
$size-key: 4;
```

Haciendo uso de esta variable, todas las medidas del proyecto irán en base a la variable. Un valor de ancho (*width:*) de 40 píxeles se pasaría como '\$size-key \* 10'. De esta manera se generan diferentes tamaños de así deseárselo, por motivos de accesibilidad, dispositivos,... con tan sólo modificar una variable.

Recogiendo el ejemplo anterior y la función *to-rem()*, comentar que la manera exacta de definir un tamaño de 40 píxeles en el proyecto es:

```
width: to-rem( $size-key * 10);
```

De esta manera la medida es en base a la variable que referencia el *base grid* pero a su vez se traduce en código como *rem*. Esto es útil dado que los diseños y el mundo del diseño en general se basa en píxeles, pero en las interfaces se encuentran muchos tipos de resoluciones y muchos

tamaños de píxeles en función del tipo de pantalla (retina, led, 4k, QHD,..) al usar *rem* y siendo este un valor relativo se asegura que todo sea escalable y pueda funcionar en cualquier tipo de pantalla.

#### **variables > \_fonts.scss:**

Variables que hacen referencia a las diferentes tipografías y tamaños de fuente en base a su función. Una vez más el nombre no es descriptivo si no que hace referencia a su rol.

```
@use 'size-key' as *;

$font-title: 'Inter', Arial, sans-serif;
$font-body: 'Roboto', sans-serif;

$font-heading-xl: $size-key * 10.5; // 42
$font-heading-lg: $size-key * 9; // 36
$font-heading-md: $size-key * 8; // 32
$font-heading-sm: $size-key * 6; // 24
$font-heading-xs: $size-key * 4.5; // 18

$font-body-lg: $size-key * 4.5; // 18
$font-body-base: $size-key * 4; // 16
$font-body-sm: $size-key * 3.5; // 14
```

#### **variables > \_variables.scss:**

Agrupar las diferentes hojas de estilos de variables en este fichero.

#### **\_abstract.scss**

Agrupar las diferentes hojas de estilos de abstract en este fichero.

### **Components**

#### **components > card-wrapper.scss**

Estilos para el componente visual que aglutina diferentes tarjetas (componentes *CardReview* o *CardStep*) y dota de comportamiento en parrilla y sus diferentes variantes en función del tamaño de resolución.

#### **\_components.scss:**

Agrupar las diferentes hojas de estilos de componentes en este fichero.

#### **styles.scss:**

Agrupar todas las diferentes hojas de estilos en este fichero, este es el archivo que finalmente será compilado.

```

src > styles > styles.scss
1  @forward 'abstract/abstract';
2  @use 'reset' as *;
3  @use 'colors' as *;
4  @use 'typography' as *;
5  @use 'grid' as *;
6  @use 'buttons' as *;
7  @use 'links' as *;
8  @use 'images' as *;
9  @use 'effects' as *;
10 @use 'components/components' as *;

```

#### 4.4.6 Modelo de datos

Como se ha mencionado anteriormente este proyecto carece de base de datos para agilizar la reproducción de la plantilla en diferentes proyectos y sus tiempos de desarrollo. Dado el volumen de datos con el que cuenta la página web se considera que el esfuerzo no compensa el tiempo de desarrollo, coste económico y mantenimiento de una base de datos.

De la misma manera, se ha mencionado enérgicamente la necesidad de mantener el código y el contenido separados para no comprometer los desarrollos o ralentizarlos. Por este motivo, encontramos todo el contenido almacenado en *keys* de JSON en el archivo `dataEs.json` (`src/assets/data/dataEs.json`) ordenados por la sección a la que pertenece el contenido.

```

{
  "HOME": {
    "HEADER": {
      "TITLE": "Disfruta de café barista en casa respetando el medio ambiente",
      "CTA_BTN_TXT": "Comprar en Amazon",
      "CTA_BTN_HREF": "https://amzn.to/3LTq630",
      "CTA_BTN_TITLE": "Ir a Amazon",
      "IMG_SRC": "/img/aeropress-preparar-cafe-2000x512.jpg 2000w, /img/aeropress-preparar-cafe-1800x512.jpg 1800w, /img/aeropress-preparar-cafe-1200x512.jpg 1200w, /img/aeropress-preparar-cafe-900x384.jpg 900w, /img/aeropress-preparar-cafe-600x420.jpg 600w, /img/aeropress-preparar-cafe-360x250.jpg 376w",
      "IMG_ALT": "Preparar café con AeroPress"
    },
    "SECTION_STEPS": {
      "STEPS_TITLE": "Cómo preparar café con Aeropress",

```

Se aprecia todo el contenido agrupado bajo la key "HOME" que pasa a ofrecer en un subnivel las diferentes secciones: "HEADER", "SECTION\_STEPS"... , a su vez, cada subgrupo cuenta con diferentes keys para definir el contenido en su valor. El nombre de cada key se ha seleccionado cuidadosamente para primero definir de qué se trata, por ejemplo IMG (abreviatura de imagen) y después qué propiedad de ese elemento se trata, por ejemplo SRC (source) de esta manera la key "IMG\_SRC" dentro de "HEADER" orienta de una manera muy clara que se espera el valor de la fuente de la imagen del header.

```
8   const data = require('./assets/data/dataEs.json');
9
10  export const AeroPressCafeApp = () => {
11    return (
12      <>
13        <HeaderCta
14          srcSet = { data.HOME.HEADER.IMG_SRC }
15          imgAlt = { data.HOME.HEADER.IMG_ALT }
16          claim = { data.HOME.HEADER.TITLE }
17          cta = { data.HOME.HEADER.CTA_BTN_TXT }
18          href = { data.HOME.HEADER.CTA_BTN_HREF }
19          title = { data.HOME.HEADER.CTA_BTN_TITLE }
20          rel= "noopener noreferrer"
21          target = '_blank'
22        />
```

Como se puede ver en la imagen superior, se importa el JSON en la línea 8, y luego cuando se llama al componente <HeaderCta>, todos los parámetros de contenido se le pasan mediante referencias a las keys del JSON.

Es importante poner en valor que todo el contenido se encuentra dentro de este archivo, dedicando un breve tiempo a modificarlo se puede conseguir una web completamente nueva. Esto junto a definir una nueva paleta de colores en colors.scss e importando unas tipografías diferentes, algo que no debe llevar más de unos minutos, genera una web completamente nueva y con identidad propia.

Este sistema también es muy útil para generar traducciones, basta con duplicar el JSON, renombrarlo, introducir el contenido en el idioma deseado, inglés por ejemplo, y simplemente hay que definir un condicional en la carga en base al idioma del navegador del usuario para cargar un archivo u otro.



## 4.4.7 Preparación y optimización de imágenes.

Como se ha visto en el modelo de negocio y la estrategia de marketing, los modelos de negocio que rodean a las páginas web nicho tienen uno de sus pilares en el posicionamiento. Un punto importante al que los buscadores dan mucha importancia es la optimización de imágenes:

### **Peso de la imagen**

Todo kilobyte de más que no está aportando calidad a la imagen está consumiendo datos y ralentizando la carga.

### **Tamaño de visualización**

Este es uno de los errores más comunes, trata sobre reproducir la imagen a un tamaño diferente al de su tamaño natural. Por ejemplo, cargar una imagen de 1200x1200 píxeles pero mediante CSS o HTML la forzamos a reproducirse a 300x300 píxeles dado que el diseño así lo requiere. Dicha imagen debería ser cargada al tamaño al que va a ser reproducida para evitar todo ese excedente de peso que no está aportando nada.

### **Parámetros**

En la guía de optimización de imágenes de *Google developers*<sup>15</sup> se encuentran diferentes parámetros que si bien no influyen tanto en el peso de la imagen, según Google son los adecuados para una imagen que va a reproducirse en una web. Por este motivo se han adaptado a un muestreo de crominancia a 4:2:0, un formato progresivo dado que tiene una tasa de compresión JPEG superior en imágenes grandes, quitar metadatos...

### **Formato**

Si bien los formatos web por tradición son JPG, PNG y GIF, Google creó (o compró según se mire) en 2010 Webp<sup>16</sup> este formato utiliza un algoritmo de compresión *intra-frame* y *Resource Interchange File Format* como contenedor de archivo, ambos utilizados para vídeo hasta la fecha. En resumen se puede decir que Webp aporta mayor calidad con menor peso que JPG y admite canales alpha. Si bien es un formato de compresión muy potente y eso hace que los buscadores lo puntúen positivamente, hay que tener presente que es un formato propio de Google, si bien ningún dato puede demostrarlo, se puede intuir que Google premia el uso de este formato por el mero hecho de usarlo.

### 4.4.7.1 ImageMagick.

Para optimizar imágenes es común utilizar programas como ImageOptim<sup>17</sup> o herramientas online como Kraken.io<sup>18</sup>, ambas muy potentes, sencillas y rápidas de usar, pero tras leer el ejemplo de la guía de optimización de imágenes de *Google Developer*, antes mencionada, y ver que usan imageMagick<sup>19</sup> se optó por investigar esta aplicación por línea de comandos y viendo que los resultados fueron muy positivos ha pasado a ser la herramienta principal de optimización.

ImageMagick es una potente aplicación por línea de comandos que permite una extensa variedad de configuraciones y tratamientos para imagen, podría decirse que es un *Photoshop* pero en la terminal de tu ordenador. Las posibilidades que ofrece imageMagick ha permitido optimizar las imágenes con los parámetros exactos que Google propone. Para resumir el uso de imageMagick decir que tras instalarlo en tu equipo se puede invocar desde la terminal con el comando *convert* seguido del nombre de la imagen a trabajar. A continuación las diferentes transformaciones y por último, si se quiere generar un archivo nuevo en lugar de afectar al original.

```
convert imagen-original.jpg -resize 100x100 nueva-imagen-transformada.jpg
```

Este comando de ejemplo devolvería una imagen nueva redimensionada a 100 por 100 píxeles.

Para la optimización de imágenes JPG que se han usado los siguientes comandos:

- **-quality 70**: Compresión.
- **-sampling-factor 4:2:0**:
- **-interlace JPEG**: Para hacer la carga de la imagen progresiva, en conexiones lentas lo primero que se visualiza es una copia completa de la imagen degradada.
- **-strip**: Para remover todos los metadatos.

#### 4.4.7.2 Uso de Webp

Como se ha visto, el uso de Webp tiene un impacto muy positivo en una página web, ahora bien, aunque se publicó en 2010, no fue hasta 2018 que el mismo Google Chrome fue compatible con su uso. Actualmente los principales navegadores soportan el formato, pero si vamos unas pocas versiones atrás es muy probable que no. Para no dejar a nadie fuera y arriesgarse a perder usuarios se ha utilizado la siguiente estructura HTML que permite alternar entre Webp o jpg en función de si se reconoce o no el formato.

```
16 <picture>
17   <source type="image/webp" srcSet={ webpSrcSet } alt={ imgAlt } />
18   <source type={'image/' + extension } srcSet={ srcSet } alt={ imgAlt } />
19   <img src={ src } alt={ imgAlt } />
20 </picture>
```

La línea 16 abre el elemento `<picture>` de HTML, este etiquetado es clave ya que permite cargar imágenes de manera diferente a la clásica etiqueta `<img>`. Dentro encontramos dos etiquetas `<source>`, la primera (línea 17) espera recibir en el atributo `srcSet` una lista de imágenes *Webp* y el tamaño de resolución en que usar una imagen u otra, esta es otra mejora frente al uso de `<img>` que sólo permite referenciar una imagen en su atributo `src`. El segundo `<source>` (línea 18) carga una serie de imágenes en formato JPG o PNG (según el parámetro que se le pase), esta línea será interpretada en el caso de que el navegador no soporte Webp pero sí soporte la etiqueta `<picture>`. Por último, y para no dejar a nadie sin visualizar la imagen, se introduce una etiqueta `<img>` cargando la versión de la imagen en JPG o PNG, esto sucederá cuando el navegador no reconozca la etiqueta `<picture>`, relativamente nueva, al no interpretar las líneas 16,17 y 18 las obviará y se quedará con la línea 19 (la `<img>`) que si la entiende. Si identifica `<picture>` y *WebP* obviará el resto de opciones tras el primer `<source>`.

Este planteamiento para trabajar con imágenes es muy potente ya que cubre todas las casuísticas posibles, pero puede hacerse un poco complicado de trabajar ya que, hay que enlazar muchas imágenes, definir varios formatos... Por ello, se ha creado un componente específico en React: *ImgWebp*. Su uso es sencillo, el componente espera dos parámetros: la lista de imágenes (`srcSet`) sólo en formato JPG o PNG y la descripción de la imagen. Pasado el `srcSet`, el componente detecta el formato (JPG o PNG) y genera una segunda lista cambiando la extensión a Webp. Hay que puntualizar que se espera que todas las imágenes estén en el formato facilitado (JPG o PNG) y en Webp en el directorio de imágenes. Finalmente el componente devuelve el HTML con ambos `<sources>` y el `<img>`.

```

JS ImgWebp.js
src > components > JS ImgWebp.js > @ ImgWebp
1 import PropTypes from 'prop-types';
2
3 export const ImgWebp = ( { srcSet, imgAlt } ) => {
4
5   const [ src, ] = srcSet.split(' ');
6   const [ , extension, ] = src.split('.');
7
8   if( extension === 'jpg' || 'png' ) {
9
10    var replace = new RegExp(extension,"g");
11    const webpSrcSet = srcSet.replace(replace, 'webp');
12
13    return (
14      <picture>
15        <source type="image/webp" srcSet={ webpSrcSet } alt={ imgAlt } />
16        <source type='image/' + extension } srcSet={ srcSet } alt={ imgAlt } />
17        <img src={ src } alt={ imgAlt } />
18      </picture>
19    )
20
21  } else if( extension === 'png' ) {
22
23    const webpSrcSet = srcSet.replace(/png/g, 'webp');
24
25    return (
26      <picture>
27        <source type="image/webp" srcSet={ webpSrcSet } alt={ imgAlt } />
28        <source type="image/png" srcSet={ srcSet } alt={ imgAlt } />
29        <img src={ src } alt={ imgAlt } />
30      </picture>
31    )
32
33  } else {
34
35    console.error('ImgWebp Component: srcSet expect a string with different .jpg or png path and width. Example: "image600.jpg 600w, image400.jpg 400w"');
36    return null;
37  }
38 }
39
40 ImgWebp.propTypes = {
41   srcSet: PropTypes.string.isRequired,
42   imgAlt: PropTypes.string
43 }

```

# 5. Implementación

## 5.1. Requisitos de instalación

### 5.1.1 Software necesarios

- Node (versión estable 16.15.0 o superior) <https://nodejs.org/en/>
- React vía npm: `npx create-react-app nombre-app`
- Navegador web (se recomienda Firefox developer edition)
- Git
- ImageMagick
- Editor de código (se recomienda Visual Studio Code)

### 5.1.2 Hardware

Para poder trabajar correctamente con React y que la aplicación funcione en local como cliente y servidor, los requisitos mínimos son los de un PC de gama baja (año 2022).

## 5.2 Instrucciones de instalación en local

### Node

El primer paso es instalar Node.js versión 14.17.5 o versión estable superior en el equipo, se puede descargar el archivo de instalación en la web oficial: <https://nodejs.org/en/> , para asegurar que la instalación se realizó correctamente escribir en una terminal bash `node -v` y debería devolver el nombre de la versión de Node, en este caso `v14.17.5`.

### GIT

El segundo paso es instalar GIT para descargar la rama actualizada y asegurarse de tener el proyecto respaldado. Descargar el archivo de instalación de <https://git-scm.com/downloads> y escribir en la terminal bash `git --version` para asegurar que se instaló correctamente en este caso `git` versión 2.15.0.

### ImageMagick

En el caso de querer tratar alguna imagen se recomienda utilizar ImageMagick y tratar las imágenes con los parámetros especificados en el apartado 4.4.5.1. Para su instalación en Windows será suficiente descargar la última versión en la web oficial <https://imagemagick.org/script/download.php> .

En caso de Mac o Linux será necesario instalar previamente homebrew escribiendo en la terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Seguidamente instalar ImageMagick mediante Homebrew escribiendo en la terminal

```
brew install imagemagick
```

### Navegador web

Se recomienda el uso de Firefox Developer Edition por las diferentes herramientas que aporta para el desarrollo y como modo de contribuir y apoyar a Mozilla, organización sin ánimo de lucro que tanto ha hecho y hace por Internet y los lenguajes HTML, CSS y JavaScript. Se puede descargar el instalador en <https://www.mozilla.org/en-US/firefox/developer/>

## Editor de código

Para poder trabajar con el código se recomienda descargar e instalar Visual Studio Code, aunque se puede trabajar sin problemas con la mayoría de editores e IDEs de desarrollo más populares. Abrir <https://code.visualstudio.com/> y hacer click en “Descargar”.

Para levantar el proyecto, clonar los archivos que figuran en la carpeta “aeropresscafe” adjuntada a la memoria, en el directorio deseado o clonar la última versión desde el repositorio abriendo el directorio deseado en la terminal y escribiendo:

```
git clone https://github.com/francastillods/renite
```

Seguidamente en la raíz del proyecto se deben instalar las dependencias a través de npm, escribid el siguiente comando.

```
npm install
```

El proyecto ya está correctamente instalado en nuestro equipo, para levantar un servidor local y poder ver los cambios en el navegador se hará uso del que ofrece Node. Para levantar el servidor se ha de escribir el siguiente comando:

```
npm run start
```

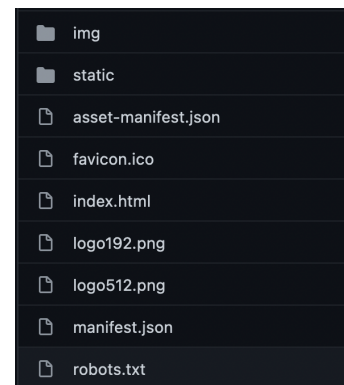
Ahora es posible acceder a la aplicación levantada en el servidor local a través del puerto 3000 escribiendo en el navegador: <http://localhost:3000>

## 5.3 Instrucciones en entorno de producción

Una vez se hayan hecho las modificaciones pertinentes y se tengan los archivos necesarios para subir al servidor de producción, se realizará el *build* de producción. Este genera los archivos HTML, CSS y JavaScript necesarios para que puedan ser interpretados por el servidor y el cliente. Para generar dichos archivos se debe escribir el siguiente comando:

```
npm run build
```

Ahora se puede observar que se ha generado (o actualizado) la carpeta *build* con los archivos necesarios. Arrastrar estos archivos al directorio asociado a la ruta web deseada y la web estará publicada.



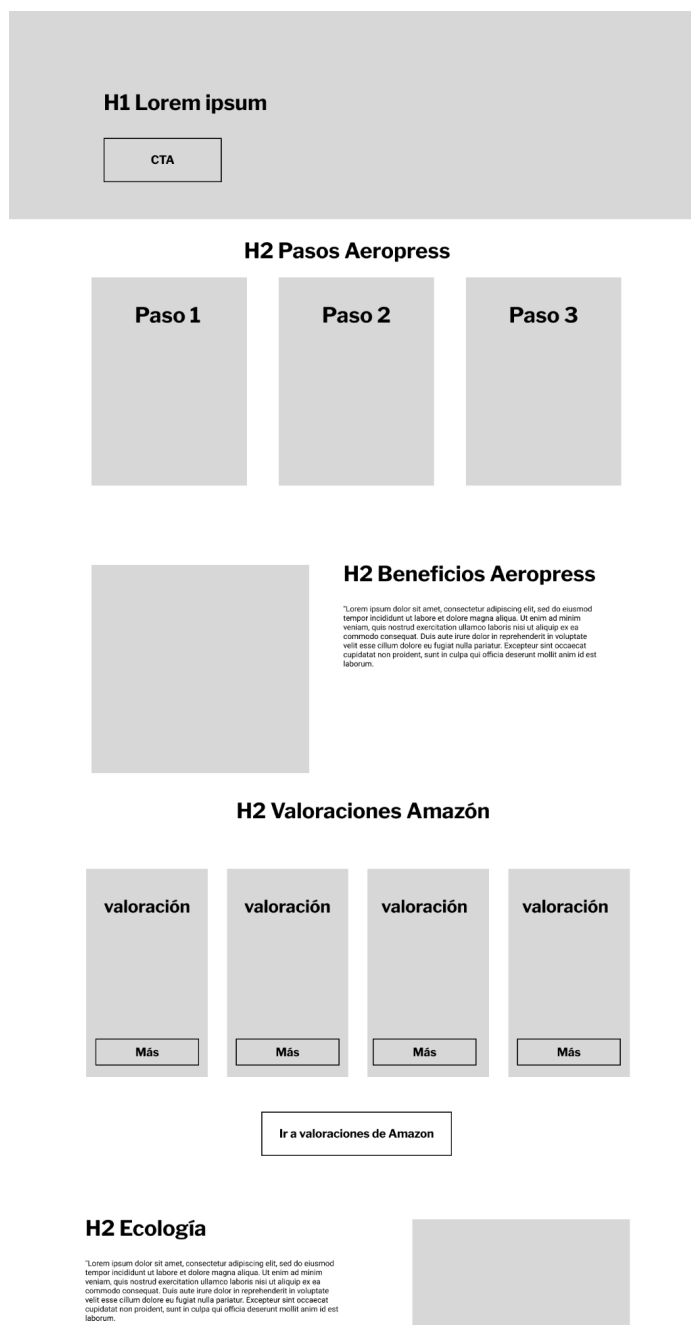
<https://www.aeropresscafe.com>, el proyecto base que ha servido como base para la creación de la plantilla React, se ha alojado en Vercel. Vercel es una plataforma destinada a frameworks de *frontend* y sitios estáticos que permite asociar una rama de Github de la que obtiene la información y a su vez permite correr *scripts* cada vez que se da una actualización en la rama asociada. Esta es una herramienta muy potente ya que permite ahorrar mucho tiempo y evita complicaciones. Cada vez que hay una actualización en la rama *Master* del proyecto, se ha configurado que se corra el comando *npm run build* automáticamente. Esto quiere decir que cada vez que se hace una actualización a la rama *Master* de Github automáticamente se actualiza la web en producción. Además al ser un único proyecto y usuario el servicio es gratuito. Para más información: <https://vercel.com/>

# 6. Demostración

## 6.1 Instrucciones de uso

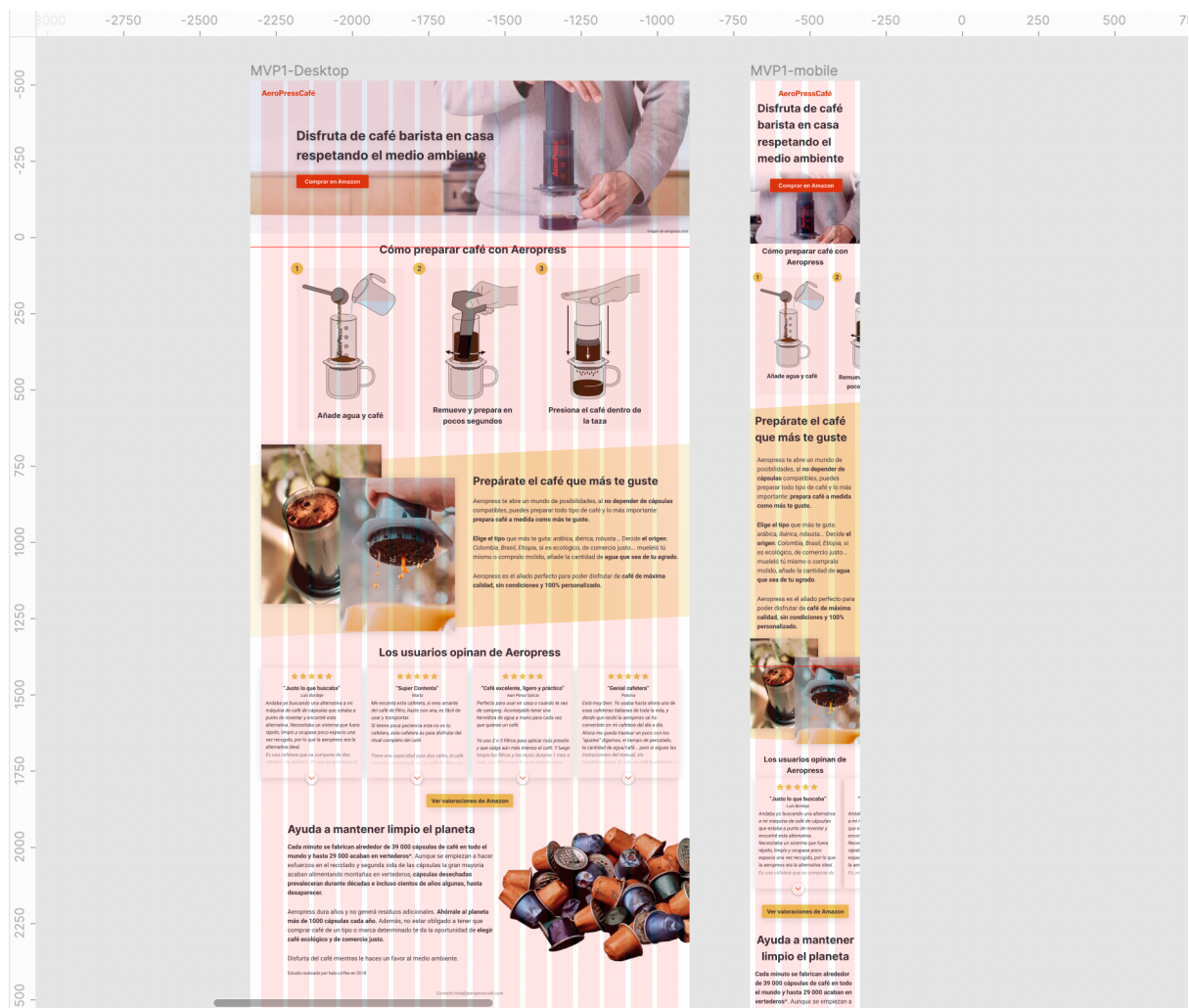
Habiendo seguido los pasos indicados en el punto 5.2 *Instrucciones de instalación en local* una vez arrancado el cliente desde la consola de comandos con el comando `npm run start` se puede acceder al navegador que habrá abierto una ventana por defecto con la url apuntando a <http://localhost:3000>. También puede visualizarse el resultado en <https://www.aeropresscafe.com>, donde el proyecto está alojado en Vercel como se apuntaba en el punto 5.3 *Instrucciones en entorno de producción*.

## 6.2. Prototipos Lo-Fi



## 6.2. Prototipos Hi-Fi

Presentada una estructura básica definida en Lo-Fi y teniendo en cuenta las características del diseño, se pasó del prototipo Lo-Fi directamente al diseño de la interfaz ya que, en este caso en particular se consideró que el esfuerzo de un prototipo Hi-Fi para testeo no estaba justificado. Dicho prototipo no podría asumir todos los desafíos que posteriormente encontraría en la fase de diseño de interfaz. Partiendo de la base de que usualmente el equipo de diseño se compone de diseñadores UX (experiencia de usuario) y de diseñadores UI (interfaz de usuario) parece una mejor gestión del esfuerzo pasar de Lo-Fi a diseño teniendo en cuenta que se pueden tener diálogos entre ambos diseñadores de darse algún problema. Puede incrementar un poco las horas de diseño UI pero compensa el tiempo mayor que sería generar un prototipo Hi-Fi que tampoco solventaría todos los desafíos de interfaz.



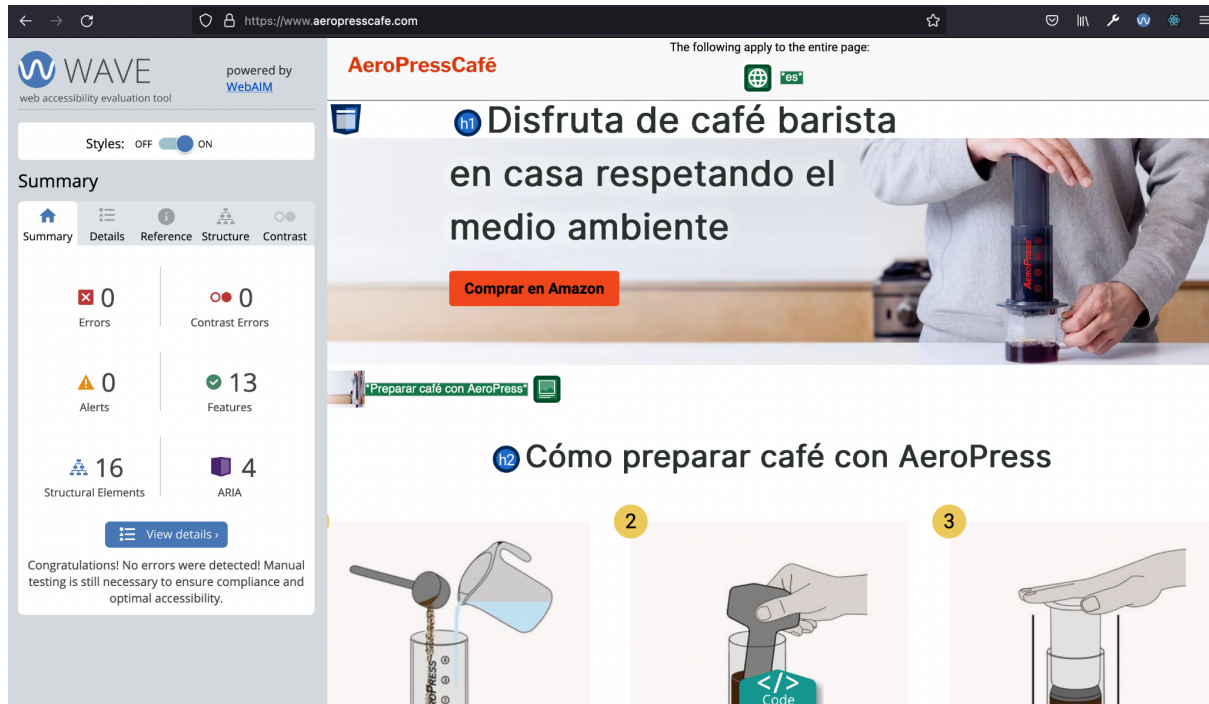
## 6.3 Test

### 6.3.1 PropTypes

Para asegurar un correcto renderizado de la web se ha hecho uso de las *PropTypes* presentadas en el punto 4.4.3.1 *Anatomía de una componente* que se encargan de testear los parámetros y funcionamiento de los diferentes componentes para asegurarse de que se cumple el funcionamiento esperado y devolver los errores pertinentes para que el desarrollador pueda subsanarlos.

### 6.3.2 Test accesibilidad

A lo largo del documento se ha hecho especial hincapié en la importancia de la accesibilidad y se ha expuesto como uno de los objetivos. Para asegurar que este aspecto ha sido cubierto se ha hecho uso de la herramienta web para testeo de accesibilidad WAVE (Web Accessibility Evaluation Tool) creada por WebAIM<sup>20</sup>, una iniciativa del *Institute for Disability Research, Policy, and Practice* en pro de la accesibilidad tecnológica. Si bien se ha utilizado la extensión para Firefox, se puede hacer uso de la herramienta a través de la web: <https://wave.webaim.org/>



Test pasado sin errores ni alertas de accesibilidad

### 6.3.3 Test usabilidad y negocio

Finalmente, se ha testado a nivel de usuario haciendo uso de la propia navegación de la web, asegurando que todos los componentes se cargan con el comportamiento deseado e imágenes y contenidos son correctos.

Se ha dado especial importancia a verificar que los enlaces de afiliado a Amazon funcionan correctamente y asocian correctamente con la cuenta de Amazon. El tag *franca-21* apunta al ID de la cuenta de afiliados por lo que el enlace es correcto.

[https://www.amazon.es/Aerobie-AeroPress-Cafetera-de-%C3%A9mbolo/dp/B000GXZ2GS?\\_mk\\_es\\_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=1KHBVMK1OUIIZ&keywords=aeropress&qid=1649777634&s=kitchen&sprefix=aeropress.kitchen.121&sr=1-2&th=1&linkCode=s11&tag=franca-21&linkId=3e236d14f52a4d47a4eaa1aab18105c9&language=es\\_ES&ref=as\\_li\\_ss\\_tl](https://www.amazon.es/Aerobie-AeroPress-Cafetera-de-%C3%A9mbolo/dp/B000GXZ2GS?_mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=1KHBVMK1OUIIZ&keywords=aeropress&qid=1649777634&s=kitchen&sprefix=aeropress.kitchen.121&sr=1-2&th=1&linkCode=s11&tag=franca-21&linkId=3e236d14f52a4d47a4eaa1aab18105c9&language=es_ES&ref=as_li_ss_tl)



## 6.4 Ejemplos de uso del producto (o guía de usuario).

### 6.4.1 Desde el punto de vista del propietario.

Para poder crear una página web nicho propia a partir de la actual se debe seleccionar el archivo `dataEs.json` ubicado en `src/assets/data/dataEs.json` que contiene todo el contenido de la página. Para cambiar el contenido y poder generar una nueva página simplemente se debe ir cambiando el valor de las diferentes *keys* del archivo JSON, a continuación se asocia de manera gráfica cada *key* con su elemento en la web.

#### HEADER

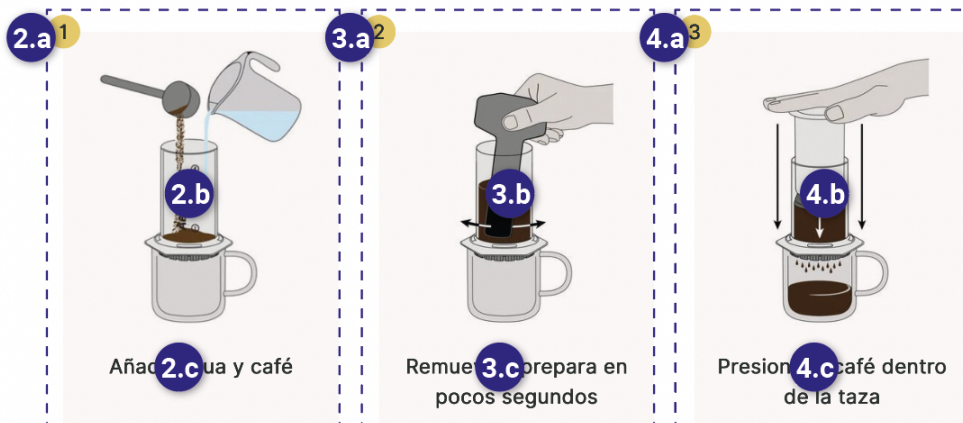


The screenshot shows a web header for "AeroPressCafé". It features a main title "1 Disfruta de café barista en casa respetando el medio ambiente", a call-to-action button "2 Comprar en Amazon", and a background image of a person using an AeroPress coffee maker, labeled "3".

```
"HEADER": {  
  1 "TITLE": "Disfruta de café barista en casa respetando el medio ambiente",  
  "CTA_BTN_TXT": "Comprar en Amazon",  
  2 "CTA_BTN_HREF": "https://amzn.to/3LTq630",  
  "CTA_BTN_TITLE": "Ir a Amazon",  
  3 "IMG_SRC": "/img/aeropress-preparar-cafe-2000x512.jpg 2000w, /img/aeropress-p",  
  "IMG_ALT": "Preparar café con AeroPress"  
},
```

## SECTION STEPS

### 1 Cómo preparar café con AeroPress



```
"SECTION_STEPS": {  
  1 "STEPS_TITLE": "Cómo preparar café con Aeropress",  
  "STEP1": {  
    2.a "NUMBER": "1",  
    "IMG_SRC": "/img/step1.jpg",  
    2.b "IMG_ALT": "paso 1",  
    2.c "TXT": "Añade agua y café"  
  },  
  "STEP2": {  
    3.a "NUMBER": "2",  
    "IMG_SRC": "/img/step2.jpg",  
    3.b "IMG_ALT": "paso 2",  
    3.c "TXT": "Remueve y prepara en pocos segundos"  
  },  
  "STEP3": {  
    4.a "NUMBER": "3",  
    "IMG_SRC": "/img/step3.jpg",  
    4.b "IMG_ALT": "paso 3",  
    4.c "TXT": "Presiona el café dentro de la taza"  
  }  
},
```

## SECTION PRESENTATION

1  2 **Prepárate el café que más te guste**

Aeropress te abre un mundo de posibilidades, al no depender de cápsulas compatibles, puedes preparar todo tipo de café y lo más importante: prepara café a medida como más te guste.

3 Elige el tipo que más te guta: arábica, ibérica, robusta... Decide el origen: Colombia, Brasil, Etiopia, si es ecológico, de comercio justo... muelelo tú mismo o compralo molido, añade la cantidad de agua que sea de tu agrado.

Aeropress es el aliado perfecto para poder disfrutar de café de máxima calidad, sin condiciones y 100% personalizado.

```
"SECTION_PRESENTATION": {  
  1 "IMG_SRC": "/img/preparar-cafe-aeropress.png",  
  2 "IMG_ALT": "imgAlt",  
  3 "TITLE": "Prepárate el café que más te guste",  
  "TXT": [  
    "Aeropress te abre un mundo de posibilidades, al no depender de cápsulas compatibles, puedes preparar todo tipo de café y lo más importante: prepara café a medida como más te guste.",  
    "Elige el tipo que más te guta: arábica, ibérica, robusta... Decide el origen: Colombia, Brasil, Etiopia, si es ecológico, de comercio justo... muelelo tú mismo o compralo molido, añade la cantidad de agua que sea de tu agrado.",  
    "Aeropress es el aliado perfecto para poder disfrutar de café de máxima calidad, sin condiciones y 100% personalizado."  
  ],  
},
```

## SECTION REVIEWS

**1a** ★★★★★

**1b** "Justo lo que buscaba"

**1c** *Luis Bordeje*

Andaba yo buscando una alternativa a mi máquina de café de cápsulas que estaba a punto de reventar y encontré esta alternativa. Necesitaba un sistema que fuera rápido, limpio y ocupase poco espacio una vez recogido, por lo que la aeropress era la alternativa ideal.

**1d** Es una cafetera que se compone de dos cilindros de plástico. En uno se introduce el agua caliente con el café molido y el otro actúa como un émbolo introduciendo aire a presión y empujando la mezcla de agua y café a

**2a** ★★★★★

**2b** "Super Contenta"

**2c** *Marta*

Me encanta esta cafetera, si eres amante del café de filtro, hazte con una, es fácil de usar y transportar.

**2d** Si tienes poca paciencia esta no es tu cafetera, esta cafetera es para disfrutar del ritual completo del café. Tiene una capacidad para dos cafés, el café como he comentado es un café de filtro, tipo americano.

Yo caliento el agua y la dejo enfriar hasta unos 80 grados, mientras enfria pongo los granos de café en el molinillo (una

**3a** ★★★★★

**3b** "Café excelente, ligero y práctico"

**3c** *van Perez Garcia*

Perfecto para usar en casa o cuando te vas de camping. Aconsejable tener una hervidora de agua a mano para cada vez que quieras un café.

**3d** Yo uso 2 o 3 filtros para aplicar más presión y que salga aún más intenso el café. Y luego limpio los filtros y los reuso durante 1 mes o más. Los filtros que lleva te durarán para años.

Es increíble cómo algo tan sencillo puede sacar un café mucho más suave y

**4a** ★★★★★

**4b** "Genial cafetera"

**4c** *Paloma*

Está muy bien. Yo usaba hasta ahora una de esas cafeteras italianas de toda la vida, y desde que recibí la aeropress se ha convertido en mi cafetera del día a día.

**4d** Ahora me queda trastear un poco con los "ajustes" digamos, el tiempo de percolado, la cantidad de agua/café... pero si sigues las instrucciones del manual, sin complicaciones, te sale un café buenísimo, y eso que uso uno premolido típico. Esa será mi próxima compra, un molinillo para moler el café al

**5** Ver valoraciones de Amazon

```
"SECTION_REVIEWS": {
  "REVIEW1": {
    1a "RATING": 5,
    1b "TITLE": "\"Justo lo que buscaba\"",
    1c "NAME": "Luis Bordeje",
    1d "TXT": [
      "Andaba yo buscando una alternativa a mi máquina de café de cápsulas que estaba a",
      "Es una cafetera que se compone de dos cilindros de plástico. En uno se introduce",
      "Su funcionamiento es sencillísimo y hay multitud de recetas en internet, en las",
      "Una vez que acabas separas las dos partes y las lavas directamente debajo del gr",
      "El café que se obtiene con este sistema es intenso, pero no esperes la intensidad
    ]
  },
  "REVIEW2": {
    2a "RATING": 5,
    2b "TITLE": "\"Super Contenta\"",
    2c "NAME": "Marta",
    2d "TXT": [
      "Me encanta esta cafetera, si eres amante del café de filtro, hazte con una, es f",
      "Si tienes poca paciencia esta no es tu cafetera, esta cafetera es para disfrutar",
      "Yo caliento el agua y la dejo enfriar hasta unos 80 grados, mientras enfria pong",
      "Muy importante, nada mas terminar de hacer el proceso, límpiala, es mucho mas fá
    ]
  },
  "REVIEW3": {
    ...
  },
  "BTN_AMZ" : "Ver valoraciones de Amazon",
  5 "BTN_AMZ_HREF" : "https://amzn.to/3LTq630",
  "BTN_AMZ_TITLE" : "Ver en Amazon"
},
```

Repetir por cada entrada

## SECTION RECYCLING

### 1 Ayuda a mantener limpio el planeta

Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo y hasta 29 000 acaban en vertederos\*. Aunque se empiezan a hacer esfuerzos en el reciclado y segunda vida de las cápsulas la gran mayoría acaban alimentando montañas en vertederos, cápsulas

2 echadas prevaleceran durante décadas e incluso cientos de años algunas, hasta desaparecer.

Aeropress dura años y no generará residuos adicionales. Ahórrale al planeta más de 1000 cápsulas cada año. Además, no estar obligado a tener que comprar café de un tipo o marca determinado te da la oportunidad de elegir café ecológico y de comercio justo.

Disfruta del café mientras le haces un favor al medio ambiente.



```
"SECTION_RECYCLING" : {  
  1 "TITLE": "Ayuda a mantener limpio el planeta",  
  2 "TXT": [  
    "Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo",  
    "Aeropress dura años y no generará residuos adicionales. Ahórrale al planeta más de 1000",  
    "Disfruta del café mientras le haces un favor al medio ambiente."  
  ],  
  5 ["IMG_SRC": "/img/reciclar-capsulas-cafe-wh.jpg",  
    "IMG_ALT": "Reciclaje de cápsulas de café"]  
}
```

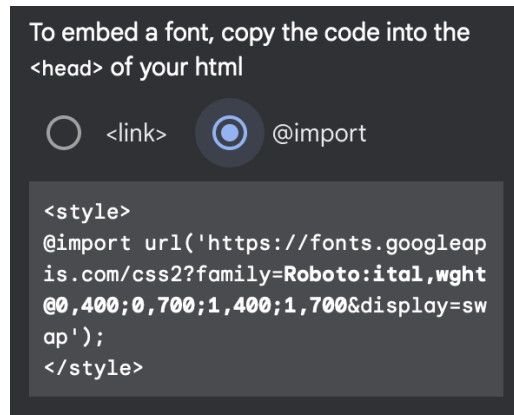
### Paleta de colores

Para poder configurar una paleta de colores que se adapte a los colores corporativos y/o deseados simplemente se ha de modificar el valor de las variables de colores albergadas en `src/styles/_colors.scss`, dicho valor a de estar en modelo de color RGB.

```
:root {  
  /* Main Palette */  
  --color-primary: 241, 71, 29;  
  --color-secondary: 235, 201, 90;  
  --color-dark: 42,50,48;  
  --color-pale: 251,247,246;  
  /* Shades */  
  --color-secondary-sh20: 188, 161, 72;  
  --color-secondary-sh40: 141, 121, 54;  
  --color-secondary-sh60: 94, 80, 36;  
  --color-secondary-sh80: 47, 40, 18;  
  /* Grey Palette */  
  --color-grey-dark: 153, 153, 153;  
  --color-grey-medium: 158, 154, 152;  
  --color-grey-light: 251, 247, 246;  
  /* B&W Palette */  
  --color-black: 0, 0, 0;  
  --color-white: 255, 255, 255;  
}
```

## Tipografías

Por último, si se desea modificar la tipografía se recomienda hacer uso del catálogo de Google Fonts (<https://fonts.google.com/>) ya que permite copiar el código de importación para scss y de esta manera se evitan posibles errores.



*Código a seleccionar.*

Una vez se dispone del código de importación facilitado por Google fonts se debe incluir en el archivo `_typography.scss` ubicado en `src/styles/_typography.scss` y sustituir la línea 3 por el nuevo código.

```
1 @use 'abstract/abstract' as *;
2
3 @import url('https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,400;0,700;1,400;1,700&dis
```

Para terminar, se determina si la nueva fuente está destinada al cuerpo del texto o a los títulos asociándola a la variable que usan los estilos. Para ello, se abre `_fonts.scss` ubicado en `src/styles/abstract/variables/_fonts.scss` y se modifica por el valor de la variable 3 o 4 por el nombre de la nueva tipografía.

```
1 @use 'size-key' as *;
2
3 $font-title: 'Inter', Arial, sans-serif;
4 $font-body: 'Roboto', sans-serif;
5
```

## 6.4.2 Desde el punto de vista del desarrollador.

Desde el punto de vista del desarrollador se puede utilizar la plantilla como base para nuevos proyectos web, a parte de los pasos anteriores para modificación de contenido, el desarrollador puede añadir nuevas secciones llamando a los componentes que alberga la plantilla. Para ello se ha de modificar el archivo `AeroPressCafeApp.js` ubicado en `src/AeroPressCafeApp.js` añadiendo aquellos componentes que se desea incorporar. El único punto a tener en cuenta son los parámetros que espera dicho componente para asegurar que se renderiza como se espera. Para ello, se puede utilizar la ayuda de Visual Studio Code que al escribir el nombre del componente informa de los parámetros esperados y su tipo.

```
< [ data-HOME-SECTION-REVIEWS-PTN-AM7 ] </>
<
  (alias) const SectionDiagonal: ({ imgSrcSet, imgAlt, title, txt }: {
  <
    imgSrcSet: any;
    imgAlt: any;
    title: any;
    txt: any;
  }) => JSX.Element
  /
  import SectionDiagonal
<SectionDiagonal>
</SectionDiagonal>
```

## 6.4.3 Desde el punto de vista del usuario.

Desde el punto de vista del usuario, se encuentra un uso de consumo de información mediante navegación web estándar basada en *scroll*, botones y desplegables.

**AeroPressCafé**

Disfruta de café barista en casa respetando el medio ambiente

[Comprar en Amazon](#)

### Cómo preparar café con AeroPress

- 1**  **Añade agua y café**
- 2**  **Remueve y prepara en pocos segundos**
- 3**  **Presiona el café dentro de la taza**

### Prepárate el café que más te guste

AeroPress te abre un mundo de posibilidades, al no depender de cápsulas compatibles, puedes preparar todo tipo de café y lo más importante: prepara café a medida como más te guste.

Elige el tipo que más te gusta: arábica, ibérica, robusta... Decide el origen: Colombia, Brasil, Etiopía, si es ecológico, de comercio justo... muelelo tú mismo o compralo molido, añade la cantidad de agua que sea de tu agrado.

AeroPress es el aliado perfecto para poder disfrutar de café de máxima calidad, sin condiciones y 100% personalizado.

★★★★★ **"Justo lo que buscaba"**  
*Luis Bordeje*

Andaba yo buscando una alternativa a mi máquina de café de cápsulas que estaba a punto de reventar y encontré esta alternativa. Necesitaba un sistema que fuera rápido, limpio y ocupase poco espacio una vez recogido, por lo que la AeroPress era la alternativa ideal.

Es una cafetera que se compone de dos cilindros de plástico. En uno se introduce el agua caliente con el café molido y el otro actúa como un émbolo introduciendo aire a presión y empujando la mezcla de agua y café a

★★★★★ **"Super Contenta"**  
*Marta*

Me encanta esta cafetera, si eres amante del café de filtro, hazte con una, es fácil de usar y transportar.

Si tienes poca paciencia esta no es tu cafetera, esta cafetera es para disfrutar del ritual completo del café. Tiene una capacidad para dos cafés, el café como he comentado es un café de filtro, tipo americano.

Yo caliento el agua y la dejo enfriar hasta unos 80 grados, mientras enfriar pongo los granos de café en el molinillo (una

★★★★★ **"Café excelente, ligero y práctico"**  
*Ivan Perez Garcia*

Perfecto para usar en casa o cuando te vas de camping. Aconsejable tener una hervidora de agua a mano para cada vez que quieras un café.

Yo uso 2 o 3 filtros para aplicar más presión y que salga aún más intenso el café. Y luego limpio los filtros y los reuso durante 1 mes o más. Los filtros que lleva te durarán para años.

Es increíble cómo algo tan sencillo puede sacar un café mucho más suave y

★★★★★ **"Genial cafetera"**  
*Paloma*

Está muy bien. Yo usaba hasta ahora una de esas cafeteras italianas de toda la vida, y desde que recibí la AeroPress se ha convertido en mi cafetera del día a día.

Ahora me queda traer un poco con los "ajustes" digamos, el tiempo de percolado, la cantidad de agua/café... pero si sigues las instrucciones del manual, sin complicaciones, te sale un café buenísimo, y eso que uso uno premolido típico. Esa será mi próxima compra, un molinillo para moler el café al

[Ver valoraciones de Amazon](#)

### Ayuda a mantener limpio el planeta

Cada minuto se fabrican alrededor de 39 000 cápsulas de café en todo el mundo y hasta 29 000 acaban en vertederos\*. Aunque se empiezan a hacer esfuerzos en el reciclado y segunda vida de las cápsulas la gran mayoría acaban alimentando montañas en vertederos, cápsulas desechadas prevalecerán durante décadas e incluso cientos de años alunas, hasta desaparecer.

AeroPress dura años y no generará residuos adicionales. Ahórrale al planeta más de 1000 cápsulas cada año. Además, no estar obligado a tener que comprar café de un tipo o marca determinado te da la oportunidad de elegir café ecológico y de comercio justo.

Disfruta del café mientras le haces un favor al medio ambiente.





# 7. Conclusiones y líneas de futuro

## 7.1. Conclusiones

### 7.1.1 Aprendizaje

El trabajo realizado me ha supuesto una gran fuente de aprendizaje desde diferentes puntos:

- He adquirido una base de conocimientos en React que me ha aportado un punto de partida suficiente para poder plantear y asumir desafíos complejos con esta herramienta.
- Al ser JavaScript el lenguaje principal, he podido profundizar mucho en el lenguaje y sobre todo en su desarrollo con EcmaScript 6. Ha sido una grata sorpresa dar con imageMagick buscando información para el proyecto y aprender a usarlo ya que es una herramienta de optimización de imágenes muy potente.
- Desde el punto de vista de diseño he asumido una base de conocimientos sobre figma y sistemas de diseño sobre la que continuar creciendo.
- Y a nivel de negocio cada vez me he convencido más de la viabilidad del proyecto a medida que desarrollaba e investigaba.

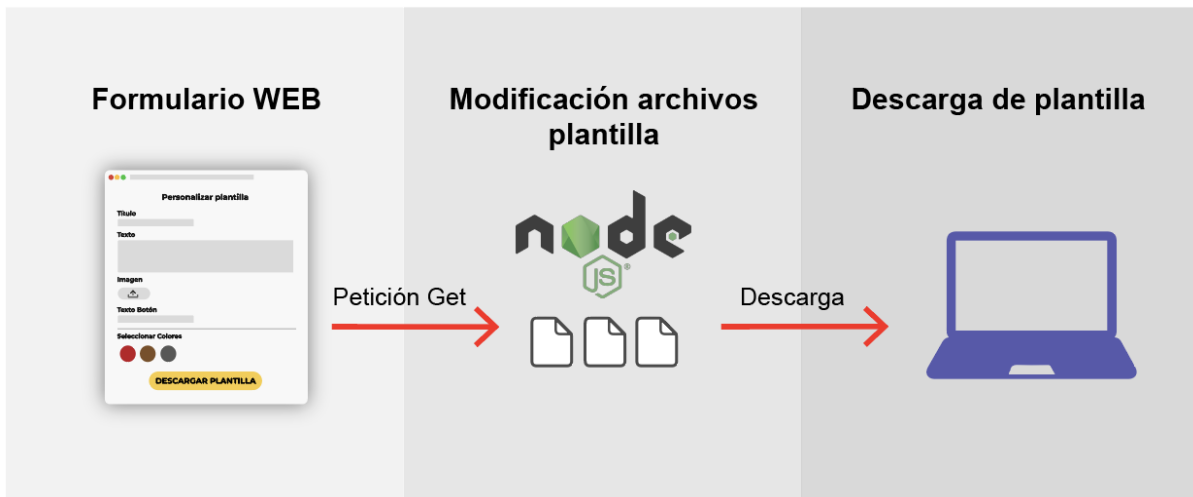
### 7.1.2 Asunción de objetivos

En mayor o menor medida se considera que se han asumido los diferentes objetivos planteados. Las inquietudes en el aprendizaje han sido cubiertas. Se han asumido las características de diseño y componentización del desarrollo. La plantilla web desarrollada permite modificarse e implementarse de manera rápida y ágil así como el sistema de diseño que permite crear diseños y propagar modificaciones rápidamente gracias a la librería creada. Se han brindado las herramientas que definían los objetivos planteados a nivel de negocio. Falta por ver, pero por suerte el tiempo lo mostrará, si el modelo económico en el que se sustenta la propuesta es viable.

## 7.2. Líneas de futuro

Finalizado el desarrollo del proyecto planteado, se ha creado un producto que facilita a muchas personas, con unos mínimos conocimientos tecnológicos, generar una página web en torno a un producto nicho en menos de una hora. Ahora bien, por sencillo que pueda ser modificar un JSON o una variable CSS, personas de sectores alejados al tecnológico o gente con poca facilidad para la tecnología pueden tener dificultades para crear su página web con Renite. Por este motivo el siguiente paso lógico para el proyecto es la generación automática de plantillas mediante formulario web.

Utilizando NodeJS, entorno de desarrollo *backend* para lenguaje JavaScript, se puede crear un programa que sobrescriba el archivo JSON con el contenido en base a una información que viene mediante petición GET de un formulario en la web.



Durante las muchas horas de desarrollo del proyecto ha dado tiempo a pensar en muchos componentes que se podrían haber incorporado a la plantilla de haber tenido más tiempo, y son candidatos a desarrollar en los siguientes meses. La siguiente lista son algunas de estas ideas:

### Módulo de vídeo

Pasándole como parámetro el enlace de un vídeo de YouTube se renderiza una caja *responsive* para reproducir el vídeo.

### Módulo de 3 columnas

Módulo de 3 columnas para presentar contenido en texto y una imagen.

### Galería de imágenes

Pasando como parámetro una array con imágenes se genera automáticamente mediante imageMagick una parrilla de miniaturas con un contenedor principal donde muestra la imagen seleccionada en grande.

### Automatización de reviews

Haciendo uso de la API de Amazon cargar el contenido de las reviews de Amazon de manera automatizada sin necesidad de introducir los datos en el documento JSON del contenido.

Por otro lado, y teniendo la plantilla React finalizada, es el momento de reciclar todo el esfuerzo realizado y generar nuevas páginas webs centradas en producto nicho, algunas ideas de líneas interesantes a explotar son:

### Cinturón vegano

Cinturón muy similar al cuero pero elaborado con caucho. La creciente conciencia de respeto animal y el atractivo estético del producto lo hacen un buen candidato como nicho a explotar. Ver producto en Amazon:

[https://www.amazon.es/Corkor-Cintur%C3%B3n-Corcho-Hebilla-Plateada/dp/B0773MZJGX/ref=sr\\_1\\_4\\_sspa?keywords=cinturon+vegano+hombre&qid=1651779448&srefix=cinturon+vegano%2Caps%2C112&sr=8-4-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExVEdlQkVFRTIxNU5EJmVuY3J5cHRIZElkPUewODk0MzMwMjQwRE0zSFBaUUIITSZlbnNyeXB0ZWRBZEIkPUewNDUxNTgxM1Q4QzdEM1dRRTTVYRyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=](https://www.amazon.es/Corkor-Cintur%C3%B3n-Corcho-Hebilla-Plateada/dp/B0773MZJGX/ref=sr_1_4_sspa?keywords=cinturon+vegano+hombre&qid=1651779448&srefix=cinturon+vegano%2Caps%2C112&sr=8-4-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExVEdlQkVFRTIxNU5EJmVuY3J5cHRIZElkPUewODk0MzMwMjQwRE0zSFBaUUIITSZlbnNyeXB0ZWRBZEIkPUewNDUxNTgxM1Q4QzdEM1dRRTTVYRyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=)

### **Escenografía para WarGames.**

Comunidades de *wargames* como Warhammer son sólidas y con un gran componente emocional a la hora de hacer las compras ya que es un hobby que genera mucha pasión. Ejemplo de producto en Amazon:

[https://www.amazon.es/Terrain-Tabletop-Miniatures-Paintable-EnderToys/dp/B077JHC4D9/ref=sr\\_1\\_4\\_2?keywords=escenografia+warhammer&qid=1651779492&srefix=escenografia+wa%2Caps%2C108&sr=8-42](https://www.amazon.es/Terrain-Tabletop-Miniatures-Paintable-EnderToys/dp/B077JHC4D9/ref=sr_1_4_2?keywords=escenografia+warhammer&qid=1651779492&srefix=escenografia+wa%2Caps%2C108&sr=8-42)

### **Comparador de kits para hacer cerveza artesanal.**

En este caso, repitiendo diferentes componentes se puede mutar la página para que muestre diferentes opciones y de esta manera no depender de un único producto. En este caso el valor del producto reside en que es un producto de iniciación a hobby y un top de ventas para regalo por lo que el componente económico no es el de más valor. Ejemplo de producto en Amazon:

[https://www.amazon.es/Cervezan%C3%ADa-elaboraci%C3%B3n-cerveza-Elabora-litros/dp/B01HPU CYME/ref=sr\\_1\\_6?keywords=kit%2Bhacer%2Bcerveza&qid=1651779694&srefix=kit%2Bhacer%2Bcer%2Caps%2C116&sr=8-6&th=1](https://www.amazon.es/Cervezan%C3%ADa-elaboraci%C3%B3n-cerveza-Elabora-litros/dp/B01HPU CYME/ref=sr_1_6?keywords=kit%2Bhacer%2Bcerveza&qid=1651779694&srefix=kit%2Bhacer%2Bcer%2Caps%2C116&sr=8-6&th=1)

# Bibliografía

1. Wikipedia contributors. (2022, 17 abril). Lean startup. Wikipedia. Recuperado 19 de abril de 2022, de [https://en.wikipedia.org/wiki/Lean\\_startup](https://en.wikipedia.org/wiki/Lean_startup)
2. Ferriss, T. (2012). Semana Laboral De 4 Horas. Rba.
3. Godin, S. (2019). La vaca púrpura: Diferénciate Para Transformar Tu Negocio (1.ª ed.). Ediciones Culturales Paidós S. A. De C. V.
4. Meta Open Source. (s. f.). React – A JavaScript library for building user interfaces. React. Recuperado 5 de mayo de 2022, de <https://reactjs.org/>
5. Initiative, W. W. A. (s. f.-b). WCAG 2 Overview. Web Accessibility Initiative (WAI). Recuperado 20 de abril de 2022, de <https://www.w3.org/WAI/standards-guidelines/wcag/>
6. Make the Web Faster |. (s. f.). Google Developers. Recuperado 20 de abril de 2022, de <https://developers.google.com/speed>
7. W3Schools. (s. f.). JavaScript ES6. Recuperado 22 de abril de 2022, de [https://www.w3schools.com/js/js\\_es6.asp](https://www.w3schools.com/js/js_es6.asp)
8. Node Foundation. (2019, 1 octubre). About. OpenJS Foundation. Recuperado 22 de abril de 2022, de <https://openjsf.org/about/>
9. Hámori, F. (2022, 6 abril). The History of React.js on a Timeline. RisingStack Engineering. Recuperado 22 de abril de 2022, de <https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>
10. Wikipedia contributors. (2022b, abril 21). Angular (web framework). Wikipedia. Recuperado 22 de abril de 2022, de [https://en.wikipedia.org/wiki/Angular\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))
11. Wikipedia contributors. (2022c, abril 22). JSON. Wikipedia. Recuperado 22 de abril de 2022, de <https://en.wikipedia.org/wiki/JSON>
12. Initiative, W. W. A. (s. f.). W3C Accessibility Standards Overview. Web Accessibility Initiative (WAI). Recuperado 25 de marzo de 2022, de <https://www.w3.org/WAI/standards-guidelines/>
13. Web Accessibility In Mind. (s. f.). WebAIM: Contrast Checker. <https://Webaim.Org>. Recuperado 29 de marzo de 2022, de <https://webaim.org/resources/contrastchecker/>
14. Google. (2018, 7 marzo). Optimize images | PageSpeed insights |. Google Developers. Recuperado 29 de marzo de 2022, de <https://developers.google.com/speed/docs/insights/OptimizeImages>
15. Optimize Images | PageSpeed Insights |. (2021, 9 diciembre). Google Developers. Recuperado 30 de abril de 2022, de <https://developers.google.com/speed/docs/insights/OptimizeImages>
16. Wikipedia contributors. (2022d, abril 22). Webp. Wikipedia. Recuperado 30 de abril de 2022, de <https://en.wikipedia.org/wiki/Webp>
17. Lesiński, K. (s. f.). ImageOptim — better Save for Web. ImageOptim makes images load faster. Recuperado 30 de abril de 2022, de <https://imageoptim.com/mac>
18. Kraken.io. (s. f.). Kraken.io Image Optimizer. Recuperado 30 de abril de 2022, de <https://kraken.io/>
19. ImageMagick Studio LLC. (s. f.). ImageMagick. ImageMagick. Recuperado 30 de abril de 2022, de <https://imagemagick.org>
20. Institute for Disability Research, Policy, and Practice. (s. f.). WebAIM: Web Accessibility In Mind. Web AIM. Recuperado 18 de mayo de 2022, de <https://webaim.org/>