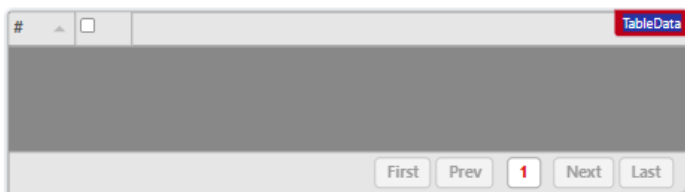


Nexus – Elemento: Table Data



¿Qué es?

Table Data es un componente de tipo **Data**. A diferencia de **Table** (que se carga manualmente con JSON), este componente se conecta directamente a una fuente de datos, ya sea una base de datos en vivo o datos estáticos. Al arrastrarlo al lienzo aparece vacío hasta que se selecciona una tabla de origen.



Propiedades

Al seleccionar Table Data en el lienzo, el panel derecho muestra “**Properties of TableData**”, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. tbd_90006ad2).
- **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).

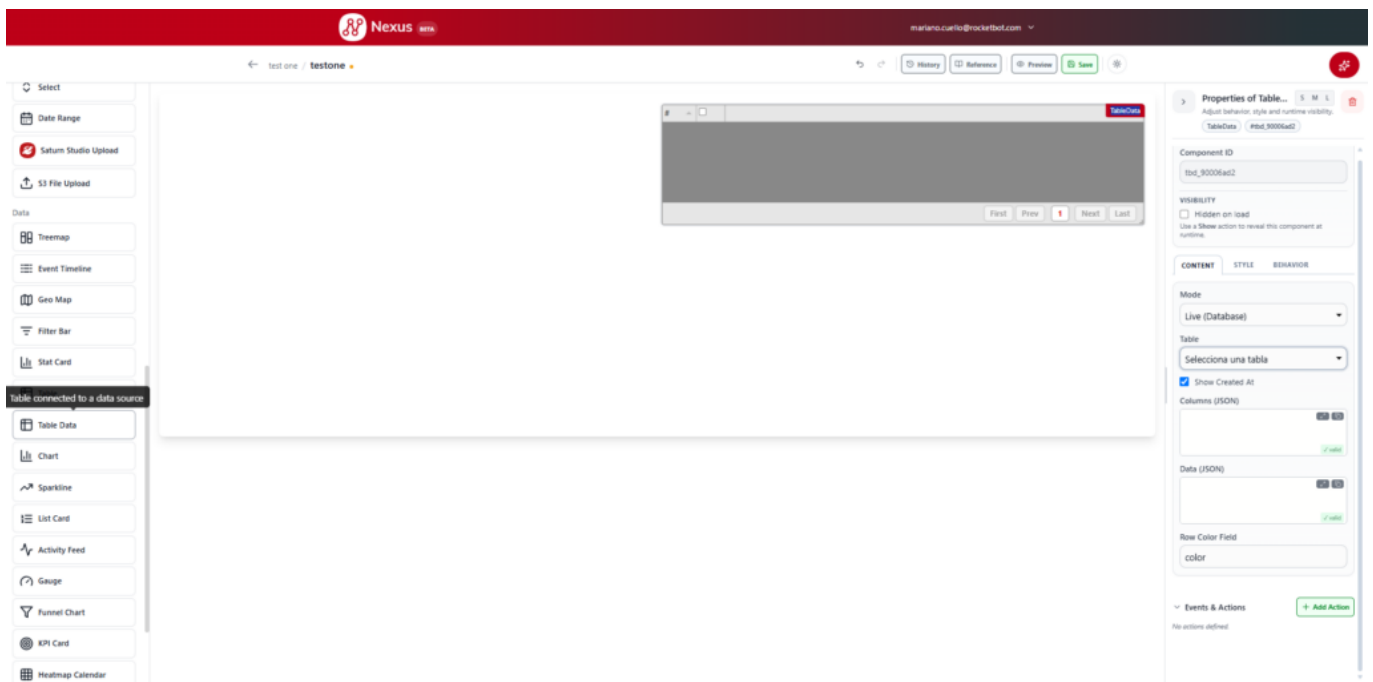
Pestaña Content:

- **Mode**: menú desplegable que define el origen de los datos. Opciones:
 - **Live (Database)**: conecta la tabla a una base de datos real de la aplicación (por ejemplo, del **Data Manager**). Requiere seleccionar una tabla en el campo **Table**; si no se selecciona ninguna, el componente no muestra nada.
 - **Static (JSON)**: carga los datos manualmente mediante JSON, de forma similar al componente **Table**.

- **Table**: menú desplegable para elegir la tabla de la base de datos a

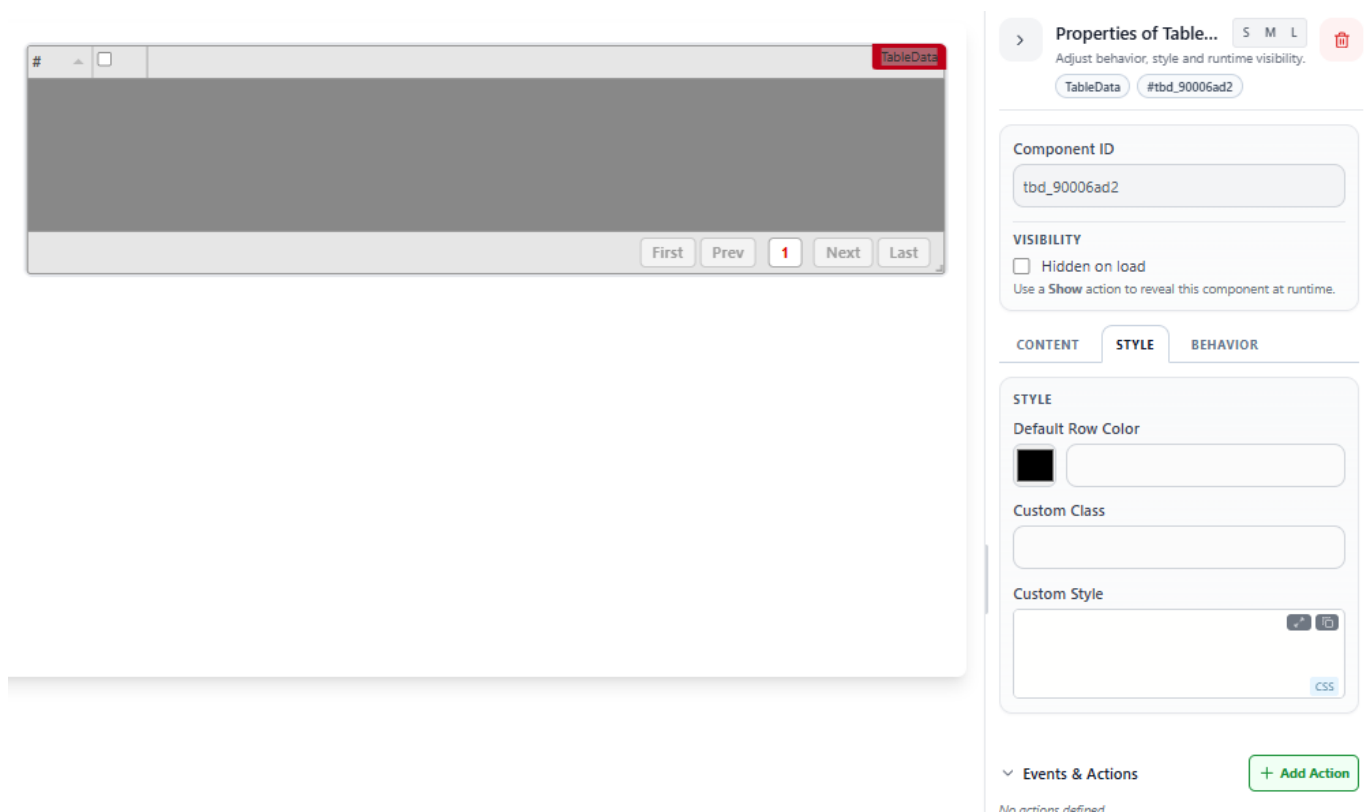
mostrar (solo aplica en modo Live).

- **Show Created At:** casilla que muestra u oculta una columna con la fecha de creación de cada registro.
- **Columns (JSON):** define las columnas a mostrar (mismo formato field / header que en Table).
- **Data (JSON):** define los datos manualmente (solo aplica en modo Static).
- **Row Color Field:** nombre del campo que define el color de cada fila (igual que en Table).



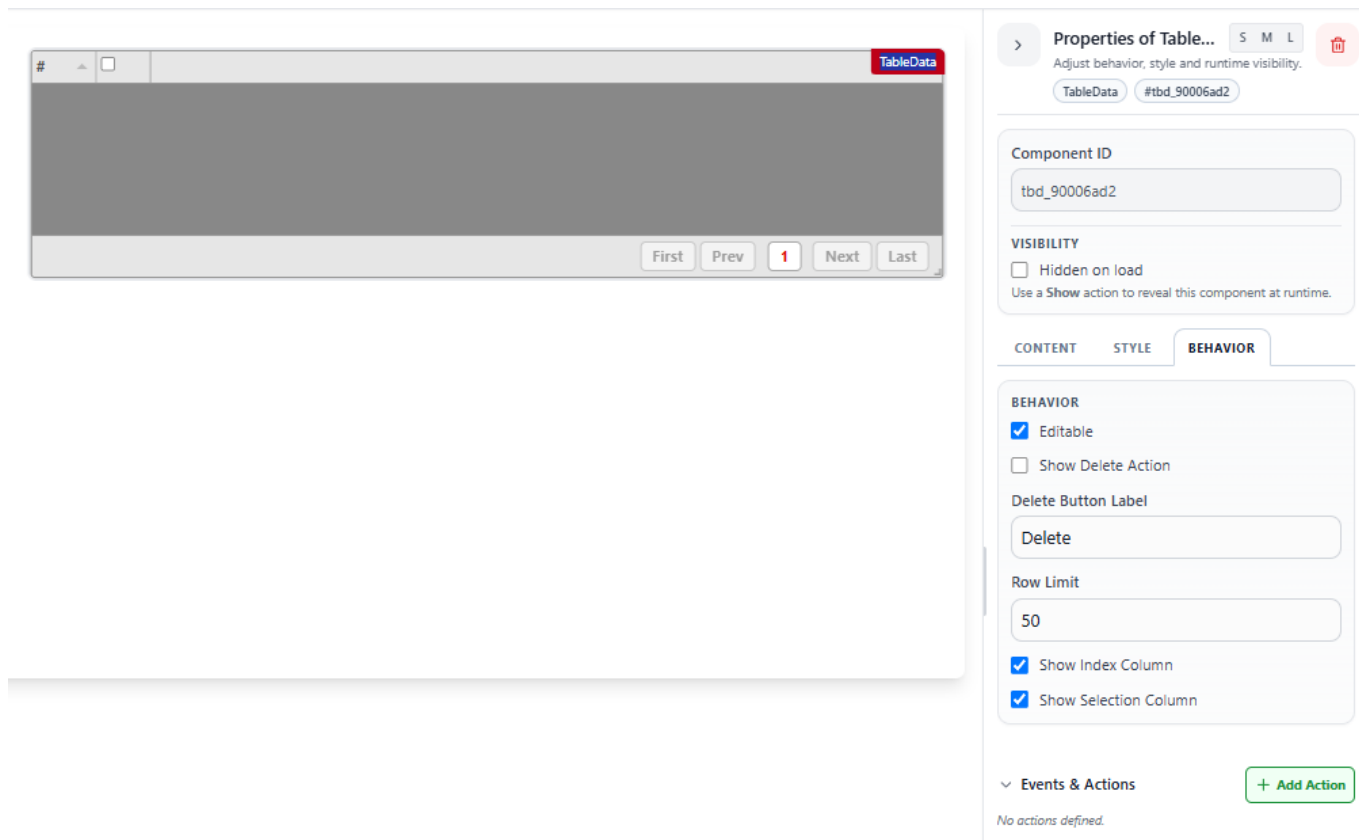
Pestaña Style:

- **Default Row Color:** color por defecto de las filas.
- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Pestaña Behavior:

- **Editable:** casilla que permite editar los valores de la tabla directamente desde la interfaz.
- **Show Delete Action:** casilla que agrega una acción de borrado por fila.
- **Delete Button Label:** texto del botón de borrado (ej. "Delete").
- **Row Limit:** cantidad máxima de filas mostradas por página (valor por defecto: 50).
- **Show Index Column:** casilla que muestra u oculta la columna de numeración (#).
- **Show Selection Column:** casilla que muestra u oculta la columna de checkboxes de selección.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#).
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.



Diferencias con Table

	Table	Table Data
Origen de datos	Solo JSON manual (estático)	Base de datos en vivo (Live) o JSON estático
Edición inline	No	Sí (con "Editable")
Acción de borrado por fila	No	Sí (con "Show Delete Action")
Paginación configurable (Row Limit)	No	Sí
Columna de índice / selección	No	Sí (configurables)

Ejemplo de uso

Para mostrar datos reales de la aplicación (por ejemplo, una tabla de usuarios cargada en **Data Manager**), se elige **Mode: Live (Database)** y se selecciona la tabla correspondiente en **Table** – sin necesidad de cargar nada manualmente en JSON. Para pruebas rápidas o datos que no cambian, se puede usar **Mode: Static (JSON)** y cargar los datos a mano, igual que en el componente Table. Activando **Editable** y **Show Delete Action**, la tabla se convierte en una mini interfaz de administración (CRUD básico) directamente sobre los datos.

Nexus – Elemento: Table



¿Qué es?

Table es un componente de tipo **Data**. Muestra una tabla de datos con columnas configurables. Al arrastrarlo al lienzo viene con un ejemplo precargado (columnas Name, Email, Status, con 2 filas de datos).

NAME	EMAIL	STATUS
John Doe	john@example.com	Active
Jane Smith	jane@example.com	Inactive

Propiedades

Al seleccionar Table en el lienzo, el panel derecho muestra “**Properties of Table**”, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. tbl_4d35b0e6).
- **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).

Pestaña Content:

- **Columns (JSON)**: campo de código donde se definen las columnas de la tabla en formato JSON. Cada columna se compone de:
 - **field**: nombre del campo de datos que representa (ej. “name”).
 - **header**: título visible de la columna (ej. “Name”).Cuenta con el editor de código expandido para más comodidad.
- **Data (JSON)**: campo de código donde se cargan los datos/filas de la tabla en formato JSON. Cada fila es un objeto con las claves correspondientes a los field definidos en Columns (ej. name, email, status).
- **Row Color Field**: nombre del campo de datos (dentro de cada fila en Data)

que define el color de esa fila (ej. color). El editor valida el JSON en tiempo real (marca “✓ valid” cuando el formato es correcto).

The screenshot displays the Nexus Studio interface. On the left is a sidebar with various component categories: Select, Date Range, Saturn Studio Upload, S3 File Upload, Data (Treemap, Event Timeline, Geo Map, Filter Bar), and a highlighted section 'Data table with configurable columns' containing Table, Table Data, Chart, Sparkline, List Card, Activity Feed, Gauge, Funnel Chart, KPI Card, and Heatmap Calendar. The main workspace shows a table with the following data:

NAME	EMAIL	STATUS
John Doe	john@example.com	Active
Jane Smith	jane@example.com	Inactive

On the right, the 'Properties of Table' panel is open, showing configuration options for the 'Table' component (ID: tbl_Act330b6). It includes sections for VISIBILITY (Hidden on load), CONTENT (Columns (JSON) and Data (JSON)), and a Row Color Field set to 'color'. The JSON fields are marked as '✓ valid'. The bottom right corner has an 'Events & Actions' section with an 'Add Action' button.

Pestaña Style:

- **Default Row Color:** color por defecto de las filas (usado si no se especifica Row Color Field por fila, o como color de fallback).
- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.

↶ ↷

History

Reference

Preview

Save

⚙

Table

NAME	EMAIL	STATUS
John Doe	john@example.com	Active
Jane Smith	jane@example.com	Inactive

>

Properties of Table

S M L

🗑

Adjust behavior, style and runtime visibility.

Table #tbl_4d35b0e6

Component ID

tbl_4d35b0e6

VISIBILITY

☐ Hidden on load

Use a **Show** action to reveal this component at runtime.

CONTENT

STYLE

BEHAVIOR

STYLE

Default Row Color

Custom Class

Custom Style

CSS

▼ Events & Actions

+ Add Action

No actions defined.

Pestaña Behavior:

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

↶ ↷

History

Reference

Preview

Save

⚙

Table

NAME	EMAIL	STATUS
John Doe	john@example.com	Active
Jane Smith	jane@example.com	Inactive

>

Properties of Table

S M L

🗑

Adjust behavior, style and runtime visibility.

Table #tbl_4d35b0e6

Component ID

tbl_4d35b0e6

VISIBILITY

☐ Hidden on load

Use a **Show** action to reveal this component at runtime.

CONTENT

STYLE

BEHAVIOR

▼ Events & Actions

+ Add Action

No actions defined.

Ejemplo de uso

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

```
[
  {
    "field": "status",
    "header": "Status"
  },
  {
    "field": "createdAt",
    "header": "Created At"
  }
]
```

Para agregar una nueva fila de datos, se hace lo mismo dentro de **Data (JSON)**, asegurándose de incluir todas las claves que coincidan con los `field` definidos en `Columns`:

```
[
  {
    "name": "Jane Smith",
    "email": "jane@example.com",
    "status": "Inactive"
  },
  {
    "name": "Carlos Ruiz",
    "email": "carlos@example.com",
    "status": "Active"
  }
]
```

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

NAME	EMAIL	STATUS
John Doe	john@example.com	Active
Jane Smith	jane@example.com	Inactive
Carlos Ruiz	carlos@example.com	Active

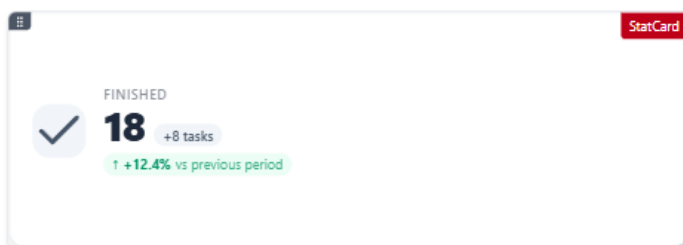
Con **Row Color Field**, si cada objeto de **Data** incluye una clave adicional (por ejemplo "color": "#fee2e2"), esa fila específica toma ese color en vez del color por defecto – útil para resaltar filas según su estado (por ejemplo, en rojo las filas con status "Inactive").

Nexus – Elemento: Stat Card



¿Qué es?

Stat Card es un componente de tipo **Data**. Muestra una tarjeta de estadística con un valor destacado, ícono, y una tendencia opcional respecto a un período anterior. Al arrastrarlo al lienzo viene con un ejemplo precargado ("Finished", valor 18, con tendencia "+12.4% vs previous period").



Propiedades

Al seleccionar Stat Card en el lienzo, el panel derecho muestra "**Properties of StatCard**", organizado en tres pestañas: **Content**, **Style** y **Behavior**.

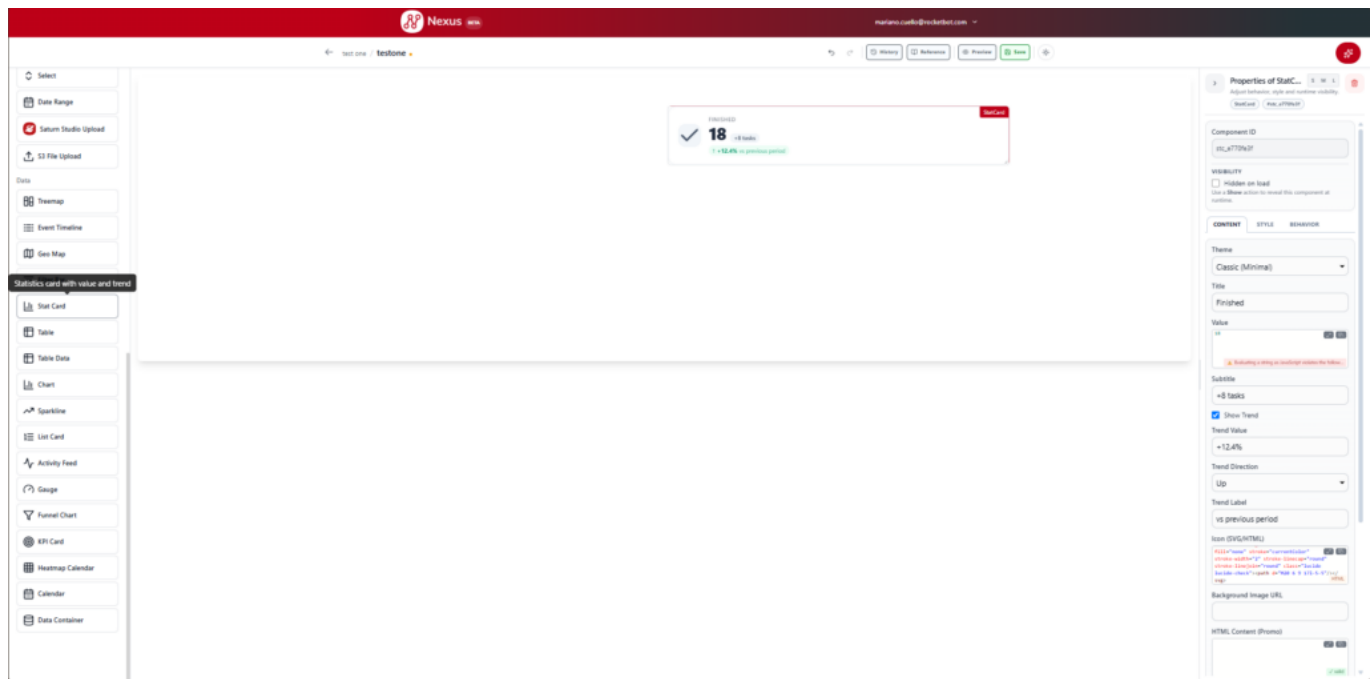
- **Component ID**: identificador único del componente, autogenerado (ej.

stc_e770fe3f).

- **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).

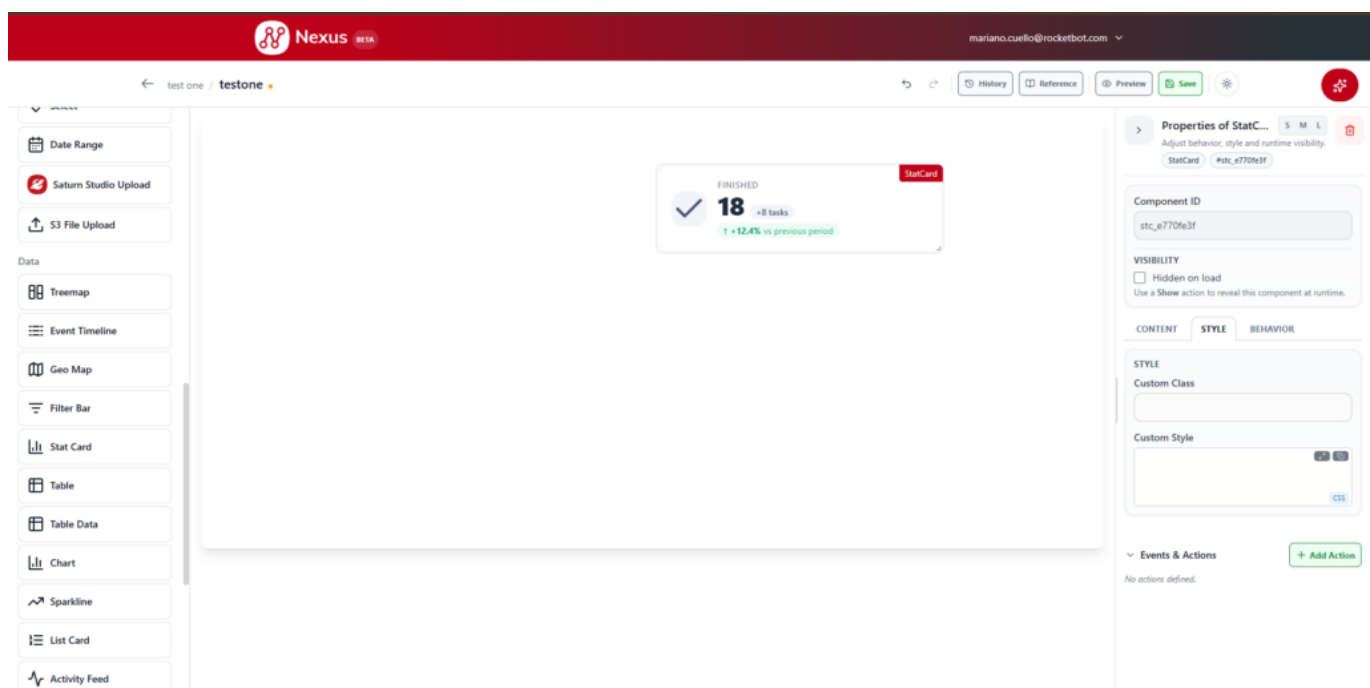
Pestaña Content:

- **Theme:** menú desplegable que define el estilo visual completo de la tarjeta. Opciones:
 - **Classic (Minimal)**
 - **Modern (Gradient)**
 - **Split (Accent Strip)**
 - **Glass (Frosted)**
 - **Metric (KPI)**
 - **Promo (Banner)**
- **Title:** título de la tarjeta (ej. "Finished").
- **Value:** valor principal destacado (ej. "18"). Cuenta con el editor de código expandido para más comodidad.
- **Subtitle:** texto secundario debajo del valor (ej. "+8 tasks").
- **Show Trend:** casilla que muestra u oculta el bloque de tendencia.
- **Trend Value:** valor de la tendencia (ej. "+12.4%").
- **Trend Direction:** menú desplegable que define la flecha/ícono de tendencia. Opciones: **Up, Down, Neutral**.
- **Trend Label:** texto que acompaña a la tendencia (ej. "vs previous period").
- **Icon (SVG/HTML):** campo de código donde se pega un ícono en formato SVG (por ejemplo, íconos de la librería Lucide). Cuenta con el editor de código expandido.
- **Background Image URL:** URL de una imagen de fondo para la tarjeta.
- **HTML Content (Promo):** campo de código HTML adicional, pensado para el Theme "Promo (Banner)".
- **Background Color (Override):** color de fondo que sobrescribe el definido por el Theme elegido.
- **Text Color (Override):** color de texto que sobrescribe el del Theme.
- **Icon BG Color (Override):** color de fondo del ícono que sobrescribe el del Theme.



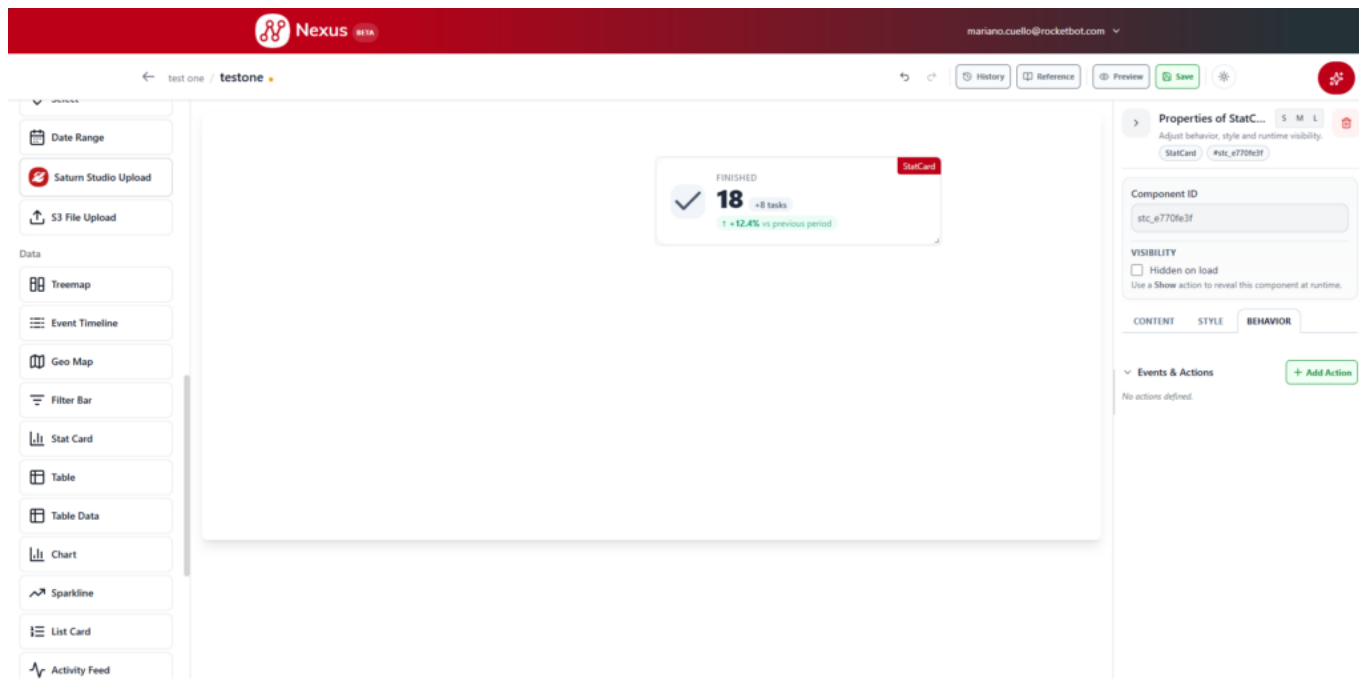
Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



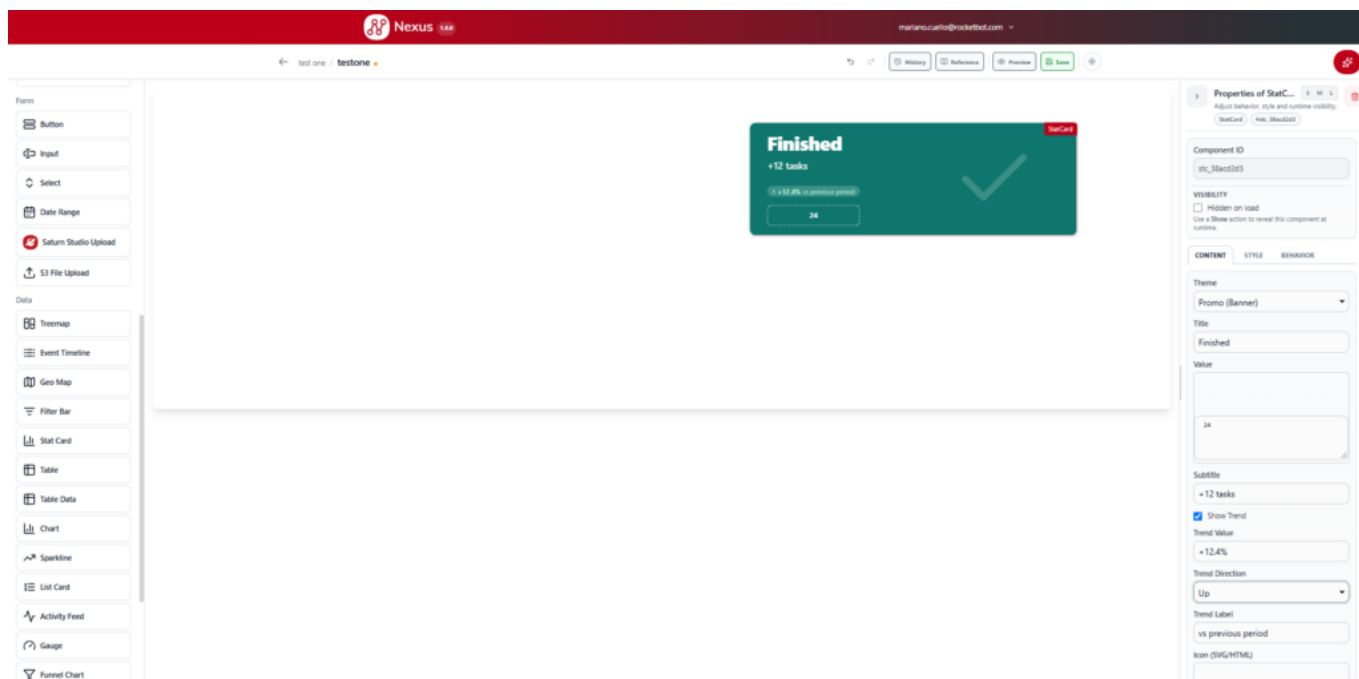
Pestaña Behavior:

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



Ejemplo de uso

Con **Theme** se elige la presentación general de la tarjeta según el contexto: por ejemplo, **Classic (Minimal)** para un dashboard sobrio, o **Promo (Banner)** cuando se necesita destacar una tarjeta con contenido promocional (usando **HTML Content**). Con **Trend Direction** se comunica visualmente si un valor mejoró (**Up**), empeoró (**Down**) o se mantuvo estable (**Neutral**), acompañado del texto de **Trend Value** y **Trend Label**. Los campos "Override" (Background Color, Text Color, Icon BG Color) permiten ajustar puntualmente los colores de una tarjeta sin perder la estructura general del Theme elegido.

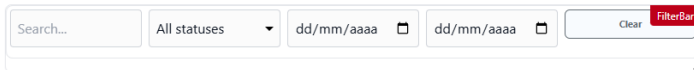


Nexus – Elemento: Filter Bar



¿Qué es?

Filter Bar es un componente de tipo **Data**. Es una barra de filtros pensada para usarse en conjunto con queries, combinando búsqueda por texto, filtro por estado y filtro por rango de fechas. Al arrastrarlo al lienzo viene con un ejemplo precargado (buscador, selector “All statuses”, dos campos de fecha y botón “Clear”).



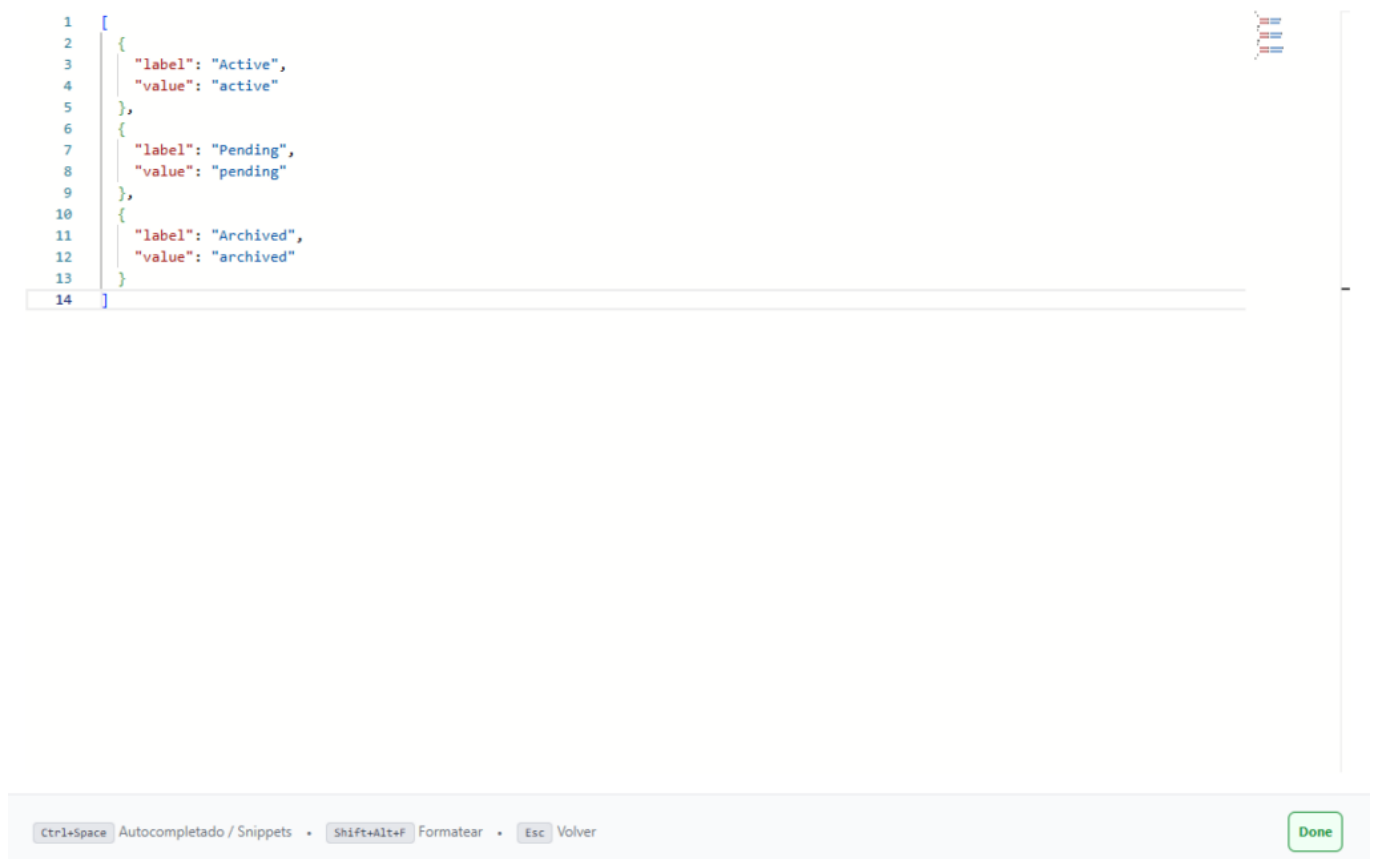
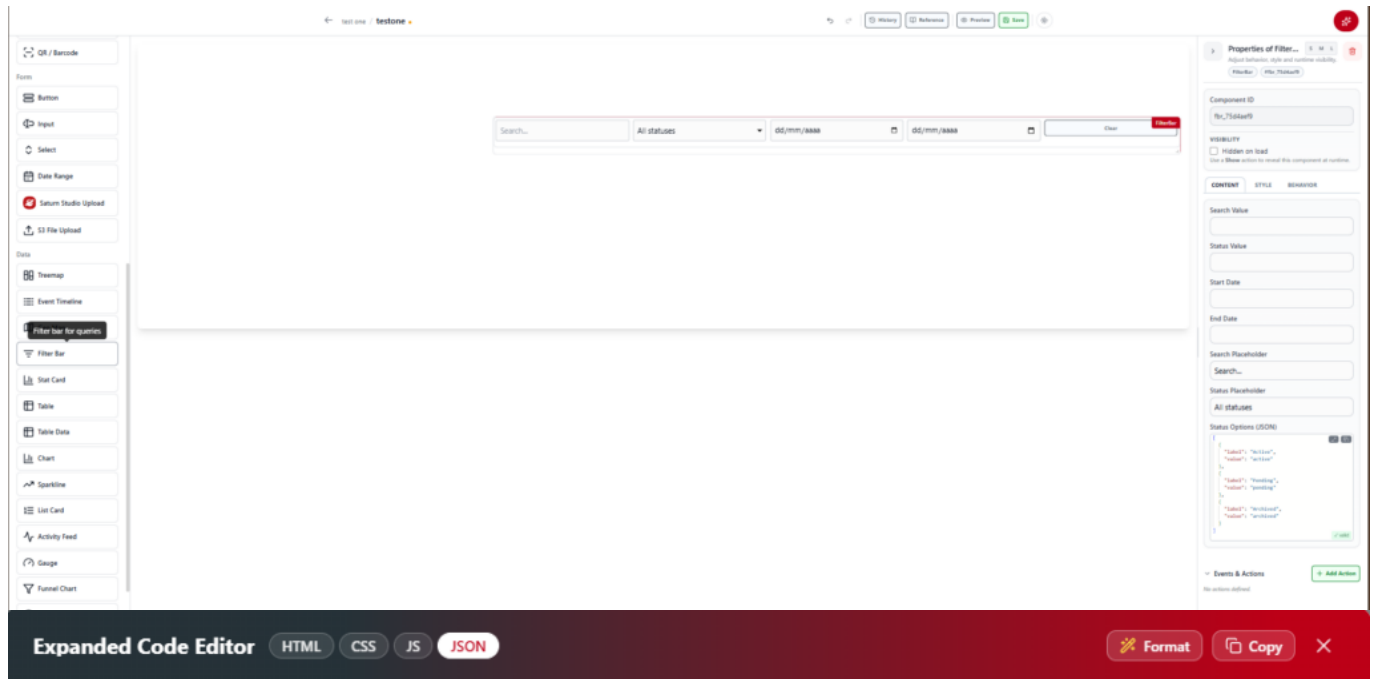
Propiedades

Al seleccionar Filter Bar en el lienzo, el panel derecho muestra “**Properties of FilterBar**”, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. fbr_75d4aef9).
- **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).

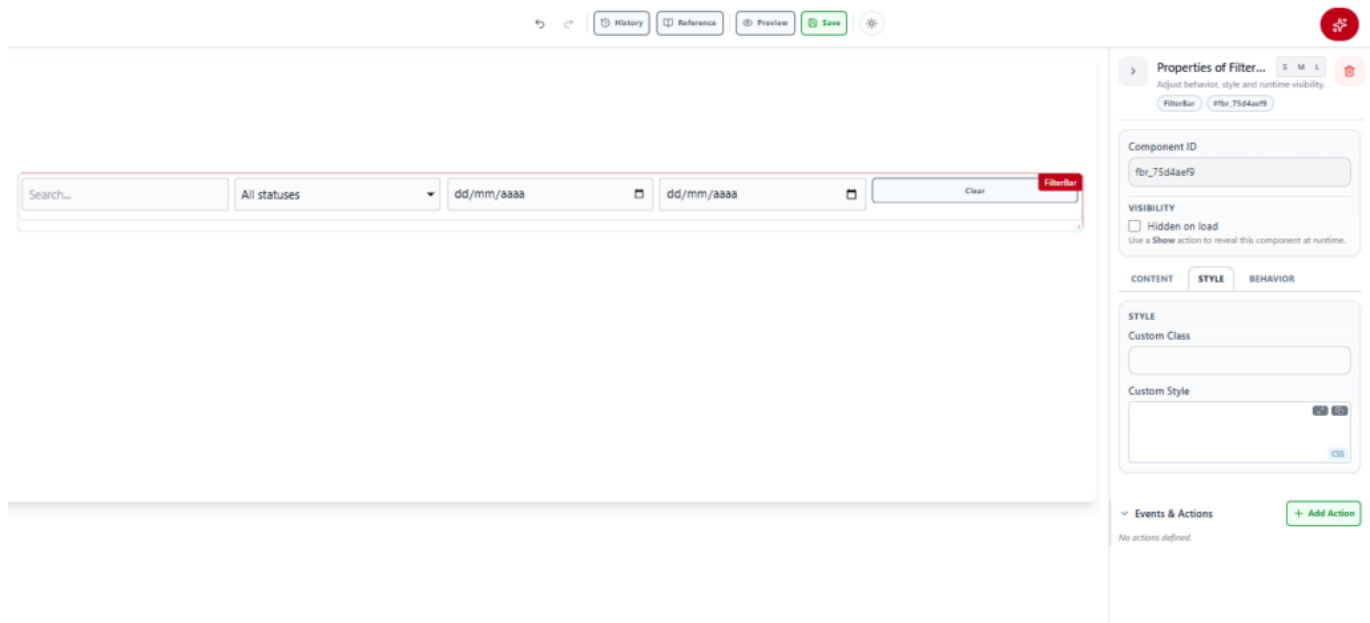
Pestaña Content:

- **Search Value**: valor por defecto del campo de búsqueda.
- **Status Value**: valor por defecto del filtro de estado.
- **Start Date / End Date**: valores por defecto del rango de fechas.
- **Search Placeholder**: texto guía del campo de búsqueda (ej. “Search...”).
- **Status Placeholder**: texto guía del selector de estado (ej. “All statuses”).
- **Status Options (JSON)**: campo de código donde se definen las opciones del filtro de estado, en el mismo formato que **Select** (label / value). Cuenta con el editor de código expandido para más comodidad.



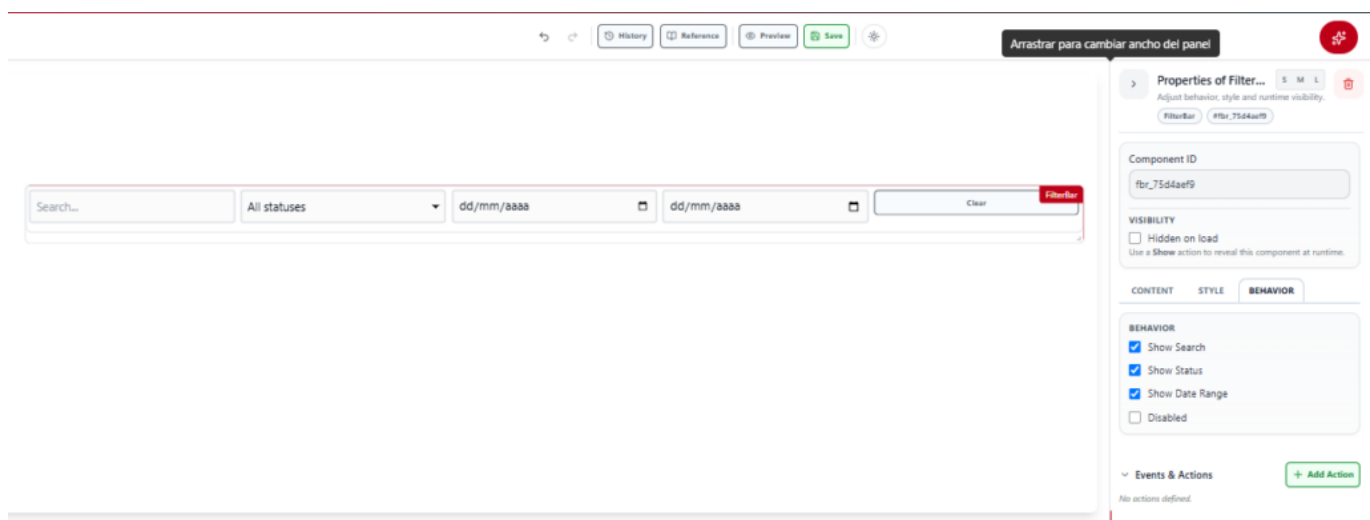
Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Pestaña Behavior:

- **Show Search:** casilla que muestra u oculta el campo de búsqueda.
- **Show Status:** casilla que muestra u oculta el filtro de estado.
- **Show Date Range:** casilla que muestra u oculta los campos de rango de fechas.
- **Disabled:** casilla que deshabilita todo el componente.
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos (por ejemplo, disparar una query al cambiar algún filtro). Ver documentación aparte: [Nexus – Events & Actions](#).



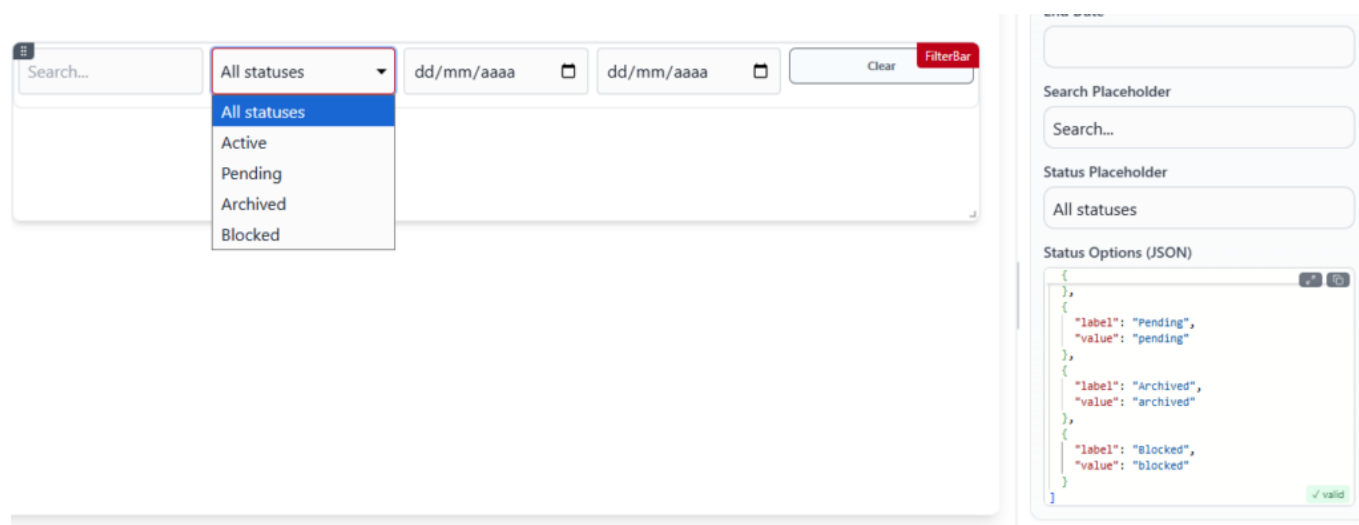
Ejemplo de uso

Para agregar una nueva opción de estado, hay que ubicarse en la última llave `}` del bloque anterior dentro de **Status Options (JSON)**, agregarle una coma, y pegar el nuevo bloque a continuación:

```
[
  {
    "label": "Archived",
    "value": "archived"
  },
  {
    "label": "Blocked",
    "value": "blocked"
  }
]
```

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

Con **Show Search / Show Status / Show Date Range** se puede simplificar la barra mostrando solo los filtros que se necesiten (por ejemplo, ocultar el rango de fechas si no aplica al listado que se está filtrando). Los valores que el usuario ingresa (Search Value, Status Value, Start/End Date) se pueden referenciar luego en una **Query** para filtrar los datos de una tabla u otro componente de datos.



The image shows a user interface for a filter bar and its configuration. On the left, a filter bar contains a search input, a status dropdown menu, and two date range inputs. The dropdown menu is open, showing options: 'All statuses', 'Active', 'Pending', 'Archived', and 'Blocked'. On the right, a configuration panel shows the 'Status Placeholder' set to 'All statuses' and a 'Status Options (JSON)' field containing a JSON array of status options. The JSON is valid, as indicated by a green checkmark.

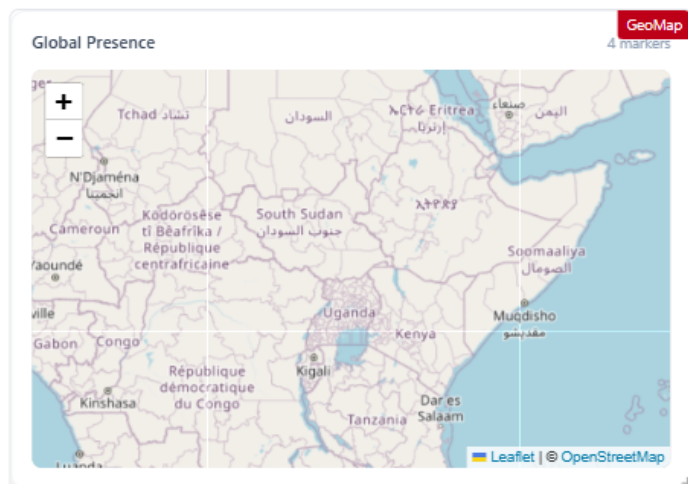
```
{
  "label": "Pending",
  "value": "pending"
},
{
  "label": "Archived",
  "value": "archived"
},
{
  "label": "Blocked",
  "value": "blocked"
}
]
```

[Nexus – Elemento: Geo Map](#)



¿Qué es?

Geo Map es un componente de tipo **Data**. Muestra un mapa geográfico interactivo (basado en **Leaflet** / **OpenStreetMap**) con marcadores ubicados por coordenadas. Al arrastrarlo al lienzo viene con un ejemplo precargado (“Global Presence”, con 4 marcadores: Santiago, Madrid, New York, Tokyo).



Propiedades

Al seleccionar Geo Map en el lienzo, el panel derecho muestra “**Properties of GeoMap**”, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

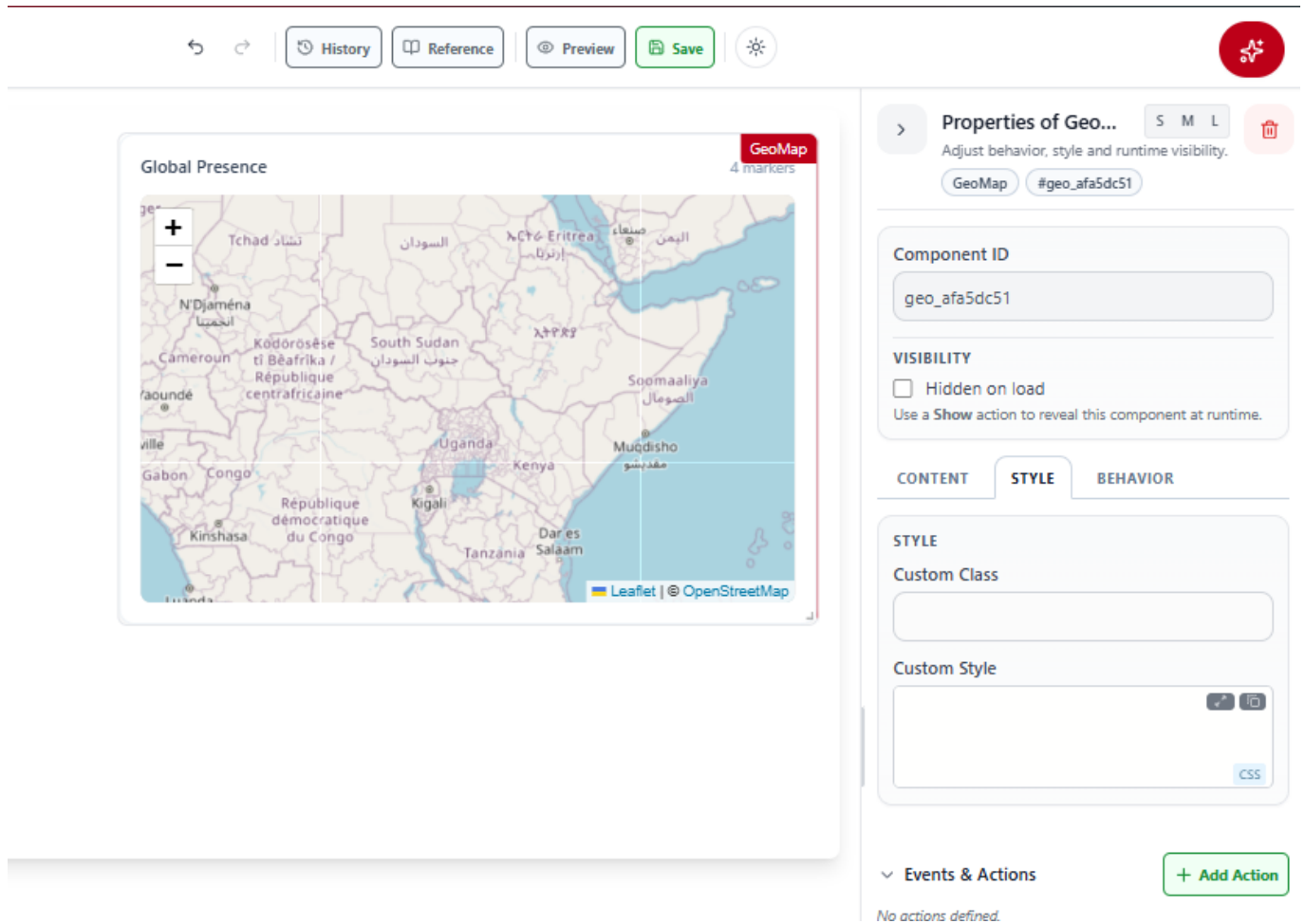
- **Component ID**: identificador único del componente, autogenerado (ej. geo_afa5dc51).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).

Pestaña Content:

- **Title**: título del mapa (ej. “Global Presence”).
- **Markers (JSON)**: campo de código donde se definen los marcadores en formato JSON. Cada marcador se compone de:
 - **label**: nombre del punto (ej. “Santiago”).
 - **lat**: latitud (ej. -33.45).
 - **lng**: longitud (ej. -70.66).
 - **color**: color del marcador en hexadecimal (ej. #2563eb).
 - **value**: valor numérico asociado al punto (ej. 42).

El campo cuenta con el mismo **editor de código expandido** visto en otros componentes (pestañas HTML/CSS/JS/JSON, formateo automático). El editor valida el JSON en tiempo real (marca “✓ valid” cuando el formato es correcto).

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Pestaña Behavior:

- **Show Labels:** casilla que muestra u oculta las etiquetas (label) de los marcadores en el mapa.
- **Max Zoom:** define el nivel máximo de zoom permitido al interactuar con el mapa (vacío por defecto).
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .

HistoryReferencePreviewSave

Global Presence

GeoMap

4 markers

Properties of Geo...

SML

GeoMap

#geo_afa5dc51

Component ID

geo_afa5dc51

VISIBILITY

☐ Hidden on load

Use a Show action to reveal this component at runtime.

CONTENTSTYLEBEHAVIOR

BEHAVIOR

☒ Show Labels

Max Zoom

Events & Actions

+ Add Action

No actions defined.

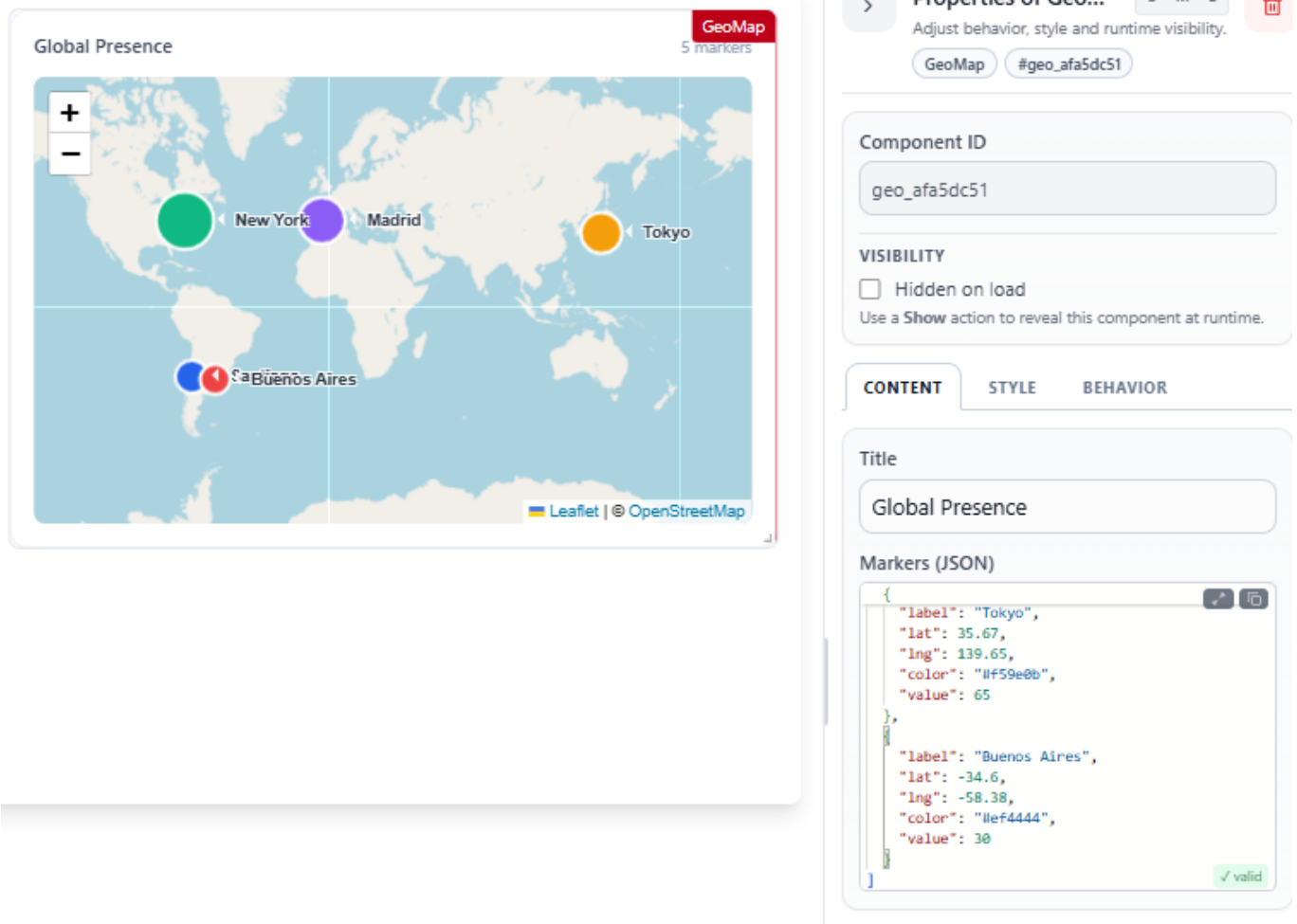
Ejemplo de uso

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

```
[
  {
    "label": "Tokyo",
    "lat": 35.67,
    "lng": 139.65,
    "color": "#f59e0b",
    "value": 65
  },
  {
    "label": "Buenos Aires",
    "lat": -34.6,
    "lng": -58.38,
    "color": "#ef4444",
    "value": 30
  }
]
```

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

Las coordenadas (lat/lng) ubican el punto en el mapa, mientras que color y value permiten diferenciar visualmente los marcadores según alguna métrica (por ejemplo, cantidad de clientes o ventas por ciudad). Con **Max Zoom** se puede limitar qué tan cerca puede acercarse el usuario al navegar el mapa.



The screenshot displays a design tool interface for a **GeoMap** component. The map on the left, titled "Global Presence", shows four markers: New York (green), Madrid (purple), Tokyo (orange), and Buenos Aires (blue). The right panel, titled "Properties of Geo...", shows the component's configuration. The "Component ID" is "geo_afa5dc51". Under "VISIBILITY", the "Hidden on load" checkbox is unchecked. The "CONTENT" tab is selected, showing the "Markers (JSON)" field with the following data:

```
{
  "label": "Tokyo",
  "lat": 35.67,
  "lng": 139.65,
  "color": "#f59e8b",
  "value": 65
},
{
  "label": "Buenos Aires",
  "lat": -34.6,
  "lng": -58.38,
  "color": "#ef4444",
  "value": 30
}
```

The JSON is marked as "valid".

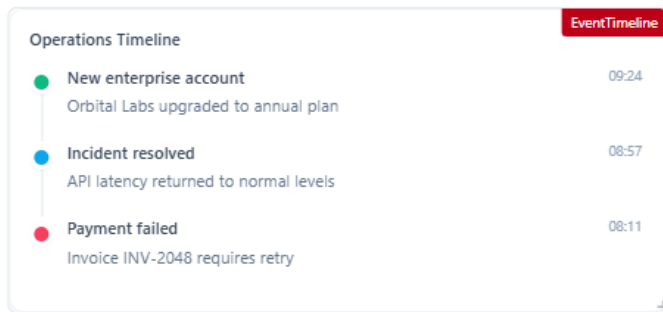
Nexus – Elemento: Event Timeline



¿Qué es?

Event Timeline es un componente de tipo **Data**. Muestra una línea de tiempo de

eventos, cada uno con título, descripción, hora y un color indicador de severidad. Al arrastrarlo al lienzo viene con un ejemplo precargado (“Operations Timeline”, con 3 eventos).



Propiedades

Al seleccionar Event Timeline en el lienzo, el panel derecho muestra **“Properties of EventTimeline”**, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. etl_42eff826).
- **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).

Pestaña Content:

- **Title**: título de la línea de tiempo (ej. “Operations Timeline”).
- **Events (JSON)**: campo de código donde se definen los eventos en formato JSON. Cada evento se compone de:
 - **title**: título del evento (ej. “New enterprise account”).
 - **description**: descripción/detalle del evento.
 - **time**: hora del evento (ej. “09:24”).
 - **variant**: color/severidad del punto indicador. Valores vistos: "success" (verde), "info" (azul), "danger" (rojo).

El campo cuenta con un botón para abrir un **editor de código expandido** (con pestañas HTML/CSS/JS/JSON, resaltado de sintaxis, autocompletado y formateo automático), útil para editar JSON extensos con más comodidad. El editor valida el JSON en tiempo real (marca “✓ valid” cuando el formato es correcto).

Form

Button

Input

Select

Date Range

Saturn Studio Upload

S3 File Upload

Data

Timeline of events with date and description

Event Timeline

Geo Map

Filter Bar

Stat Card

Table

Table Data

Chart

Sparkline

List Card

Activity Feed

Gauge

test one / testone

History

Reference

Preview

Save

Operations Timeline

New enterprise account

Orbital Labs upgraded to annual plan

09:24

Incident resolved

API latency returned to normal levels

08:57

Payment failed

Invoice INV-2048 requires retry

08:11

Properties of Event...

Adjust behavior, style and runtime visibility.

EventTimeline

det_42ef8226

Component ID

et_42ef8226

Visibility

☐ Hidden on load

Use a Show action to reveal this component at runtime.

Content

Style

Behavior

Title

Operations Timeline

Events (JSON)

"title": "New enterprise account",

"description": "Orbital Labs upgraded to annual plan",

"time": "09:24",

"variant": "success"

"title": "Incident resolved",

"description": "API latency returned to normal levels",

"time": "08:57",

"variant": "info"

"title": "Payment failed",

"description": "Invoice INV-2048 requires retry",

"time": "08:11",

"variant": "danger"

Events & Actions

No actions defined

Add Action

Expanded Code Editor

HTML

CSS

JS

JSON

Format

Copy

X

1

[

2

{

3

"title": "New enterprise account",

4

"description": "Orbital Labs upgraded to annual plan",

5

"time": "09:24",

6

"variant": "success"

7

},

8

{

9

"title": "Incident resolved",

10

"description": "API latency returned to normal levels",

11

"time": "08:57",

12

"variant": "info"

13

},

14

{

15

"title": "Payment failed",

16

"description": "Invoice INV-2048 requires retry",

17

"time": "08:11",

18

"variant": "danger"

19

}

20

}

Ctrl+Space Autocompletado / Snippets

Shift+Alt+F Formatear

Esc Volver

Done

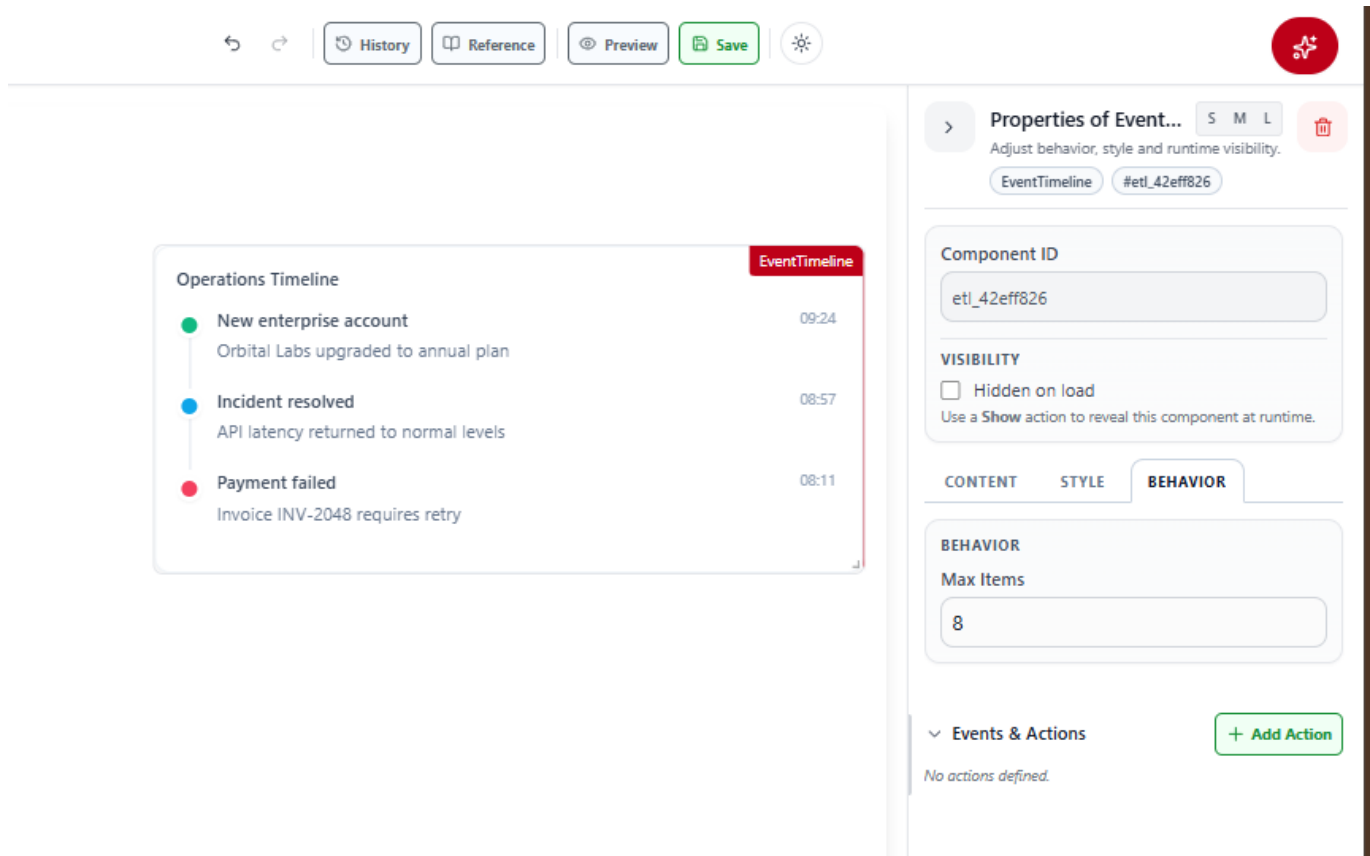
Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.

The screenshot displays the Axure RP interface. On the left, a canvas shows a component titled 'Operations Timeline' with a red 'EventTimeline' label. The timeline lists three events: 'New enterprise account' (09:24), 'Incident resolved' (08:57), and 'Payment failed' (08:11). On the right, the 'Properties of Event...' panel is open, showing the 'EventTimeline' component with ID '#eti_42eff826'. The 'STYLE' tab is selected, showing fields for 'Custom Class' and 'Custom Style'. The 'BEHAVIOR' tab is also visible, showing 'Events & Actions' with a '+ Add Action' button.

Pestaña Behavior:

- **Max Items:** cantidad máxima de eventos que se muestran en la línea de tiempo (valor de ejemplo: 8).
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .



Ejemplo de uso

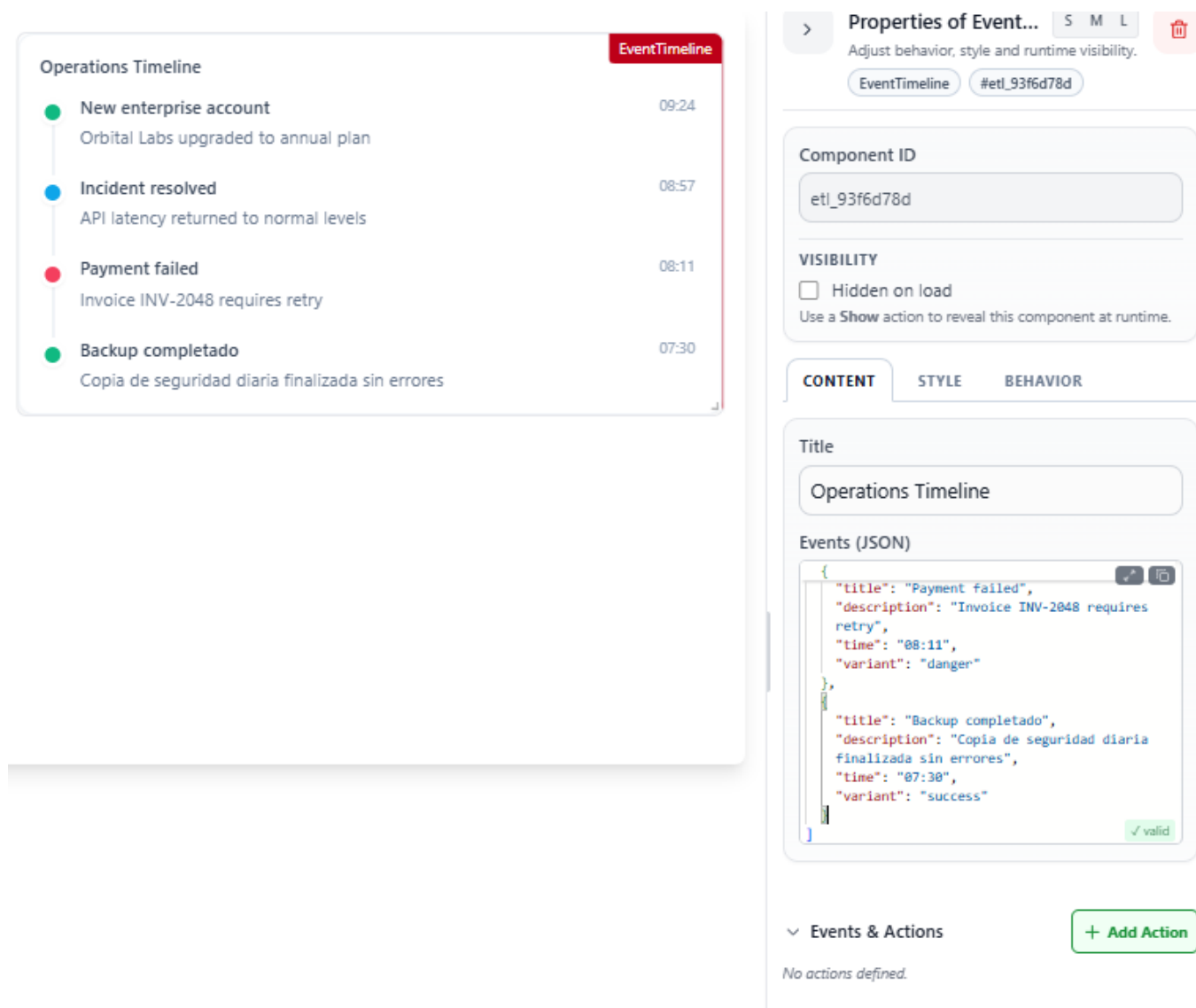
Para agregar un nuevo evento, hay que ubicarse en la última llave `}` del bloque anterior dentro de **Events (JSON)**, agregarle una coma, y pegar el nuevo bloque a continuación:

```
[
  {
    "title": "Payment failed",
    "description": "Invoice INV-2048 requires retry",
    "time": "08:11",
    "variant": "danger"
  },
  {
    "title": "Backup completado",
    "description": "Copia de seguridad diaria finalizada sin errores",
    "time": "07:30",
    "variant": "success"
  }
]
```

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

Con `variant` se le da significado visual a cada evento: `success` para eventos

positivos, info para eventos neutros/informativos, y danger para errores o incidentes. Con **Max Items** se puede limitar cuántos eventos se muestran a la vez, útil cuando la lista de eventos crece mucho.



The screenshot displays the **EventTimeline** component in a design tool. On the left, a preview of the component shows a vertical timeline with four events:

- New enterprise account** (green dot) at 09:24: Orbital Labs upgraded to annual plan
- Incident resolved** (blue dot) at 08:57: API latency returned to normal levels
- Payment failed** (red dot) at 08:11: Invoice INV-2048 requires retry
- Backup completado** (green dot) at 07:30: Copia de seguridad diaria finalizada sin errores

On the right, the **Properties of EventTimeline** panel is shown. It includes a **Component ID** field with the value `etl_93f6d78d`. Under the **VISIBILITY** section, there is a checkbox for **Hidden on load** (unchecked) and a note: "Use a **Show** action to reveal this component at runtime." Below this are tabs for **CONTENT**, **STYLE**, and **BEHAVIOR**. The **CONTENT** tab is active, showing a **Title** field with "Operations Timeline" and an **Events (JSON)** field containing two event objects:

```
{
  "title": "Payment failed",
  "description": "Invoice INV-2048 requires retry",
  "time": "08:11",
  "variant": "danger"
},
{
  "title": "Backup completado",
  "description": "Copia de seguridad diaria finalizada sin errores",
  "time": "07:30",
  "variant": "success"
}
```

A green "valid" indicator is present at the bottom right of the JSON field. At the bottom of the panel, there is a section for **Events & Actions** with a "+ Add Action" button and the text "No actions defined."

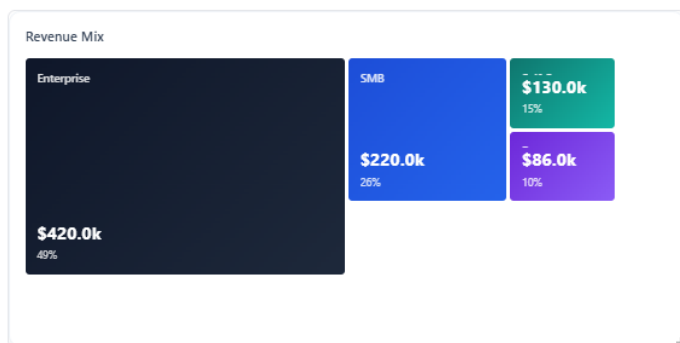
Nexus – Elemento: Treemap



¿Qué es?

Treemap es un componente de tipo **Data**. Muestra una visualización jerárquica de datos, donde cada bloque representa un valor proporcional al total (mientras mayor el valor, más grande el bloque). Al arrastrarlo al lienzo

viene con un ejemplo precargado (“Revenue Mix”, con categorías Enterprise, SMB, Self-Serve, Partners).



Propiedades

Al seleccionar Treemap en el lienzo, el panel derecho muestra **“Properties of Treemap”**, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

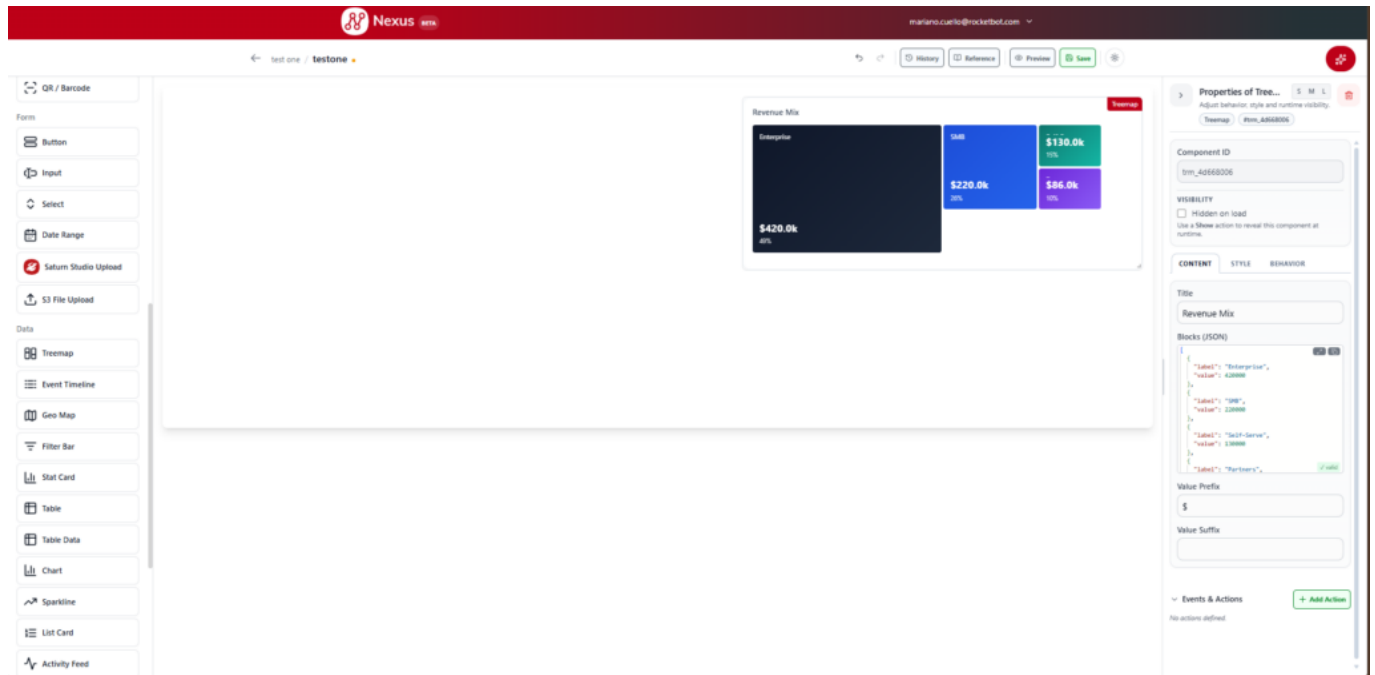
- **Component ID**: identificador único del componente, autogenerado (ej. trm_4d668006).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).

Pestaña Content:

- **Title**: título del gráfico (ej. “Revenue Mix”).
- **Blocks (JSON)**: campo de código donde se definen los bloques del treemap en formato JSON. Cada bloque se compone de:
 - **label**: nombre de la categoría (ej. “Enterprise”).
 - **value**: valor numérico que determina el tamaño proporcional del bloque (ej. 420000).

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

- **Value Prefix**: texto que se antepone al valor mostrado en cada bloque (ej. \$).
- **Value Suffix**: texto que se agrega después del valor mostrado en cada bloque.



Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.

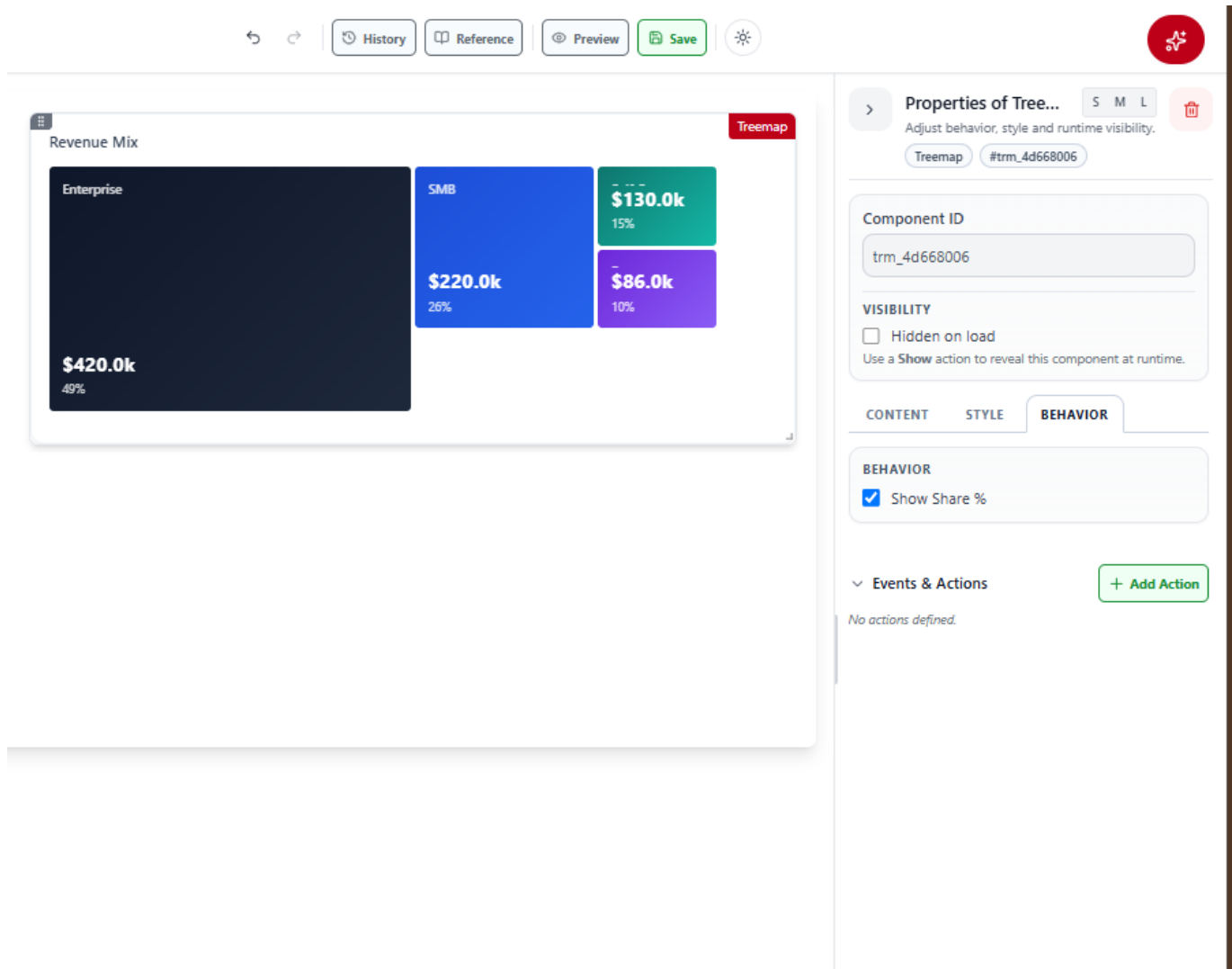
The screenshot displays a software interface with a top navigation bar containing icons for History, Reference, Preview, Save, and a settings icon. The main workspace shows a treemap titled "Revenue Mix" with a red "Treemap" label in the top right corner. The treemap consists of four colored rectangles representing different revenue segments:

Segment	Revenue	Share %
Enterprise	\$420.0k	49%
SMB	\$220.0k	26%
Mid-Market	\$130.0k	15%
Other	\$86.0k	10%

The right sidebar contains the "Properties of Tree..." panel, which includes tabs for S, M, and L views, and a delete icon. Below these are sections for Component ID (trm_4d668006), VISIBILITY (with a checkbox for "Hidden on load"), and tabs for CONTENT, STYLE, and BEHAVIOR. The STYLE tab is active, showing fields for Custom Class and Custom Style. At the bottom, there is an "Events & Actions" section with a "+ Add Action" button and the text "No actions defined."

Pestaña Behavior:

- **Show Share %:** casilla que muestra el porcentaje que representa cada bloque respecto al total (ej. "49%" debajo de "\$420.0k").
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .



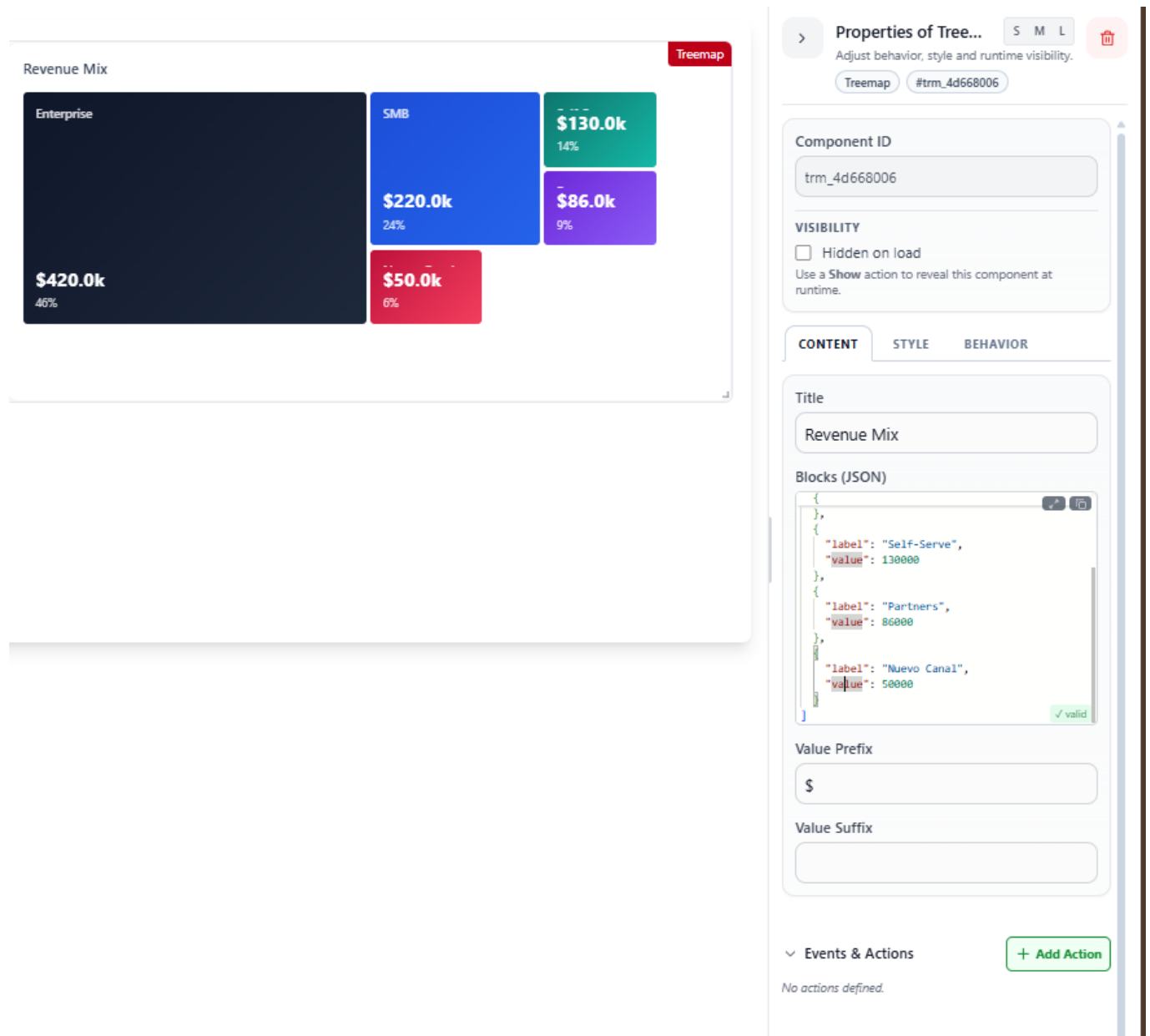
Ejemplo de uso

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

```
[
  {
    "label": "Partners",
    "value": 86000
  },
  {
    "label": "Nuevo Canal",
    "value": 50000
  }
]
```

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

Con **Value Prefix** y **Value Suffix** se puede dar formato al valor mostrado sin modificar el JSON – por ejemplo, prefijo \$ para montos en dólares, o sufijo kg para unidades de peso. Con **Show Share %** activado, cada bloque muestra automáticamente el porcentaje que representa sobre el total de todos los valores cargados.

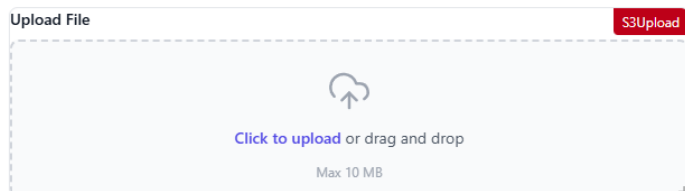


Nexus – Elemento: S3 File Upload



¿Qué es?

S3 File Upload es un componente de tipo **Form**. Permite subir archivos directamente a un bucket de **Amazon S3**, mediante un DataSource S3 configurado en la aplicación. Al arrastrarlo al lienzo viene con un ejemplo precargado (“Upload File”, con zona de arrastre y límite de tamaño).



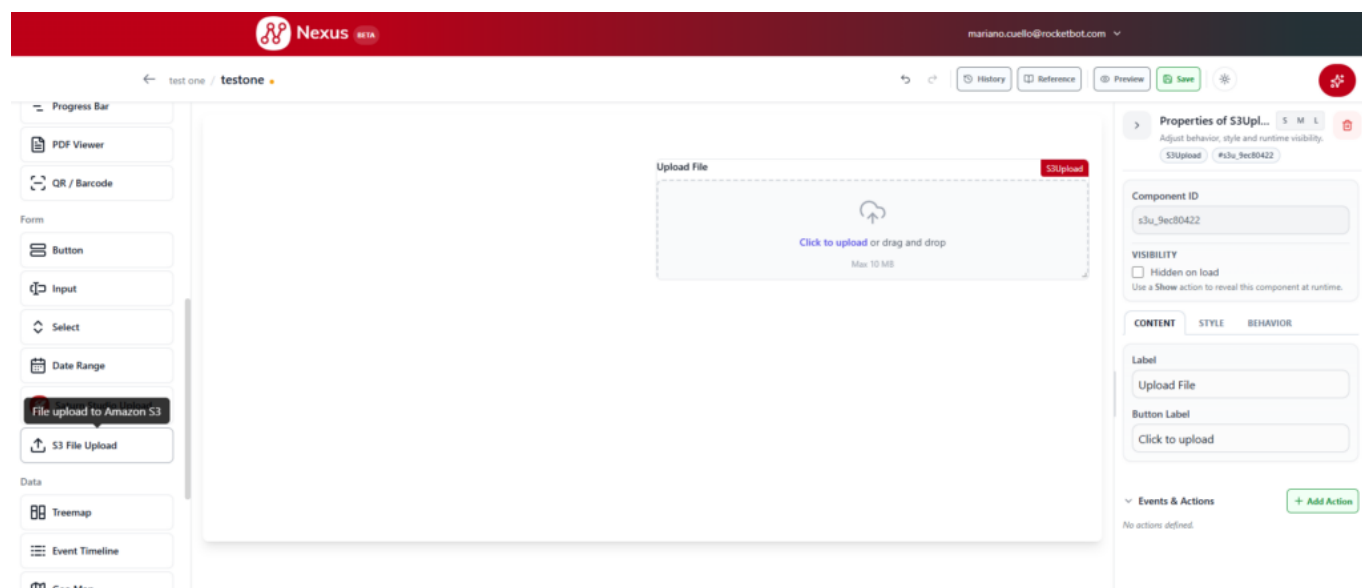
Propiedades

Al seleccionar el componente en el lienzo, el panel derecho muestra **“Properties of S3Upload”**, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. s3u_9ec80422).
- **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).

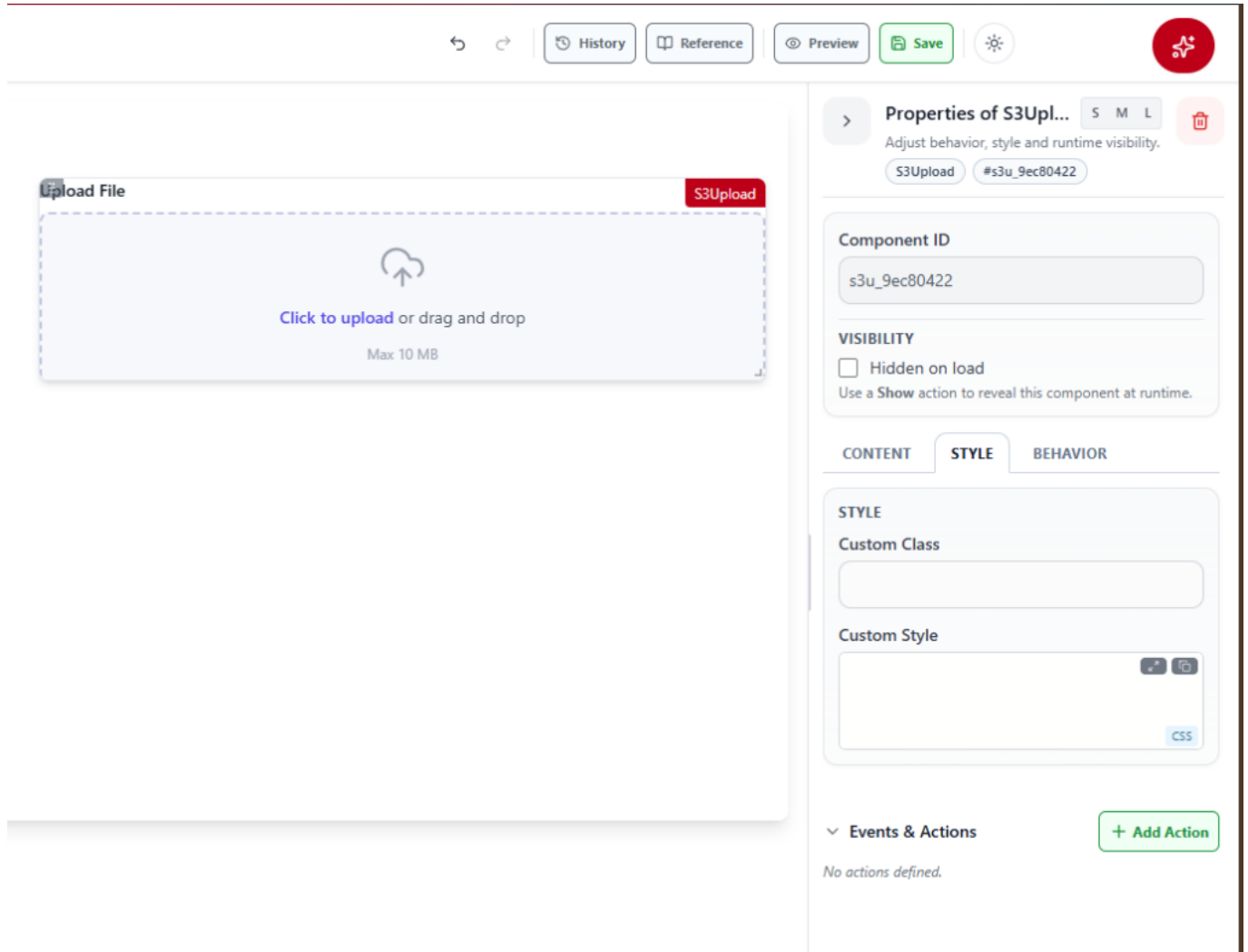
Pestaña Content:

- **Label**: título que se muestra sobre la zona de carga (ej. “Upload File”).
- **Button Label**: texto del enlace/botón para subir el archivo (ej. “Click to upload”).



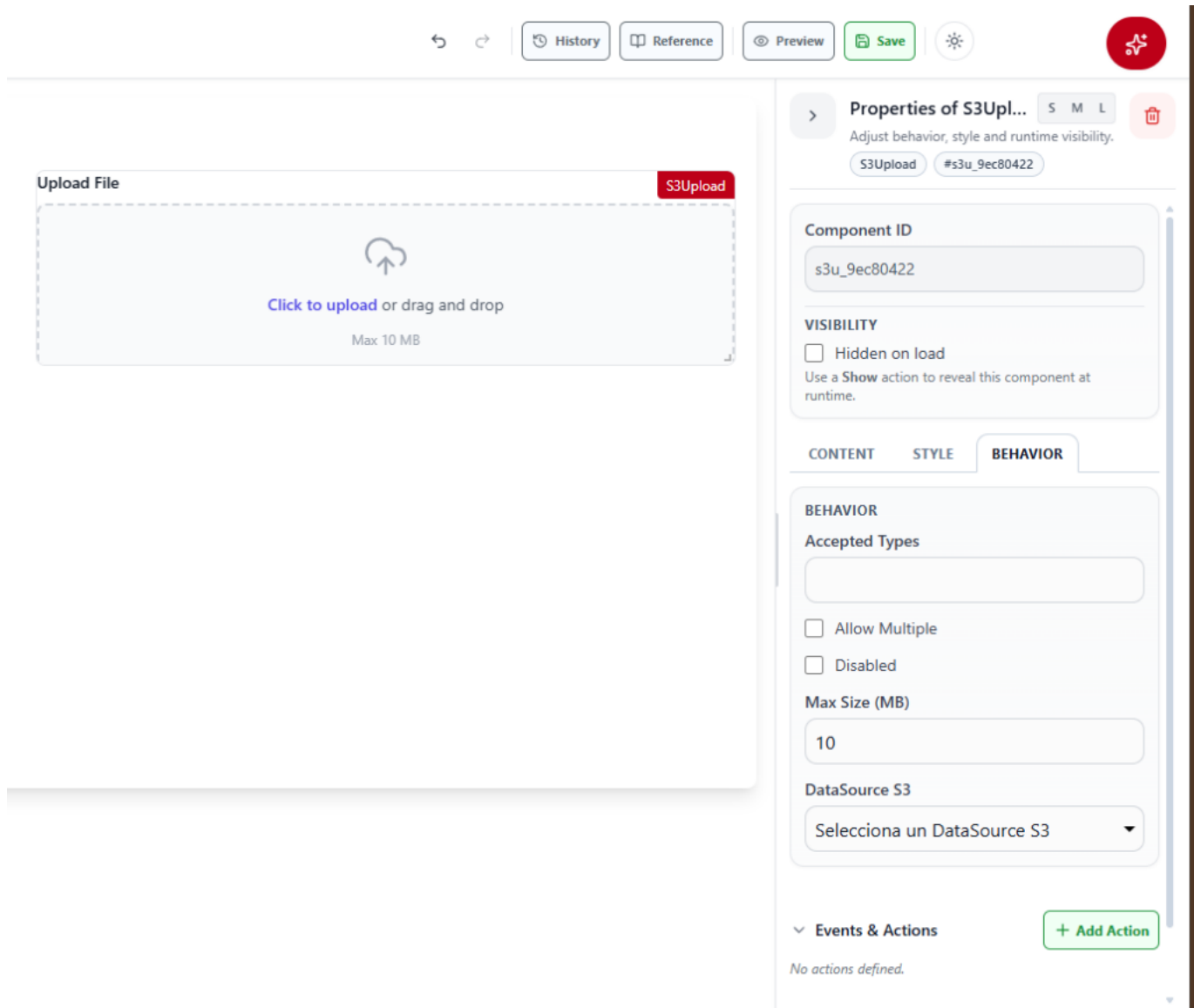
Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Pestaña Behavior:

- **Accepted Types:** campo para definir qué tipos de archivo se aceptan (vacío por defecto, lo que permite cualquier tipo).
- **Allow Multiple:** casilla que permite subir más de un archivo a la vez.
- **Disabled:** casilla que deshabilita el componente.
- **Max Size (MB):** tamaño máximo permitido por archivo, en megabytes (valor por defecto: 10).
- **DataSource S3:** menú desplegable para seleccionar un DataSource S3 ya configurado, que define a qué bucket de Amazon S3 se suben los archivos.
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .



Diferencias con Saturn Studio Upload

Ambos componentes son muy similares en estructura, pero se diferencian en:

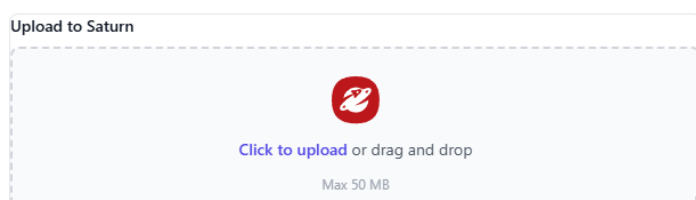
- **S3 File Upload** tiene la opción **Allow Multiple** (subir varios archivos a la vez), que **Saturn Studio Upload** no tiene.
- **S3 File Upload** se conecta a un bucket de Amazon S3 mediante **DataSource S3**, mientras que **Saturn Studio Upload** se conecta a un proceso/robot mediante **DataSource Saturn**.

Nexus – Elemento: Saturn Studio Upload



¿Qué es?

Saturn Studio Upload es un componente de tipo **Form**. Permite subir archivos que se procesan mediante un **DataSource Saturn** configurado en la aplicación, vinculado a un proceso/robot de **Saturn Studio** (la herramienta de automatización de Rocketbot). Al arrastrarlo al lienzo viene con un ejemplo precargado ("Upload to Saturn", con zona de arrastre y límite de tamaño).



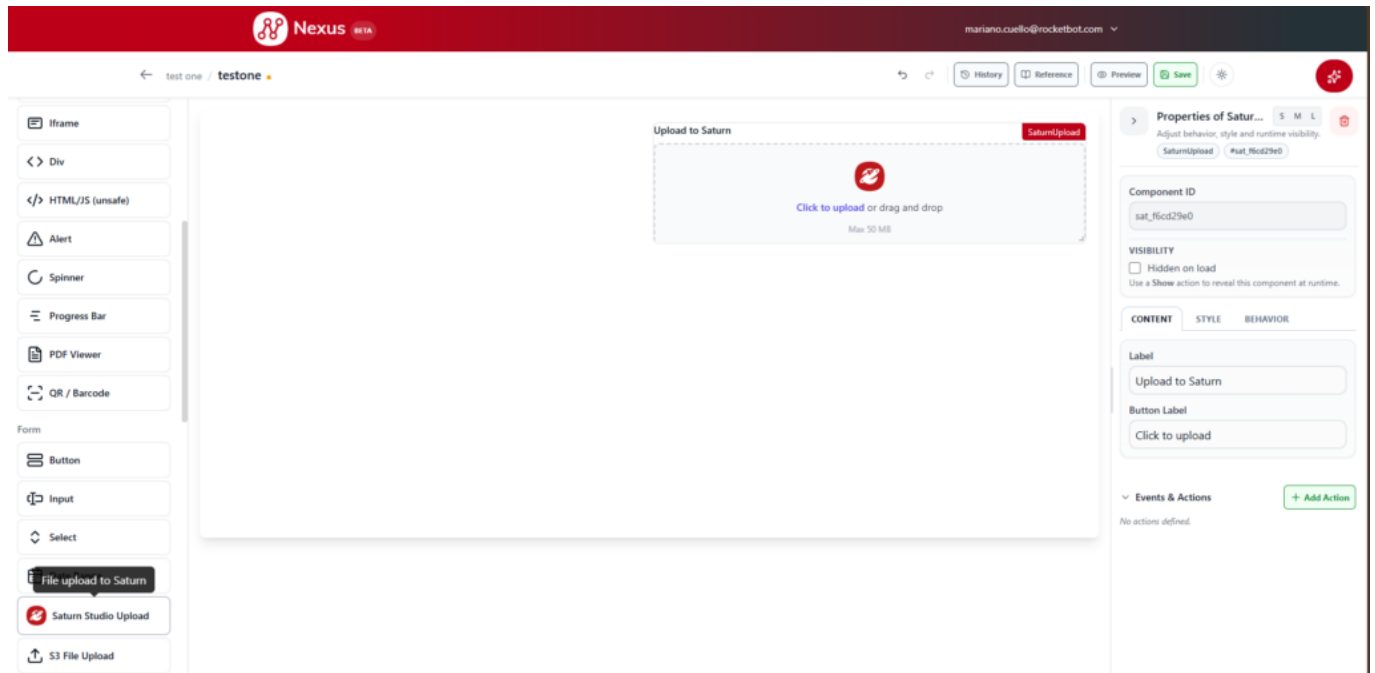
Propiedades

Al seleccionar el componente en el lienzo, el panel derecho muestra **"Properties of SaturnUpload"**, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **Component ID**: identificador único del componente, autogenerado (ej. sat_f6cd29e0).
- **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).

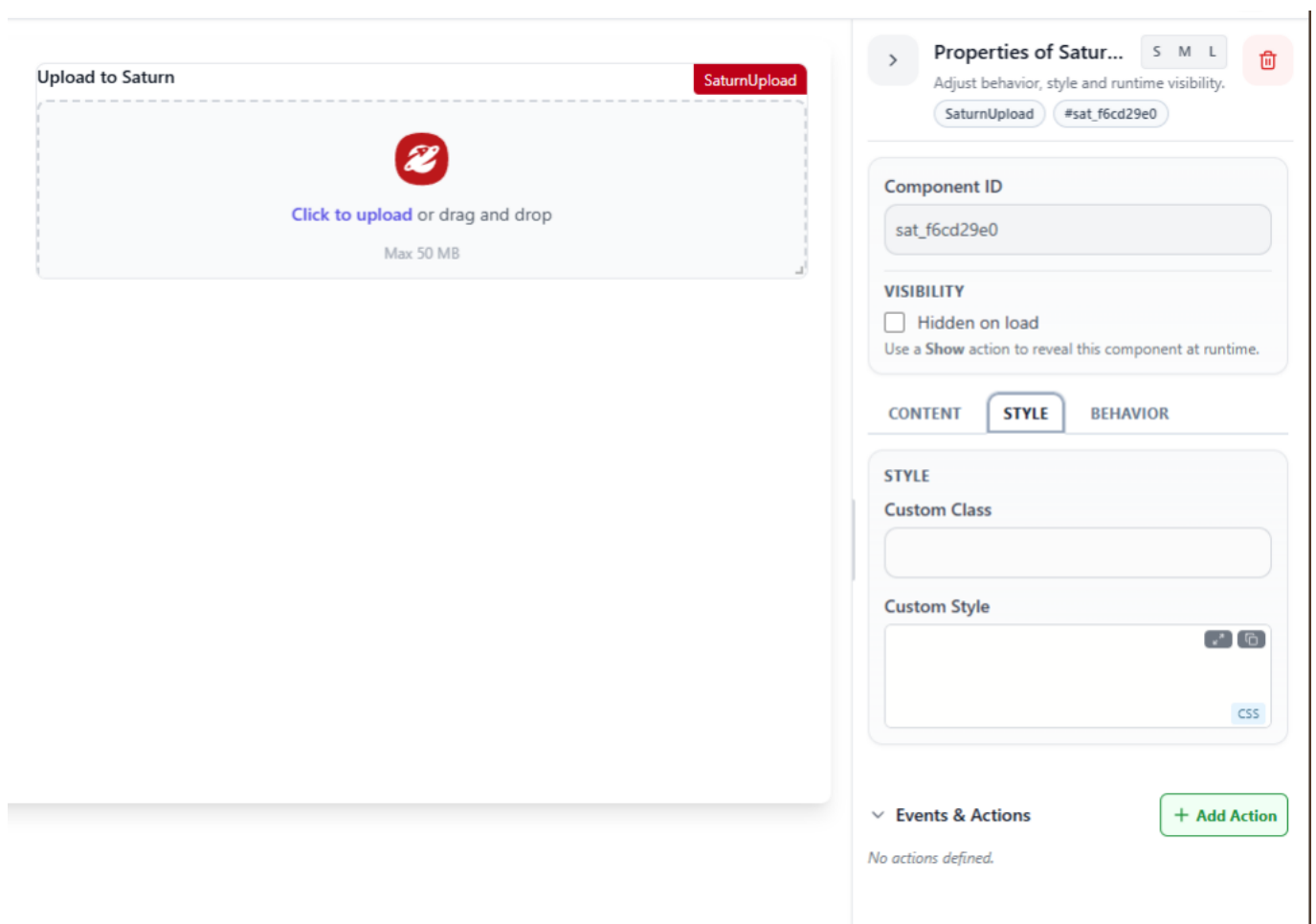
Pestaña Content:

- **Label**: título que se muestra sobre la zona de carga (ej. "Upload to Saturn").
- **Button Label**: texto del enlace/botón para subir el archivo (ej. "Click to upload").



Pestaña Style:

- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Pestaña Behavior:

- **Accepted Types:** campo para definir qué tipos de archivo se aceptan (vacío por defecto, lo que permite cualquier tipo).
- **Disabled:** casilla que deshabilita el componente.
- **Max Size (MB):** tamaño máximo permitido por archivo, en megabytes (valor por defecto: 50).
- **DataSource Saturn:** menú desplegable para seleccionar un **DataSource** ya configurado que conecta con Saturn Studio. Este DataSource se define/configura en otra parte de la aplicación (no dentro del componente), y es lo que determina a qué robot/proceso de Saturn Studio se envía el archivo subido.
- **Delete Component** (ícono de papelera, arriba): elimina el componente del lienzo.
- **Events & Actions:** sistema para definir comportamientos interactivos. Ver documentación aparte: [Nexus – Events & Actions](#) .

The screenshot displays the Saturn Studio interface. At the top, there is a navigation bar with buttons for History, Reference, Preview, Save, and a settings icon. A red circular icon with a star is also present.

The main workspace shows a component titled "Upload to Saturn" with a red Saturn logo and the text "Click to upload or drag and drop" and "Max 50 MB". A red button labeled "SaturnUpload" is in the top right corner of the component.

The right sidebar shows the "Properties of Saturn..." panel. It includes tabs for S, M, and L, and a trash icon. The panel is divided into sections:

- Properties of Saturn...**: Adjust behavior, style and runtime visibility. It shows "SaturnUpload" and "#sat_f6cd29e0".
- VISIBILITY**: A checkbox for "Hidden on load" with a note: "Use a Show action to reveal this component at runtime."
- CONTENT**, **STYLE**, and **BEHAVIOR** tabs. The **BEHAVIOR** tab is active.
- BEHAVIOR** section:
 - Accepted Types**: A text input field.
 - Disabled**: A checkbox.
 - Max Size (MB)**: A text input field with the value "50".
 - DataSource Saturn**: A dropdown menu with the text "Selecciona un DataSource Saturn".
- Events & Actions**: A section with a "+ Add Action" button and the text "No actions defined."

Nexus – Elemento: Date Range



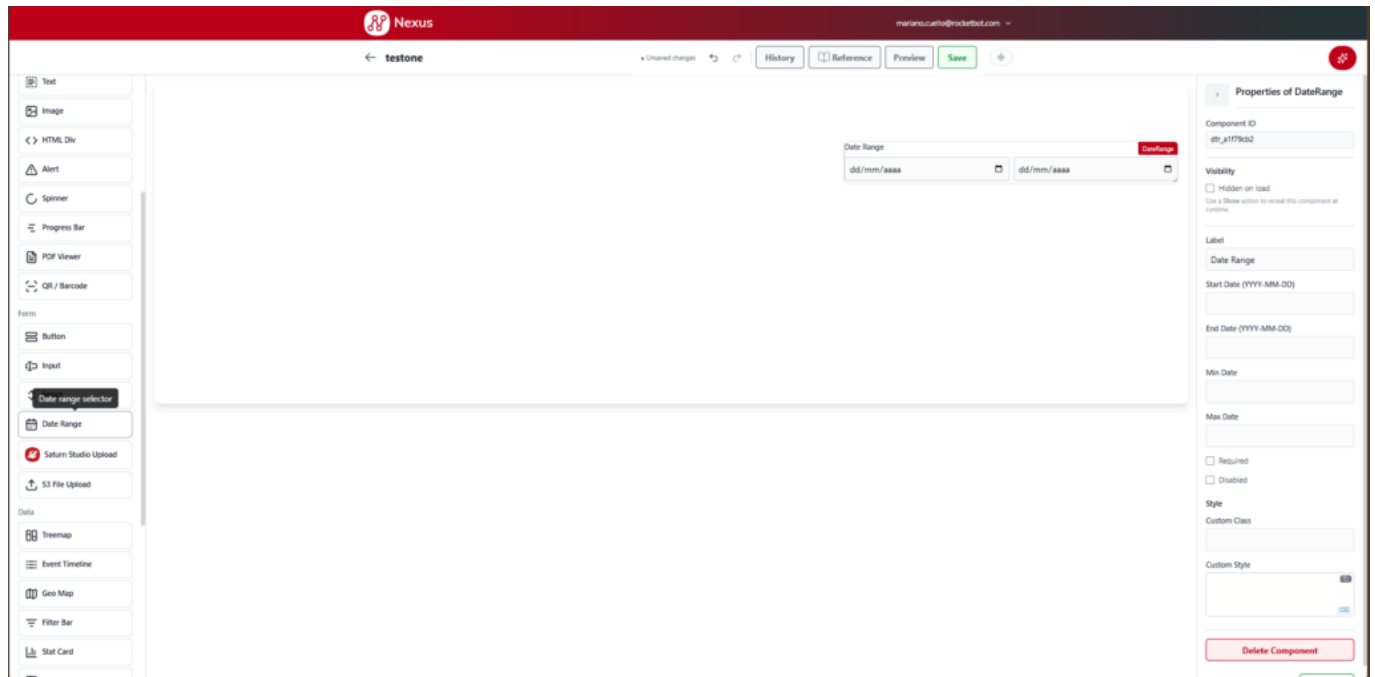
¿Qué es?

Date Range es un componente de tipo **Form**. Permite seleccionar un rango de fechas (fecha de inicio y fecha de fin) mediante dos selectores de calendario. Al arrastrarlo al lienzo viene con un ejemplo precargado (“Date Range”, ambos campos vacíos).

Propiedades

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

- **Component ID**: identificador único del componente, autogenerado (ej. dtr_alf79cb2).
- **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).
- **Label**: etiqueta que se muestra encima del componente (ej. “Date Range”).
- **Start Date (YYYY-MM-DD)**: valor por defecto para la fecha de inicio, en formato Año-Mes-Día.
- **End Date (YYYY-MM-DD)**: valor por defecto para la fecha de fin, en el mismo formato.
- **Min Date**: fecha mínima seleccionable (límite inferior del rango permitido).
- **Max Date**: fecha máxima seleccionable (límite superior del rango permitido).
- **Required**: casilla que marca el campo como obligatorio.
- **Disabled**: casilla que deshabilita el componente (queda inactivo, no editable).
- **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

Con **Start Date** y **End Date** se puede precargar un rango de fechas por defecto (formato YYYY-MM-DD, por ejemplo 2026-01-01). Con **Min Date** y **Max Date** se restringe el rango de fechas que el usuario puede elegir – por ejemplo, para no permitir seleccionar fechas futuras o anteriores a cierto límite. Es útil para filtrar datos por período, en combinación con componentes de tipo **Data** (como Table o Chart) mediante queries.