



Aptugo

Elementos Básicos Web

Guía de estudio

Index

Introducción	3
Front-End	4
¿Qué es Front-End?	4
Inspect	5
Árbol de Navegación	6
Páginas	7
Drag & Drop	11
Elementos	12
Sidebar	12
Nav Link	13
Autolinking	13
Div	14
Typography	14
Text	16
Container	18
Simple Link	19
Image	21
Categorías	22
Basic	22
Experimental	22
Geo	23
Interact	23
Layouts	24
Material-UI	25
Programming	25
Elementos Árbol de Navegación UX/UI	26
Maquetado	29
Conclusión	30

Introducción

Cuando comenzamos a programar un proyecto de software y entramos en la etapa **Código (Coding)**, es aquí donde se escribe todo el código que hará realidad las ideas y funciones que pensamos durante la planificación.

Para poder organizar de manera eficiente el trabajo y lograr que cada parte del proyecto cumpla su función correctamente, el desarrollo de software se divide en dos grandes capas, **Back End y Front End**.

Esta separación tiene un motivo muy claro, permitir que cada parte del sistema se encargue de una tarea específica y pueda evolucionar o modificarse sin que todo el proyecto se vea afectado. En otras palabras, estas dos capas existen para hacer que nuestros desarrollos sean **más organizados, escalables y más fáciles de mantener**. Esto significa que si en el futuro queremos cambiar la apariencia o agregar nuevas funciones, podremos hacerlo sin necesidad de romper o rehacer todo lo que ya funciona.

El **Back End** es la parte invisible para el usuario, pero esencial para que todo funcione. Aquí es donde se gestiona la lógica del sistema, el almacenamiento de datos y la comunicación entre servidores. Aunque en este curso no profundizaremos en esa área, es importante que entendamos su existencia, ya que todo proyecto bien construido necesita un Back End que garantice que la información viaja y se procesa de forma correcta.

A partir de este momento, entraremos en las capas visuales de la aplicación, aquellas que interaccionan directamente con quien las utiliza. Hablamos de textos, imágenes, colores, estilos, elementos dinámicos como carruseles y pop-ups. Nos referimos al **Front End** de nuestra aplicación.



Es importante destacar que puede existir Front-End sin Back-End, más no a la inversa. Toda base de datos necesita de una plataforma visual en la cual proyectarse y ser manipulada, por lo que las siguientes lecciones serán de gran utilidad para garantizar el empleo y funcionalidad del proyecto.

Front-End

¿Qué es el Front-End?

El Front-End es la parte del desarrollo que involucra los elementos visuales que vemos a través de nuestro navegador web.

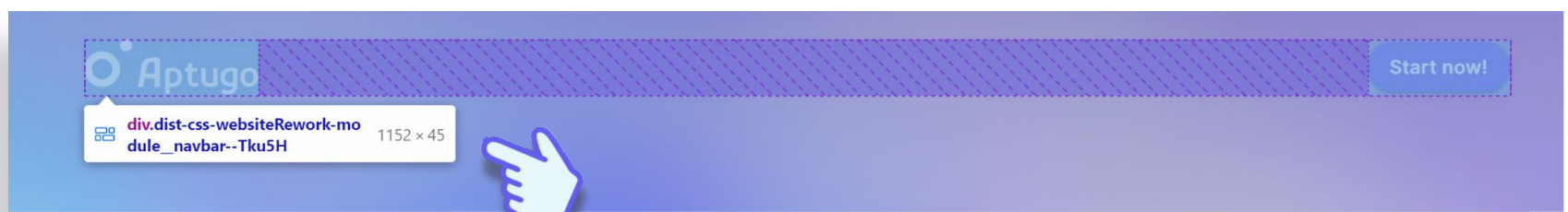
El desarrollo Front-End no solo se relaciona con que una aplicación o un sitio se vean estéticos, sino también con todo lo que se refiere a la experiencia del usuario. Es en esta capa donde se diseñan y construyen las pantallas, botones y animaciones que forman parte de la experiencia visual de nuestra aplicación..

Podemos destacar dentro de Front-End los siguientes puntos fundamentales:

1. Páginas
2. Elementos
3. Maquetación
4. CSS

A medida que avancen las lecciones, iremos profundizando en cada uno de ellos, con el fin de utilizarlos en pos de nuestro propio proyecto.

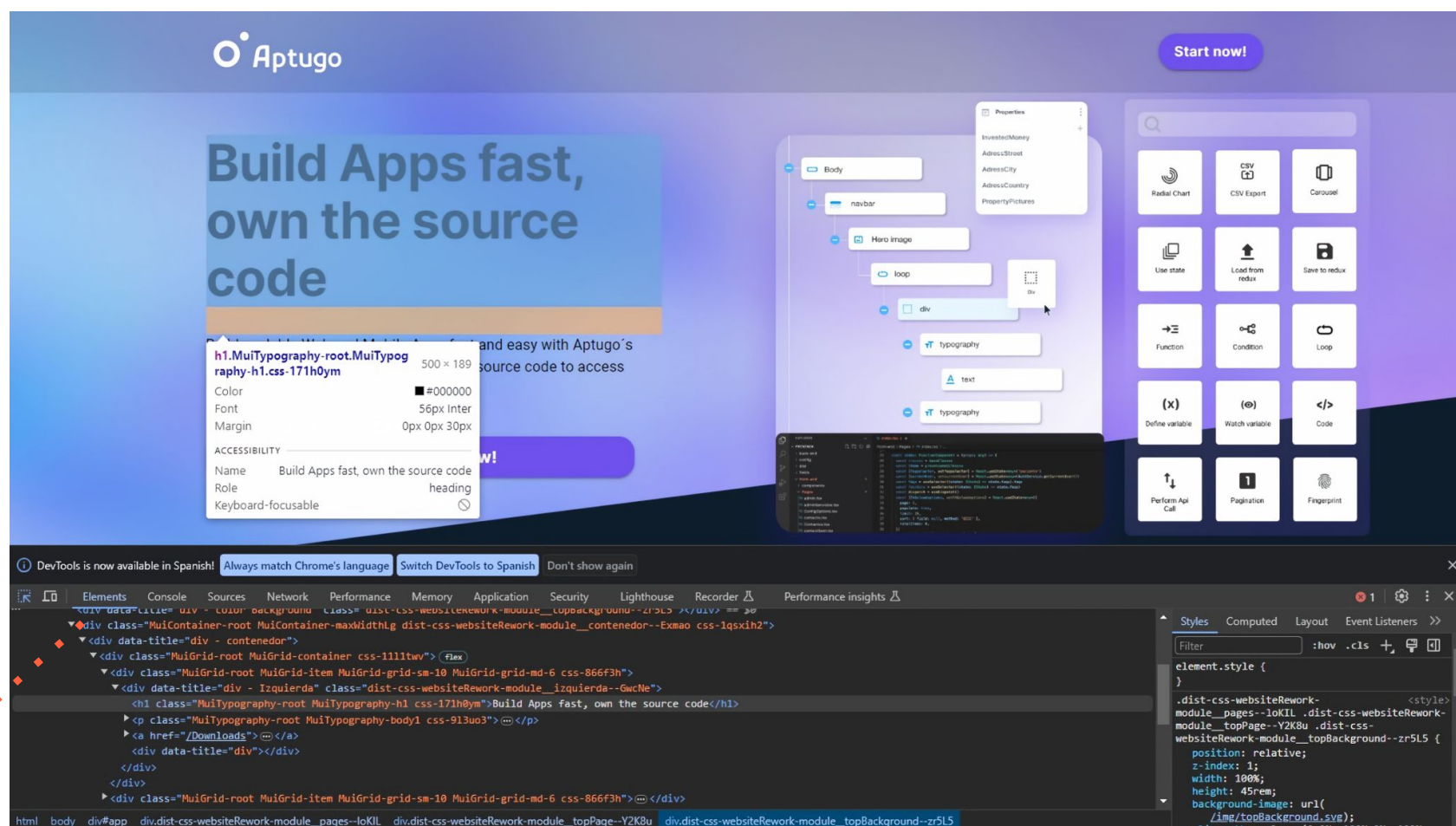
Tomemos como ejemplo el sitio web de Aptugo y comencemos por inspeccionar una pequeña parte: el área de la barra de navegación o "Navbar".



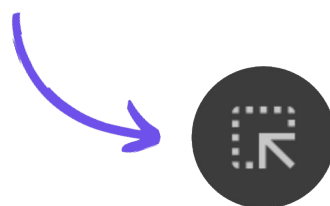
A partir de este ejemplo distinguiremos distintos componentes que son parte de la utilización cotidiana de un desarrollador web Front-End.

Inspect

La mayoría de los navegadores web incluyen una ventana desplegable que otorga información más detallada sobre lo que está sucediendo al cargarse una página web. Esta es una herramienta sencilla que usan los programadores y que nos da indicios de cómo está armada una aplicación.

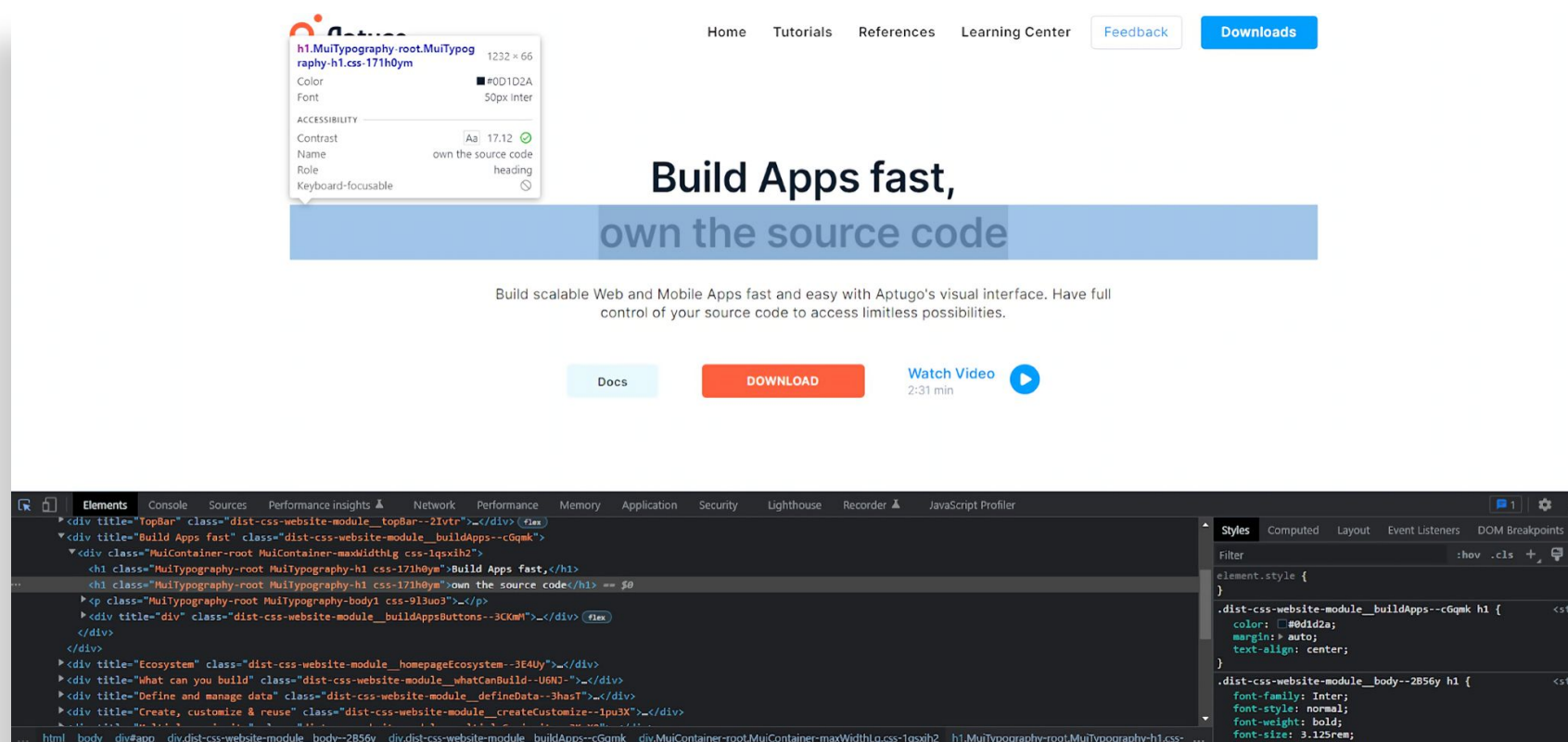


En el caso del navegador web Chrome al ejecutar Ctrl+Shift+I o F12 o haciendo click derecho, opción Inspeccionar se despliega "DevTools" y se abrirá en la pestaña **Elements**, con la cual haciendo click en el ícono de la izquierda:



"Select an element in the page to inspect it" es posible reconocer qué tipo de elemento visual se muestra en la página que estemos navegando.

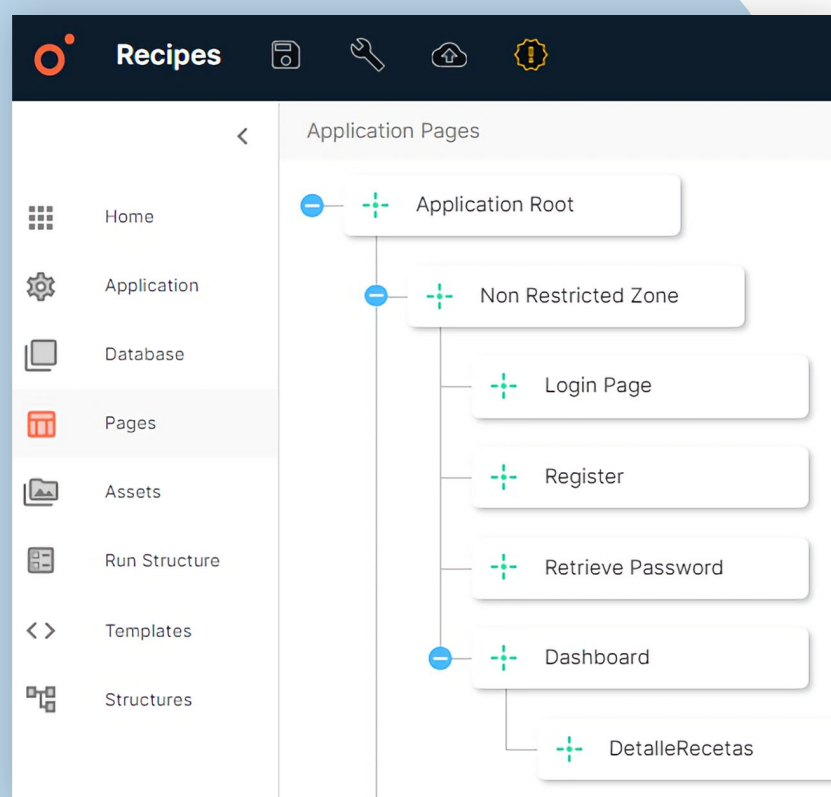
Nuevamente, si consideramos el sitio web de Aptugo y seleccionamos el texto **"own the source code"** encontramos un elemento de tipo texto que estudiaremos en profundidad más adelante. **Podemos replicar este ejercicio en cualquier página web.**



Árbol de Navegación

En Aptugo, podemos abordar y visualizar el desarrollo Front-End a través de un árbol de navegación.

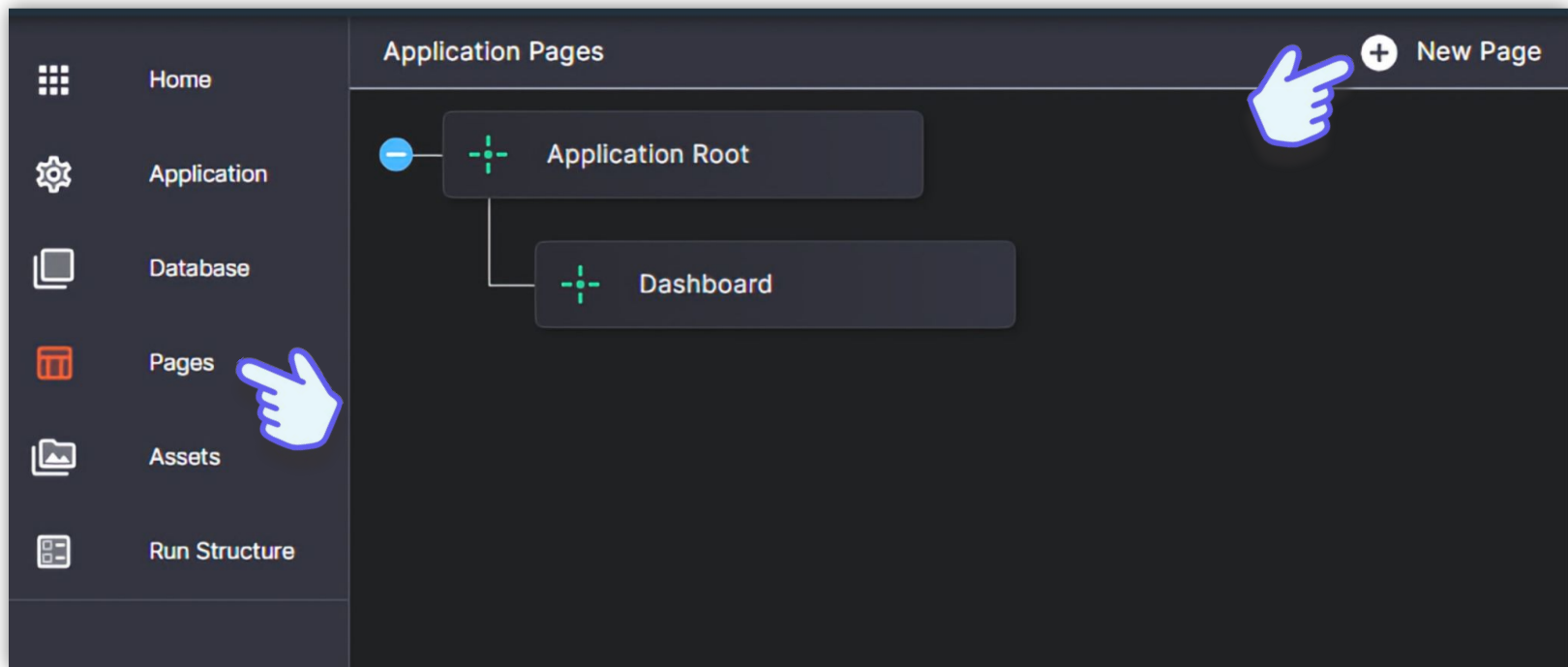
El **árbol de navegación de Aptugo** se accede por medio de la opción **Pages** desde donde se podrá desplegar cada uno de los componentes ya sean Páginas o Elementos.



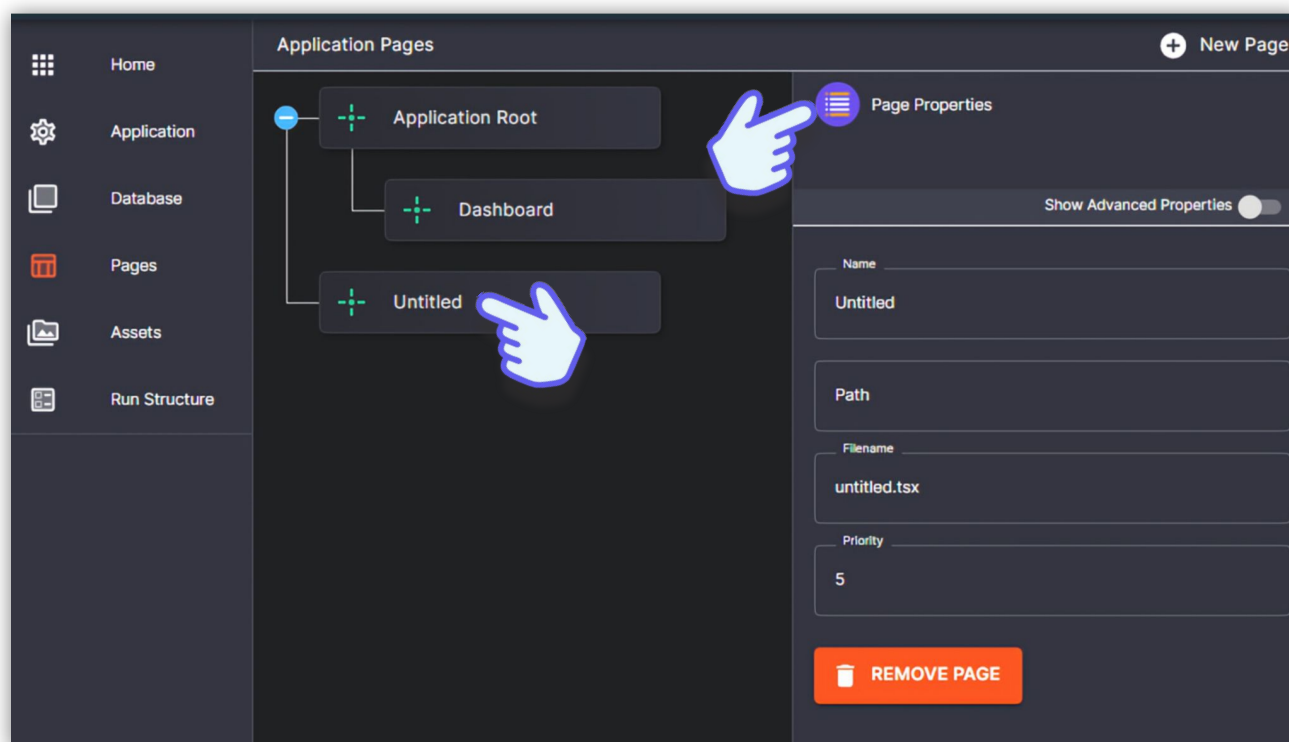
Páginas

Cuando creamos un proyecto nuevo, **Aptugo** nos genera automáticamente la página principal del proyecto con el nombre **Dashboard**. Para generar nuevas páginas debemos seguir los siguientes pasos:

Primero debemos posicionarnos en el apartado **Pages**, luego en la parte superior derecha, nos aparecerá un botón que dirá **"New Page"** (Nueva Página).



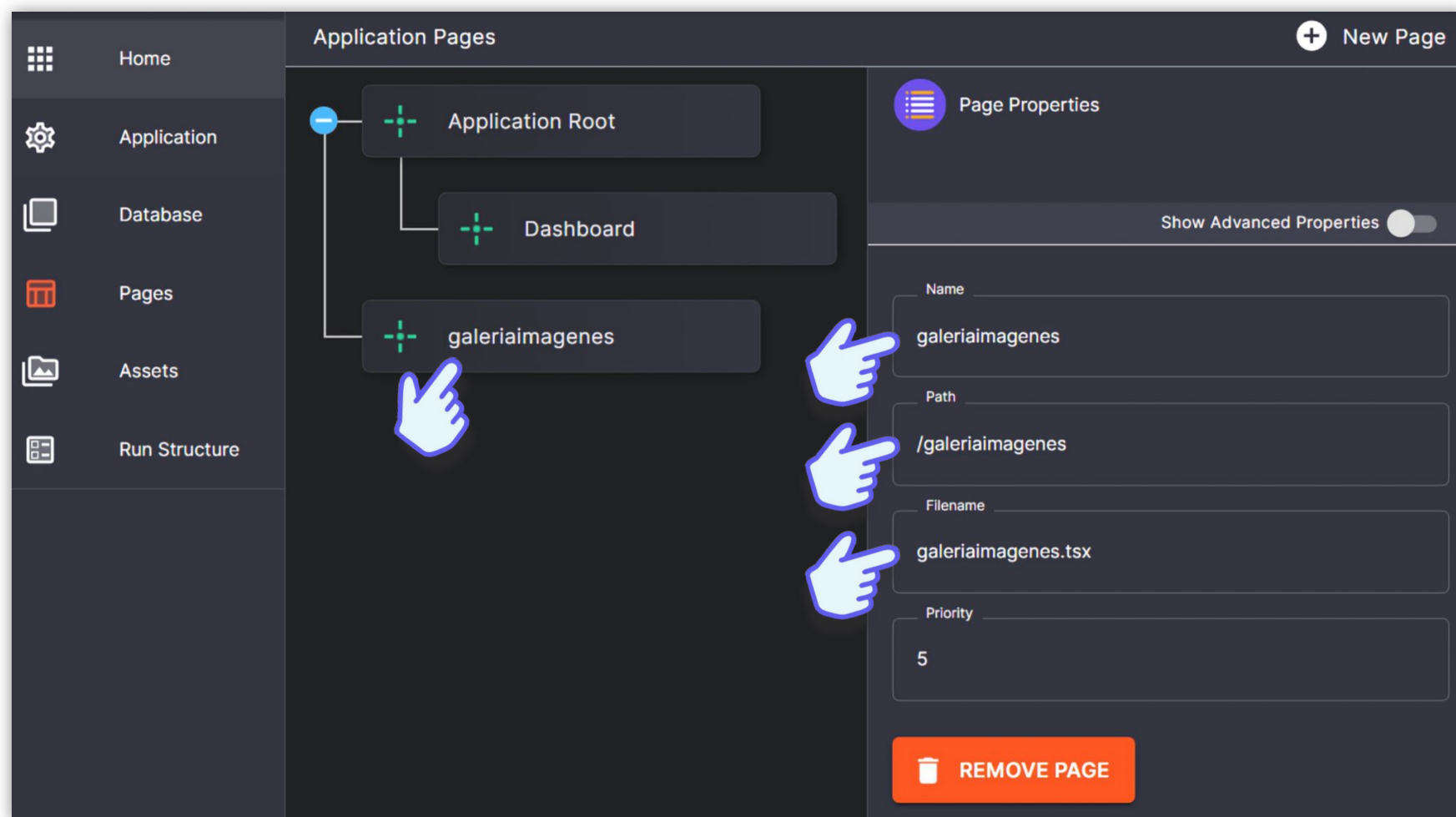
Al darle un click izquierdo se nos generará una nueva página con el nombre **"Untitled"**. Al darle un click en el nombre de la nueva página se nos desplegará las propiedades de dicha página.



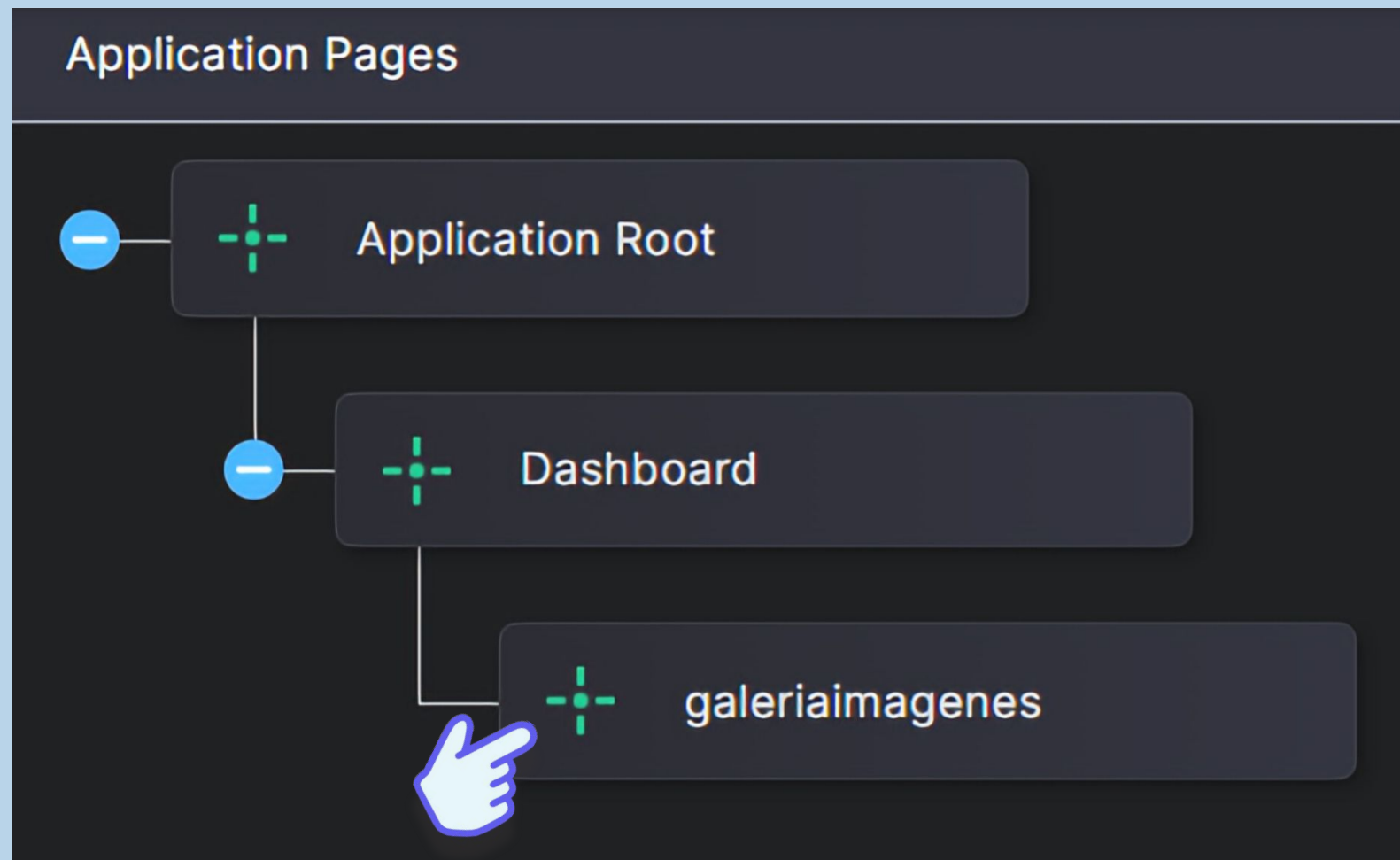
Ahora debemos cambiar el **"Name"** (nombre de la página) por el que queramos. En este campo es recomendable escribir sin dejar espacios ni utilizar caracteres especiales, como por ejemplo, un acento o la letra ñ y escribir todo en minúscula.

Luego tendremos que completar el **"Path"** (la ruta), para ello podemos utilizar lo mismo que escribimos en el nombre pero agregando el símbolo barra **" / "** en su inicio como veremos en la siguiente imagen. Pero en este campo, a diferencia del campo **"Name"**, es obligatorio escribir sin dejar espacios ni utilizar caracteres especiales, como por ejemplo un acento o la letra ñ y escribir todo en minúscula.

Y por último debemos editar la palabra **"untitled"** del campo **"Filename"** (Nombre del Archivo), podemos utilizar la misma palabra que escribimos en el nombre, dejando la extensión **".tsx"**, pero en este campo, a diferencia del campo **"Name"**, es obligatorio escribir sin dejar espacios ni utilizar caracteres especiales, como por ejemplo un acento o la letra ñ y escribir todo en minúscula.



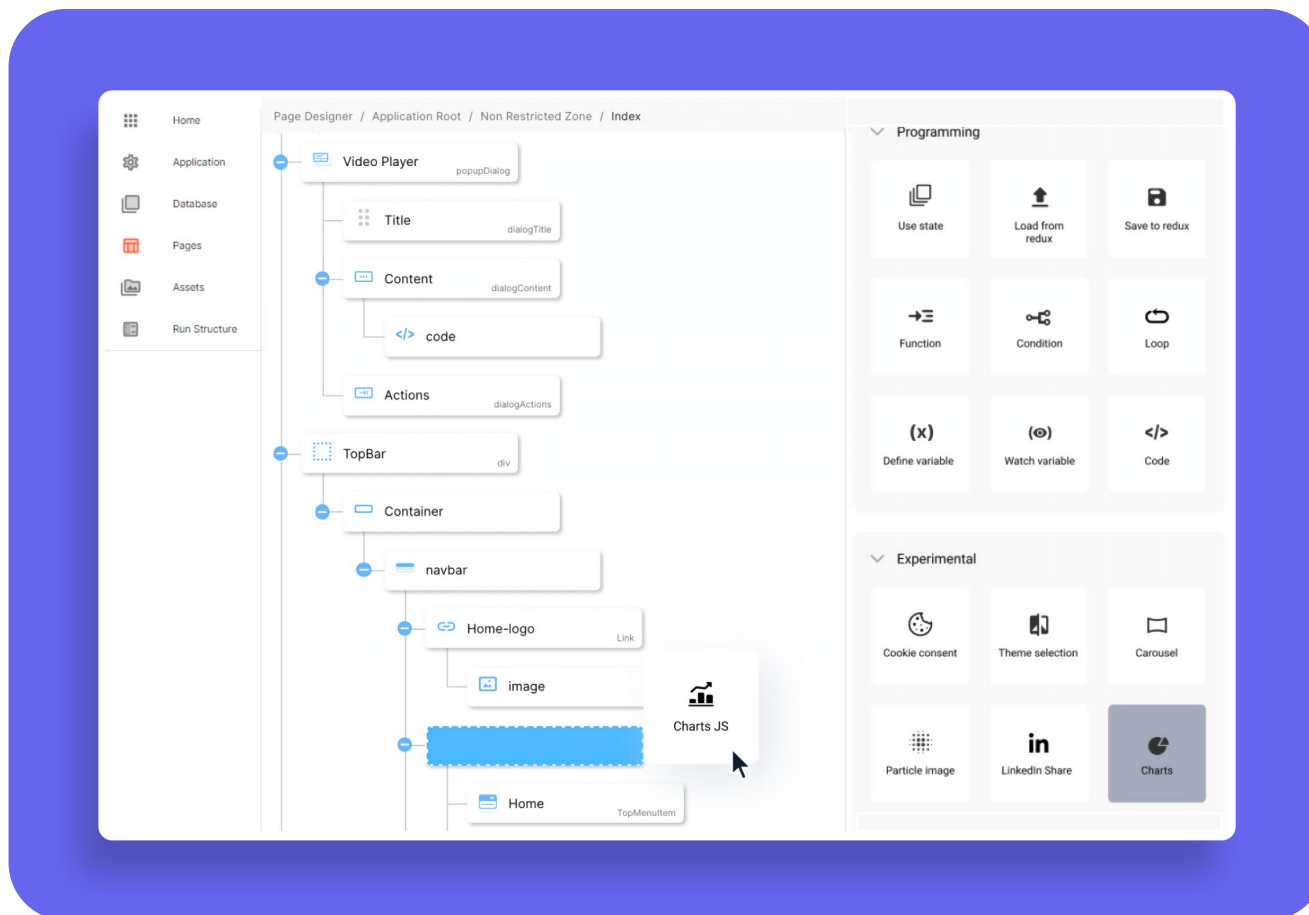
A continuación debemos mover la página creada, manteniendo el clic sobre el símbolo verde y arrastrándolo a la ubicación deseada. Es recomendable inicialmente ubicarla como hijo de Dashboard.



¡Y de esta simple manera podremos crear nuevas páginas para nuestros proyectos!

Drag & Drop

En **Aptugo**, dentro de cada página, vamos a ir distribuyendo distintos elementos, utilizando la técnica **Drag & Drop (arrastrando y soltando)** para ubicar la información y funcionalidades, según vayamos necesitando.



Con mayor detalle en la imagen anterior puede verse que:

- A la izquierda, la palabra Pages, seleccionada de color **naranja** siendo la cuarta sección de la columna.
- En el centro, se despliega el árbol de navegación, que muestra páginas ó elementos según sea el caso.
- En el apartado superior, la siguiente ruta: "Page Designer / Application Root / Non Restricted Zone / Dashboard" que permitirá orientarte al momento de reconocer si se está trabajando sobre una página ó elemento.
- A la derecha, figura la sección Elements, que contiene la librería de elementos de Aptugo clasificados en diferentes categorías.

Elementos

Los elementos son los componentes que despliegan cierta información o ejecutan determinadas acciones según sea el caso.

Conozcamos algunos de los elementos básicos que podemos encontrar en una página funcional:

Sidebar

El Sidebar es el espacio que está al lado del contenido de una página web, en este caso se utiliza como barra de navegación lateral.

Name: Aquí se coloca el nombre descriptivo del elemento, para poder visualizarlo mejor dentro del árbol de elementos en Aptugo.

Color Schema: Despliega un menú de opciones que se utiliza para darle un color predeterminado.

Open Variable: Expresión booleana (true "verdadero" o false "falso") o variable que será evaluada para determinar si se muestra o no el sidebar (obligatorio). Es un campo avanzado lo veremos más adelante en profundidad.

Open/Close handler: Expresión booleana (true "verdadero" o false "falso") o variable que será evaluada para determinar si se muestra o no el sidebar (obligatorio). Es un campo avanzado lo veremos más adelante en profundidad.


A configuration panel for the Sidebar element. It has a dark background with white text. The panel contains four sections: 1. "Name" with a text input field containing "sidebar". 2. "Color Schema" with a dropdown menu showing "Greens". 3. "Open Variable" with a text input field containing "...". 4. "Open/Close handler" with a text input field containing "...".

Para garantizar el adecuado funcionamiento del elemento Sidebar, podemos agregar dos elementos más:

Nav Link

Contiene los enlaces (links) dentro de una barra de navegación como menú ubicado normalmente a un costado izquierdo (Sidebar).



Element Properties

Show Advanced Properties ☐

Name

navLink

destination

/

Destination

/

ClassName


ClassName (active)

Text

Home

☐ Cascades


COPY

 REMOVE ELEMENT

Autolinking

Mostrará todas las páginas que sean hijos, en este caso, de Dashboard.



Element Properties

Show Advanced Properties ☐

Name


autolinking

From Link

Dashboard

☐ Cascades

COPY

 REMOVE ELEMENT

Div

Es un elemento agrupador que puede contener múltiples elementos de otros tipos, como textos o imágenes, dentro de sí mismo. Su objetivo principal es la organización de los contenidos dentro de una página.



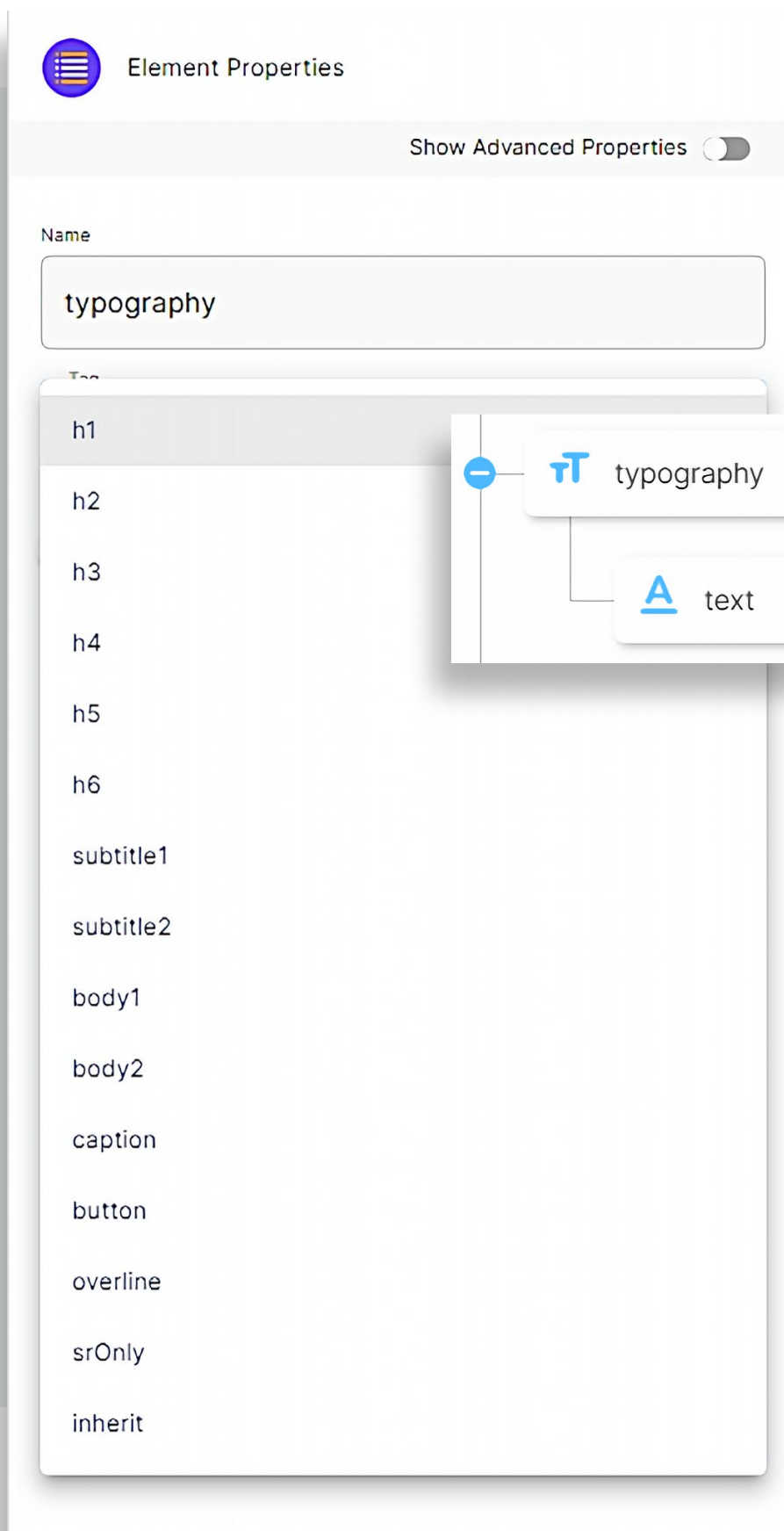
Typography

El elemento Typography nos permite elegir un estilo predeterminado para nuestros textos. También incluye por defecto otro elemento text, donde podremos escribir el texto que queramos mostrar con el estilo anteriormente seleccionado



Name: Nombre del elemento.

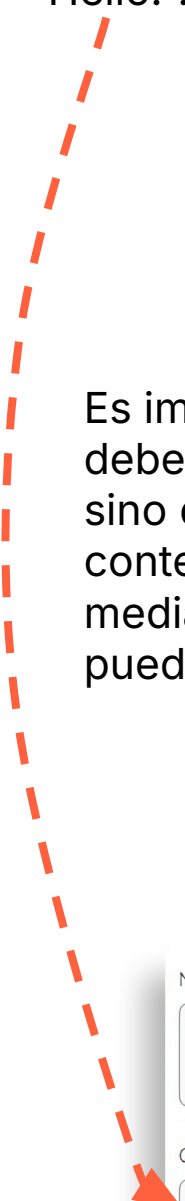
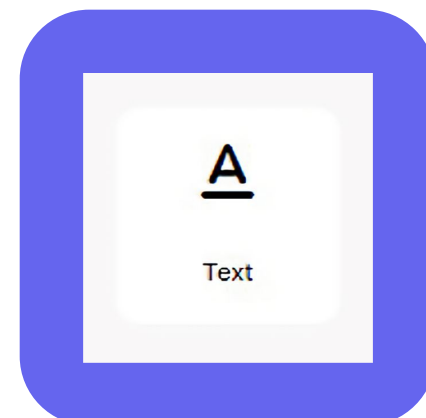
Tag: Diferentes estilos de tipografía, según el protagonismo que tenga el texto. El tag h1 posee el tamaño de letra más grande, y va disminuyendo hasta body2.



Text

Es un elemento que permite mostrar un determinado texto dentro de una página. En este caso nos muestra el texto "Hello!".

Es importante destacar que no necesariamente debe estar vinculado a un elemento Typography, sino que puede utilizarse de forma independiente, contenido dentro de un div o container. Además, mediante el último campo (ClassName), se le puede agregar estilos



Name


Content

 ...

ClassName

 ...

☐ Cascades

COPY  **REMOVE ELEMENT**

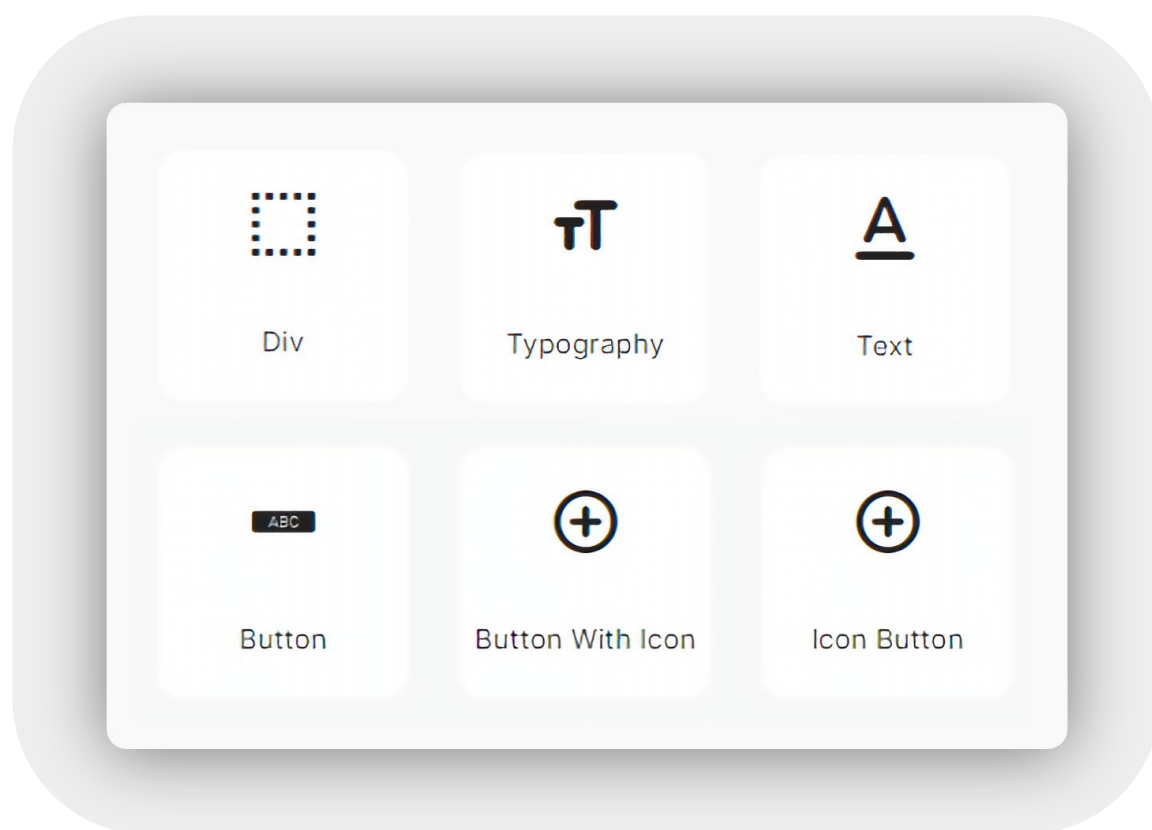
Name: Nombre del elemento.

Content: Aquí se añade el contenido que se quiere visualizar en la página.

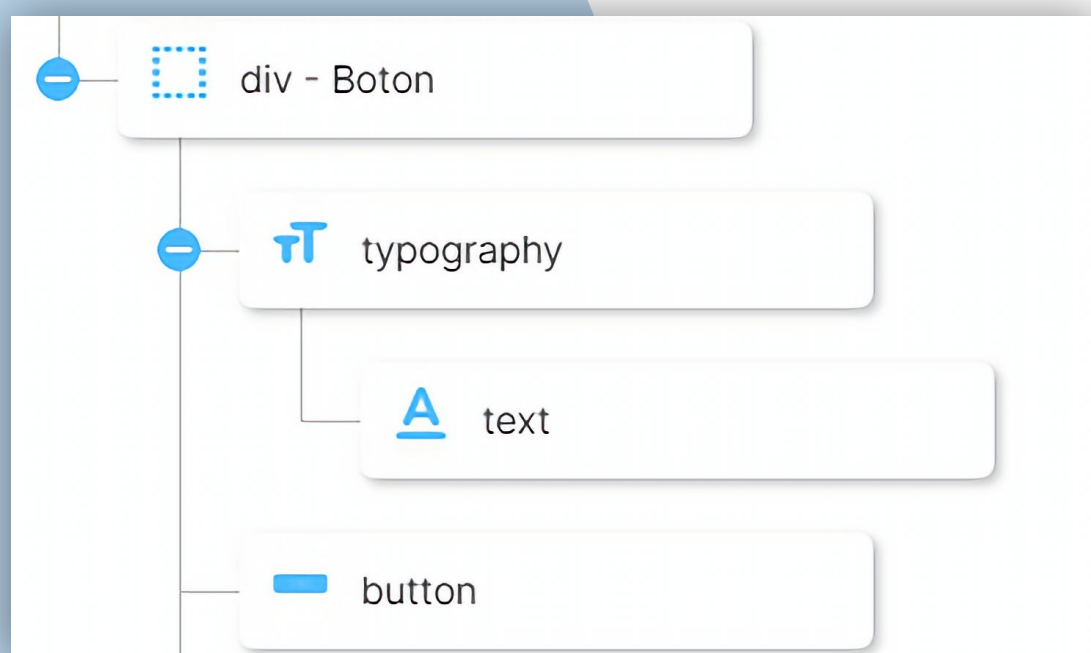
ClassName: Permite darle estilos en CSS.

En el siguiente ejemplo observamos una estructura básica para insertar un **texto** acompañado de un **botón**.

Los elementos que incluye son los siguientes:



Y la disposición en el árbol de navegación sería de la siguiente manera:



Container

Es un elemento semejante al Div con propiedades incluidas por defecto.



Name: Nombre del elemento

Max Width: Determina el ancho del contenedor

- Lg
- Md
- Sm
- Xl
- Xs
- false

ClassName: Permite darle estilos en CSS.

Name

Container

className

Max Width

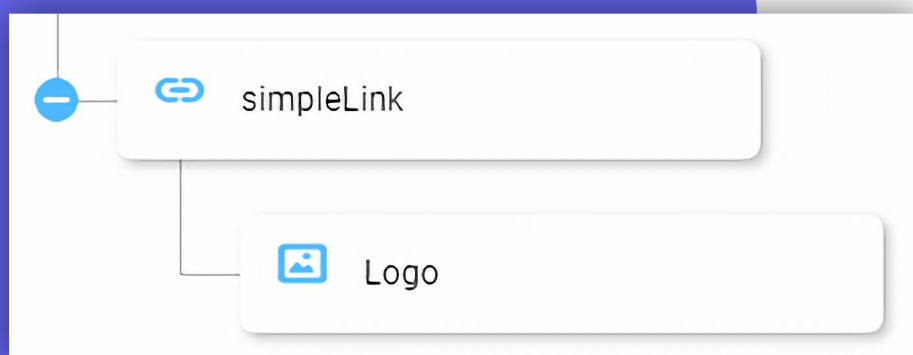
▼

Simple Link

Este elemento **habilita la incorporación de enlaces, tanto internos como externos, en nuestra página web.**

Puede asociarse con otros elementos tales como imágenes, textos o botones, siempre y cuando los coloquemos como hijos.

En el siguiente ejemplo, se puede apreciar el orden en el que deben estar los elementos para vincular la imagen de un logo a un link.



Dentro de las posibles configuraciones, hallamos:

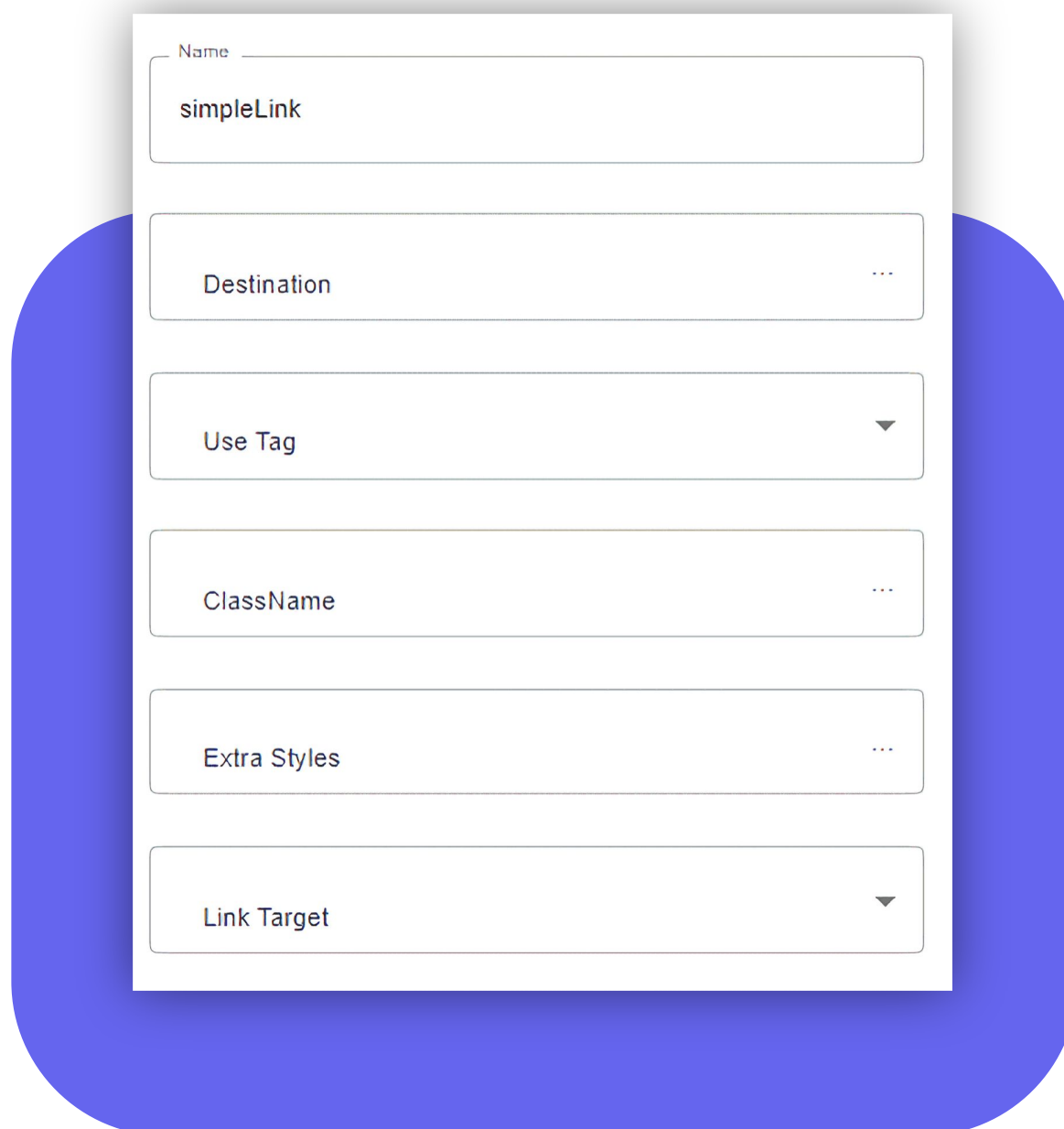
Name: Nombre del elemento

Destination: Dirección a la cual se dirigirá el enlace, el mismo, según el tipo, puede llevar a una página dentro de la aplicación (por ende, este campo se completará con el path de la página correspondiente, de la siguiente manera: /pathDeLaPagina, o solo / si deseamos ir a la página principal o dashboard) o bien a un enlace externo, el cual debe escribirse tal cual se visualiza en el navegador, incluyendo https:// y el dominio de la página web.

Use Tag:

- **Navlink:** Se emplea para dirigir el enlace a una página dentro de la aplicación.
- **A:** Se selecciona al querer dirigir el enlace a un sitio externo.

- **ClassName:** Permite el agregado de estilos con CSS.
- **Extra Styles:** Habilita la opción de insertar estilos de manera manual.
- **Link Target:** Establece el formato en el cual se abrirá el link, si será dentro de la misma página (por ejemplo, `_self`) o en una página nueva (`_blank`) :
 - **_self:** Es la opción por defecto, abre el link en la misma ventana.
 - **_blank:** Abre el enlace en una nueva pestaña o ventana del navegador.
 - **_Parent:** Carga la URL en el contexto de navegación padre (parent) del actual. Si no existe el padre, este se comporta del mismo modo que `_self`.
 - **_Top:** Muestra el enlace en pantalla completa en la misma ventana.



The image shows a white dialog box with rounded corners, set against a blue background. The dialog box contains several input fields and dropdown menus for configuring a link. The fields are arranged vertically and are as follows:

- Name:** A text input field containing the value "simpleLink".
- Destination:** A text input field with a three-dot menu icon on the right.
- Use Tag:** A dropdown menu with a downward arrow icon on the right.
- ClassName:** A text input field with a three-dot menu icon on the right.
- Extra Styles:** A text input field with a three-dot menu icon on the right.
- Link Target:** A dropdown menu with a downward arrow icon on the right.

Image

Permite el uso de imágenes a nuestro proyecto.

Configuraciones que podemos realizar:

Name: nombre del elemento.

Use an Asset: nos permite elegir entre las imágenes que cargamos previamente en Aptugo. Las mismas se pueden ingresar arrastrando el archivo y soltándolo en la pestaña Asset → Images.

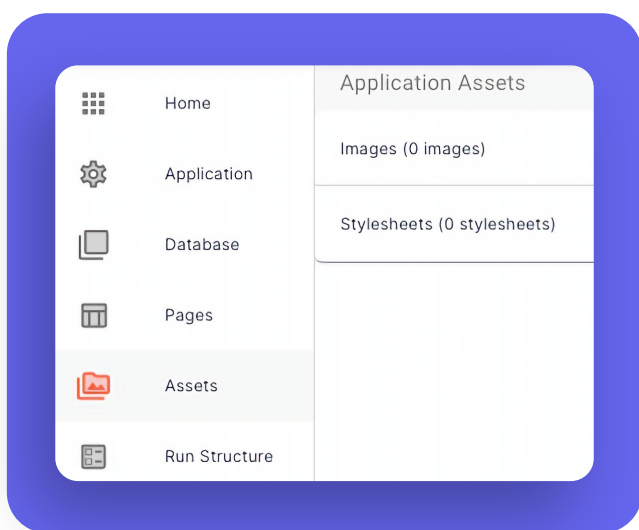
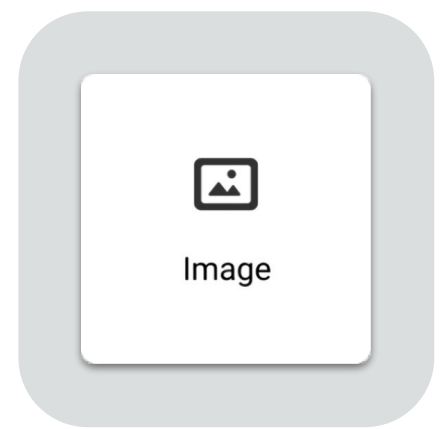


Image Path: Permite incluir imágenes a través de una URL o ruta.

Image Path (WebP): Otra manera de representar la ruta de la imagen, se suele dejar en blanco.

ClassName: Permite el agregado de estilos con CSS.

Alt text: Se utiliza para colocar una descripción de texto a la imagen.

Width: Configura el ancho de la imagen.

Height: Determina el alto de la misma.

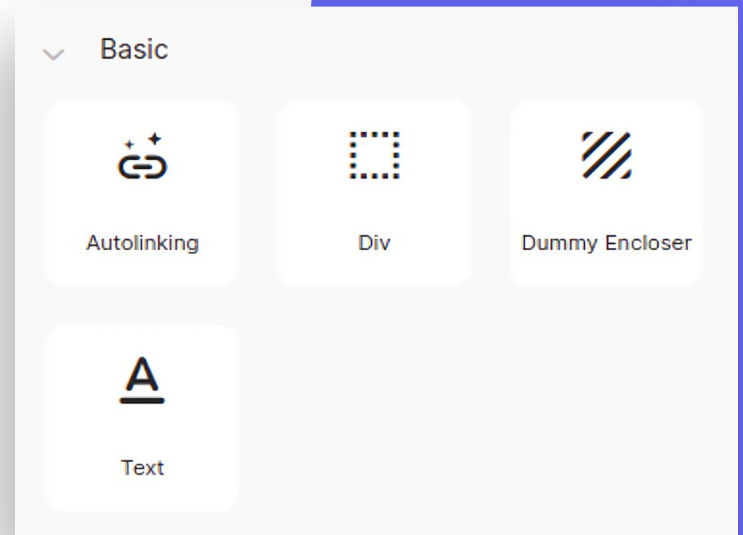
A vertical form for configuring an image element. It consists of several input fields and a dropdown menu. The fields are: "Name" (text input with "image" entered), "Use an asset" (dropdown menu), "Image Path" (text input with a three-dot menu icon), "Image Path (WebP)" (text input with a three-dot menu icon), "ClassName" (text input with a trash icon), "Alt Text" (text input with a three-dot menu icon), "Width" (text input with a three-dot menu icon), and "Height" (text input with a three-dot menu icon).

Categorías

Veamos un poco más en detalle los **elementos que ya están incluidos en la librería de Aptugo** y que podemos reutilizar arrastrando desde la sección de elementos hasta nuestro árbol de navegación.

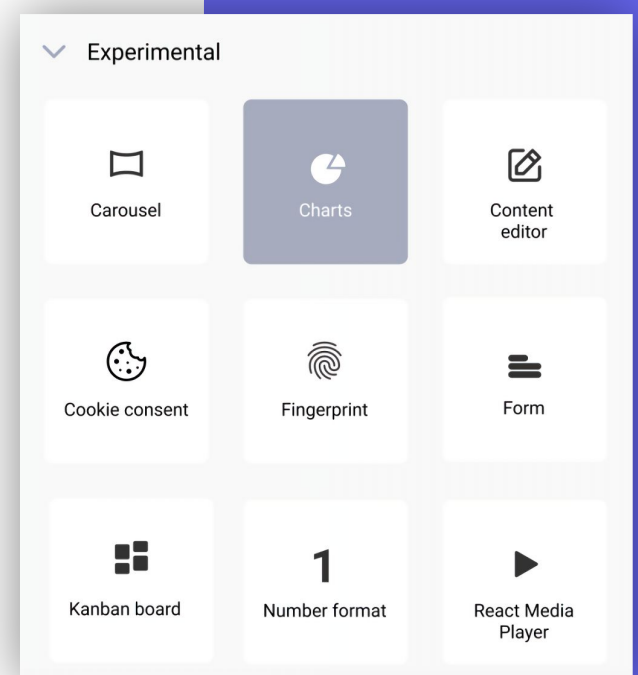
Basic

Esta categoría reúne elementos básicos como puede ser un Text (texto). Al desplegar la sección Basic dentro de Pages en Aptugo encontrarás distintos elementos que pertenecen a la categoría. Casi cualquier desarrollo que realices contendrá muchos elementos básicos a través de todas sus secciones.



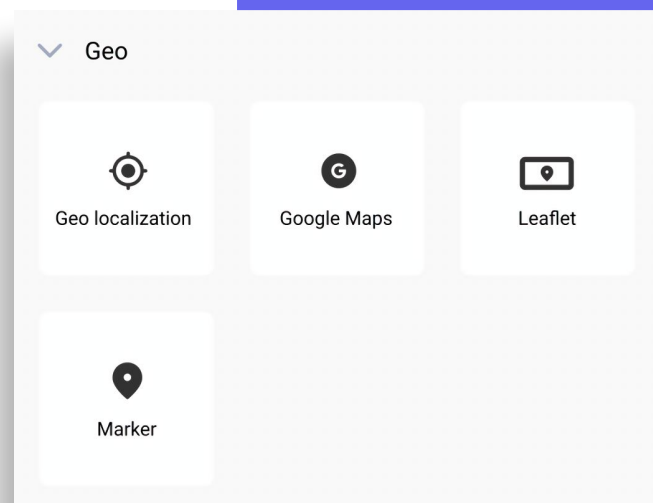
Experimental

La categoría Experimental, concentra elementos que permiten incorporar funcionalidades que aún se encuentran en estado de prueba en Aptugo. Es una de las secciones más interesantes y contiene elementos tan variados como **ChartJS** que permite incorporar distintos tipos de gráficos, **Kanban Board**, o **React Media Player**.



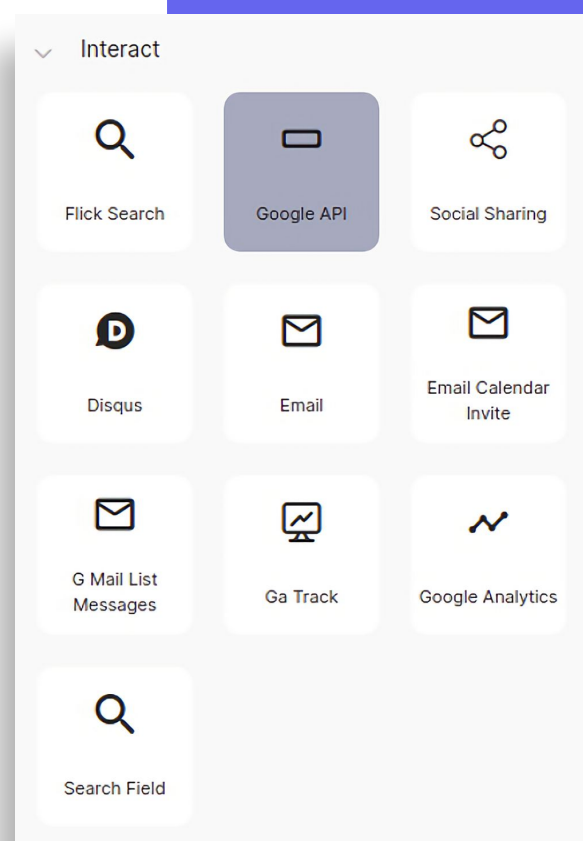
Geo

Reúne elementos que hacen referencia a geolocalización. El elemento **Leaf Let** permite mostrar un mapa interactivo es uno de los más utilizados de esta sección. Al desplegar la sección **Geo** dentro de Pages en Aptugo encontrarás todos los elementos que pertenecen a la categoría.



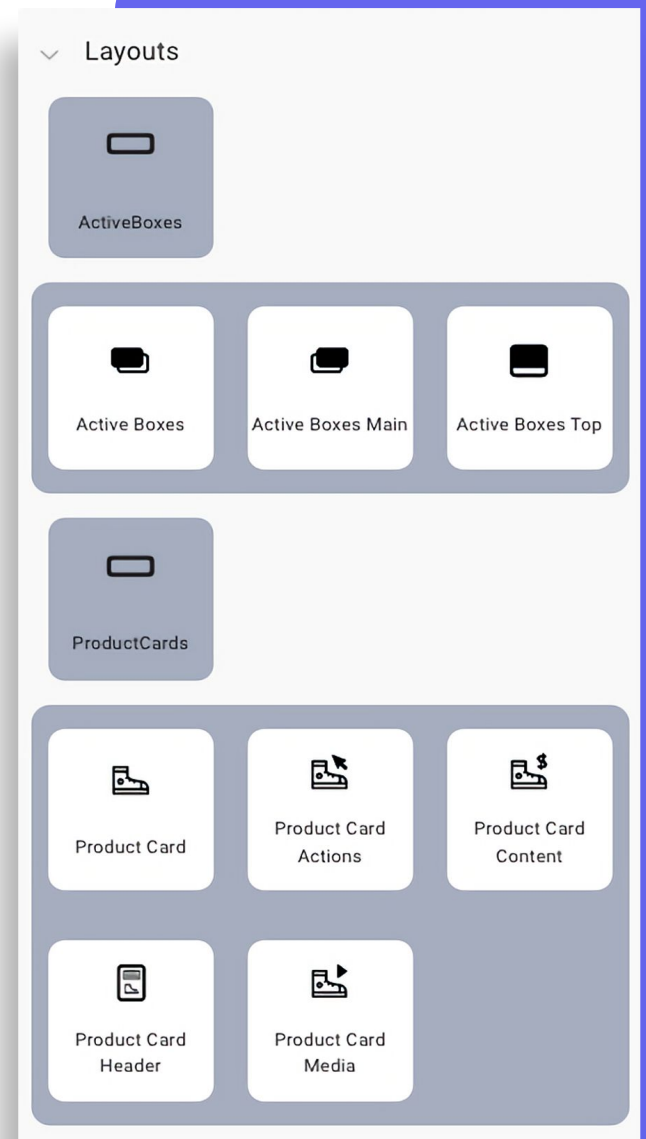
Interact

Esta categoría contiene elementos que permiten obtener o vincular cierta información externa con cada una de las páginas que construyamos. El elemento **Google Analytics**, por ejemplo, permite obtener información respecto a la cantidad de visitantes que navegan un conjunto de páginas. Encontrarás también otros elementos interesantes como **Social Sharingo** **Email**.



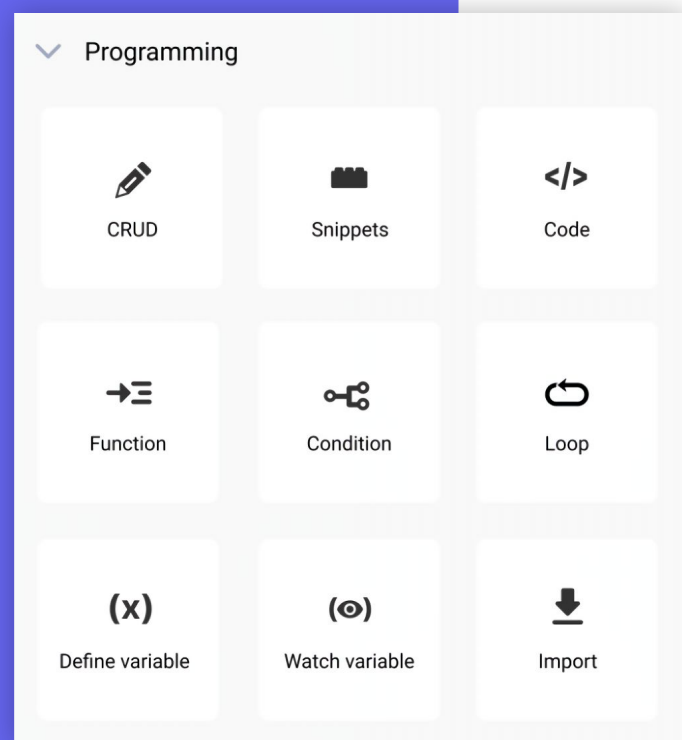
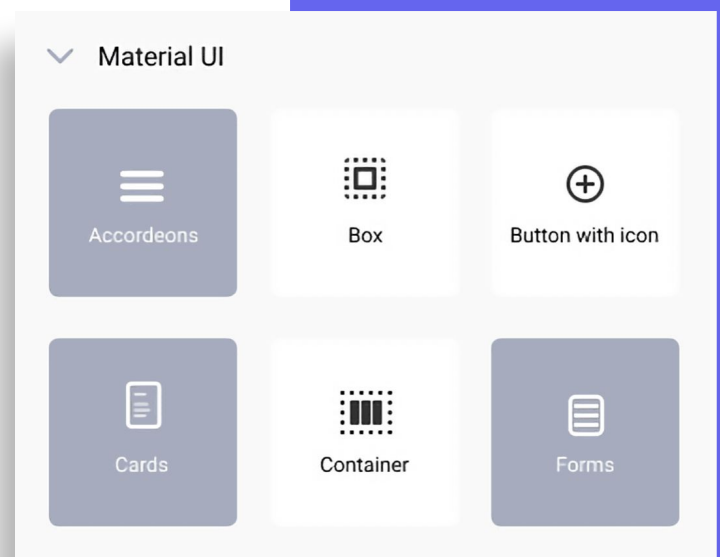
Layouts

Contiene elementos que hacen referencia a la disposición y organización general de los elementos de alto nivel. Al desplegar la sección Layout dentro de Pages en Aptugo encontrarás todos los elementos que pertenecen a la categoría.



Material-UI

Contiene elementos muy utilizados sobre los cuales Aptugo tiene basada su estructura general de elementos. El elemento Button (botón) que permite mediante un click realizar una determinada acción es un ejemplo. Los elementos **Checkbox**, **Accordion**, **Card**, y **Dialogue** son algunos de los elementos que encontrarás al desplegar esta categoría.



Programming

Esta sección contiene elementos que permiten incorporar lógica de programación a las páginas que construyamos. A modo de ejemplo, el elemento **Condition** permite evaluar una condición de verdad (Verdadero o Falso) y determinar, según el resultado obtenido, cierto camino que deba tomar el flujo de un usuario al navegar un determinado sitio.

Elementos

Árbol de Navegación

UX/UI

Cómo evaluamos en los primeros ejemplos, existe una correlación directa entre el árbol y cada uno de los elementos que visualizará el usuario final. Ésta última puede estar vinculada tanto a la experiencia del usuario (UX) como a la interfaz en la cual navegará para utilizar la aplicación (UI).

A continuación se muestra un ejemplo donde se incluyen elementos de tipo texto y checkbox.

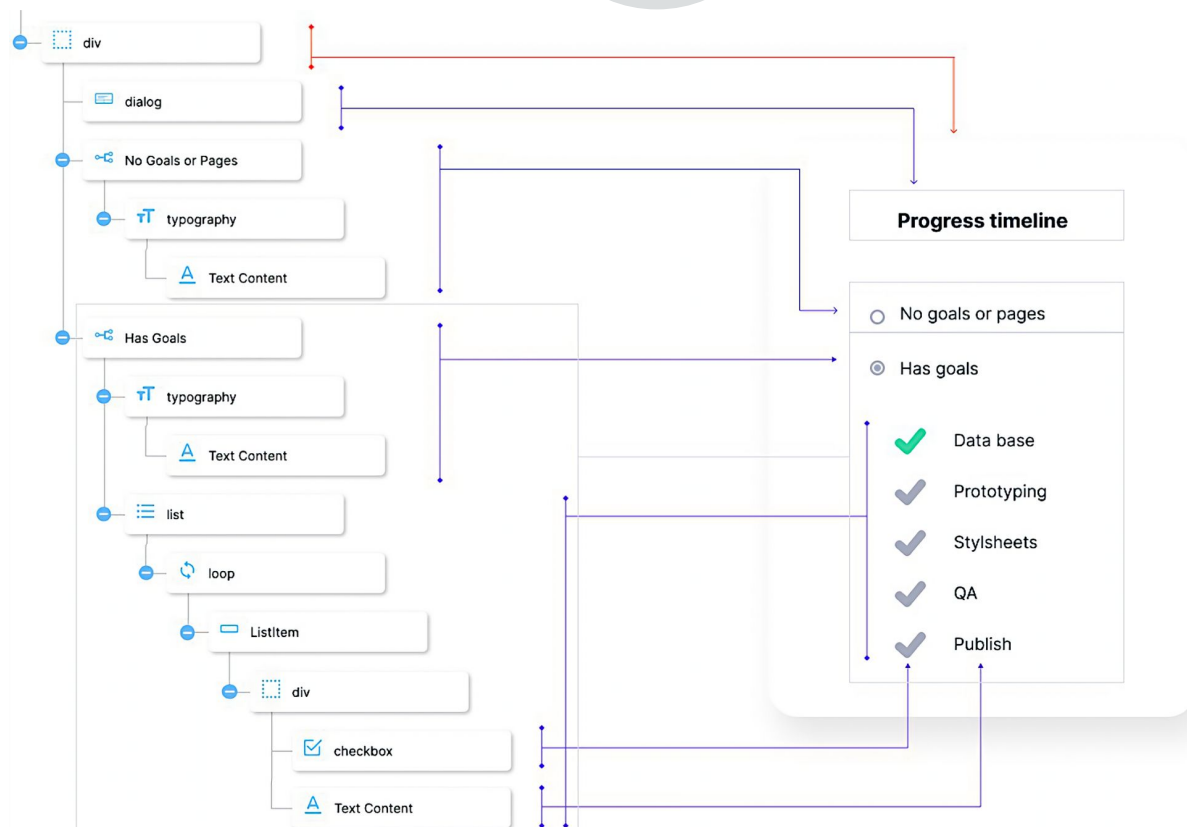
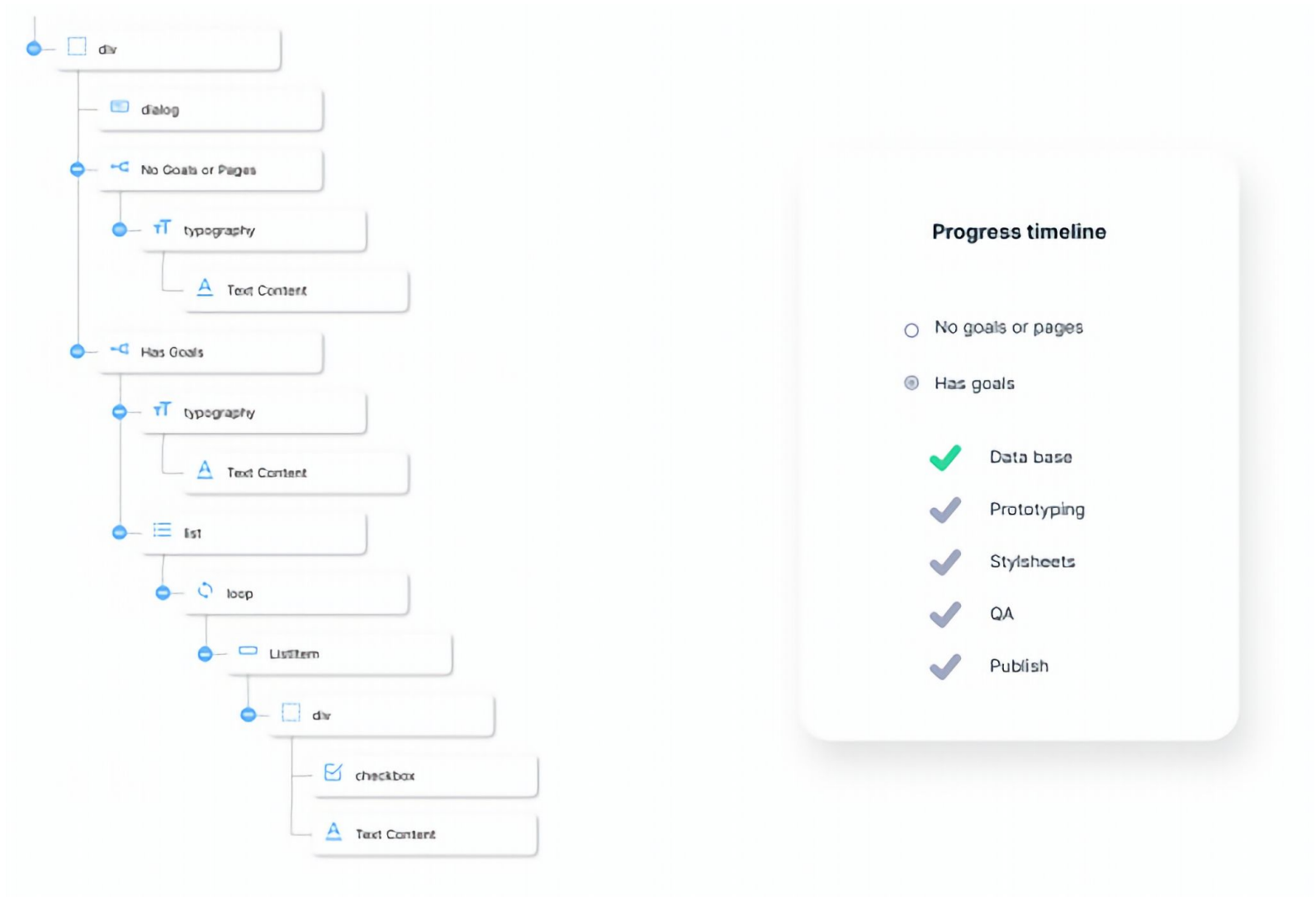
¡Esta categoría no se la esperaban!

Uno de los principales diferenciales de Aptugo con otras herramientas y plataformas visuales existentes, es que Aptugo nos permite crear nuestros propios elementos reutilizables.

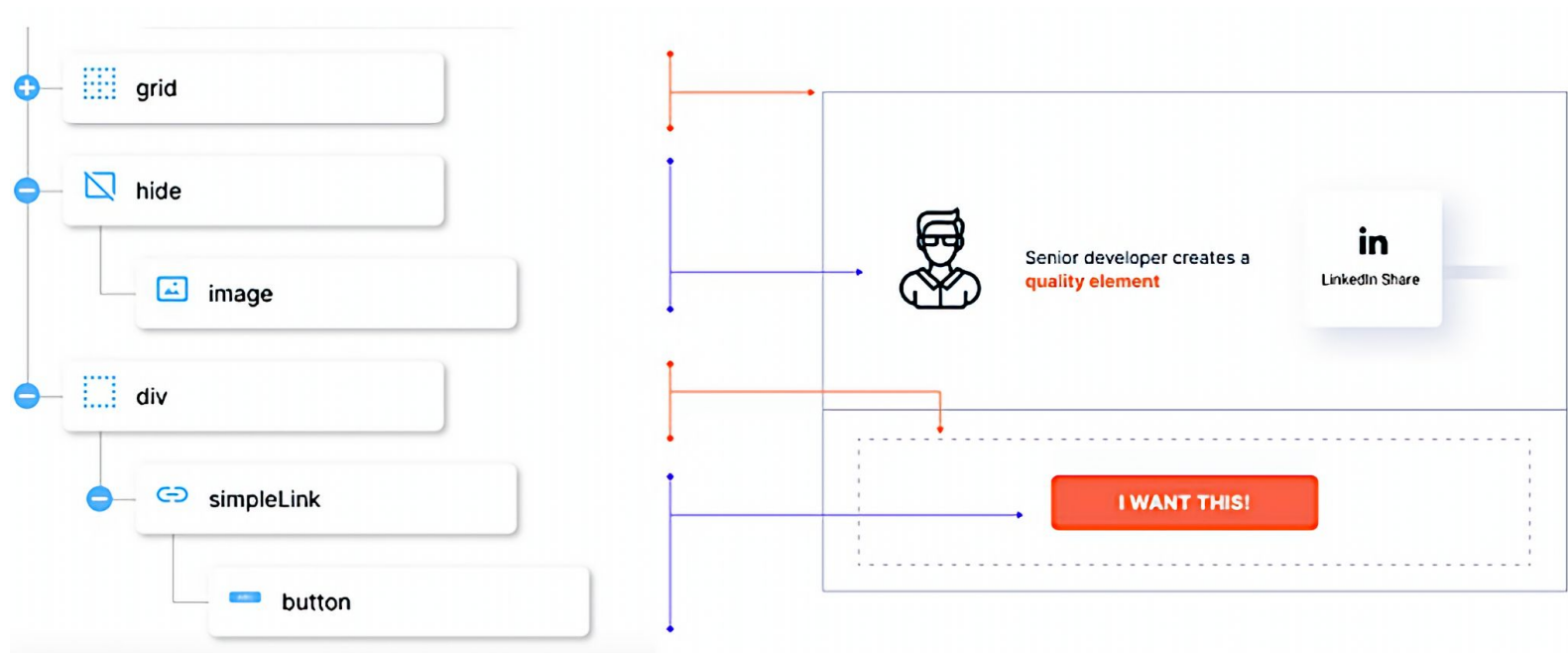
Supongamos que necesitamos un elemento que vincule nuestra aplicación con un sistema de pagos en criptomonedas y que estaremos utilizando este sistema de pagos varias veces en diferentes proyectos.

Aptugo nos permite crear con nuestro propio código, elementos nuevos y a nuestra medida, para implementar ese sistema de pagos.

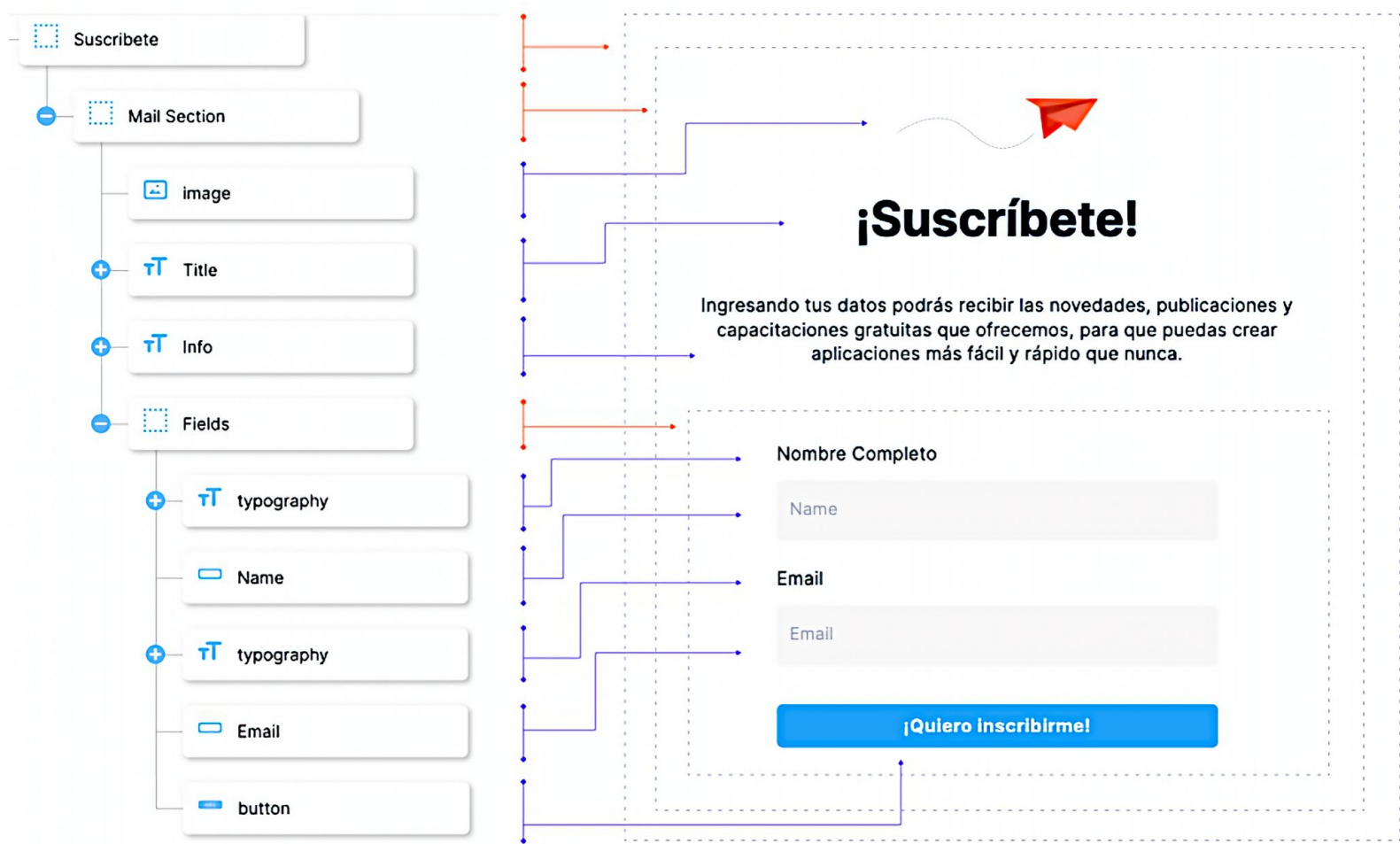
Al igual que el resto de la librería de Aptugo, podemos reutilizar lo creado y compartirlo con otros colegas y desarrolladores.



Y la relación entre el árbol y sus elementos se mantiene como base del desarrollo Front-End en Aptugo. **A continuación se muestran dos ejemplos ilustrativos, de menor**

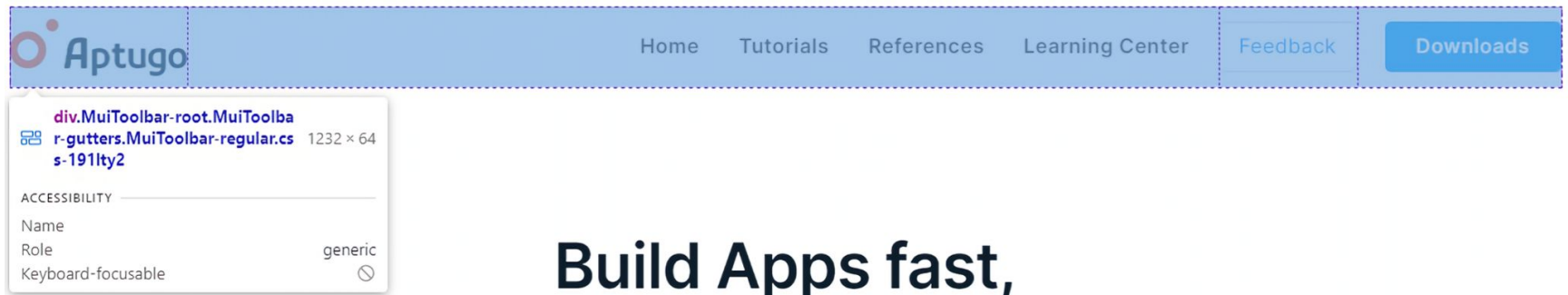


a mayor complejidad en su construcción:



Maquetado

Ahora que hemos visto los diferentes elementos que podemos utilizar, y la relación elementos-arbol-usuario, retomemos el ejemplo del header (encabezado) del sitio web de Aptugo:



¿Cuáles son cada uno de estos elementos que lo componen? A simple vista podemos distinguir una imagen, seguida de tres enlaces con distintos textos y por último un botón de Login. ¡Estos son sólo algunos de los elementos que podemos crear desde Aptugo!

De la misma forma, podés inspeccionar otra página web e intentar maquetarla con los elementos y conceptos que aprendimos hoy.

Conclusión

Como mencionamos anteriormente, para desarrollar la parte de Front-End de nuestra aplicación basta con combinar los elementos adecuados en el orden correcto. En otras palabras, **la construcción de la interfaz de usuario se basa en bloques dentro de un árbol que incluyen un nombre descriptivo y su elemento correspondiente.**

El desarrollo de Front-End es muy amplio. Esta guía de estudio nos brinda algunos conceptos elementales de Front-End y los relaciona con la disposición de sus componentes en Aptugo.

El Front-End estará mostrando con la utilización de sus componentes (páginas y elementos) los datos que reciba desde el Back-End por medio de la API que se conecta con la base de datos. La combinación de los mismos dependerá de la aplicación en concreto, cuya estructura podemos visualizar mediante el inspector. Cada elemento tiene un objetivo, y será el medio por el cual se vincularán el usuario con el Back-End, jugando un papel muy importante dentro de la funcionalidad del proyecto.

¿Qué sucede si arrastro elementos sin ningún sentido y los pongo uno abajo del otro?

Es altamente probable que Aptugo no entienda lo que se pretende hacer. Por eso mismo es siempre conveniente detenerse unos segundos a pensar en el desarrollo de la aplicación, que componentes se necesitarán y en qué orden deben estar.



Es aconsejable hacer cambios incrementales, guardando los progresos, e ir testeando a través de las funciones de **Save y Build** para ir cotejando los resultados sobre la aplicación.