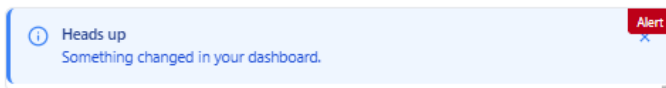


Nexus – Elemento: Alert



¿Qué es?

Alert es un componente de tipo **Display**. Muestra un mensaje de aviso o notificación con distintos niveles de severidad, cada uno con su propio color e ícono. Al arrastrarlo al lienzo viene con un mensaje de ejemplo precargado (“Heads up / Something changed in your dashboard”).

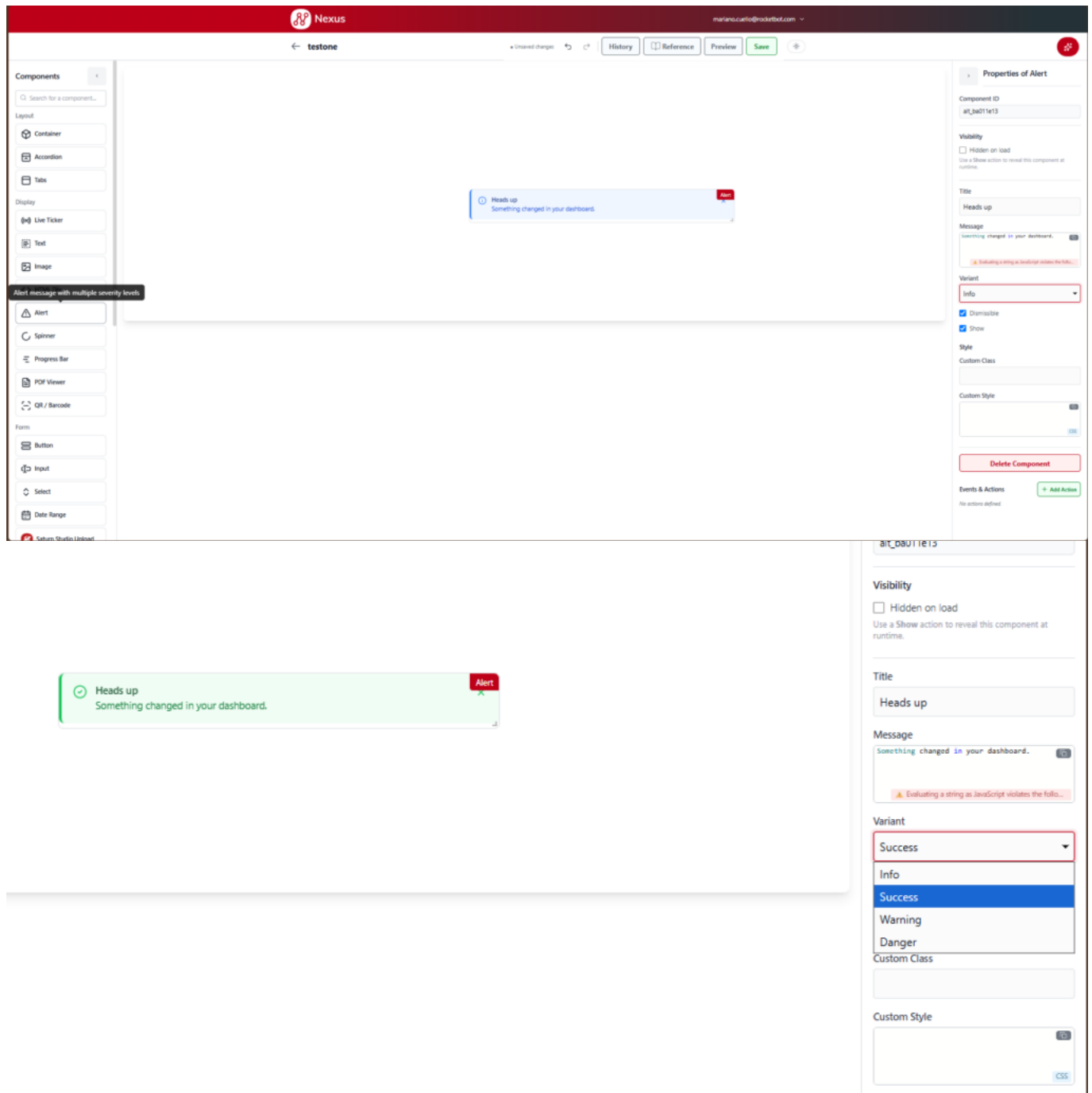


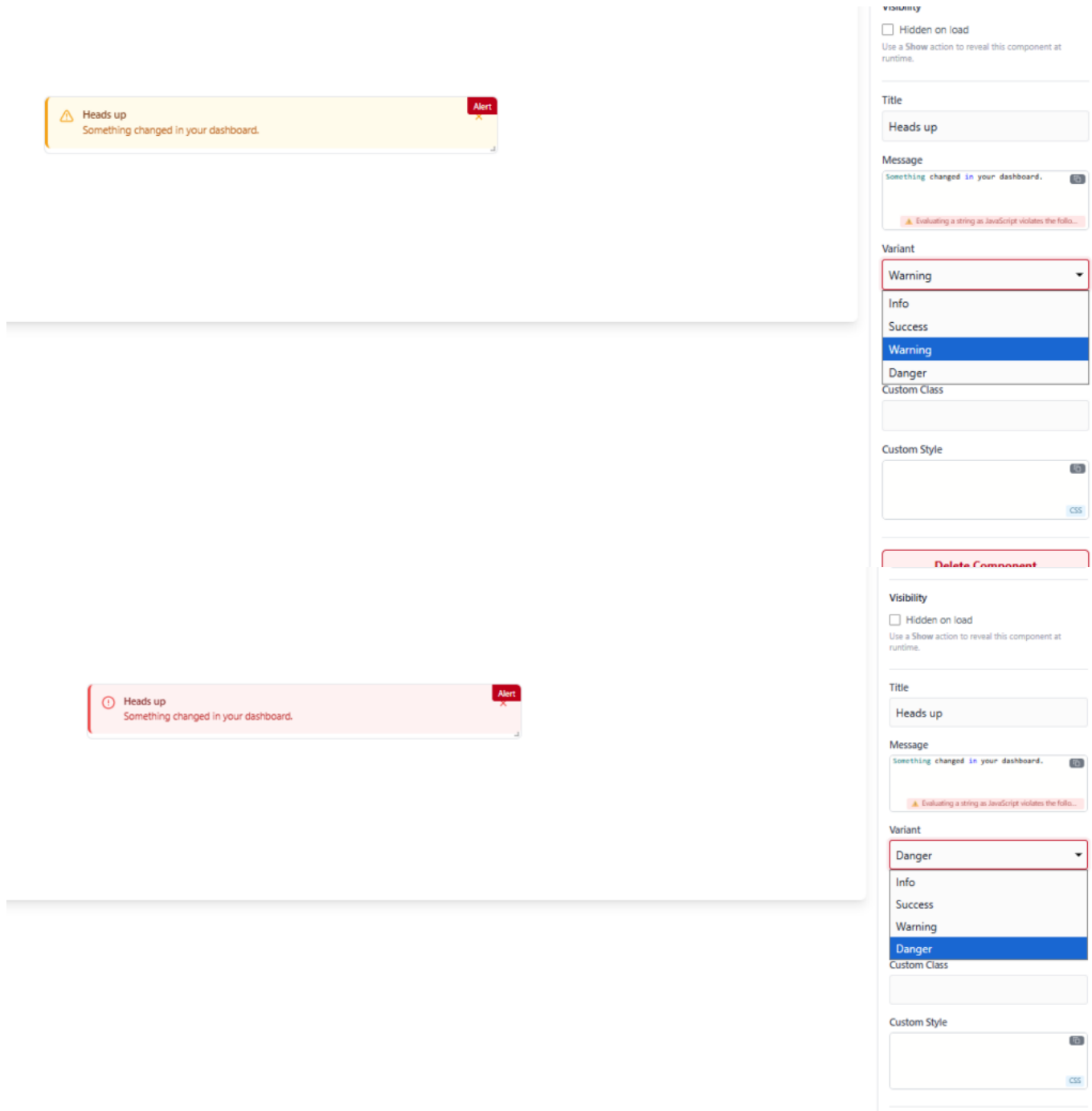
Propiedades

Al seleccionar Alert en el lienzo, el panel derecho muestra “**Properties of Alert**” con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. alt_ba011e13).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Title**: título del mensaje de alerta (ej. “Heads up”).
- **Message**: texto del cuerpo del mensaje (ej. “Something changed in your dashboard.”).
- **Variant**: menú desplegable que define el tipo/severidad de la alerta, cambiando color e ícono. Opciones:
 - **Info**: alerta informativa, en color azul.
 - **Success**: alerta de éxito, en color verde.
 - **Warning**: alerta de advertencia, en color amarillo/naranja.
 - **Danger**: alerta de error/peligro, en color rojo.
- **Dismissible**: casilla que habilita un botón (ícono x) para que el usuario pueda cerrar la alerta manualmente.
- **Show**: casilla que controla si la alerta se muestra o no.
- **Style → Custom Class**: campo para asignar una clase CSS personalizada.
- **Style → Custom Style**: campo de texto para CSS personalizado aplicado directamente a este componente.

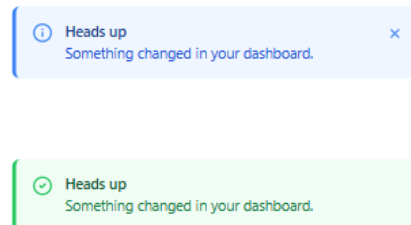
- **Delete Component:** elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .





Ejemplo de uso

Se puede usar **Variant** para comunicar distintos tipos de mensajes según el contexto: por ejemplo, **Success** al confirmar que una acción se completó correctamente, **Warning** para advertir sobre algo que requiere atención, o **Danger** para señalar un error. Con **Dismissible** activado, el usuario puede cerrar la alerta haciendo clic en el ícono ×; si se desactiva, la alerta permanece visible de forma fija mientras **Show** esté activo.



Nexus – Elemento: HTML Div



¿Qué es?

HTML Div es un componente de tipo **Display**. Es un contenedor genérico de HTML, que permite insertar código HTML personalizado directamente dentro de la pantalla. Al arrastrarlo al lienzo viene con un contenido de ejemplo precargado (“HTML content here”).

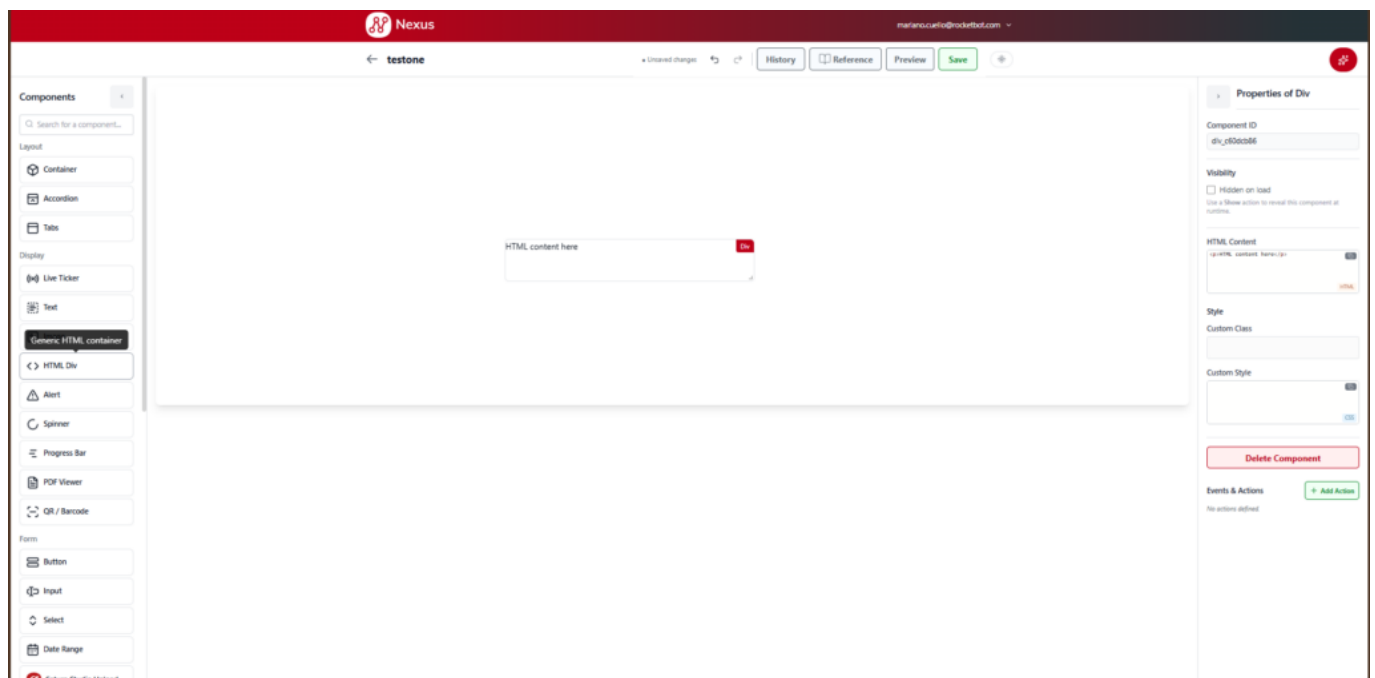
HTML content here

Div

Propiedades

Al seleccionar HTML Div en el lienzo, el panel derecho muestra “**Properties of Div**” con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. div_c60dcb86).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **HTML Content**: campo de código donde se escribe el contenido en HTML directamente (ej. <p>HTML content here</p>).
- **Style → Custom Class**: campo para asignar una clase CSS personalizada.
- **Style → Custom Style**: campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component**: elimina el componente del lienzo.
- **Events & Actions**: sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#).



Ejemplo de uso

El contenido se escribe directamente en HTML dentro de **HTML Content**. Por ejemplo:

```
html
```

```
<p>Este es un párrafo con <strong>texto en negrita</strong> y un <a href="https://www.rocketbot.com">link</a>.</p>
```

A diferencia del componente **Text** (que usa Markdown), HTML Div permite

escribir HTML puro, útil para estructuras más complejas o personalizadas que no se logran fácilmente con Markdown.

Unsaved changes ↶ ↷

History

Reference

Preview

Save

⚙️

Div

Este es un párrafo con **texto en negrita** y un [link](#).

Properties of Div

Component ID

div_cf0dcb86

Visibility

☐ Hidden on load

Use a Show action to reveal this component at runtime.

HTML Content

<p>Este es un párrafo con texto en negrita y un link.</p>

Style

Custom Class

Custom Style

Delete Component

Events & Actions

+ Add Action

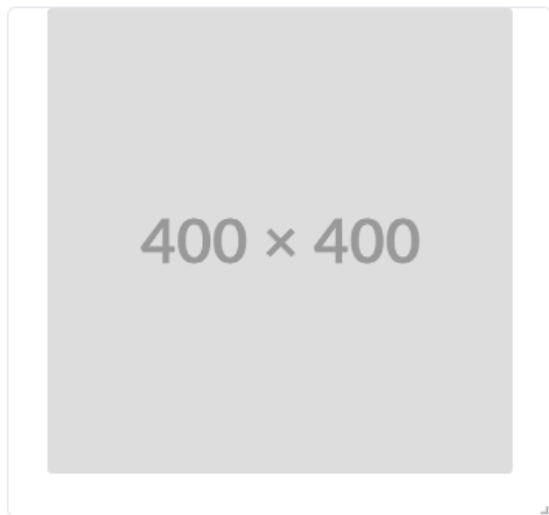
No actions defined.

Nexus – Elemento: Image



¿Qué es?

Image es un componente de tipo **Display**. Permite insertar una imagen con URL configurable. Al arrastrarlo al lienzo viene con una imagen de ejemplo tipo placeholder (400×400).



Propiedades

Al seleccionar Image en el lienzo, el panel derecho muestra **“Properties of Image”** con las siguientes opciones:

- **Component ID:** identificador único del componente, autogenerado (ej. img_ba0309b3).
- **Visibility → Hidden on load:** casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Image URL:** campo donde se coloca la URL de la imagen a mostrar (ej. <https://placeholder.co/400>).
- **Alt Text:** texto alternativo de la imagen (accesibilidad / se muestra si la imagen no carga).
- **Variant:** menú desplegable que define el tamaño con el que se muestra la imagen dentro de su espacio disponible. Opciones:
 - **Thumbnail:** la imagen se muestra en tamaño reducido, tipo miniatura.
 - **Body:** la imagen se muestra en un tamaño intermedio, como parte del contenido normal de la pantalla.
 - **Header:** la imagen se muestra ocupando el ancho completo disponible, como una imagen de cabecera.
- **Align:** menú desplegable que define la alineación de la imagen dentro de su espacio. Opciones: **Left, Center, Right**.
- **Style → Custom Class:** campo para asignar una clase CSS personalizada.
- **Style → Custom Style:** campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component:** elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .

testone

Unsaved changes

History

Reference

Preview

Save

Properties of Image

Component ID
img_ba3309b3

Visibility
☐ Hidden on load
Use a Show action to reveal this component at runtime.

Image URL
https://placeholder.co/400

Alt Text
Image

Variant
Thumbnail

Align
Center

Style
Custom Class

Custom Style

Delete Component

Events & Actions
[+ Add Action](#)
No actions defined.

Nexus

mariano.cuello@rocketbot.com

testone

Unsaved changes

History

Reference

Preview

Save

Properties of Image

Component ID
img_ba3309b3

Visibility
☐ Hidden on load
Use a Show action to reveal this component at runtime.

Image URL
https://placeholder.co/400

Alt Text
Image

Variant
Body

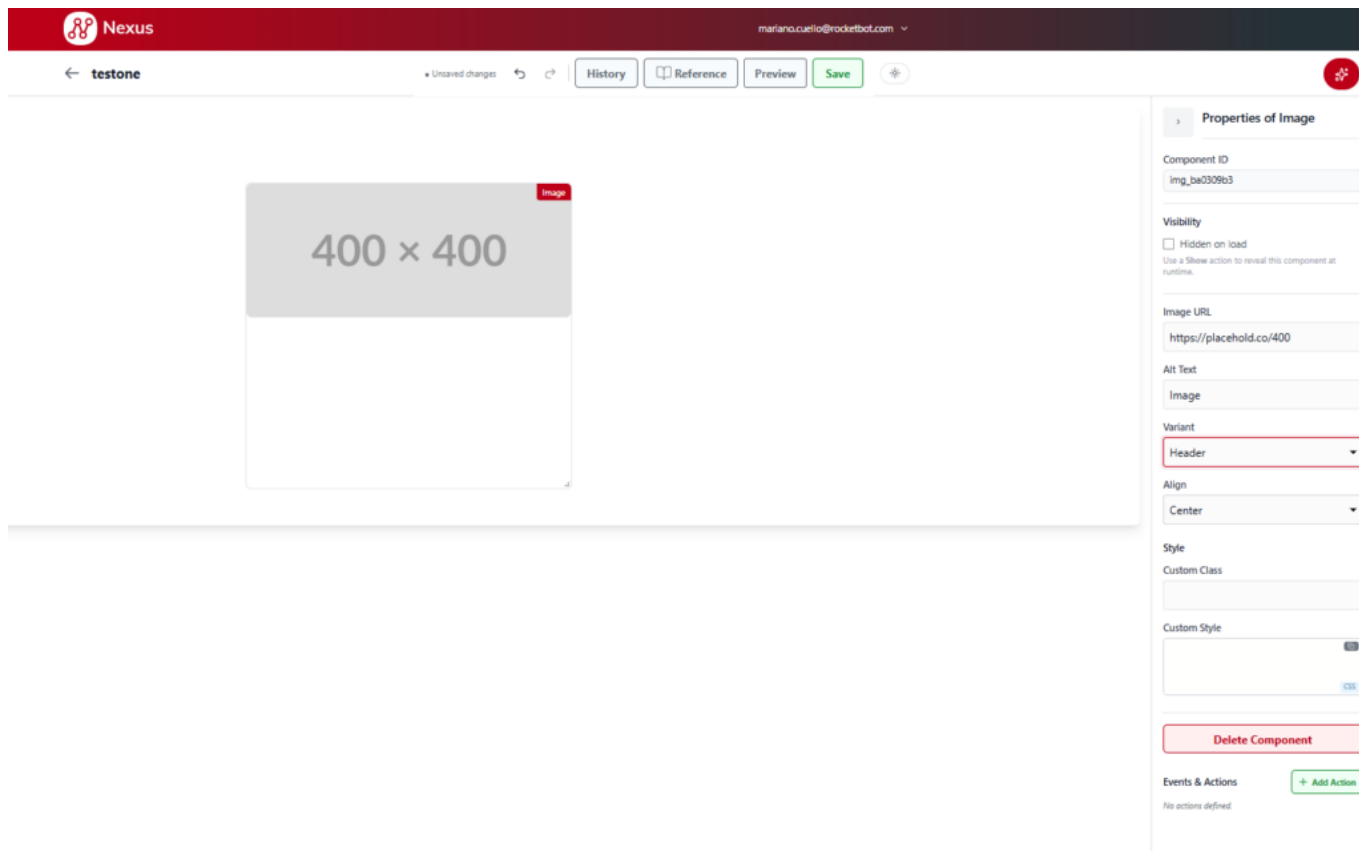
Align
Center

Style
Custom Class

Custom Style

Delete Component

Events & Actions
[+ Add Action](#)
No actions defined.

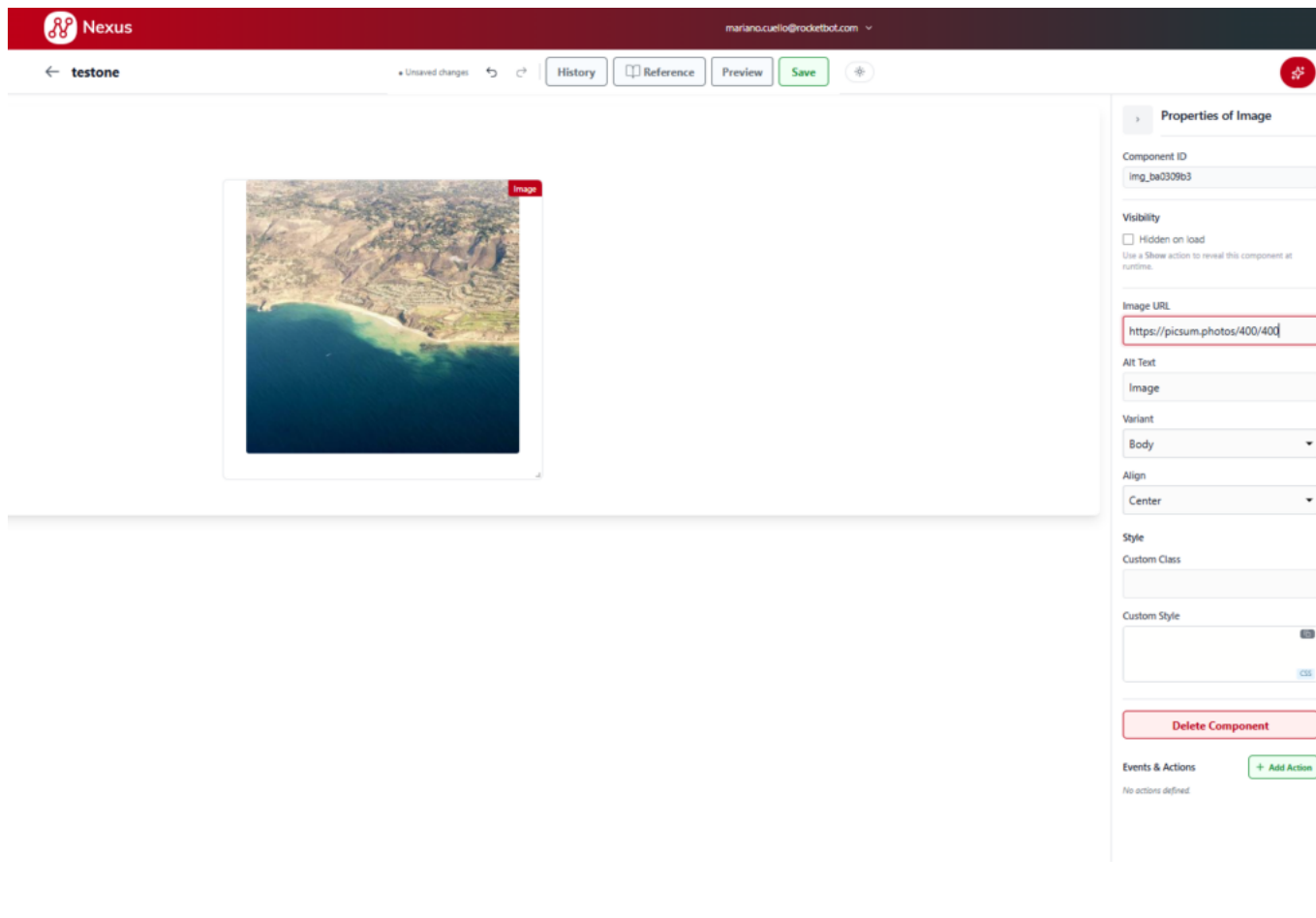


Ejemplo de uso

Para mostrar una imagen propia, se reemplaza el valor de **Image URL** por la dirección de la imagen deseada (debe ser una URL pública accesible). Por ejemplo, se puede usar una imagen libre de copyright de prueba:

<https://picsum.photos/400/400>

Con **Variant** se ajusta el tamaño según el uso: **Thumbnail** para íconos o miniaturas, **Body** para imágenes dentro del contenido, y **Header** para imágenes destacadas de ancho completo. Con **Align** se controla si queda alineada a la izquierda, centrada o a la derecha.



Nexus – Elemento: Text



¿Qué es?

Text es un componente de tipo **Display**. Permite mostrar texto estático o dinámico con formato, usando sintaxis **Markdown**. Al arrastrarlo al lienzo viene con un contenido de ejemplo precargado (título, texto con negrita y link, y una lista).

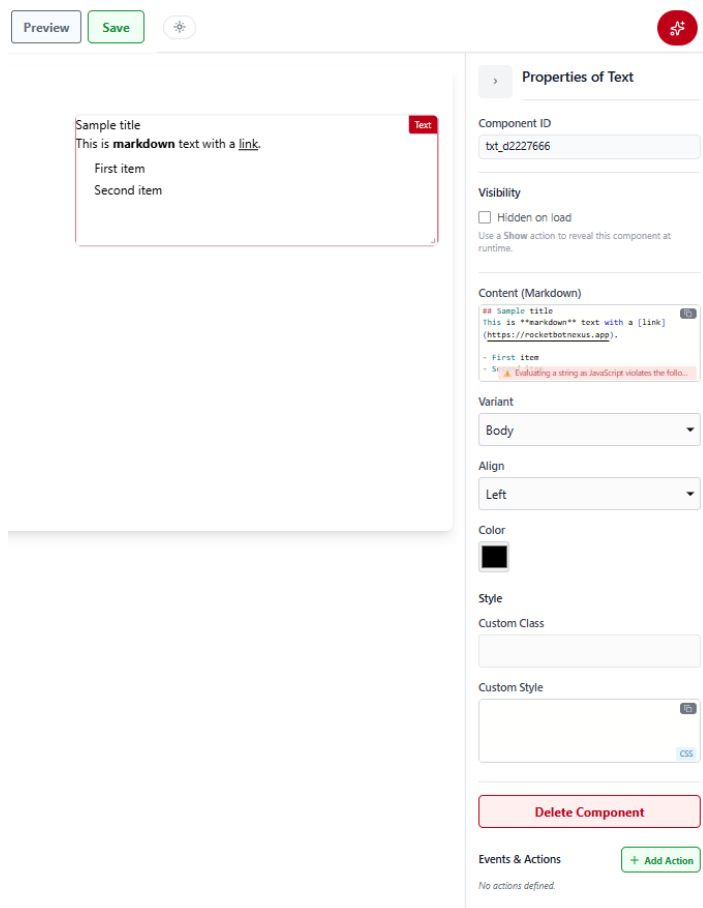
Sample title
This is **markdown** text with a [link](#).

First item
Second item

Propiedades

Al seleccionar Text en el lienzo, el panel derecho muestra “**Properties of Text**” con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. txt_aef2f592).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Content (Markdown)**: campo de texto donde se escribe el contenido usando sintaxis Markdown (títulos con #, negrita con ****texto****, links con [texto](url), listas con -, etc.).
- **Variant**: menú desplegable que define el estilo tipográfico del texto. Opciones: **Heading 1, Heading 2, Heading 3, Body, Caption**.
- **Align**: menú desplegable que define la alineación del texto. Opciones: **Left, Center, Right**.
- **Color**: selector de color para el texto, con paleta visual y campos numéricos **R / G / B** para definir el color de forma manual.
- **Style → Custom Class**: campo para asignar una clase CSS personalizada.
- **Style → Custom Style**: campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component**: elimina el componente del lienzo.
- **Events & Actions**: sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .



Ejemplo de uso

El contenido se escribe en Markdown dentro de **Content**. Por ejemplo:

markdown

```
# Bienvenido a Nexus
```

Con este componente podés escribir texto con ****negrita****, **cursiva** y enlaces, como este a la [documentación de Rocketbot](https://docs.rocketbot.com/).

- Primer ítem
- Segundo ítem

Con **Variant** se define si ese bloque de texto se comporta como un encabezado (Heading 1/2/3), texto de cuerpo (Body) o texto pequeño auxiliar (Caption). Con **Align** se controla la alineación horizontal, y con **Color** el color del texto.

• Unsaved changes ↶ ↷

HistoryReferencePreviewSave⚙

Bienvenido a Nexus

Con este componente podés escribir texto con **negrita**, *cursiva* y enlaces, como este a la [documentación de Rocketbot](#).

Primer ítem

Segundo ítem

Text

Properties of Text

Component ID

txt_aef2f592

Visibility

☐ Hidden on load

Use a Show action to reveal this component at runtime.

Content (Markdown)

```
# Bienvenido a Nexus
Con este componente podés escribir texto con
**negrita**, *cursiva* y enlaces, como este
a la [documentación de Rocketbot](https://docs.rocketbot.com/).
```

Variant

Body

Align

Left

Color

Nexus – Elemento: Live Ticker



¿Qué es?

Live Ticker es un componente de tipo **Display**. Muestra una cinta de datos con desplazamiento continuo (scroll horizontal automático), ideal para destacar métricas o indicadores en tiempo real. Al arrastrarlo al lienzo viene con datos de ejemplo precargados (Revenue, NPS, Open incidents).

Unsaved changes | History | Reference | Preview | Save | ⚙️

Live Ticker

LIVE | \$142.4k +8.2% NPS 68 +3 Open incidents 4 -2 Revenue \$142.4k +8.2% NPS 68 +3 Open incidents 4 -2

Properties of LiveTicker

Component ID: ltk_ca180ff8

Visibility: ☐ Hidden on load
Use a Show action to reveal this component at runtime.

Ticker Label: LIVE

Items (JSON)

```
{
  {
    "title": "Revenue",
    "value": "$142.4k",
    "delta": "+8.2%",
    "deltaDirection": "up"
  },
  {
    "title": "NPS",
    "value": "68",
    "delta": "+3",
    "deltaDirection": "up"
  },
  {
    "title": "Open incidents",
    "value": "4",
    "delta": "-2",
    "deltaDirection": "down"
  }
}
```

Behavior: Loop Duration (sec): 24

Style: Custom Class: Custom Style:

Delete Component

Propiedades

Al seleccionar Live Ticker en el lienzo, el panel derecho muestra **“Properties of LiveTicker”** con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. ltk_ca180ff8).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Ticker Label**: texto que aparece fijo al inicio de la cinta (ej. LIVE).
- **Items (JSON)**: campo de código donde se define el contenido de la cinta en formato JSON. Cada ítem se compone de:
 - **title**: nombre del dato (ej. “Revenue”).
 - **value**: valor mostrado (ej. “\$142.4k”).
 - **delta**: variación mostrada junto al valor (ej. “+8.2%).
 - **deltaDirection**: dirección de la variación. Acepta "up" (variación positiva, se muestra en **verde**) y "down" (variación negativa, se muestra en **rojo**).

El editor valida el JSON en tiempo real (marca “✓ valid” cuando el formato es correcto).

- **Behavior → Loop Duration (sec):** tiempo en segundos que tarda la cinta en completar un ciclo de desplazamiento (valor de ejemplo: 24).
- **Style → Custom Class:** campo para asignar una clase CSS personalizada.
- **Style → Custom Style:** campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component:** elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .



Properties of LiveTicker

Component ID

ltk_ca180ff8

Visibility

☐ Hidden on load

Use a Show action to reveal this component at runtime.

Ticker Label

LIVE

Items (JSON)

```
[
  {
    "title": "Revenue",
    "value": "$142.4k",
    "delta": "+8.2%",
    "deltaDirection": "up"
  },
  {
    "title": "NPS",
    "value": "68",
    "delta": "+3",
    "deltaDirection": "up"
  },
  {
    "title": "Open incidents",
    "value": "12",
    "delta": "-1",
    "deltaDirection": "down"
  }
]
```

✓ valid

Behavior

Loop Duration (sec)

24

Style

Custom Class

Custom Style

CSS

Delete Component

Events & Actions

+ Add Action

Ejemplo de uso

Para agregar un nuevo ítem a la cinta, hay que ubicarse en la última llave `}` del bloque anterior dentro de **Items (JSON)**, agregarle una coma, y pegar el nuevo bloque a continuación:

```
[
  {
    "title": "Open incidents",
    "value": "4",
    "delta": "-2",
    "deltaDirection": "down"
  },
  {
    "title": "Usuarios activos",
    "value": "312",
    "delta": "+5",
    "deltaDirection": "up"
  }
]
```

⚠️ Importante: la coma va pegada a la llave `}` del bloque anterior, no al principio del bloque nuevo.

The screenshot displays a software interface with a top toolbar containing buttons for 'Unsaved changes', 'History', 'Reference', 'Preview', 'Save', and a settings icon. The main workspace shows a 'LiveTicker' component with a red label. The ticker displays a horizontal list of items: 'LIVE', 'NPS 68 +3', 'Open incidents 4 -2', 'Usuarios activos 312 +5', 'Revenue \$142.4k +8.2%', 'NPS 68 +3', and 'Open incidents'. On the right, the 'Properties of LiveTicker' panel is open, showing the 'Component ID' as 'ltk_ca180f9b'. The 'Items (JSON)' section displays the JSON configuration for the ticker, which matches the example code provided in the text. The JSON is valid, as indicated by a green checkmark and the word 'valid'.

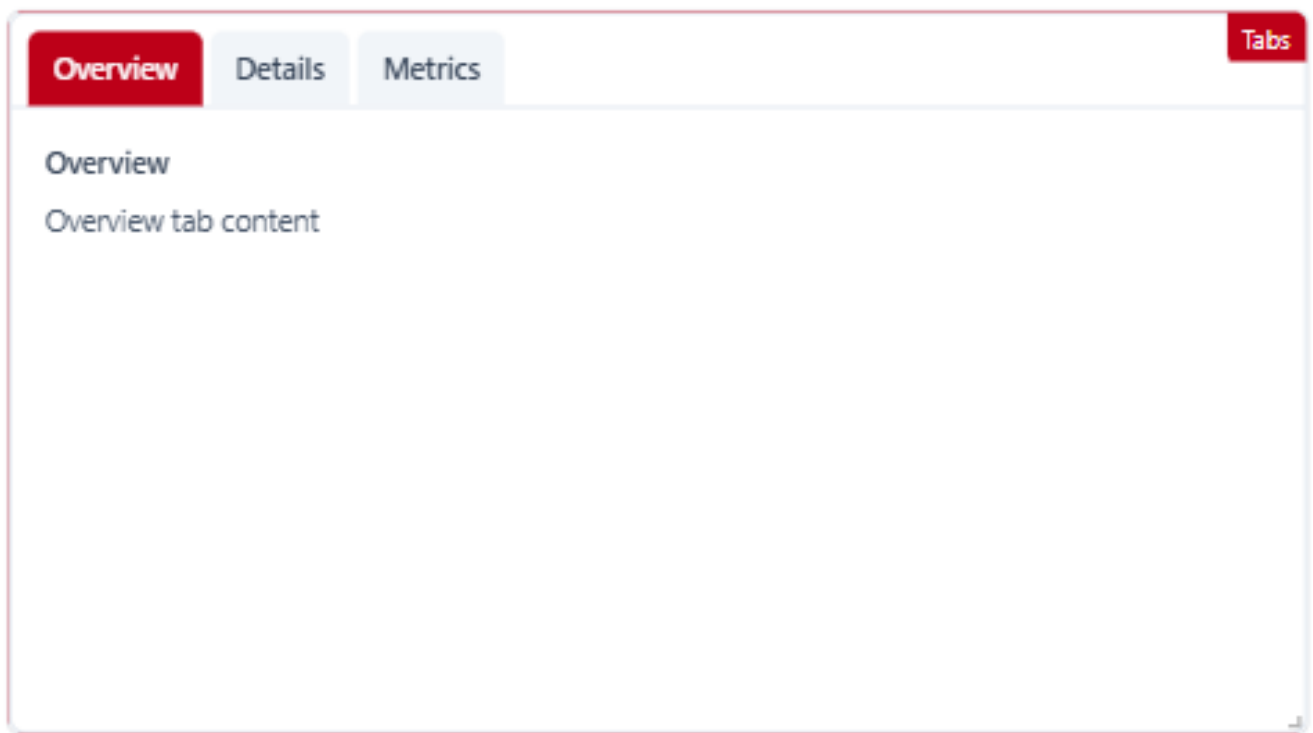
```
{
  "title": "Open incidents",
  "value": "4",
  "delta": "-2",
  "deltaDirection": "down"
},
{
  "title": "Usuarios activos",
  "value": "312",
  "delta": "+5",
  "deltaDirection": "up"
}
]
```

Nexus – Elemento: Tabs



¿Qué es?

Tabs es un componente de tipo **Layout** (estructura). Organiza contenido en pestañas de navegación, mostrando una sección de contenido a la vez según la pestaña seleccionada. Al arrastrarlo al lienzo, viene con 3 pestañas de ejemplo precargadas (Overview, Details, Metrics).



Propiedades

Al seleccionar Tabs en el lienzo, el panel derecho muestra “**Properties of Tabs**” con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. tab_79660d88).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Tabs (JSON)**: campo de código donde se define el contenido en formato JSON. Cada pestaña se compone de:

- **label:** texto visible de la pestaña.
- **value:** identificador interno de la pestaña.
- **content:** contenido que se muestra al seleccionar esa pestaña.

El editor valida el JSON en tiempo real (marca “✓ valid” cuando el formato es correcto).

- **Default Tab Value:** define qué pestaña se muestra seleccionada por defecto al cargar la pantalla, referenciando el value de esa pestaña (ej. overview).
- **Style → Variant:** menú desplegable que define el estilo visual de las pestañas. Opciones:
 - **Underline:** las pestañas se muestran como texto simple, con la pestaña activa subrayada/resaltada en color.
 - **Pill:** las pestañas se muestran como botones tipo “píldora”, con la pestaña activa marcada con fondo de color sólido.
- **Style → Custom Class:** campo para asignar una clase CSS personalizada.
- **Style → Custom Style:** campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component:** elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#).



Properties of Tabs

Component ID

tab_79660d88

Visibility

☐ Hidden on load

Use a Show action to reveal this component at runtime.

Tabs (JSON)

```
[
  {
    "label": "Overview",
    "value": "overview",
    "content": "Overview tab content"
  },
  {
    "label": "Details",
    "value": "details",
    "content": "Details tab content"
  },
  {
    "label": "Metrics",
    "value": "metrics",
    "content": "Metrics tab content"
  }
]
```

✓ valid

Default Tab Value

overview

Style

Variant

Pill

Custom Class

Custom Style

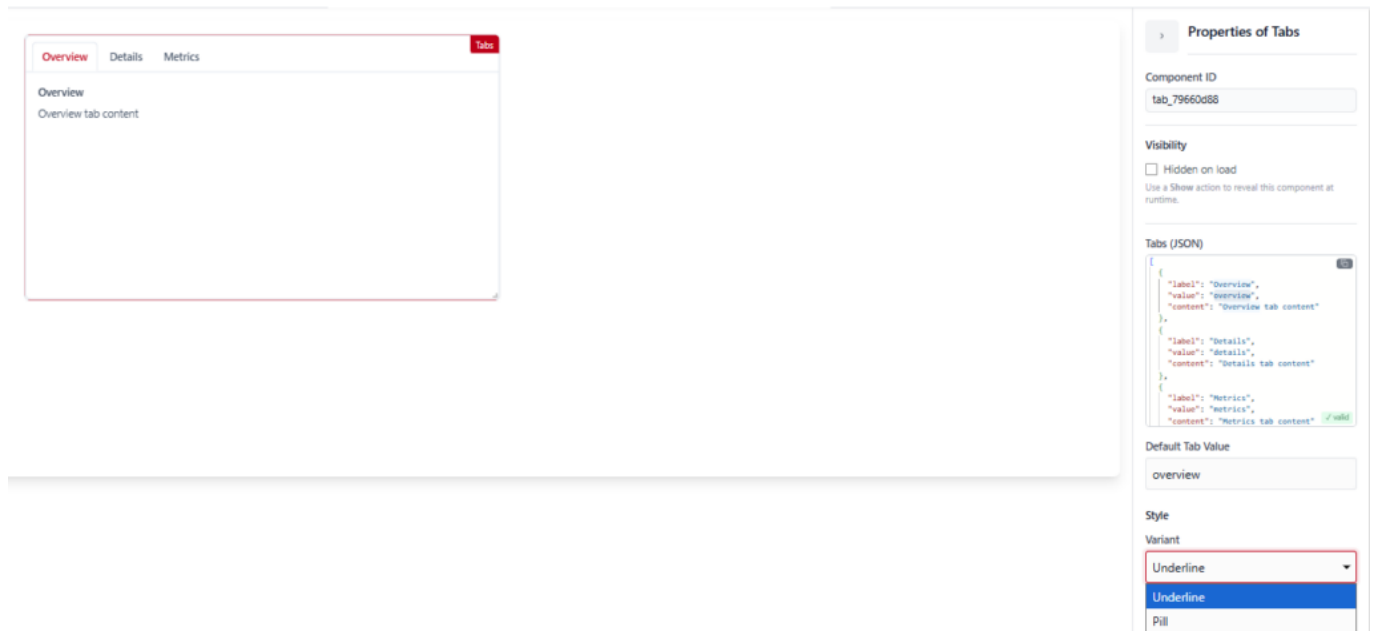
CSS

Delete Component

Events & Actions

+ Add Action

No actions defined.



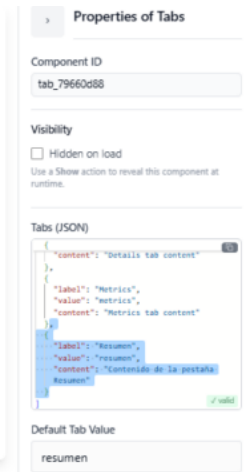
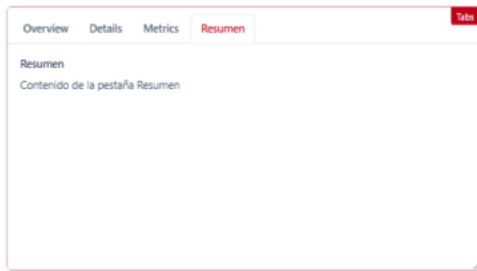
Ejemplo de uso

Para agregar una nueva pestaña, hay que ubicarse en la **última llave }** del **bloque anterior** dentro del JSON de **Tabs**, y agregarle una coma. Después, pegar el nuevo bloque a continuación. Quedaría así:

```
[
  {
    "label": "Metrics",
    "value": "metrics",
    "content": "Metrics tab content"
  },
  {
    "label": "Resumen",
    "value": "resumen",
    "content": "Contenido de la pestaña Resumen"
  }
]
```

⚠️ Importante: la coma va **pegada a la llave }** del **bloque anterior** (el último que ya existía en el JSON), no al principio del bloque nuevo. Si el bloque anterior era el último de la lista, hay que agregarle esa coma antes de pegar el nuevo bloque debajo.

Para que una pestaña específica aparezca seleccionada al abrir la pantalla, hay que colocar su value en el campo **Default Tab Value** (ej. resumen).

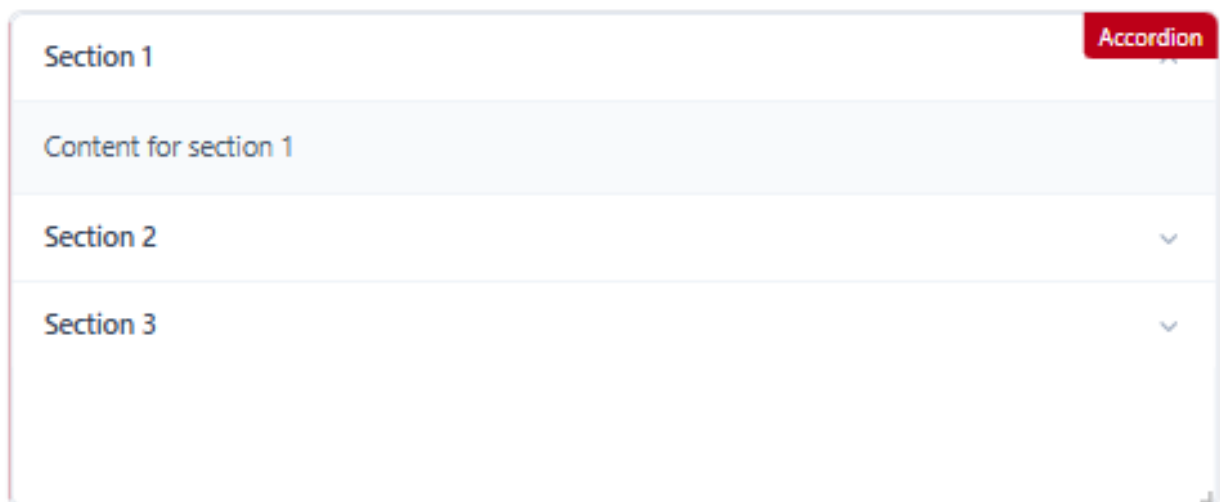


Nexus – Elemento: Accordion



¿Qué es?

El **Accordion** es un componente de tipo **Layout** (estructura). Muestra contenido organizado en **secciones colapsables**, cada una con un título y un contenido que se expande o contrae al hacer clic. Al arrastrarlo al lienzo, viene con 3 secciones de ejemplo precargadas (Section 1, Section 2, Section 3).



Propiedades

Al seleccionar el Accordion en el lienzo, el panel derecho muestra **"Properties of Accordion"** con las siguientes opciones:

- **Component ID:** identificador único del componente, autogenerado (ej. acc_d0ebb60d).
- **Visibility → Hidden on load:** casilla para que el Accordion esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).
- **Sections (JSON):** campo de código donde se define el contenido del Accordion en formato JSON. Cada sección se compone de:
 - **title:** título de la sección.
 - **content:** contenido/texto que se muestra al expandir la sección.
 - **open:** solo puede usarse en la **primera sección** del arreglo; agregarla en cualquier otra sección genera un error de validación del JSON.

El editor valida el JSON en tiempo real (se ve la marca "✓ valid" cuando el formato es correcto).

- **Behavior → Allow Multiple Open:** casilla que define si se puede tener más de una sección abierta a la vez.
 - **Desactivado:** al abrir una sección, las demás se cierran automáticamente (solo una abierta a la vez).
 - **Activado:** permite tener todas las secciones abiertas al mismo tiempo, sin límite.
- **Style → Custom Class:** campo para asignar una clase CSS personalizada.
- **Style → Custom Style:** campo de texto para CSS personalizado aplicado directamente a este componente.
- **Delete Component:** elimina el Accordion del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#).



Properties of Accordion

Component ID

acc_6c8b6984

Visibility

☐ Hidden on load

Use a [Show](#) action to reveal this component at runtime.

Sections (JSON)

```
[
  {
    "title": "Section 1",
    "content": "Content for section 1",
    "open": true
  },
  {
    "title": "Section 2",
    "content": "Content for section 2"
  },
  {
    "title": "Section 3",
    "content": "Content for section 3"
  }
]
```

✓ valid

Behavior

☐ Allow Multiple Open

Style

Custom Class

Custom Style

CSS

Delete Component

Events & Actions

+ Add Action

No actions defined.

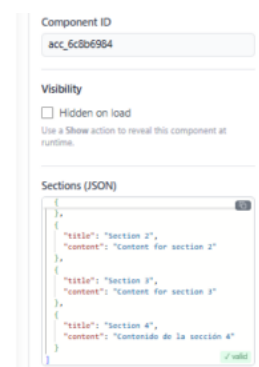
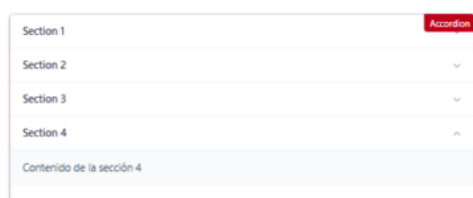
Ejemplo de uso

Para agregar una nueva sección, se copia un bloque completo dentro del JSON de **Sections** (incluyendo la coma que lo separa del bloque anterior) y se modifican sus valores de title y content. Por ejemplo, para sumar una cuarta sección:

```
[
  {
    "title": "Section 4",
    "content": "Contenido de la sección 4"
  }
]
```

⚠️ Importante: la coma va **pegada a la llave }** del bloque anterior (el último que ya existía en el JSON), no al principio del bloque nuevo. Si el bloque anterior era el último de la lista, hay que agregarle esa coma antes de pegar el nuevo bloque debajo.

El Accordion no tiene límite de 3 secciones – se pueden agregar tantas como se necesite repitiendo este bloque.

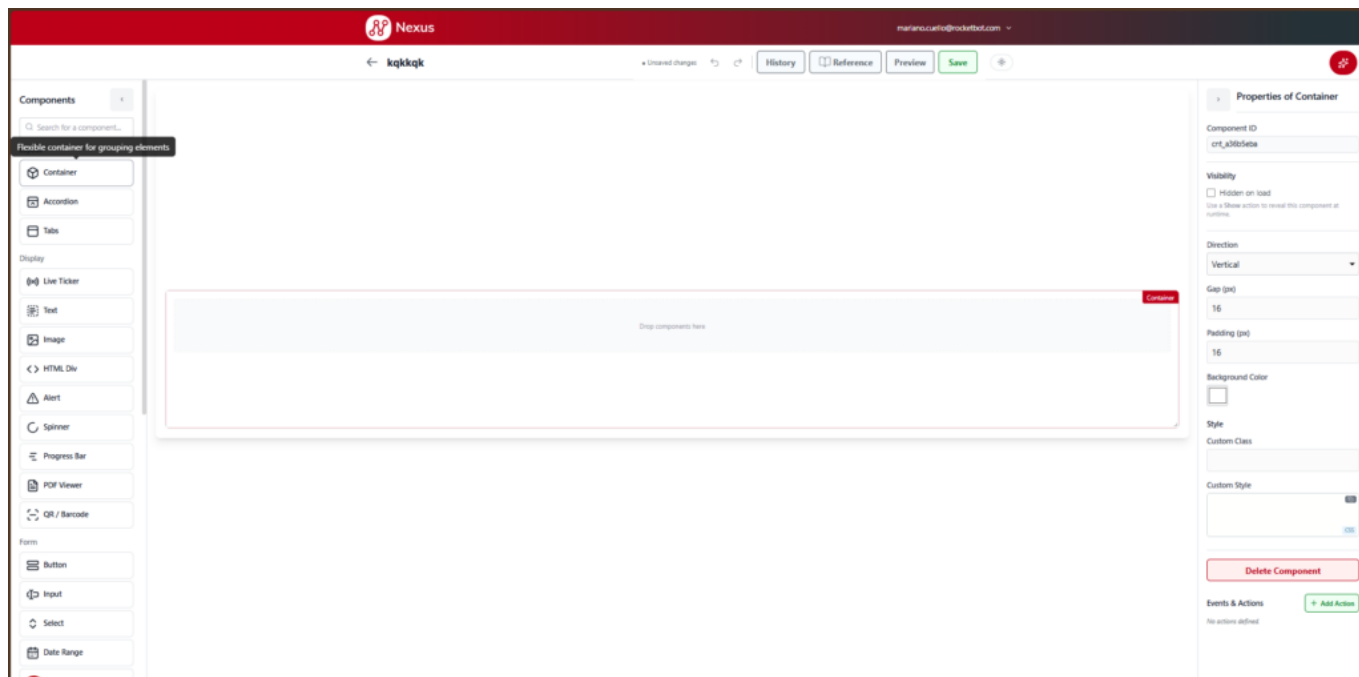


Nexus – Elemento: Container



¿Qué es?

El **Container** es un componente de tipo **Layout** (estructura). Funciona como una caja flexible para agrupar otros componentes dentro del lienzo. Al arrastrarlo, aparece como un bloque vacío con el mensaje “Drop components here”, listo para recibir otros elementos en su interior.



Propiedades

Al seleccionar el Container en el lienzo, el panel derecho muestra **"Properties of Container"** con las siguientes opciones:

- **Component ID**: identificador único del componente, autogenerado (ej. cnt_a36b5eba). Se usa para referenciarlo desde otras partes de la app (queries, acciones, scripts).
- **Visibility → Hidden on load**: casilla para que el Container esté oculto al cargar la pantalla. Puede mostrarse luego en tiempo de ejecución mediante una acción de tipo **Show** (ver sección Events & Actions).
- **Direction**: define la orientación en que se acomodan los componentes puestos dentro del Container. Opciones confirmadas: **Vertical** y **Horizontal**.
- **Gap (px)**: espacio entre los elementos internos del contenedor, en píxeles (valor por defecto: 16).
- **Padding (px)**: espacio interno entre el borde del contenedor y su contenido, en píxeles (valor por defecto: 16).
- **Background Color**: color de fondo del contenedor.
- **Style → Custom Class**: campo para asignar una clase CSS personalizada al componente.
- **Style → Custom Style**: campo de texto para escribir CSS personalizado aplicado directamente a este componente.
- **Delete Component**: botón para eliminar el Container del lienzo.
- **Events & Actions**: sistema para definir comportamientos interactivos (qué pasa cuando ocurre determinado evento sobre el componente). Ver documentación aparte: [Nexus – Events & Actions](#)

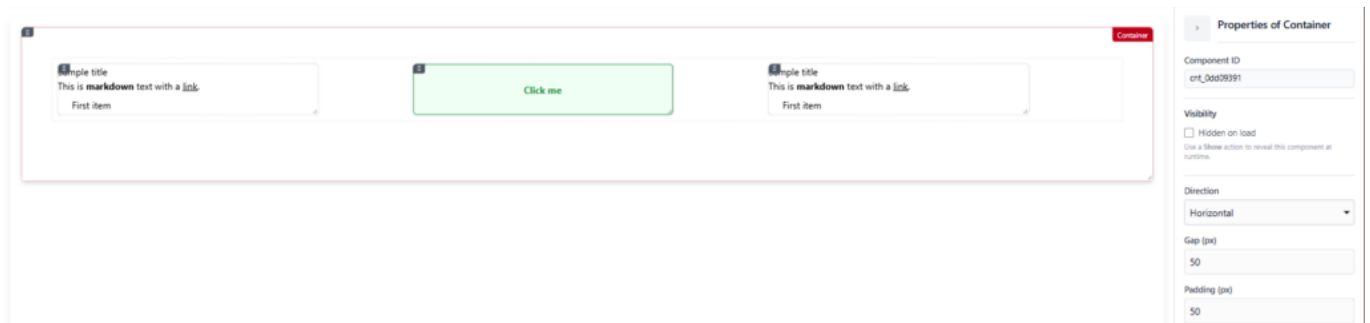
Ejemplo de uso

Al colocar varios componentes dentro del Container (por ejemplo, textos,

botones u otros elementos), estos quedan agrupados como una unidad dentro del lienzo. El panel de propiedades sigue mostrando las mismas opciones del Container aunque tenga contenido adentro.

Las propiedades **Gap** y **Padding** son las que permiten controlar visualmente el espaciado del grupo:

- Aumentar el **Gap** separa más los componentes entre sí.
- Aumentar el **Padding** agranda el espacio entre el borde del Container y los componentes de adentro, empujándolos hacia el centro.



Nexus – Screens



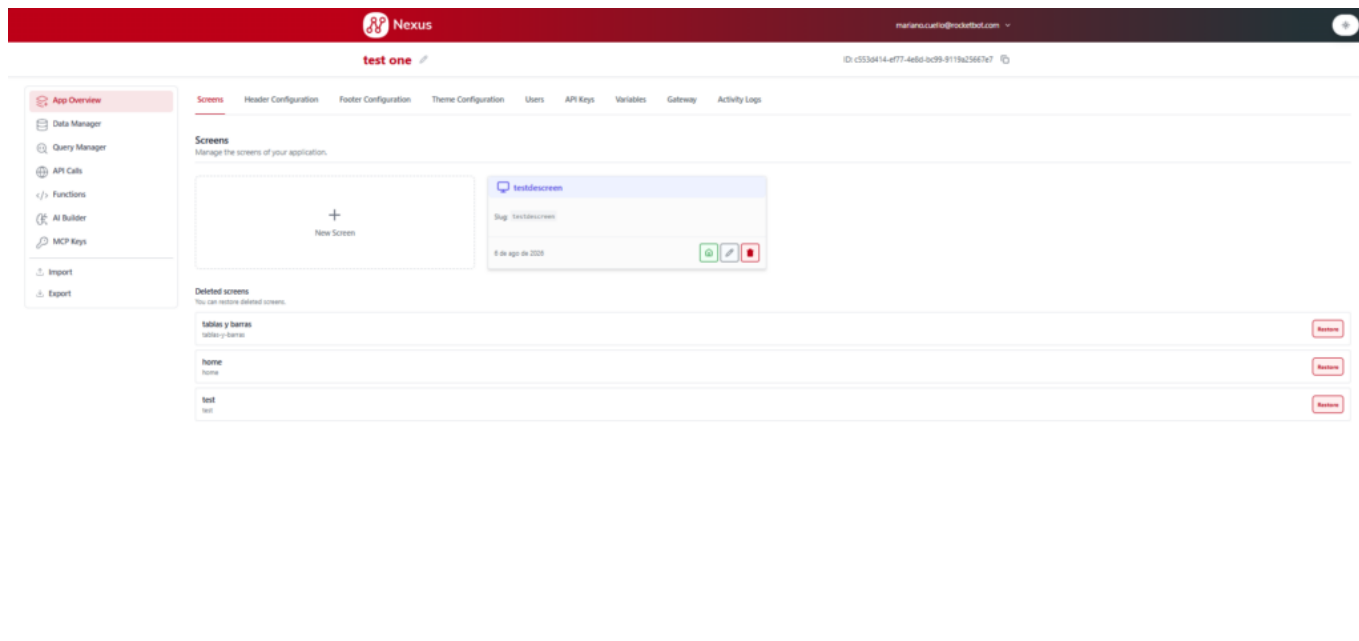
Las **Screens** (pantallas) son las vistas individuales que componen una aplicación en Nexus. Cada aplicación puede tener múltiples screens, construidas mediante un editor visual de tipo “arrastrar y soltar”.

1. Gestión de Screens

La sección **Screens** muestra el listado de pantallas existentes, con opción de crear una nueva. También hay un apartado de **Deleted screens**, con botón **Restore** para recuperarlas.

Cada screen en el listado muestra nombre, slug, fecha de última edición, y tres acciones:

- **Ícono de casa:** convierte a esa screen en la pantalla principal (Home Screen) de la app.
- **Ícono de lápiz:** editar la screen.
- **Ícono de tachito:** eliminar la screen.



2. Crear una nueva Screen

Al hacer clic en “+ New Screen” se abre un formulario:

- **Screen Name:** nombre de la pantalla.
- **Slug:** identificador único (autogenerado, editable).
- **Set as default screen:** define esta pantalla como principal desde su creación.
- **Initial Template:** permite partir de una plantilla en vez de un lienzo vacío:
 - *None (empty canvas):* lienzo vacío.
 - *Analytics Dashboard:* pantalla de ejemplo con filtros, tarjetas de estadísticas (Revenue, Orders, Conversion) y gráfico de barras.
 - *Admin CRUD:* pantalla de ejemplo con buscador/filtros, tabla de administración con acciones por fila y paginación.
 - *Landing + CTA:* pantalla tipo landing, con título, texto, botón de acción e imagen/preview.

Create New Screen



Screen Name *

e.g., Home, Dashboard, Profile

Slug

auto-generated from name

Leave empty to auto-generate from name.

☐ Set as default screen

The first screen users see when opening the app

Initial Template

None (empty canvas)

Optional prebuilt layout to speed up screen creation

Cancel

Create Screen

Initial Template

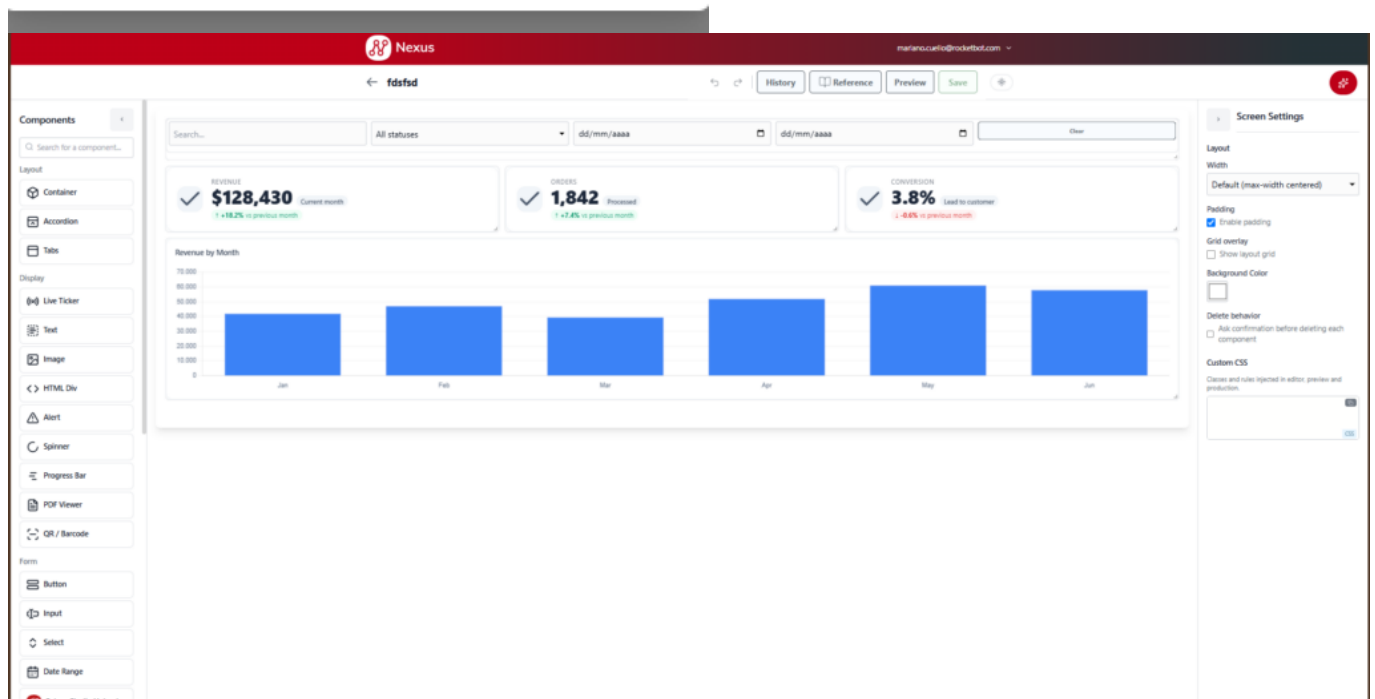
None (empty canvas)

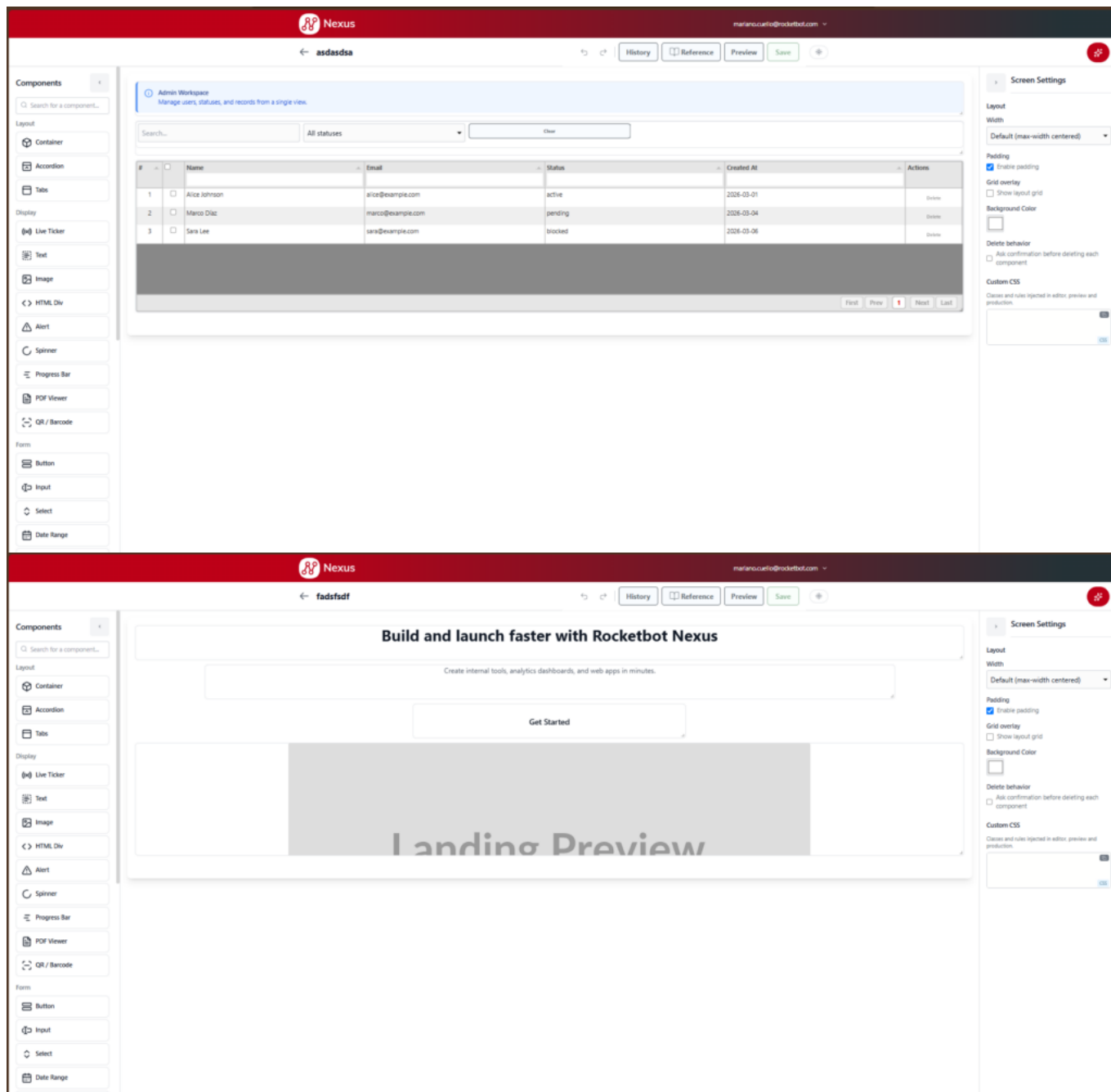
None (empty canvas)

Analytics Dashboard

Admin CRUD

Landing + CTA

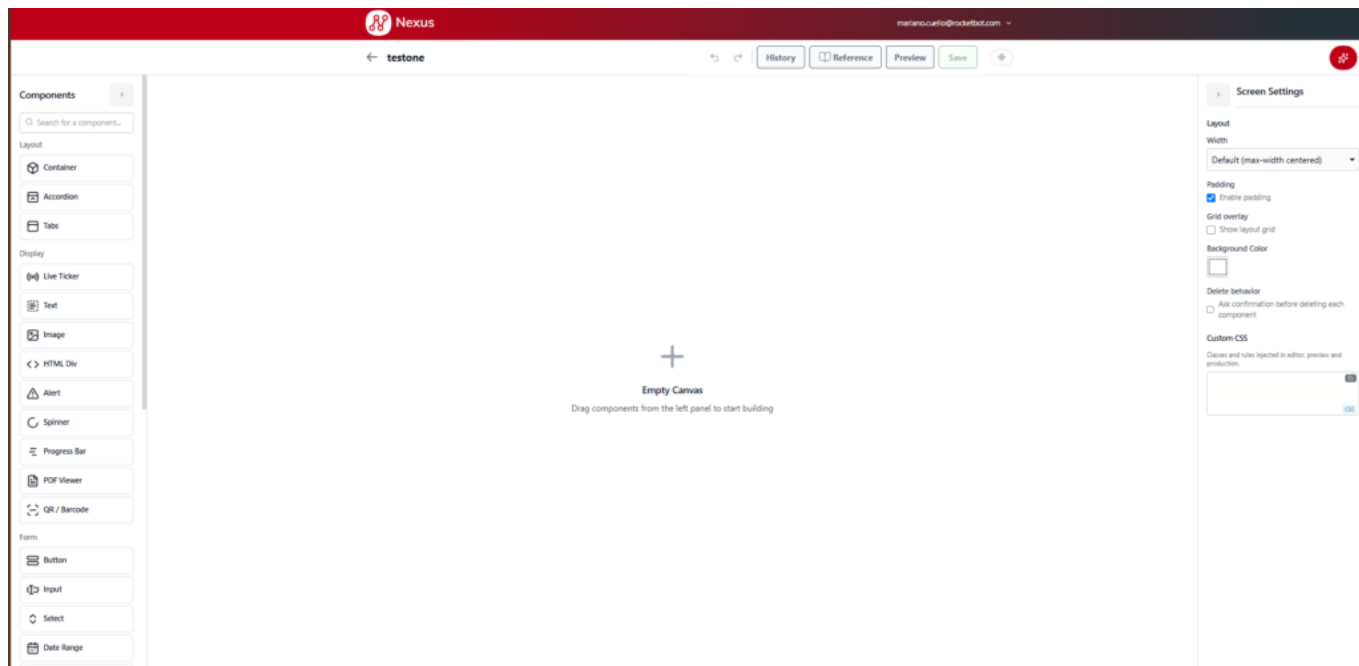




3. El Editor de Screens

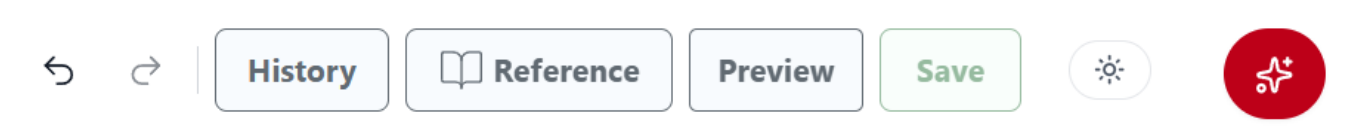
El editor se divide en tres áreas:

- **Izquierda – Components:** panel con todos los elementos disponibles para arrastrar al lienzo, organizados por categorías (Layout, Display, Form, Data).
- **Centro – Lienzo (Canvas):** área de diseño donde se arma la pantalla.
- **Derecha – Propiedades:** al no haber nada seleccionado, muestra **Screen Settings** (ancho, padding, grid overlay, color de fondo, comportamiento de borrado, CSS personalizado). Al seleccionar un componente del lienzo, muestra las propiedades específicas de ese componente.



Herramientas de la barra superior:

- **History:** deshacer/rehacer cambios.
- **Reference:** panel con funciones, queries, variables y componentes disponibles (\$fn, \$query, \$var, \$comp).
- **Preview:** vista previa de la pantalla.
- **Save:** guardar cambios.
- **Ícono sol/luna:** alterna modo claro/oscuro del editor.
- **Ícono circular rojo (estrella):** abre un asistente de IA (chatbot) dentro del editor.



4. Componentes disponibles

Cada componente, al arrastrarse al lienzo, despliega en el panel derecho sus propias **propiedades de configuración** (y en muchos casos, la opción de [Events & Actions](#) para definir comportamientos interactivos). El detalle de configuración de cada uno se documenta por separado.

Layout

- **Container:** agrupa otros componentes en un bloque. ([documentación detallada](#))
- **Accordion:** contenido organizado en secciones expandibles. ([documentación detallada](#))
- **Tabs:** organiza contenido en pestañas. ([documentación detallada](#))

Display

- **Live Ticker:** cinta de contenido en movimiento. ([documentación detallada](#))
- **Text:** bloque de texto. ([documentación detallada](#))
- **Image:** inserción de imágenes. ([documentación detallada](#))
- **HTML Div:** bloque de HTML personalizado. ([documentación detallada](#))
- **Alert:** mensaje de aviso/notificación. ([documentación detallada](#))
- **Spinner:** indicador de carga. ([documentación detallada](#))
- **Progress Bar:** barra de progreso. ([documentación detallada](#))
- **PDF Viewer:** visor de archivos PDF. ([documentación detallada](#))
- **QR / Barcode:** componente relacionado a códigos QR o de barras. ([documentación detallada](#))

Form

- **Button:** botón de acción. ([documentación detallada](#))
- **Input:** campo de entrada de texto. ([documentación detallada](#))
- **Select:** menú de selección desplegable. ([documentación detallada](#))
- **Date Range:** selector de rango de fechas. ([documentación detallada](#))
- **Saturn Studio Upload:** carga de archivos vinculada a Saturn Studio. ([documentación detallada](#))
- **S3 File Upload:** carga de archivos a almacenamiento S3. ([documentación detallada](#))

Data

- **Treemap:** visualización jerárquica de datos. ([documentación detallada](#))
- **Event Timeline:** línea de tiempo de eventos. ([documentación detallada](#))
- **Geo Map:** mapa geográfico. ([documentación detallada](#))
- **Filter Bar:** barra de filtros. ([documentación detallada](#))
- **Stat Card:** tarjeta con un dato estadístico. ([documentación detallada](#))
- **Table:** tabla de datos. ([documentación detallada](#))
- **Table Data:** variante de tabla. ([documentación detallada](#))
- **Chart:** gráfico de datos. ([documentación detallada](#))
- **Sparkline:** mini gráfico de tendencia. ([documentación detallada](#))
- **List Card:** tarjeta con listado de ítems. ([documentación detallada](#))
- **Activity Feed:** feed de actividad reciente. ([documentación detallada](#))
- **Gauge:** medidor tipo velocímetro. ([documentación detallada](#))
- **Funnel Chart:** gráfico de embudo. ([documentación detallada](#))
- **KPI Card:** tarjeta de indicador clave. ([documentación detallada](#))
- **Heatmap Calendar:** calendario tipo mapa de calor. ([documentación detallada](#))
- **Calendar:** calendario. ([documentación detallada](#))
- **Data Container:** contenedor para componentes de datos. ([documentación detallada](#))

Nexus – Events & Actions



¿Qué es?

Events & Actions es el sistema de Nexus para definir comportamientos interactivos en los componentes. Se encuentra en la pestaña **Behavior** de cada componente, mediante el botón + **Add Action**. Al agregarla, se abre un panel “New Action” donde se configura un disparador (Trigger) y una acción a ejecutar (Action Type). Según el Action Type elegido, el panel muestra dinámicamente los campos correspondientes a esa acción.

Trigger

Menú desplegable que define qué evento dispara la acción. Opciones disponibles:

- **On Click:** se ejecuta al hacer clic sobre el componente.
- **On Load:** se ejecuta al cargar el componente/pantalla.
- **On Change:** se ejecuta cuando cambia el valor del componente.
- **On Event:** se ejecuta ante un evento personalizado.

Action Type

Menú desplegable con el tipo de acción a ejecutar. Opciones disponibles:

- Navigate
- Show Alert
- Execute Query
- Execute API Call
- Set Value
- Run Script
- Show Components
- Hide Components
- Enable Components
- Disable Components
- Refresh Component

Cada una de estas opciones despliega campos adicionales específicos, detallados a continuación.

Navigate

- **Target URL / Path:** ruta o URL de destino a la que navega al ejecutarse la acción (ej. “/dashboard”).

Show Alert

- **Message:** texto del mensaje a mostrar (ej. “Operation successful”).
- **Type:** menú desplegable con el tipo de alerta, que determina su estilo visual. Opciones: –Select a type–, Info, Success, Warning, Error.

Execute Query

- **Query:** menú desplegable donde se selecciona la consulta a ejecutar (placeholder “Select a query...”).
- **Parameters:** sección para definir los parámetros que recibe la consulta. Con el botón + **Add** se agrega una fila por parámetro, compuesta por:
 - **field (column):** nombre del campo/columna al que corresponde el parámetro.
 - Valor asociado: referencia al dato que se pasa a ese campo (ej. \$comp.inp_1, tomando el valor de otro componente).
 - Cada fila tiene un ícono de papelera para eliminarla.
- **If executed successfully (✓):** bloque para configurar, mediante un botón “+”, una o más acciones que se ejecuten automáticamente si la query se ejecuta con éxito (por defecto: “No actions configured”).
- **If there is an error (×):** bloque equivalente para configurar acciones que se ejecuten si la query falla.

Execute API Call

- **API Call:** menú desplegable donde se selecciona la llamada a API a ejecutar (placeholder “Select an API call...”).
- **Parameters:** sección para definir los parámetros que recibe la API call. Con el botón + se agrega una fila por parámetro, compuesta por:
 - **param name:** nombre del parámetro.
 - Valor asociado: referencia al dato que se pasa a ese parámetro (ej. \$comp.inp_1).
 - Cada fila tiene un ícono de papelera para eliminarla.

- **If executed successfully (✓):** bloque para configurar, mediante un botón "+", acciones que se ejecuten automáticamente si la API call responde con éxito.
- **If there is an error (×):** bloque equivalente para configurar acciones que se ejecuten si la API call falla.

Set Value

- **Assignments:** sección para definir las asignaciones de valor a ejecutar. Con el botón + **Add** se agrega un bloque **TARGET COMPONENT**, compuesto por:
 - **Select target component...:** menú desplegable que lista los componentes disponibles en el lienzo (ej. "cnt_01bd8b73 (Container)").
 - **PROPERTY:** propiedad del componente a modificar (ej. "value").
 - **VALUE / EXPR:** valor o expresión a asignar (placeholder "\$comp.inp_1 or text"), pudiendo ser un valor fijo o una referencia al dato de otro componente.
 - Cada bloque tiene un ícono de papelera para eliminarlo.
- **+ Add another assignment:** permite agregar múltiples bloques TARGET COMPONENT, para asignar valores a varios componentes en una sola acción.

Run Script

- **Function:** menú desplegable donde se selecciona la función/script a ejecutar (placeholder "Select a function...").
- **Parameters:** sección para definir los parámetros que recibe la función. Con el botón + se agrega una fila por parámetro, compuesta por:
 - **param name:** nombre del parámetro.
 - Valor asociado: referencia al dato que se pasa a ese parámetro (ej. \$comp.inp_1).
 - Cada fila tiene un ícono de papelera para eliminarla.
- **If executed successfully (✓):** bloque (resaltado en verde) para configurar acciones que se ejecuten si la función termina con éxito.
- **If there is an error (×):** bloque (resaltado en rojo) para configurar acciones que se ejecuten si la función falla.

Show Components

Muestra componentes que estaban ocultos (por ejemplo, los que tienen activada

la casilla “Hidden on load”).

- **Components:** sección donde se eligen los componentes a mostrar. Con el botón + se agrega una fila por componente, compuesta por:
 - **Select component...:** menú desplegable que lista los componentes disponibles en el lienzo (ej. “cal_03579022 (Calendar)”).
 - Un ícono de papelera para eliminar la fila.
- Si no hay ningún componente seleccionado, el panel indica “No components selected. Add at least one.” (hay que agregar al menos uno).
- **From a Function:** Nexus muestra un ejemplo de código de referencia (no es un campo configurable). Sirve para ejecutar esta misma acción desde un script de tipo Run Script, devolviendo lo siguiente:

```
return {
  _uiActions: [
    {
      type: 'Show',
      componentIds: ['id1']
    }
  ]
};
```

Hide Components

Ocultar componentes que están visibles en pantalla.

- **Components:** sección donde se eligen los componentes a ocultar. Con el botón + se agrega una fila por componente, compuesta por:
 - Un menú desplegable que lista los componentes disponibles en el lienzo (ej. “cal_03579022 (Calendar)”).
 - Un ícono de papelera para eliminar la fila.
- **From a Function:** Nexus muestra un ejemplo de código de referencia (no es un campo configurable). Sirve para ejecutar esta misma acción desde un script de tipo Run Script, devolviendo lo siguiente:

```
return {
  _uiActions: [
    {
      type: 'Hide',
```

```
        componentIds: ['id1']
    }
]
};
```

Enable Components

Habilita componentes que estaban deshabilitados, para que vuelvan a poder usarse.

- **Components:** misma estructura que Hide Components (botón +, desplegable de componentes y papelera para eliminar cada fila), con el texto de ayuda “Select the components to enable.”
- **From a Function:** Nexus muestra un ejemplo de código de referencia (no es un campo configurable). Sirve para ejecutar esta misma acción desde un script de tipo Run Script, devolviendo lo siguiente:

```
return {
  _uiActions: [
    {
      type: 'Enable',
      componentIds: ['id1']
    }
  ]
};
```

Disable Components

Deshabilita componentes para que no puedan usarse (por ejemplo, evitar que se interactúe con ellos).

- **Components:** misma estructura que Hide Components, con el texto de ayuda “Select the components to disable.”
- **From a Function:** Nexus muestra un ejemplo de código de referencia (no es un campo configurable). Sirve para ejecutar esta misma acción desde un script de tipo Run Script, devolviendo lo siguiente:

```
return {
  _uiActions: [
    {
      type: 'Disable',
      componentIds: ['id1']
    }
  ]
};
```

Refresh Component

Vuelve a cargar componentes, para que actualicen su información.

- **Components to Refresh:** sección donde se eligen los componentes a refrescar. Con el botón + **Add** se agrega una fila por componente, compuesta por:
 - Un menú desplegable que lista los componentes disponibles en el lienzo (ej. "cal_03579022 (Calendar)").
 - Un ícono de papelera para eliminar la fila.
- Si no se selecciona ningún componente, se refrescan todos los componentes de la pantalla ("Leave empty to refresh all components on the screen").