



Aptugo

CSS Básico Web

Guía de estudio

Index

Hablemos de Páginas Web	3
La estrella de los estilos: CSS	3
Reglas y sintaxis	5
Stylesheets en Aptugo	7
Vinculando estilos y elementos	10
Conclusión	16

Hablemos de Páginas Web

En mayor o menor medida, todos tenemos una idea de que es una página web. La asociamos con Internet, texto, video e imágenes.

Todo eso que hace que una página se vea bien y sea agradable de navegar, está relacionado directamente con el diseño y los estilos que se aplican a sus elementos.

¿Quién se encarga de que todo se vea bien? Ahí es donde entra en juego el protagonista de esta clase: **CSS**.

La estrella de los estilos: CSS

CSS es el lenguaje que usamos para darle estilo a los elementos visuales. Permite cambiar colores, fuentes, tamaños, posiciones, márgenes, fondos, bordes, y mucho más. Gracias al CSS podemos hacer que una página o juego se vea profesional, coherente y visualmente atractiva.

En esta clase vamos a comenzar a explorar cómo funciona este lenguaje, cómo se aplica dentro de Aptugo, y qué podemos modificar con él para que cada elemento de nuestra interfaz refleje nuestro estilo y nuestras ideas.

CSS - Significa **Cascading Stylesheets** (hojas de estilo en cascada). Este lenguaje permite crear reglas para intervenir los diferentes elementos de una aplicación, cambiando sus características visuales.

Cascading - La cascada permite establecer reglas con cierto orden de importancia, **permitiendo que se apliquen y sobreescriban sobre otros elementos (a modo de cascada)**. En caso que haya reglas incompatibles, ésta permite establecer cuál debe aplicarse, por ejemplo: en caso que haya dos sentencias de tipografía, se impondrá la última escrita.

Stylesheets - Los archivos CSS son llamados '**Stylesheets**' (hojas de estilo), tienen extensión .css, y permiten establecer reglas de estilos en una hoja en particular. Así, independientemente de las páginas que tengamos, podemos establecer todos los estilos de en una misma hoja e ir asignándolos según corresponda.

El CSS es una de las herramientas que cubre las limitaciones de los elementos, y está principalmente asociada a la propiedad de herencia descrita anteriormente. Hace uso de ella para establecer propiedades en cascada.

SCSS	CSS
<pre>1 section { 2 height: 100px; 3 width: 100px; 4 5 .class-one { 6 height: 50px; 7 width: 50px; 8 9 .button { 10 color: #074e68; 11 } 12 } 13 }</pre>	<pre>1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 section .class-one { 7 height: 50px; 8 width: 50px; 9 } 10 11 section .class-one .button { 12 color: #074e68; 13 }</pre>



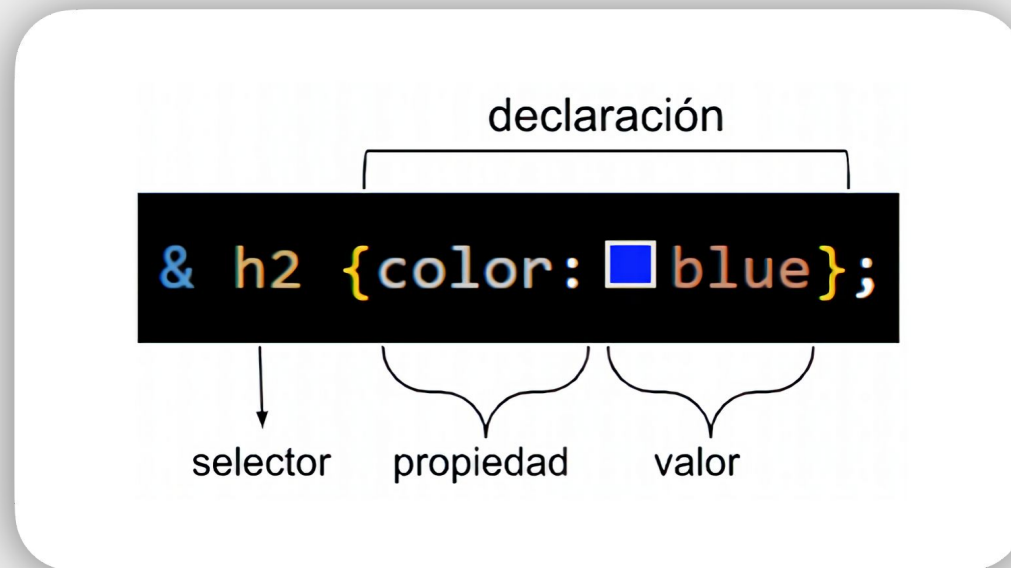
Es importante destacar que, en Aptugo se emplean hojas de estilo **.scss**. El mismo es un metalenguaje de CSS, vinculado a SASS (Hojas de estilo sintácticamente impresionantes) el cual se traduce a CSS, confiriéndole la capacidad de incluir estructuras y efectos más complejos y dinámicos.

Reglas y Sintaxis

Un documento CSS está compuesto por una colección de Reglas. Las reglas, a su vez, se componen de un selector, el cual es una palabra que hace referencia al elemento del árbol a la cual se le aplicarán las propiedades, y contiene una serie de declaraciones asociadas a él. Sucesivamente, cada declaración hace referencia a una propiedad.

El siguiente ejemplo se corresponde a un documento CSS, el cual afectará a una sección o Div que contendrá dos elementos textos, con formato h2 y h3.

<pre>& .aboutInfo { display: flex; flex-direction: column; align-items: center; padding: 0 0 100px; & h2 { font-style: normal; font-weight: 300; font-size: 45px; line-height: 132.8%; text-align: center; margin: auto; color: #0D1D2A; padding: 62px 0 18px; } & h3 { font-style: normal; font-weight: 300; font-size: 20px; line-height: 132.4%; text-align: center; margin: auto; color: #7C7C7C; padding: 0 0 80px; } }</pre>	<p>Selector padre</p> <p>Reglas del selector Padre</p> <p>Selector Hijo 1</p> <p>Reglas del selector hijo 1</p> <p>Selector Hijo 2</p> <p>Reglas del selector hijo 2</p>
--	--



Veamos más detalladamente cómo se escribe una regla de CSS.

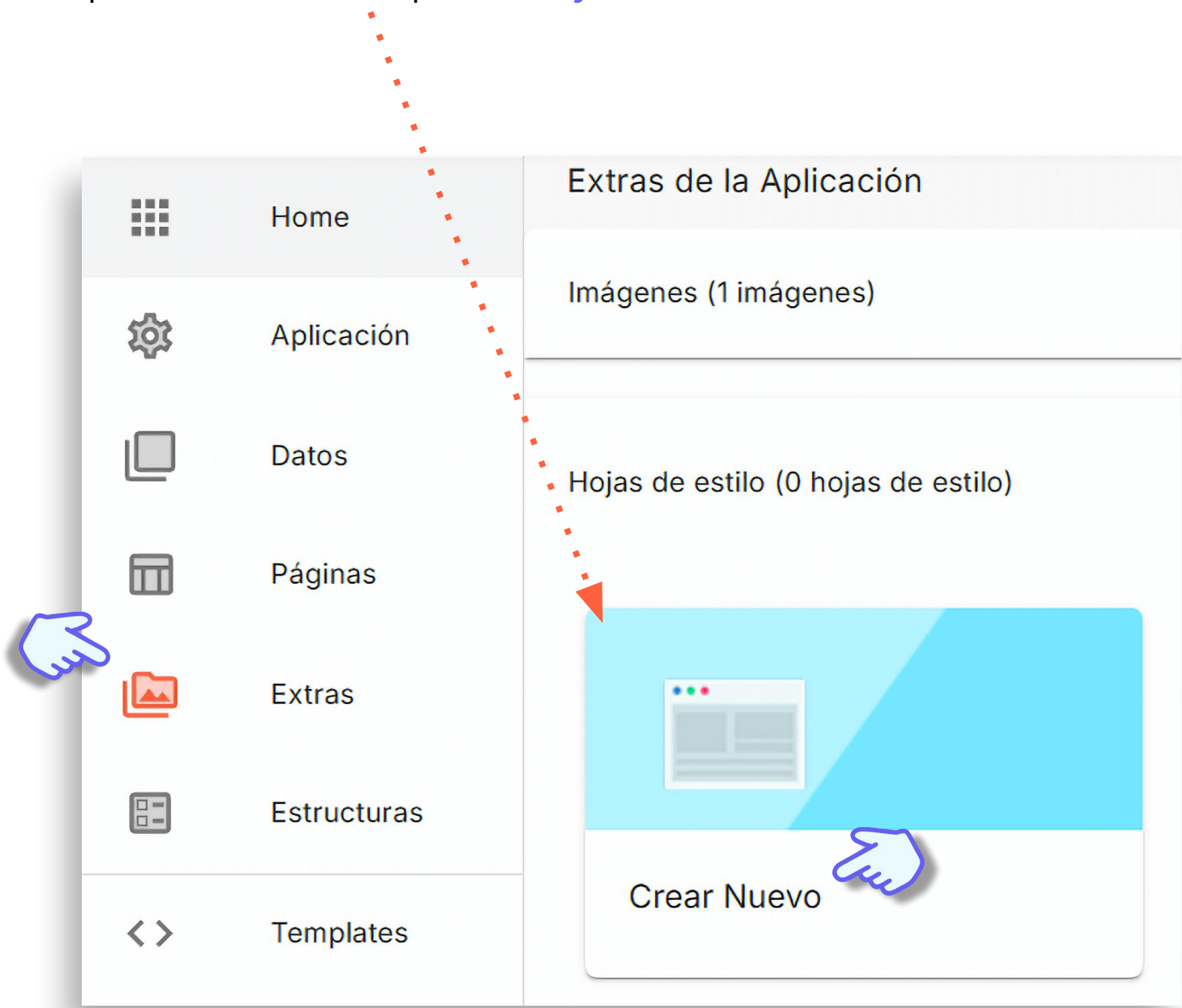
- Comenzamos con un selector, el cual indica el elemento a sentenciar (en este caso, el estilo de un texto h2). El **ampersand** ("&") **del inicio es una característica propia de los archivos .scss** (la combinaciones entre CSS y SASS que describimos en la sección anterior, y son usados en Aptugo). Dicho carácter permite indicar que el selector declarado es hijo de otro anterior, sin necesidad de reescribirlo.

En este caso el **selector h2 pertenece dentro de un elemento contenedor (del tipo div, container, card...)**. Como podemos observar en la imagen final de la página anterior, dentro del selector ".aboutInfo" .

- Toda **declaración** comienza y **termina con llaves**.
- **Dentro** de la declaración "h2" **escribimos la propiedad** (color, tipografía, padding, margin...), seguido de dos puntos (":") .
- Posteriormente, a la propiedad establecida le determinamos un valor. En caso de ser la propiedad "**color**", el valor corresponderá al color elegido, el cual puede escribirse de manera literal como por medio de formato **Hex en RGB**, como observamos en la imagen superior. De ser propiedades mensurables (asociadas a tamaños), la misma consistirá en **un número y una unidad (px, rem, %)**.
- **Finalizamos** la regla mediante **un punto y coma (";")** .

Stylesheets en Aptugo

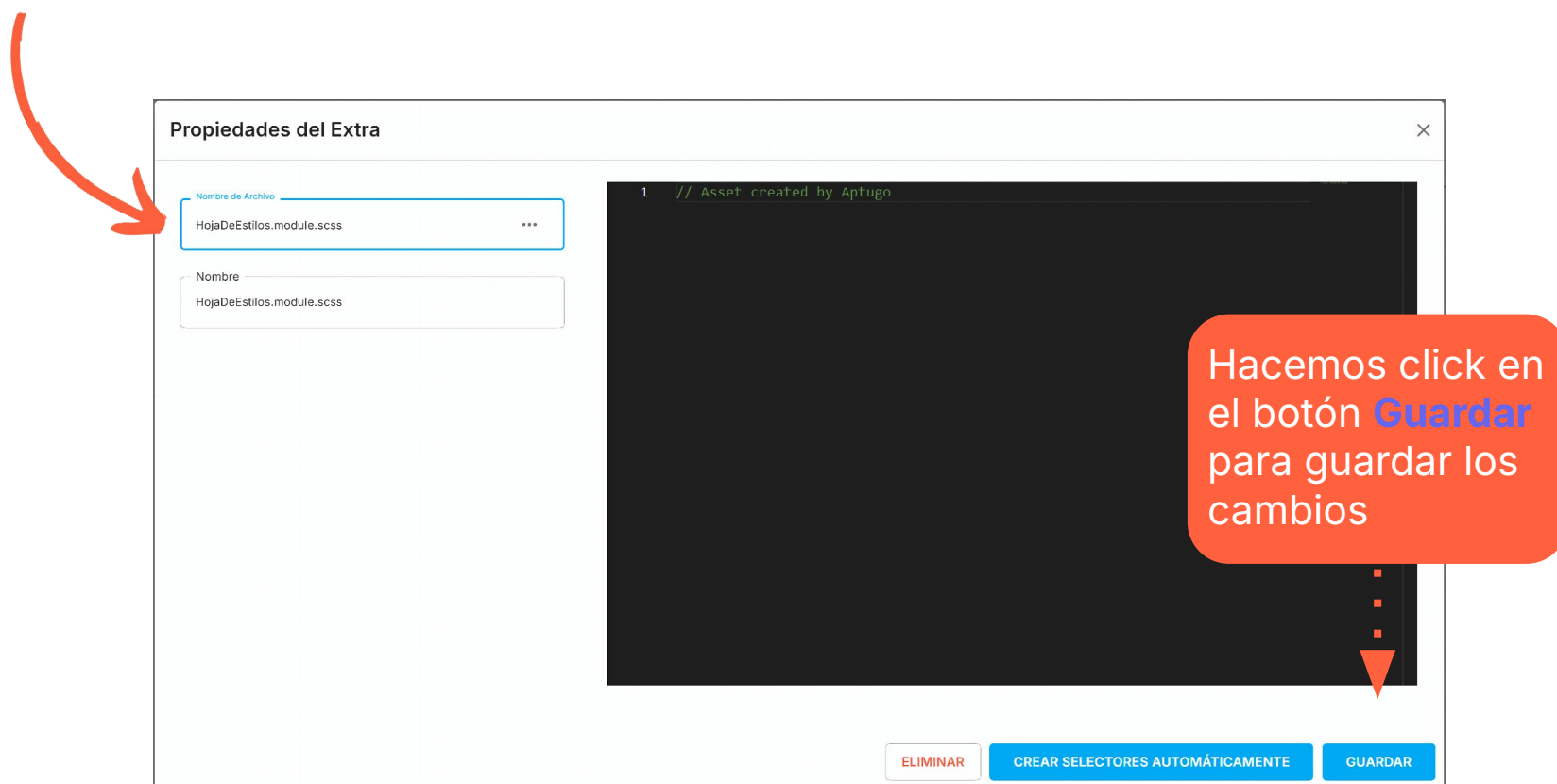
Las **stylesheets** en Aptugo se pueden visualizar en la sección **Assets**, específicamente en el apartado **Stylesheets**



Para poder utilizarlas, se debe hacer click en **create new**. Una vez hecho esto, se nos devolverá la pantalla abajo representada.

A la izquierda, encontramos los campos para nombrar nuestra hoja de estilos. Es importante que cambiemos sus nombres y que ambas lleven el mismo nombre, delimitado de la siguiente manera:

hojaDeEstilos.module.scss

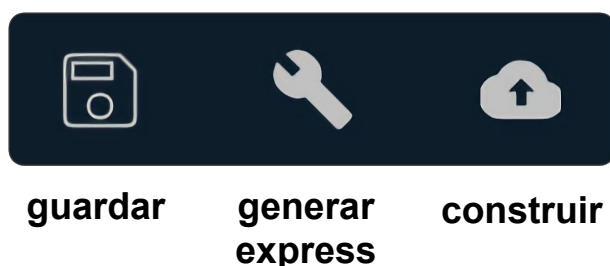


El siguiente paso es escribir **las diferentes reglas y sentencias**, lo cual puede realizarse en la sección derecha de la pantalla.



Recordemos hacer click en **"guardar"** con cierta frecuencia, ya que la sección no posee autoguardado.

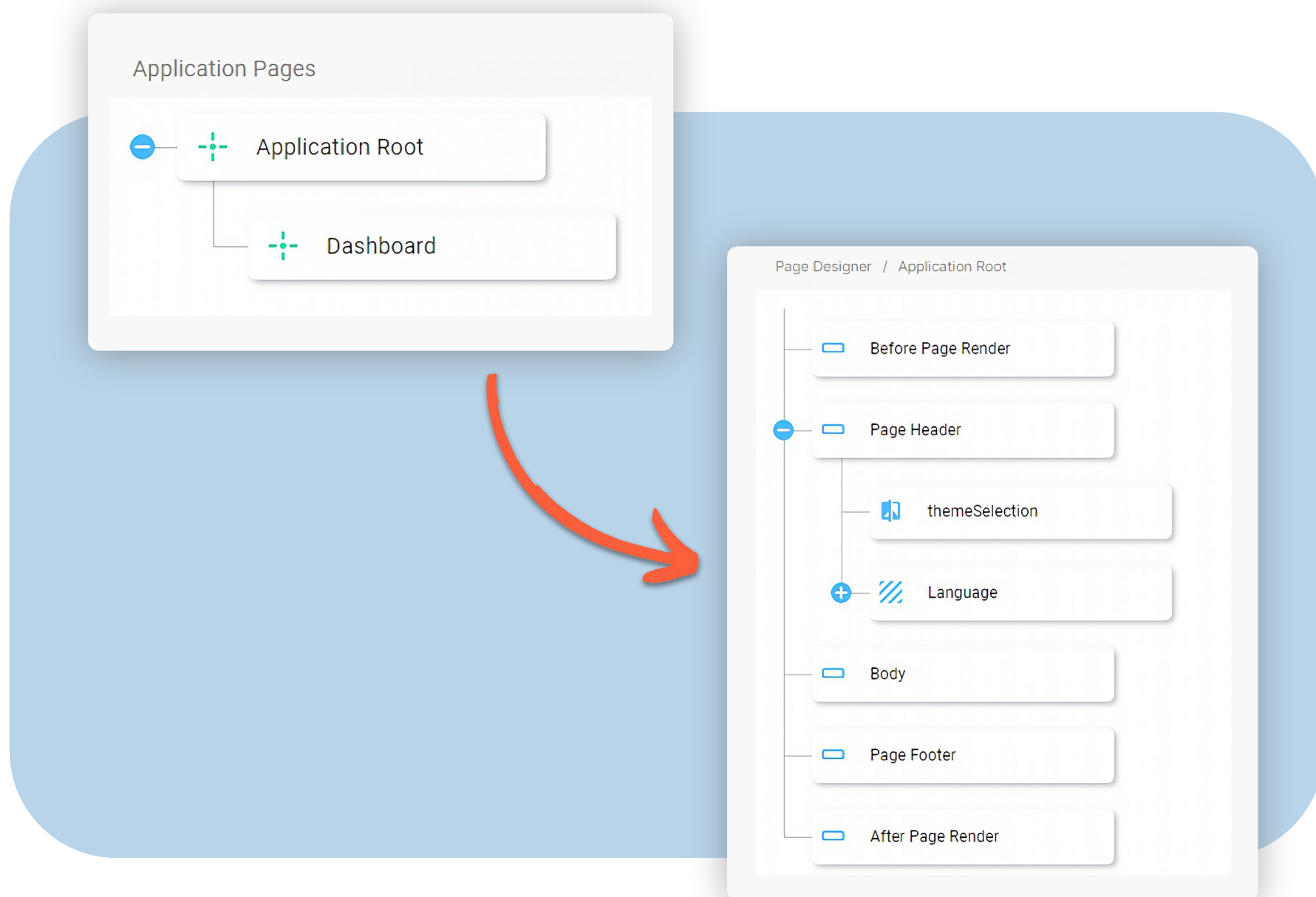
Una vez escrito todo el código y clickeado **"guardar"**, resta hacer click en cualquier punto externo de la ventana, **guardar y construir** para que los cambios se compilen en la aplicación. Recordamos los botones indicados para ello:



Vinculando estilos y elementos

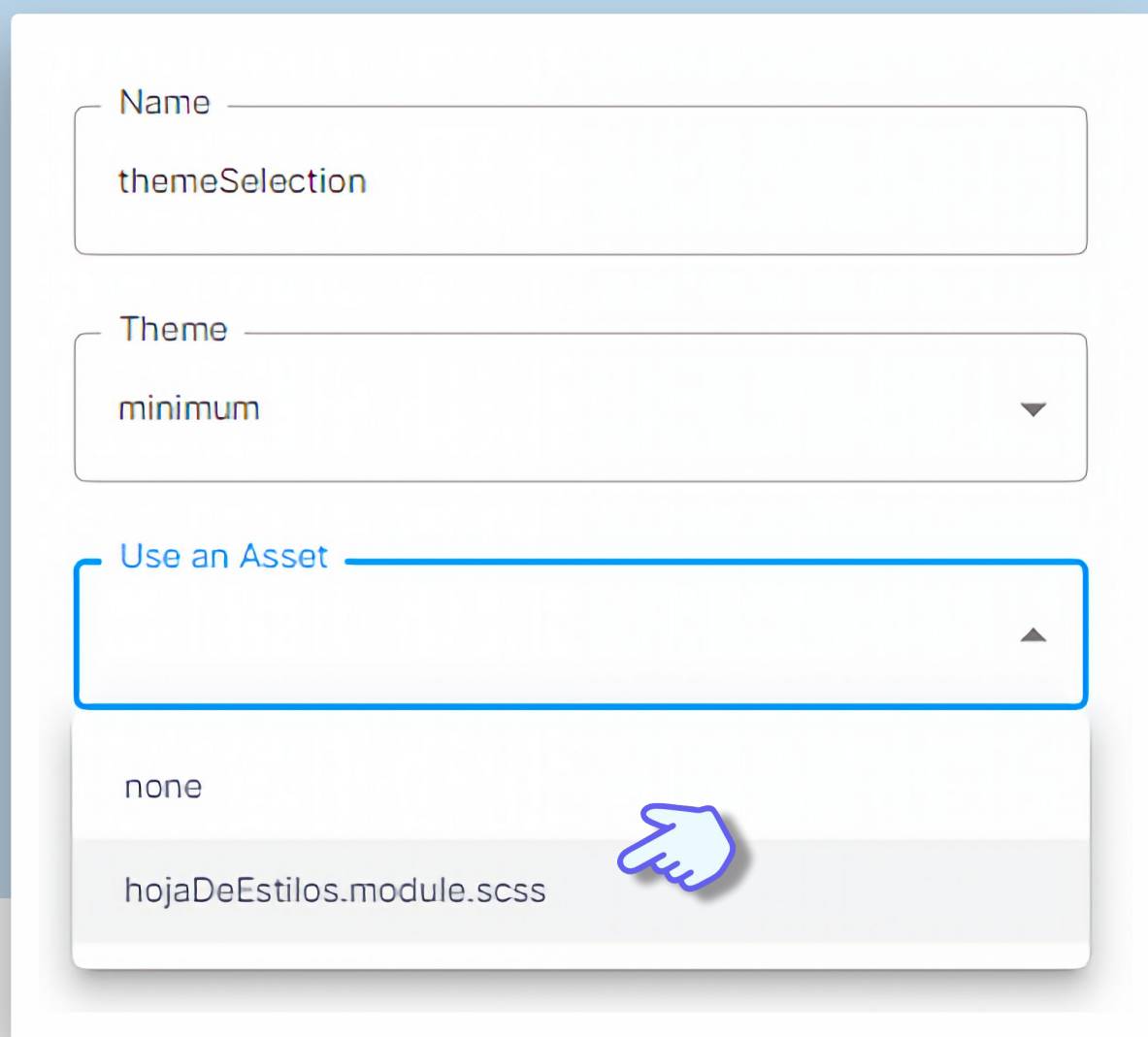
Nada de lo que escribamos en el código se reflejará en los elementos **si no le decimos Aptugo que lo haga**. Para lograrlo, se deben seguir una serie de pasos:

En primer lugar, seleccionamos la hoja de estilos: **nos dirigimos a la sección Pages de Aptugo e ingresamos en Application Root**. Dentro de **Page Header** encontraremos el elemento **themeSelection**.



Al hacer click en el elemento, se nos presentarán las propiedades en la **barra lateral derecha**.

Por defecto, Aptugo nos provee de un tema o estilo llamado minimum. **Para poder usar nuestra propia hoja de estilos**, debemos seleccionarla del campo **Use an Asset** que se encuentra debajo. Una vez hecho esto, **guardamos y compilamos** por medio de **construir (build)**.

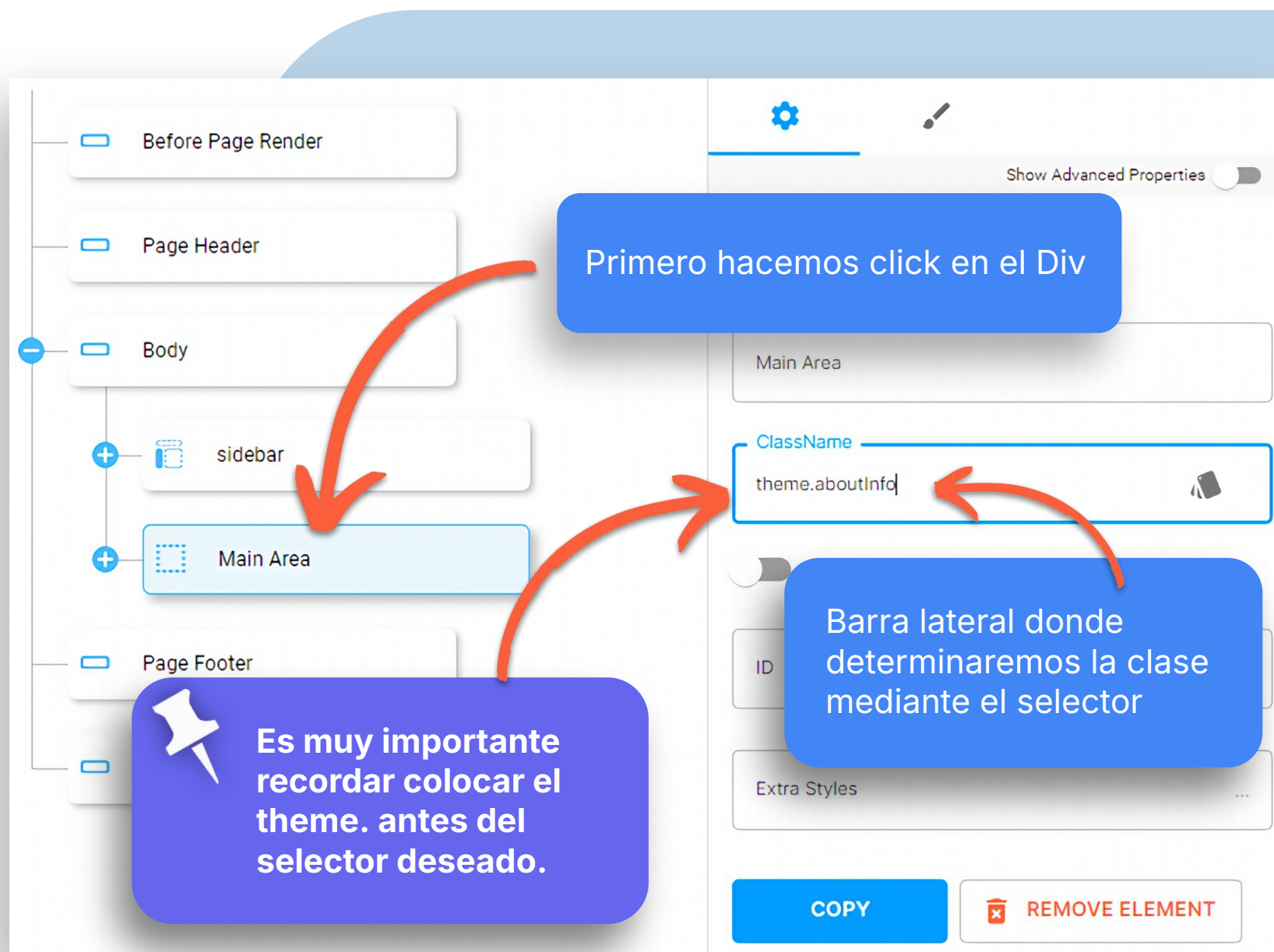


The image shows a configuration window with three input fields. The first field, labeled 'Name', contains the text 'themeSelection'. The second field, labeled 'Theme', contains the text 'minimum' and has a downward arrow on the right. The third field, labeled 'Use an Asset', is highlighted with a blue border and has an upward arrow on the right. Below this field is a dropdown menu with two options: 'none' and 'hojaDeEstilos.module.scss'. A blue hand icon is pointing at the 'hojaDeEstilos.module.scss' option, indicating it is being selected.

En las imágenes anteriores, la **stylesheet** creada recibió el nombre de **"hojaDeEstilos"**. Al seleccionarla, la misma queda escrita dentro del campo.

Seguidamente, debemos asignar las clases a los elementos. Podemos entenderlas como los diferentes estilos que nosotros definimos e identificamos por medio de los selectores. Cada elemento dentro de Aptugo que puede ser modificado cuenta con un campo llamado `ClassName` donde podemos asignarles el estilo. Debemos escribir **theme.** y el nombre del selector que hayamos incluido en la stylesheet, tal cual lo escribimos, respetando las mayúsculas o cualquier otro identificador.

A modo de ejemplo, el **.aboutInfo** que encontramos en las secciones anteriores, puede verse de esta forma:



Un poco más de cerca... 👍

Repitiendo lo anterior con cada elemento, ¡el resto es código!

A continuación, te dejamos algunas apreciaciones y propiedades útiles que te ayudarán a realizar las diferentes reglas de tu proyecto.

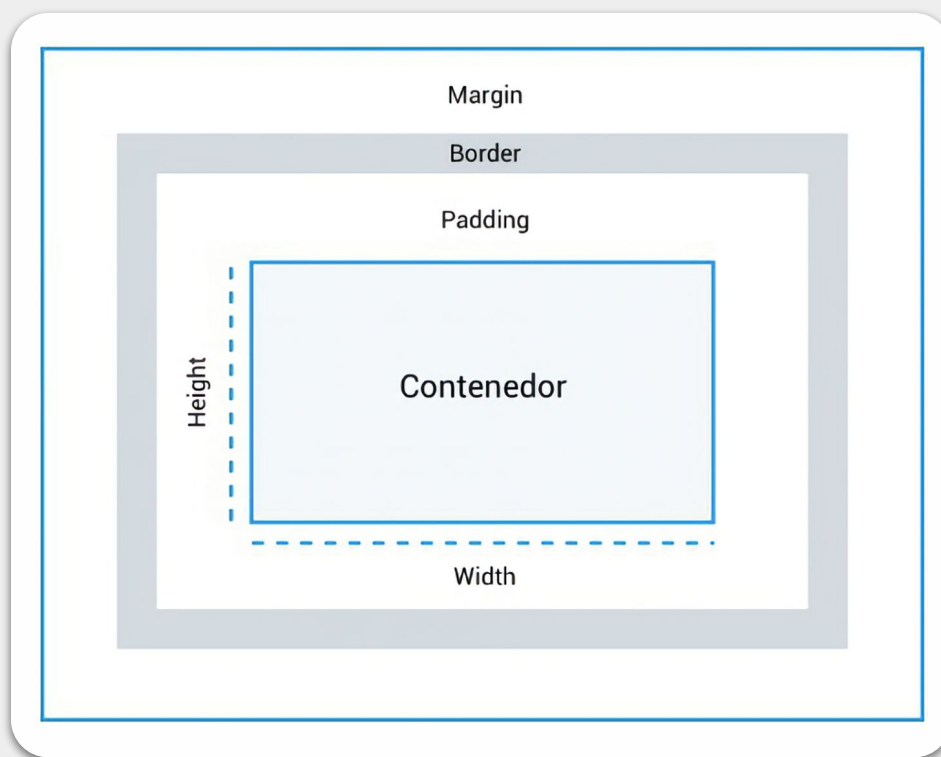
Modelo de caja

El modelo de caja permite dimensionar las páginas web como un conjunto de cajas que contienen elementos. Así, podemos delimitar los diferentes bordes y espacios que hay entre ellos, establecemos un perímetro en el cual habitan y se desempeñan.

Cada caja tiene un área del contenido, y podemos definir las siguientes:

- **Content (contenido):** El área donde se muestra el contenido, cuyo tamaño puede cambiarse.
- **Padding (relleno):** Espacio entre el contenido y el borde.
- **Border (borde):** Espacio entre el padding y el margin.
- **Margin (margen):** Permite especificar el espacio entre los elementos.

A su vez, las áreas se dividen en 4 sectores: top (superior), bottom (inferior), left (izquierda) y right (derecha) lo cual permite establecer tamaños idénticos o diferenciados según sea necesario.



Propiedades útiles

- **Width (ancho):** Establece el ancho del elemento.
- **Height (alto):** Determina el alto del elemento.
- **Margin (margen):** Modifica el espacio entre los elementos.
- **Padding (relleno):** Establece cada uno de los rellenos horizontales y verticales de un elemento.
- **Font-family (familia de fuentes):** Permite determinar la familia tipográfica.

Nota: Para poder usar diferentes fuentes, la primera línea de la stylesheet debe poseer un **import** de las tipografías. Por ejemplo, encontramos aquí el import de una tipografía de **Google Fonts**:

```
@import url('https://fonts.googleapis.com/css2?family=Inter&display=swap');
```

- **Font-size (tamaño de la fuente):** determina el tamaño de fuente (px, em, rem, %)
- **Font-weight (grosor de la fuente):** permite establecer el peso de la fuente (normal, bold, light)
- **Font-style (estilo de la fuente):** determina si la fuente es normal (normal) o itálica (italic)
- **Color:** permite colorear.
- **Text-align (alineación de texto):** nos permite ubicar el texto en una posición.
- **Background-color (color del fondo):** establece color de fondo.

Conclusión

El estilo es un aspecto muy importante a la hora de diseñar un proyecto de software. Puede potenciar sus funcionalidades, transmitir significado e identidad. **Es la forma en la cual se presentan los datos e interactúa directamente con el usuario.** Mediante diferentes sentencias y reglas del lenguaje CSS, podemos aportar esa cuota creativa a nuestra aplicación, modificando las características visuales de textos, imágenes y estructuras.

Es crucial ser organizado con los diferentes niveles y selectores del lenguaje, asignar adecuadamente las clases a los elementos correspondientes, e ir poco a poco aprendiendo las propiedades y valores que más se ajustan a nuestras necesidades. El mundo CSS es enorme, por lo que requiere de tiempo para investigarlo y experimentarlo.

El estilo puede ser el **factor diferencial en el éxito de una página**, especialmente en proyectos más visuales como portafolios o catálogos. Su desarrollo adecuado brinda prolijidad y nivel al proyecto. Es recomendable no descuidarlo e ir mejorándolo de manera escalable, como a los aspectos funcionales de toda aplicación.

Como documentación extra, les dejamos algunas páginas webs con propiedades CSS para que sigan investigando:



<https://css-tricks.com/almanac/properties/>

<https://www.quackit.com/css/properties/>

<https://www.w3schools.com/css/>

<https://developer.mozilla.org/es/docs/Web/CSS>