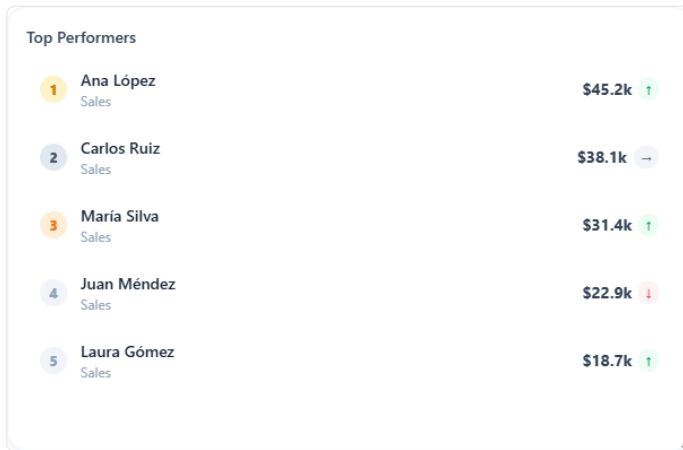


Nexus – Elemento: List Card



¿Qué es?

List Card es un componente de tipo **Data**. Muestra una lista de ítems tipo ranking, cada uno con nombre, subtítulo, valor y una tendencia (flecha de subida, bajada o neutra). Al arrastrarlo al lienzo viene con un ejemplo precargado (“Top Performers”, 5 vendedores con sus ventas).



Propiedades

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

- **Component ID**: identificador único del componente, autogenerado (ej. `lst_420dde73`).
- **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 lista (ej. “Top Performers”).
- **Data (array of items)**: campo de código donde se definen los ítems en formato JSON. Cada ítem se compone de:
 - **label**: nombre del ítem (ej. “Ana López”).
 - **sublabel**: texto secundario debajo del nombre (ej. “Sales”).

- **value:** valor numérico asociado (ej. 45200).
- **trend:** dirección de la tendencia. Valores vistos: "up", "down", "neutral".

Cuenta con el editor de código expandido para más comodidad. El editor valida el JSON en tiempo real (marca "✓ valid" cuando el formato es correcto).

- **Show Rank:** casilla que muestra u oculta el número de posición/ranking junto a cada ítem.
- **Value Prefix:** texto que se antepone al valor mostrado (ej. \$).
- **Value Suffix:** texto que se agrega después del valor mostrado.
- **Max Items:** cantidad máxima de ítems que se muestran en la lista (valor por defecto: 10).

The screenshot displays the Saturn Studio interface. On the left is a sidebar with various widget categories: Data (Treemap, Event Timeline, Geo Map, Filter Bar, Stat Card, Table, Table Data, Chart), Items list in card format, List Card, Activity Feed, Gauge, Funnel Chart, KPI Card, Heatmap Calendar, and Calendar. The main workspace shows a 'Top Performers' list card with the following data:

Rank	Name	Value	Trend
1	Ana López	\$45.2k	up
2	Carlos Ruiz	\$38.5k	down
3	María Silva	\$31.4k	up
4	Juan Méndez	\$22.9k	down
5	Laura Gómez	\$18.7k	up

On the right, the 'Properties of ListCard' panel is open, showing configuration options for the selected widget. The 'CONTENT' tab is active, displaying the JSON data source and the 'Show Rank' checkbox, which is checked. Other visible settings include 'Value Prefix' set to '\$' and 'Max Items' set to 10.

Expanded Code Editor HTML CSS JS JSON Format Copy X

```
1 [{"label": "Ana López", "sublabel": "Sales", "value": 45200, "trend": "up"}, {"label": "Carlos Ruiz", "sublabel": "Sales", "value": 38100, "trend": "neutral"}, {"label": "María Silva", "sublabel": "Sales", "value": 31400, "trend": "up"}, {"label": "Juan Méndez", "sublabel": "Sales", "value": 22900, "trend": "down"}, {"label": "Laura Gómez", "sublabel": "Sales", "value": 18700, "trend": "up"}]
```

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.

HistoryReferencePreviewSave

Top Performers

1

Ana López

Sales

\$45.2k

↑

2

Carlos Ruiz

Sales

\$38.1k

→

3

María Silva

Sales

\$31.4k

↑

4

Juan Méndez

Sales

\$22.9k

↓

5

Laura Gómez

Sales

\$18.7k

↑

ListCard

Properties of ListCa...SML

Adjust behavior, style and runtime visibility.

ListCard#lst_420dde73

Component ID

lst_420dde73

VISIBILITY

☐ Hidden on load

Use a Show action to reveal this component at runtime.

CONTENTSTYLEBEHAVIOR

STYLE

Custom Class

Custom Style

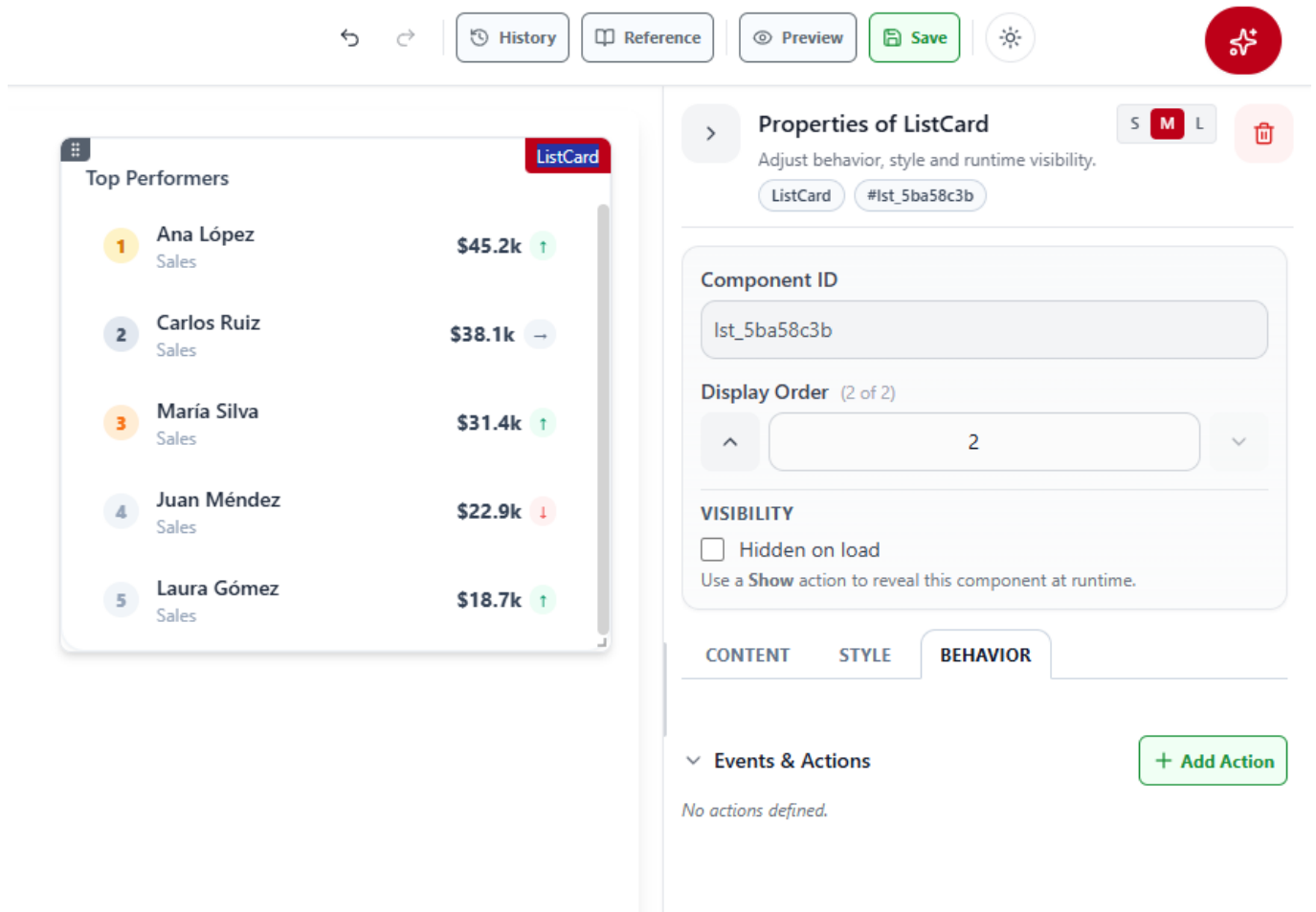
Events & Actions

+ 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.



Ejemplo de uso

Para agregar un nuevo ítem a la lista, hay que ubicarse dentro del array en **Data**, agregar una coma después del último objeto }, y pegar el nuevo bloque a continuación:

```
[{"label":"Pedro Fernández","sublabel":"Sales","value":15300,"trend":"up"}]
```

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

Top Performers

1	Ana López	\$45.2k	↑
	Sales		
2	Carlos Ruiz	\$38.1k	→
	Sales		
3	María Silva	\$31.4k	↑
	Sales		
4	Juan Méndez	\$22.9k	↓
	Sales		
5	Laura Gómez	\$18.7k	↑
	Sales		
6	Pedro Fernández	\$15.3k	↑
	Sales		

Con **trend** se indica visualmente si ese valor mejoró (up), empeoró (down) o se mantuvo estable (neutral) respecto a un período anterior. Con **Value Prefix** y **Value Suffix** se da formato al valor sin modificar el JSON (por ejemplo, \$ como prefijo para montos en dólares). Con **Show Rank** se puede optar por mostrar u ocultar la numeración de posición, útil si la lista no representa necesariamente un ranking sino solo un listado de ítems con valores.

Nexus – Elemento: Sparkline



¿Qué es?

Sparkline es un componente de tipo **Data**. Muestra un mini gráfico de tendencia, minimalista y sin ejes visibles, ideal para mostrar la evolución de un valor en poco espacio. Al arrastrarlo al lienzo viene con un ejemplo precargado (gráfico de línea, “Last 10 days”).



Propiedades

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

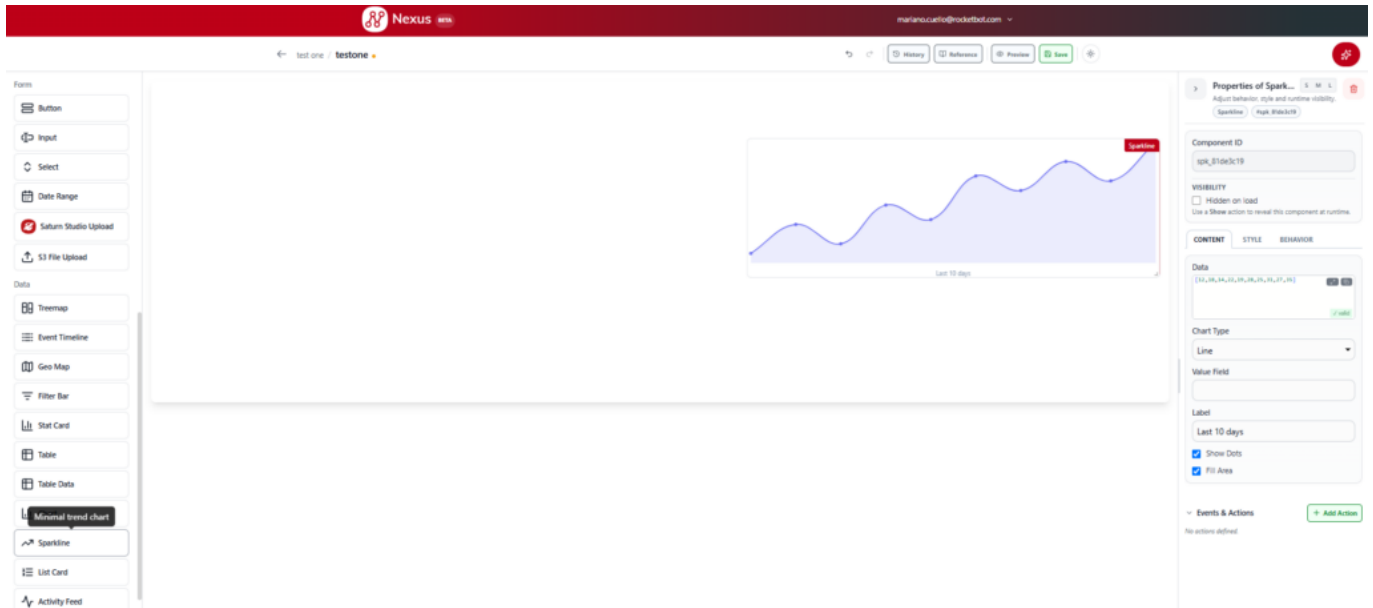
- **Component ID**: identificador único del componente, autogenerado (ej. spk_8lde3c19).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).

Pestaña Content:

- **Data**: array simple de números que representa los puntos del gráfico (ej. [12,18,14,22,19,28,25,31,27,35]).
- **Chart Type**: menú desplegable que define el tipo de gráfico. Opciones: **Line**, **Bar**.
- **Value Field**: nombre del campo a usar como valor, cuando los datos provienen de una estructura más compleja (objetos) en vez de un array simple de números (vacío por defecto).
- **Label**: texto descriptivo debajo del gráfico (ej. “Last 10 days”).
- **Show Dots**: casilla que muestra u oculta los puntos sobre la línea del

gráfico.

- **Fill Area:** casilla que rellena de color el área debajo de la línea del gráfico.



Pestaña Style:

- **Color:** color de la línea/barras del gráfico, con selector visual y campo hexadecimal (ej. #6366f1).
- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.



Properties of Spark... S M L

Adjust behavior, style and runtime visibility.

Sparkline #spk_81de3c19

Component ID
spk_81de3c19

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

CONTENT **STYLE** **BEHAVIOR**

STYLE
Color: #6366f1
Custom Class:
Custom Style:

Events & Actions

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.



Properties of Spark... S M L

Adjust behavior, style and runtime visibility.

Sparkline #spk_81de3c19

Component ID
spk_81de3c19

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

CONTENT **STYLE** **BEHAVIOR**

Events & Actions

No actions defined.

Ejemplo de uso

Sparkline es ideal para acompañar una métrica (por ejemplo, dentro de una **Stat Card** o al lado de un **KPI Card**) mostrando su tendencia reciente sin ocupar el espacio de un gráfico completo. Con **Data** se carga la serie de valores directamente como array de números (ej. [12,18,14,22,19,28,25,31,27,35]), y con **Chart Type** se elige si se representa como línea continua o como barras. Activando **Fill Area** y **Show Dots** se le da más presencia visual al gráfico, útil cuando el Sparkline es un elemento protagonista y no solo un acompañamiento discreto.

Nexus – Elemento: Chart



¿Qué es?

Chart es un componente de tipo **Data**. Muestra un gráfico de datos con múltiples tipos de visualización disponibles. Al arrastrarlo al lienzo viene con un ejemplo precargado (“Monthly Revenue”, gráfico de barras con datos de Enero a Mayo).



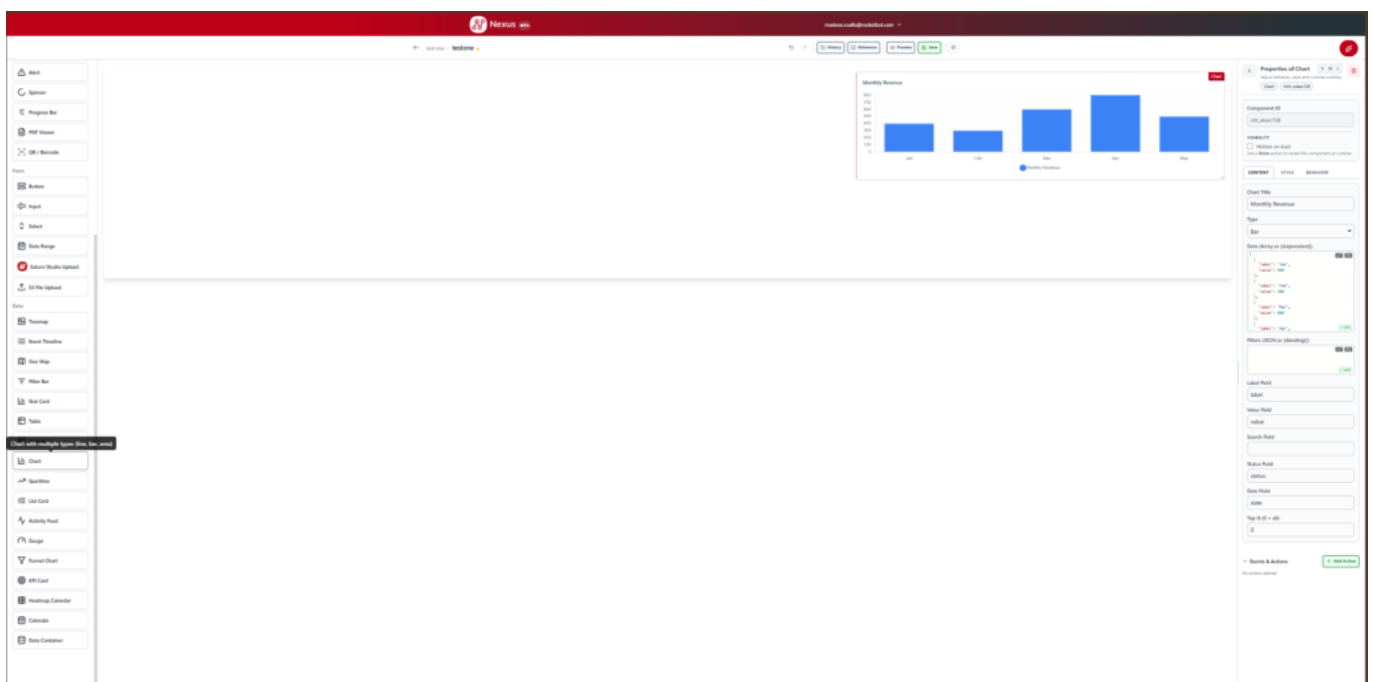
Propiedades

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

- **Component ID:** identificador único del componente, autogenerado (ej. cht_a4aec728).
- **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:

- **Chart Title:** título del gráfico (ej. “Monthly Revenue”).
- **Type:** menú desplegable que define el tipo de gráfico. Opciones: **Bar**, **Line**, **Pie**, **Doughnut**.
- **Data (Array or {{expression}}):** campo de código donde se cargan los datos del gráfico en formato JSON (array de objetos label / value), o alternativamente mediante una expresión/binding dinámico ({{...}}). Cuenta con el editor de código expandido.
- **Filters (JSON or {{binding}}):** campo para aplicar filtros a los datos mostrados, en JSON o mediante binding.
- **Label Field:** nombre del campo que se usa como etiqueta de cada dato (por defecto: label).
- **Value Field:** nombre del campo que se usa como valor de cada dato (por defecto: value).
- **Search Field:** nombre del campo usado para búsqueda/filtrado por texto (vacío por defecto).
- **Status Field:** nombre del campo usado para filtrar por estado (por defecto: status).
- **Date Field:** nombre del campo usado para filtrar por fecha (por defecto: date).
- **Top N (0 = all):** limita el gráfico a mostrar solo los N valores más altos (0 = mostrar todos).



Expanded Code EditorHTMLCSSJSJSONFormatCopyX

```
1 {
2   "label": "Jan",
3   "value": 400
4 },
5 {
6   "label": "Feb",
7   "value": 300
8 },
9 {
10  "label": "Mar",
11  "value": 600
12 },
13 {
14  "label": "Apr",
15  "value": 800
16 },
17 {
18  "label": "May",
19  "value": 500
20 }
21 }
22
```

Ctrl+Space Autocompletado / Snippets • Shift+Alt+F Formatear • Esc VolverDone

Pestaña Style:

- **Show Legend:** casilla que muestra u oculta la leyenda del gráfico.
- **Stacked:** casilla que apila las series de datos unas sobre otras (relevante en gráficos de barras/líneas con múltiples series).
- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.

↶ ↷

History

Reference

Preview

Save

⚙

Monthly Revenue

Chart



>

Properties of Chart

S M L

🗑

Adjust behavior, style and runtime visibility.

Chart

#cht_a4aec728

Component ID

cht_a4aec728

VISIBILITY

☐ Hidden on load

Use a Show action to reveal this component at runtime.

CONTENT

STYLE

BEHAVIOR

STYLE

☒ Show Legend

☐ Stacked

Custom Class

Custom Style

CSS

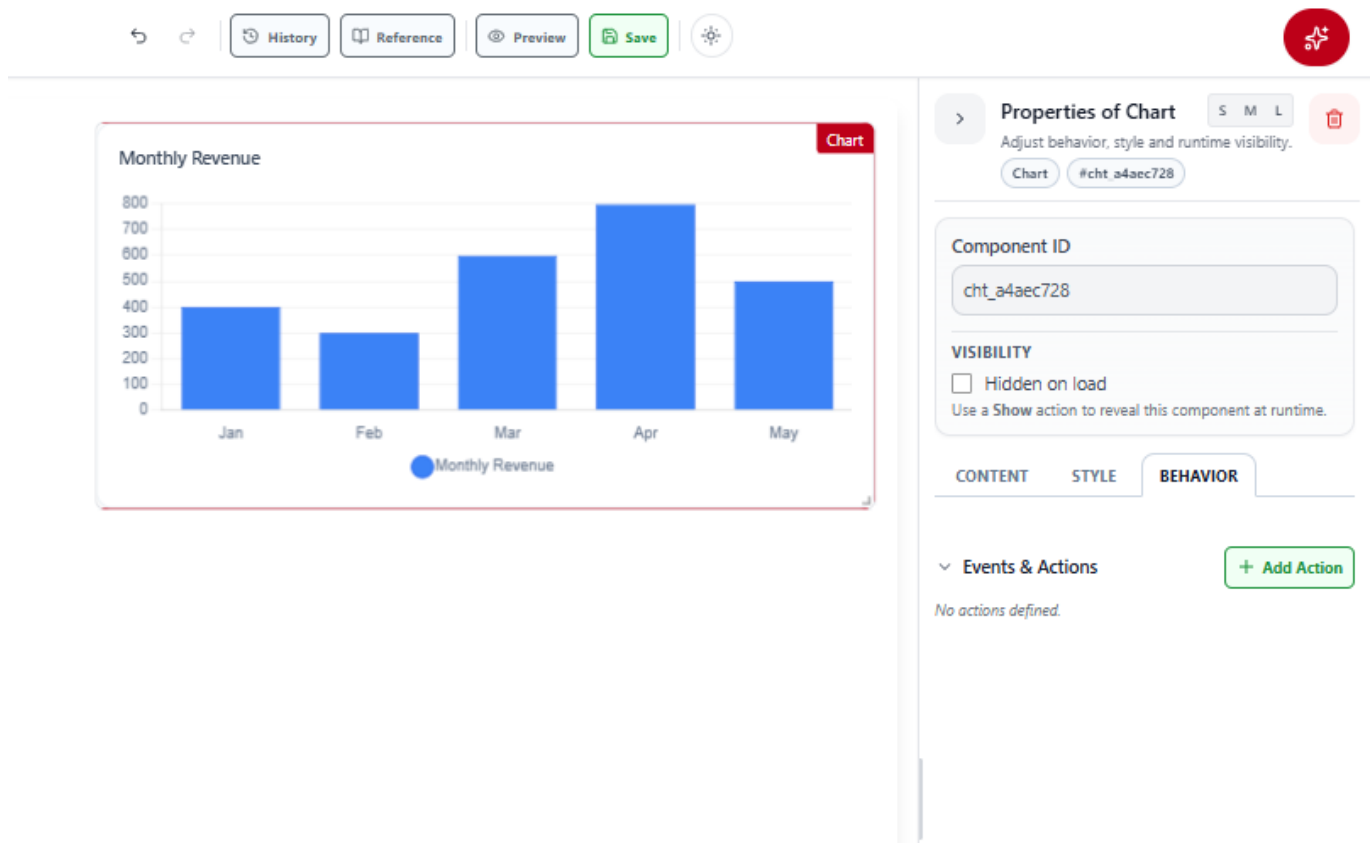
▼ Events & Actions

+ 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.



Ejemplo de uso

Para agregar un nuevo dato al gráfico, hay que ubicarse en la última llave `}` del bloque anterior dentro de **Data**, agregarle una coma, y pegar el nuevo bloque a continuación:

```
[
  {
    "label": "May",
    "value": 500
  },
  {
    "label": "Jun",
    "value": 650
  }
]
```

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



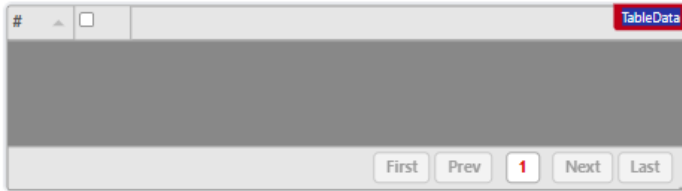
Con **Type** se elige la mejor forma de representar los datos: **Bar** o **Line** para tendencias en el tiempo, y **Pie** o **Doughnut** para mostrar proporciones dentro de un total. Los campos **Label Field** y **Value Field** permiten reutilizar el mismo componente con datos que usen claves distintas a label/value (por ejemplo, datos que vengan de una query con nombres de columna diferentes), sin tener que transformar el JSON. Con **Filters**, **Search Field**, **Status Field** y **Date Field**, el gráfico puede conectarse a un **Filter Bar** para actualizarse dinámicamente según lo que el usuario filtre.

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.



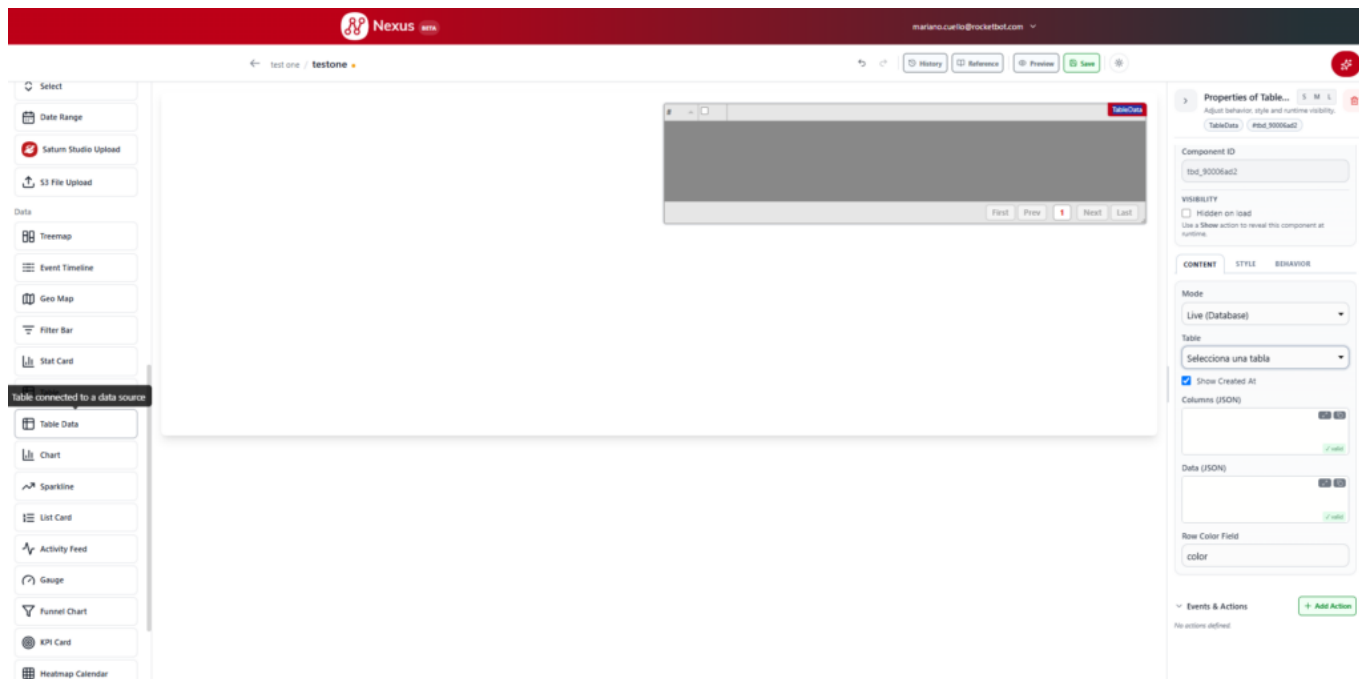
Propiedades

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).

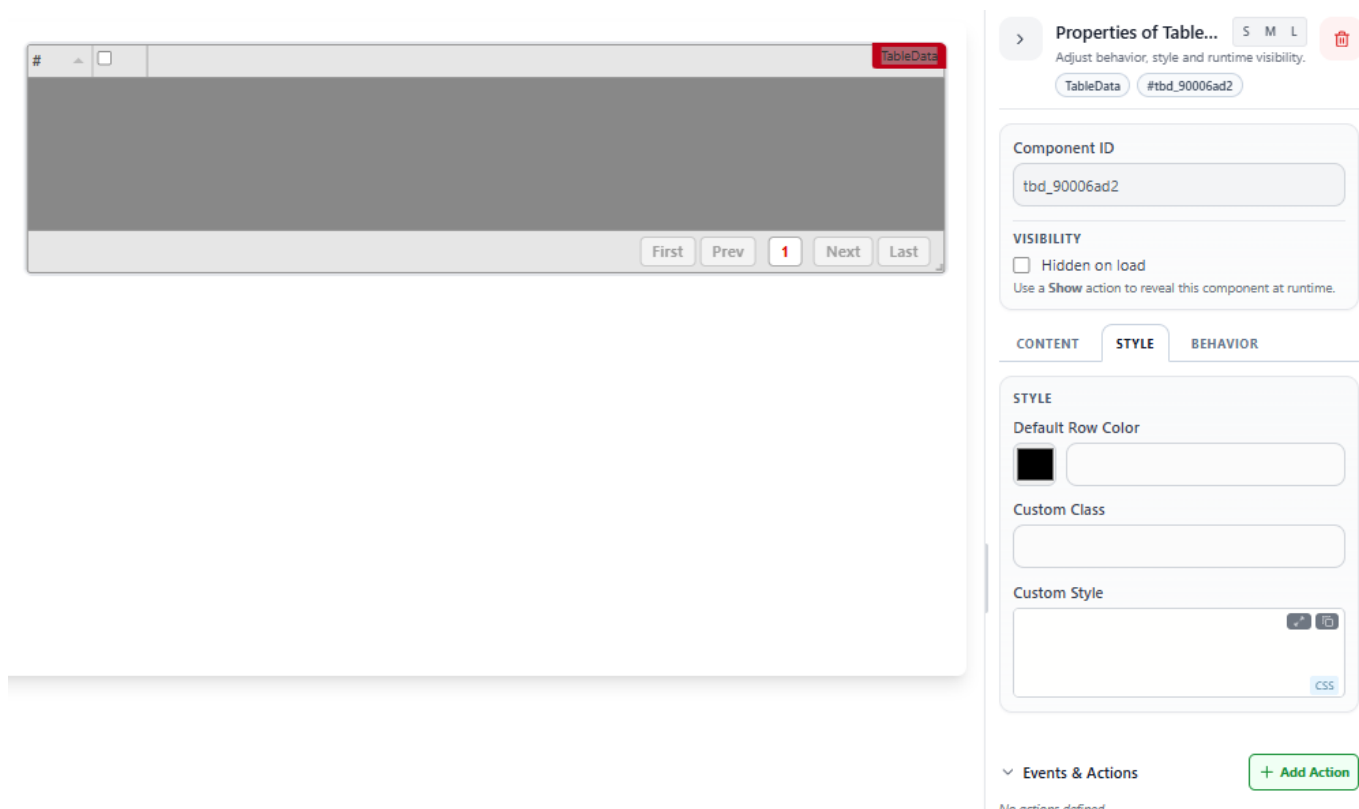
Pestaña Content:

- **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.

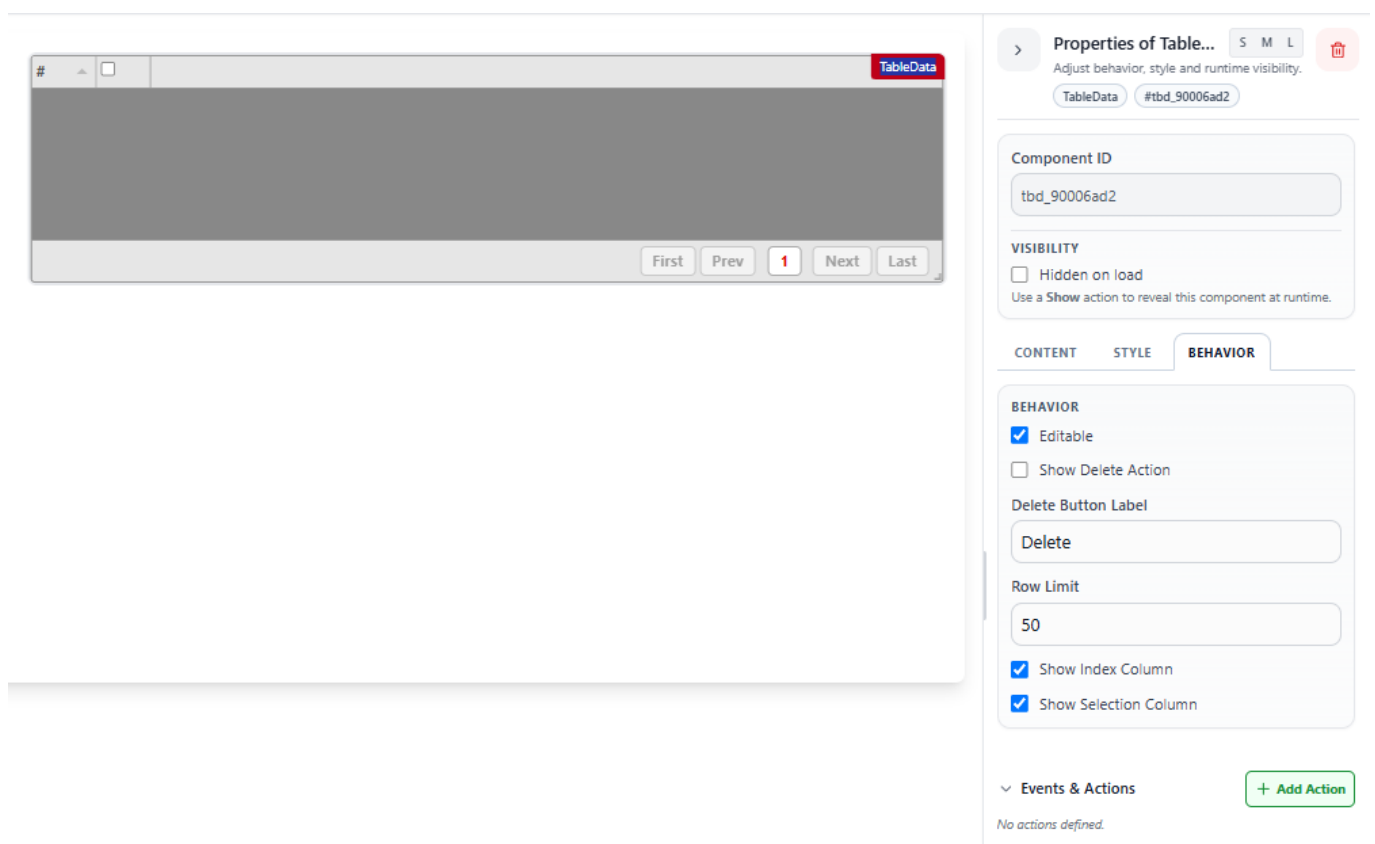


Pestaña Behavior:

- **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

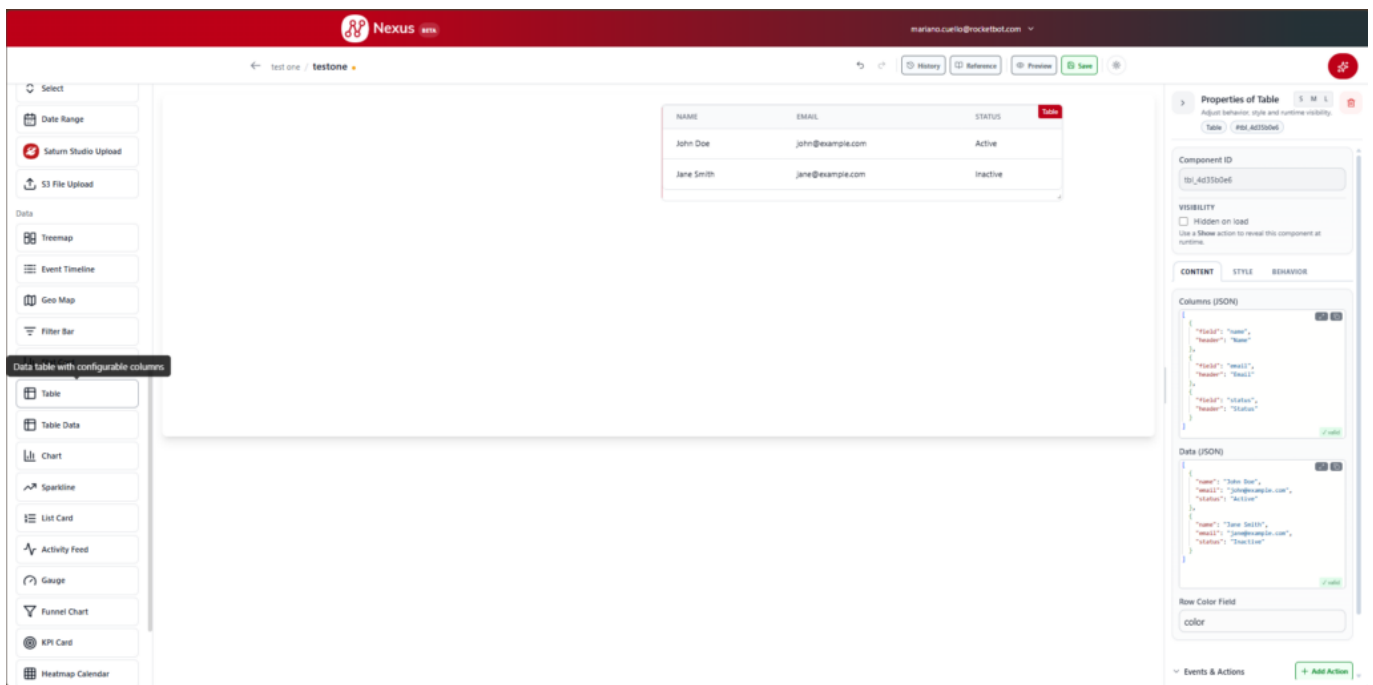
Propiedades

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).



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.

↶ ↷

HistoryReferencePreviewSave⚙️

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

Properties of TableSML🗑️

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.

CONTENTSTYLEBEHAVIOR

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.

↶ ↷

HistoryReferencePreviewSave⚙️

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

Properties of TableSML🗑️

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.

CONTENTSTYLEBEHAVIOR

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

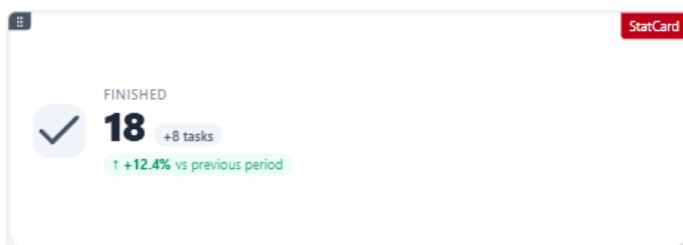
Con **Row Color Field**, si cada objeto de **Data** incluye una clave adicional (por ejemplo "color": "#fee2e2"), esa fila específica toma ese color en vez del color por defecto – útil para resaltar filas según su estado (por ejemplo, en rojo las filas con status "Inactive").

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**.

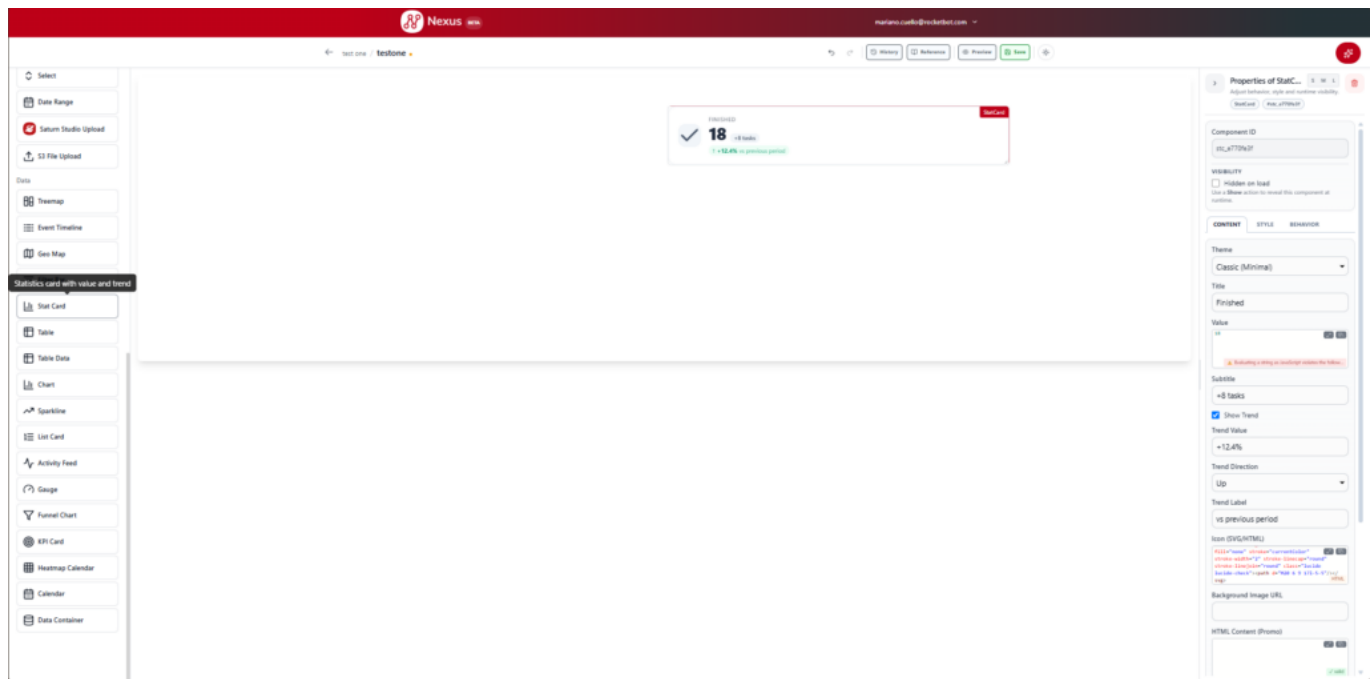
- **Component ID**: identificador único del componente, autogenerado (ej.

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).

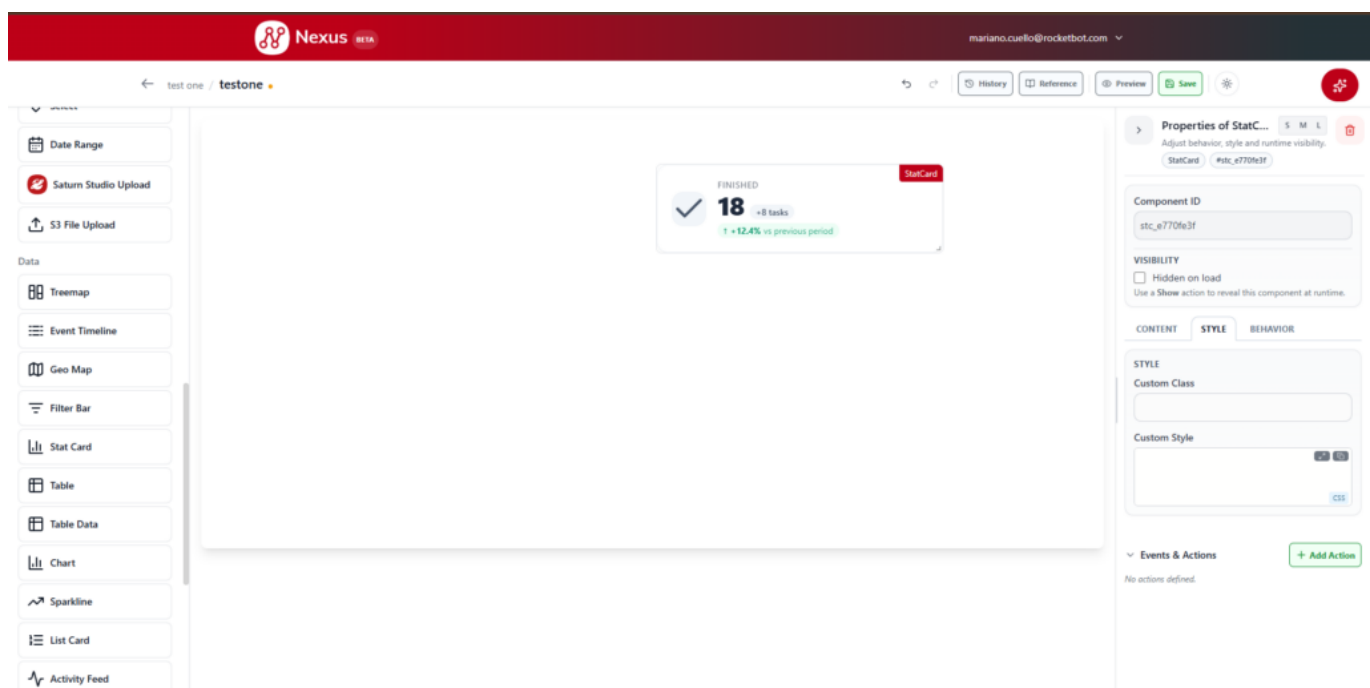
Pestaña Content:

- **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.



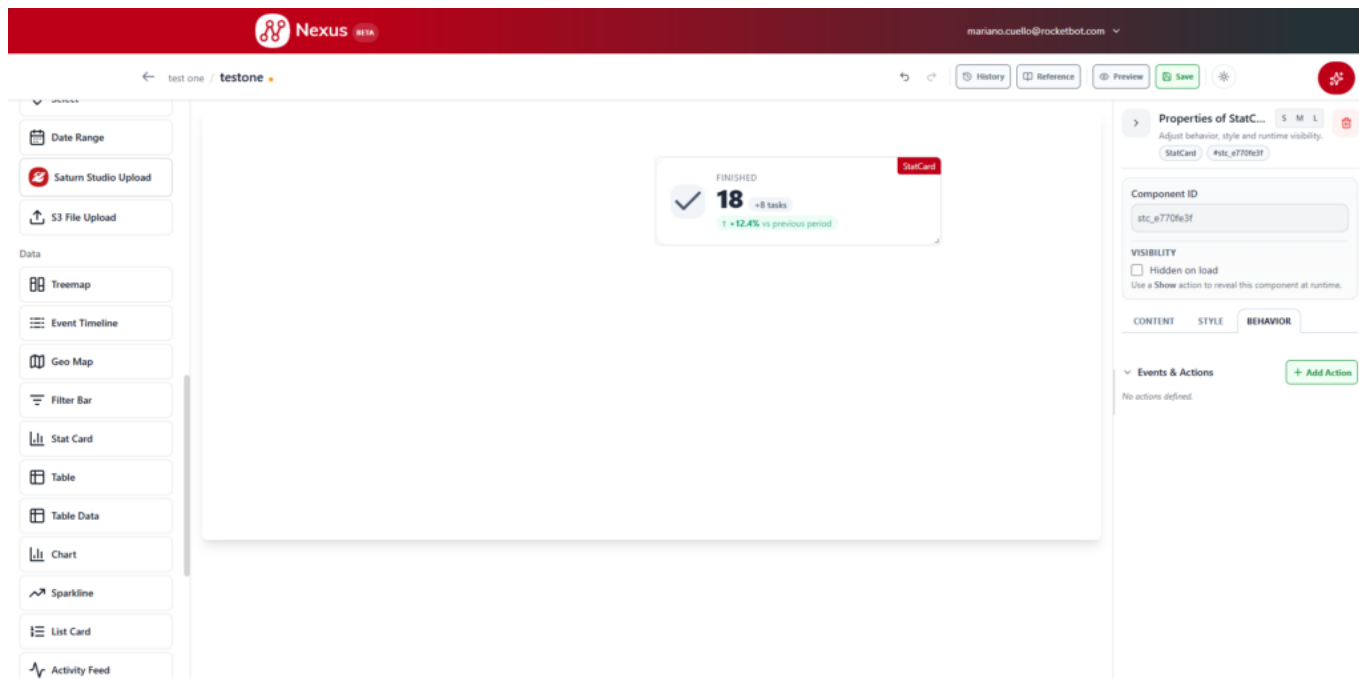
Pestaña Style:

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



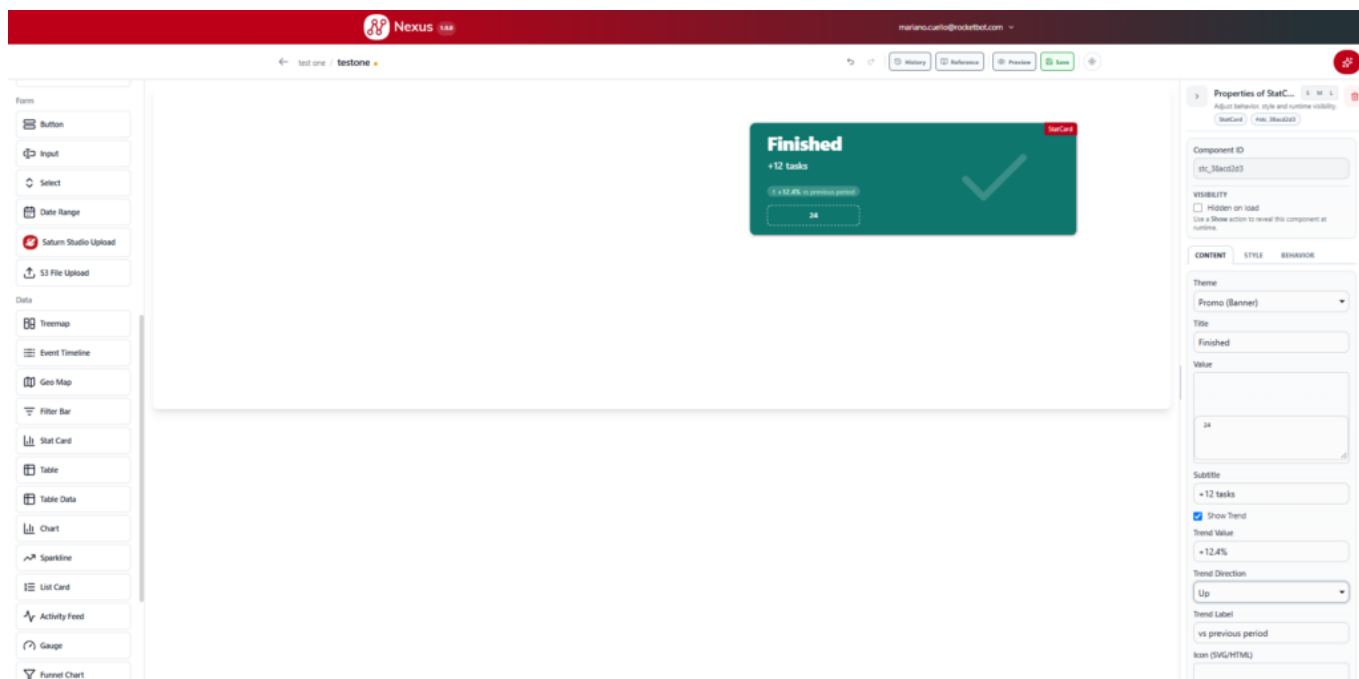
Pestaña Behavior:

- **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.

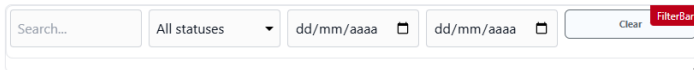


Nexus – Elemento: Filter Bar



¿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”).



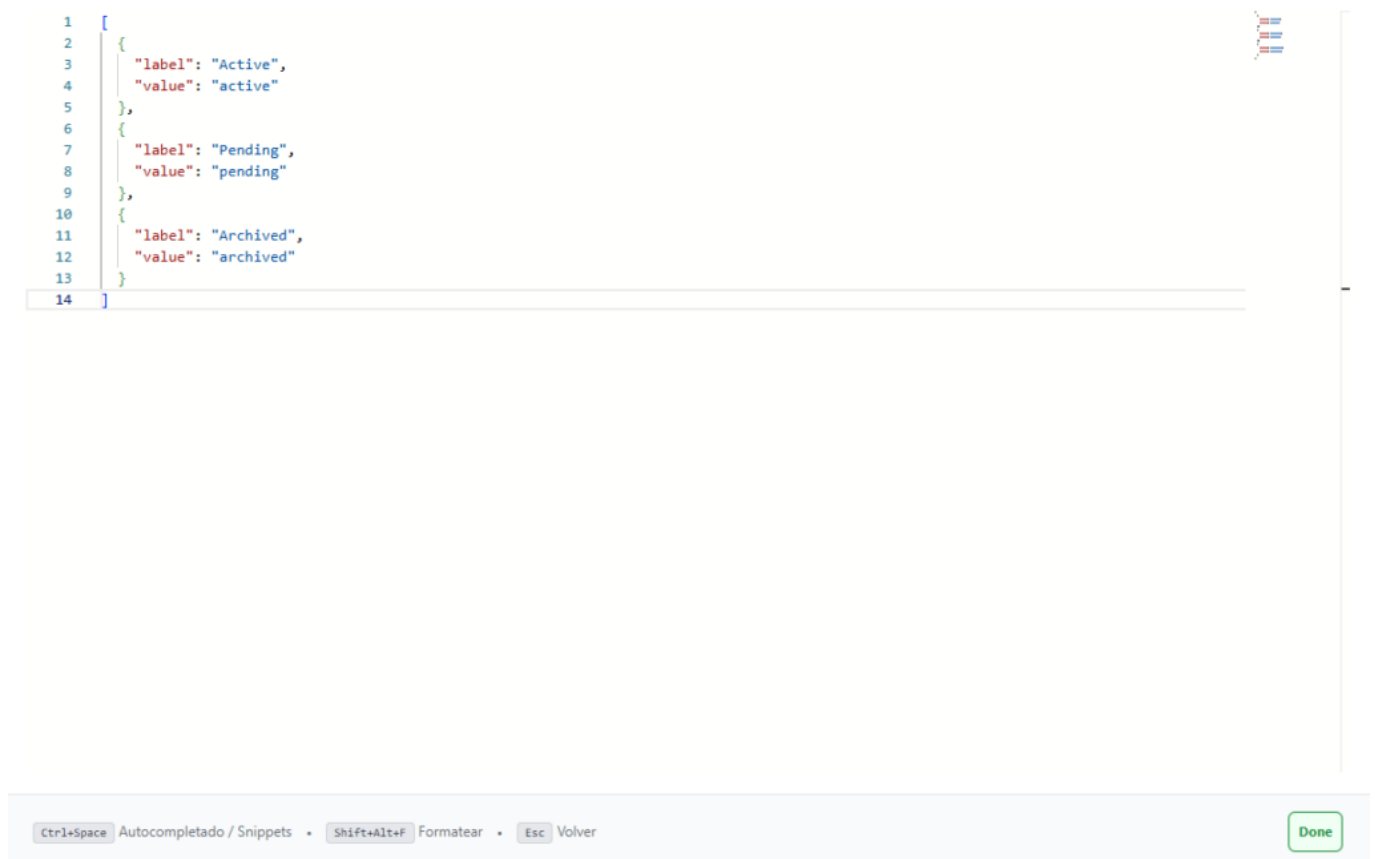
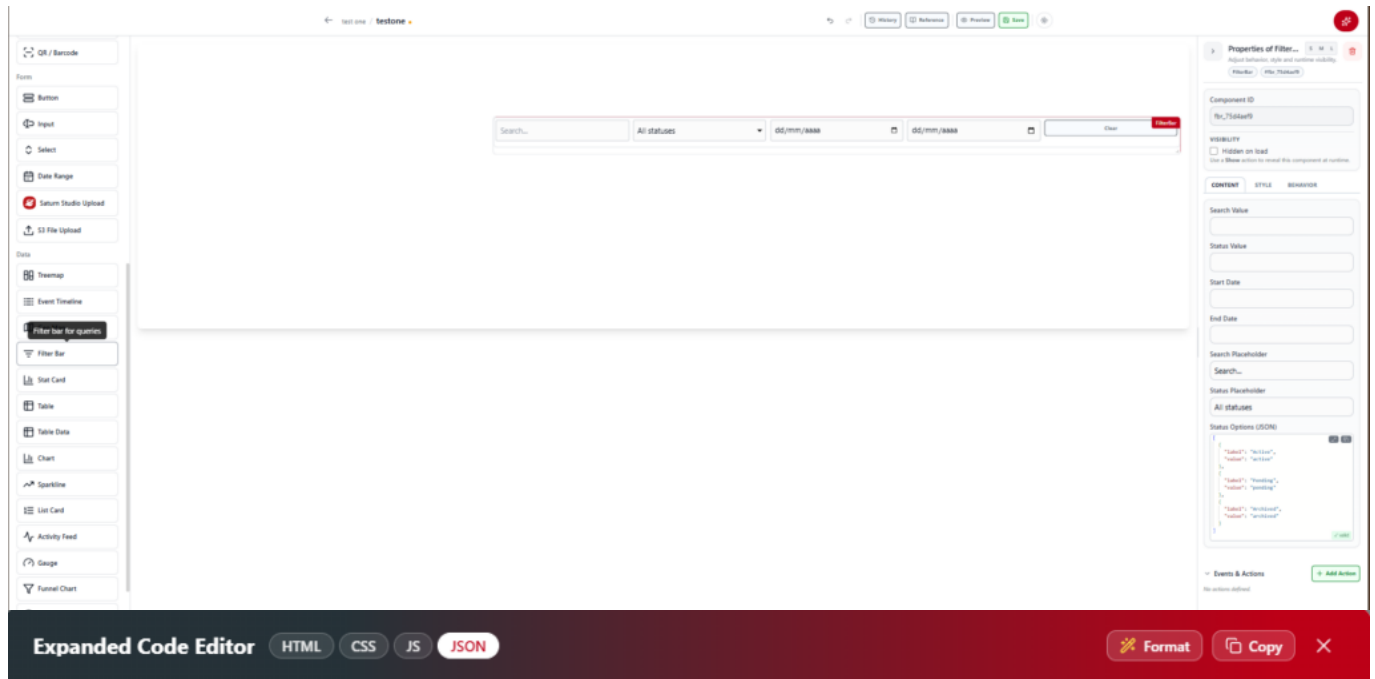
Propiedades

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).

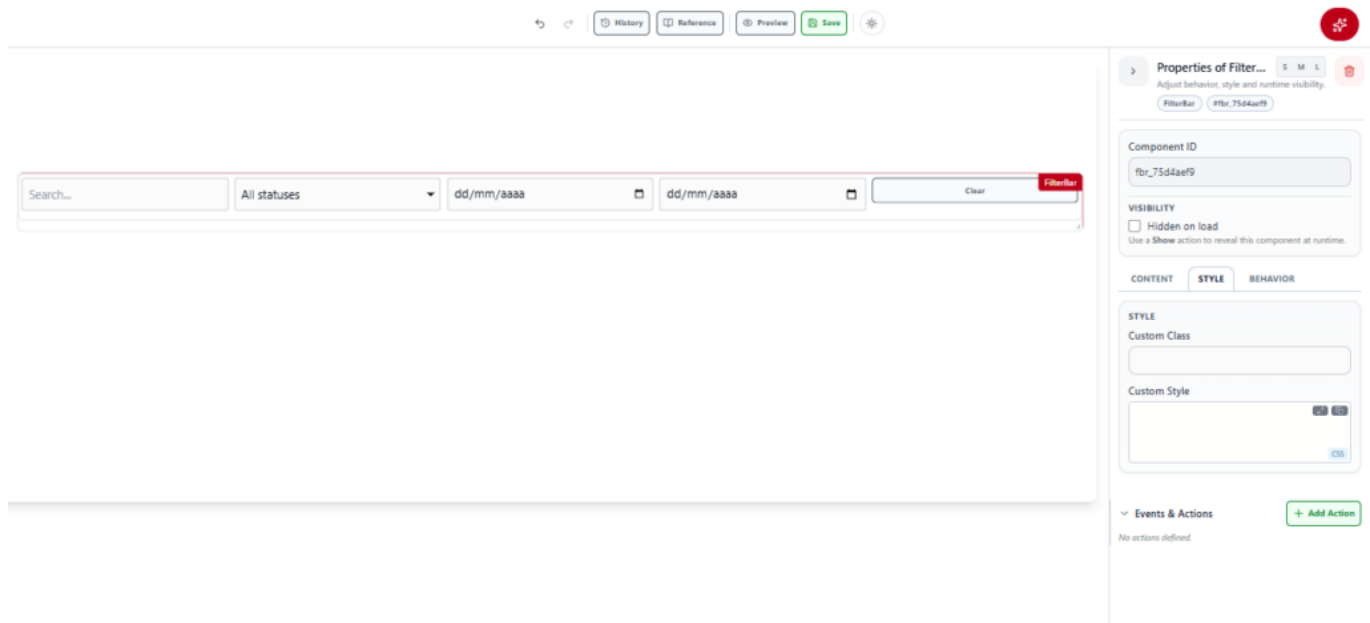
Pestaña Content:

- **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.



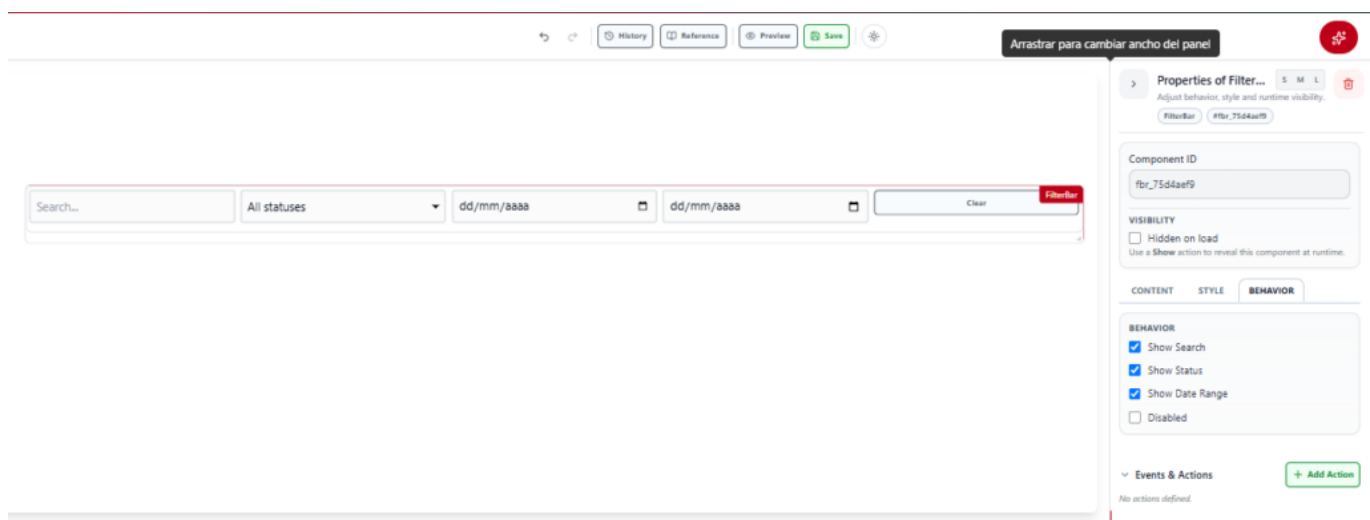
Pestaña Style:

- **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](#).



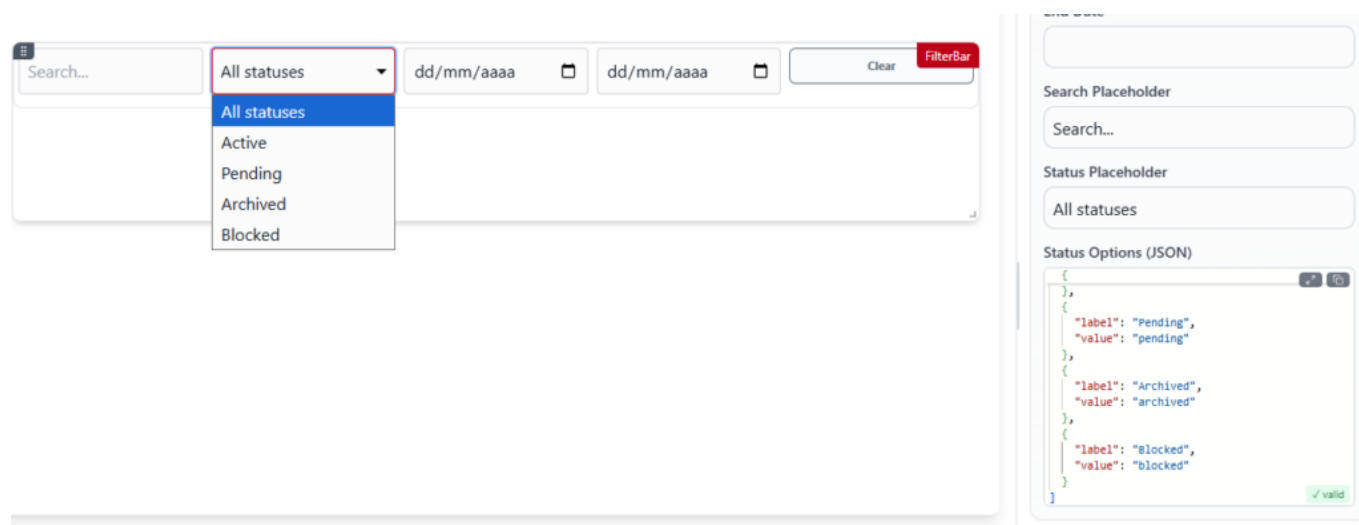
Ejemplo de uso

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 filter bar on the left and its configuration panel on the right. The filter bar includes a search input, a status dropdown menu (currently showing 'All statuses' with options: All statuses, Active, Pending, Archived, Blocked), two date range inputs (dd/mm/yyyy), a 'Clear' button, and a 'FilterBar' button. The configuration panel on the right shows the 'Status Placeholder' set to 'All statuses' and the 'Status Options (JSON)' field containing the following JSON array:

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

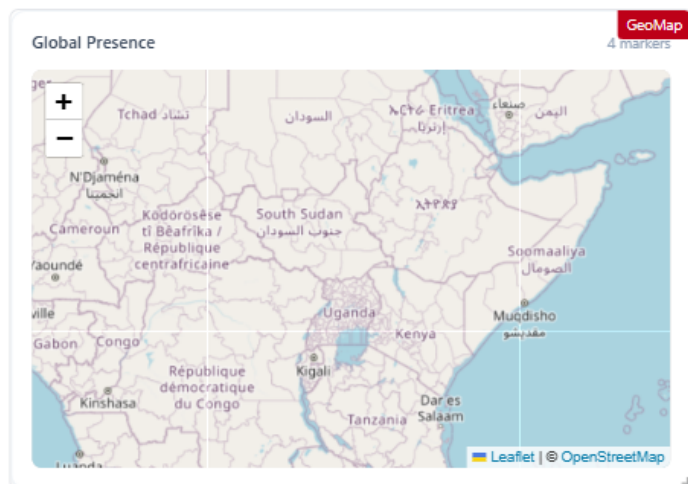
The JSON is marked as 'valid'.

[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

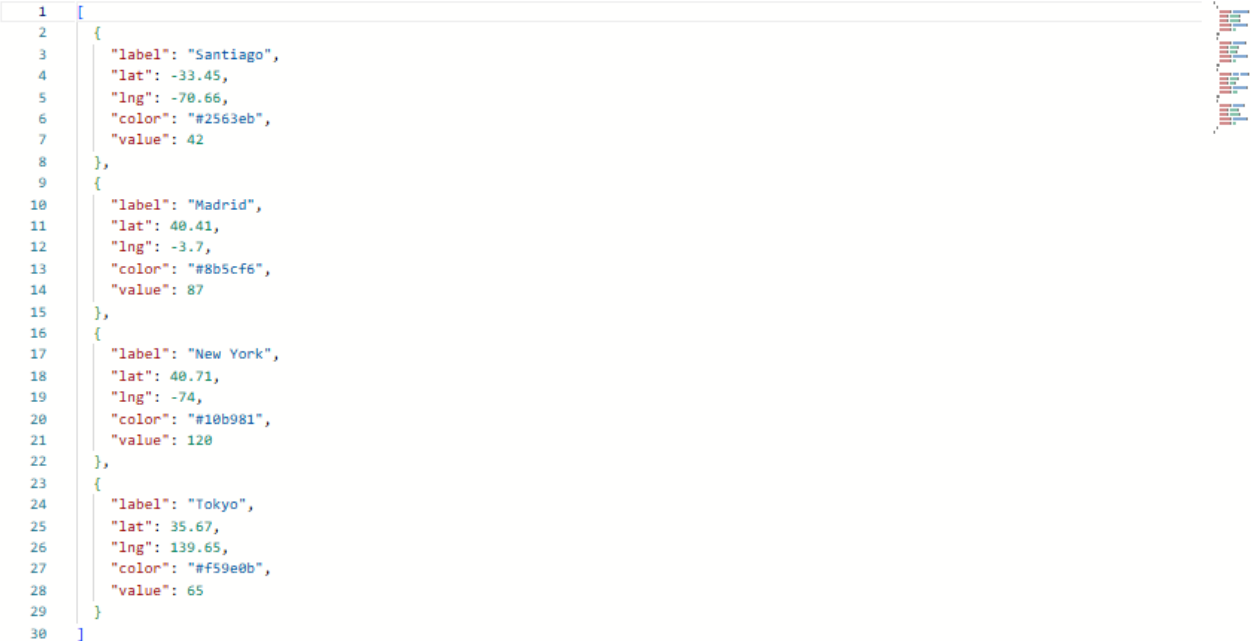
Al seleccionar Geo Map en el lienzo, el panel derecho muestra “**Properties of GeoMap**”, organizado en tres pestañas: **Content**, **Style** y **Behavior**.

- **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).



Pestaña Style:

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

The screenshot displays the GeoMap component in a design tool. The main canvas shows a map of East Africa with labels in Spanish and Arabic. The right sidebar shows the 'Properties of GeoMap' panel with tabs for Content, Style, and Behavior. The Style tab is active, showing fields for 'Custom Class' and 'Custom Style'. The Behavior tab shows 'Events & Actions' with a '+ Add Action' button.

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](#) .

↶

↷

History

Reference

Preview

Save

⚙️

GeoMap

4 markers

Global Presence

>

Properties of Geo...

S M L

🗑️

Adjust behavior, style and runtime visibility.

GeoMap

#geo_afa5dc51

Component ID

geo_afa5dc51

VISIBILITY

☐ Hidden on load

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

CONTENT

STYLE

BEHAVIOR

BEHAVIOR

☒ Show Labels

Max Zoom

⌵ Events & Actions

+ Add Action

No actions defined.

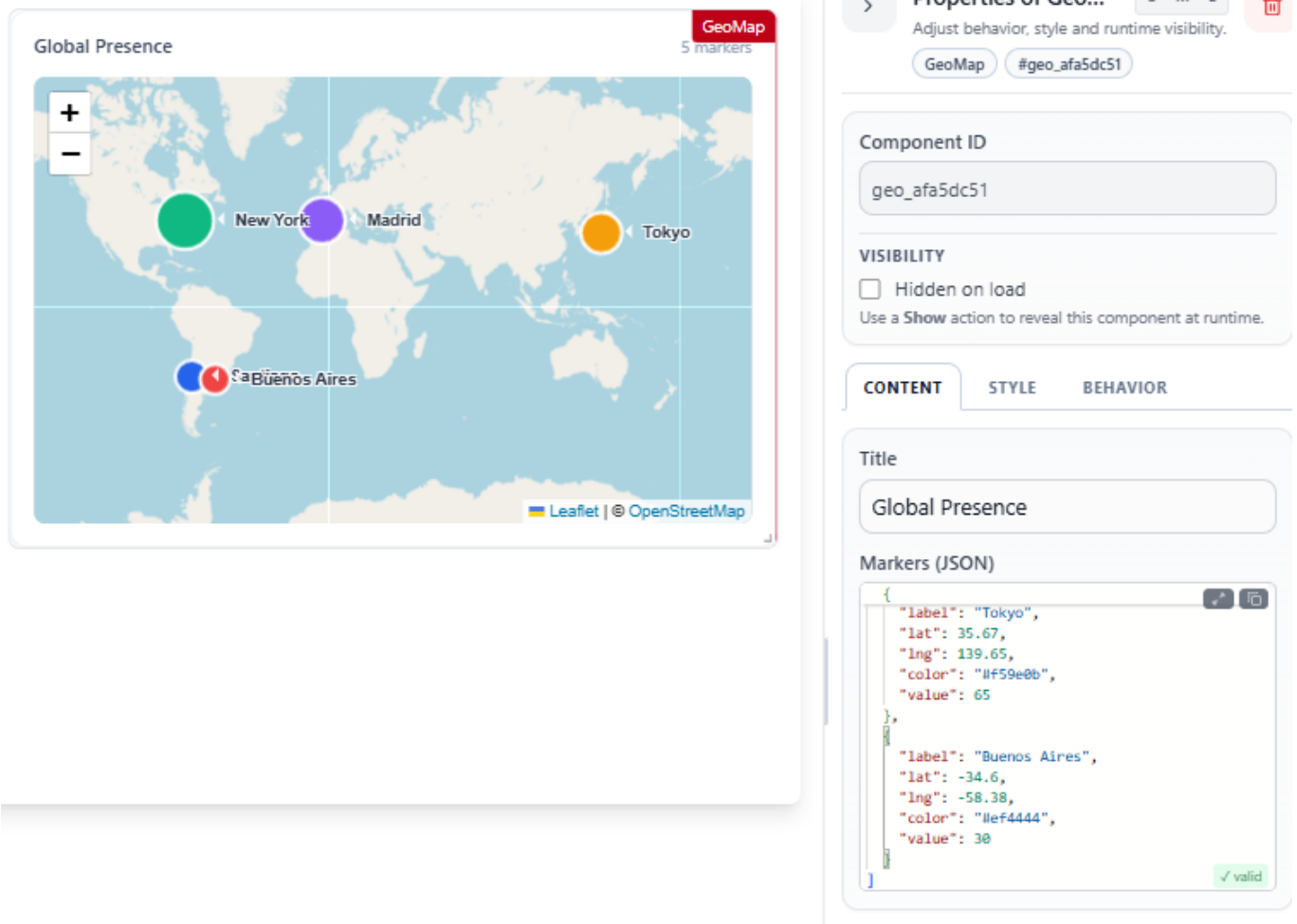
Ejemplo de uso

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 shows the 'Properties of GeoMap' component with a Component ID of 'geo_afa5dc51'. The 'Markers (JSON)' section shows the data for the markers:

```
{
  "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
}
```

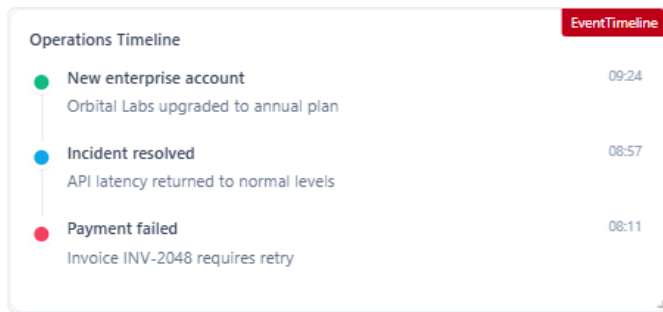
Nexus – Elemento: Event Timeline



¿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).

RP

Nexus

beta

mariano.carrillo@rockethub.com

test one / teststone

HistoryReferencePreviewSave

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

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.

EventTimelineDet_42eff526

Component ID

etl_42eff526

Visibility

Hidden on load

Use a Show action to reveal this component at runtime.

ContentStyleBehavior

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

HTMLCSSJSJSON

FormatCopy

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

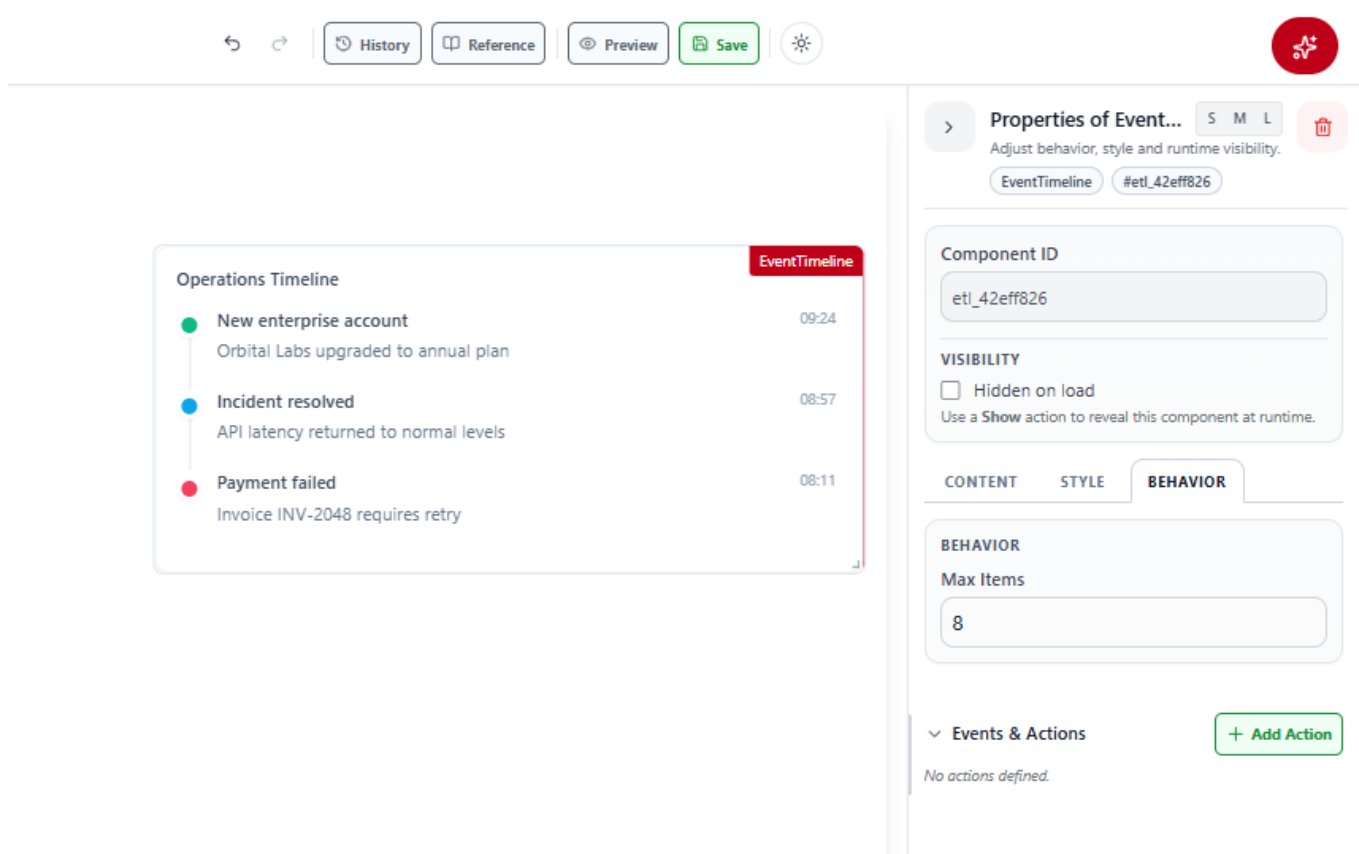
Pestaña Style:

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

The screenshot displays the 'EventTimeline' component in a design tool interface. The main canvas shows a vertical timeline titled 'Operations Timeline' with three events: 'New enterprise account' (09:24), 'Incident resolved' (08:57), and 'Payment failed' (08:11). The right sidebar contains the 'Properties of Event...' panel, which is currently on the 'STYLE' tab. This panel includes fields for 'Component ID' (eti_42eff826), a 'VISIBILITY' section with a 'Hidden on load' checkbox, and a 'STYLE' section with 'Custom Class' and 'Custom Style' input fields. Below the style section is an 'Events & Actions' section with a '+ Add Action' button. The top of the interface features navigation buttons: 'History', 'Reference', 'Preview', 'Save', and a settings icon.

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 is a preview of the component, titled "Operations Timeline", which lists four events with colored circular markers and timestamps:

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

On the right is the "Properties of Event..." panel, which includes:

- Component ID**: etl_93f6d78d
- VISIBILITY**: A checkbox for "Hidden on load" is unchecked. A note states: "Use a **Show** action to reveal this component at runtime."
- Tabs**: "CONTENT", "STYLE", and "BEHAVIOR". The "CONTENT" tab is active.
- Title**: Operations Timeline
- Events (JSON)**: A code editor showing two JSON 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 "valid" status is shown at the bottom right of the editor.
- Events & Actions**: A section with a "+ Add Action" button and the text "No actions defined."

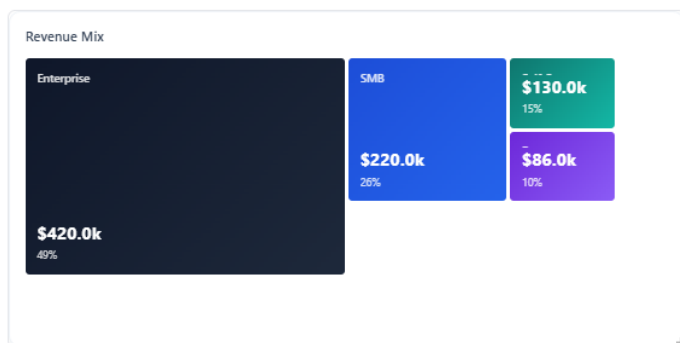
Nexus – Elemento: Treemap



¿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**.

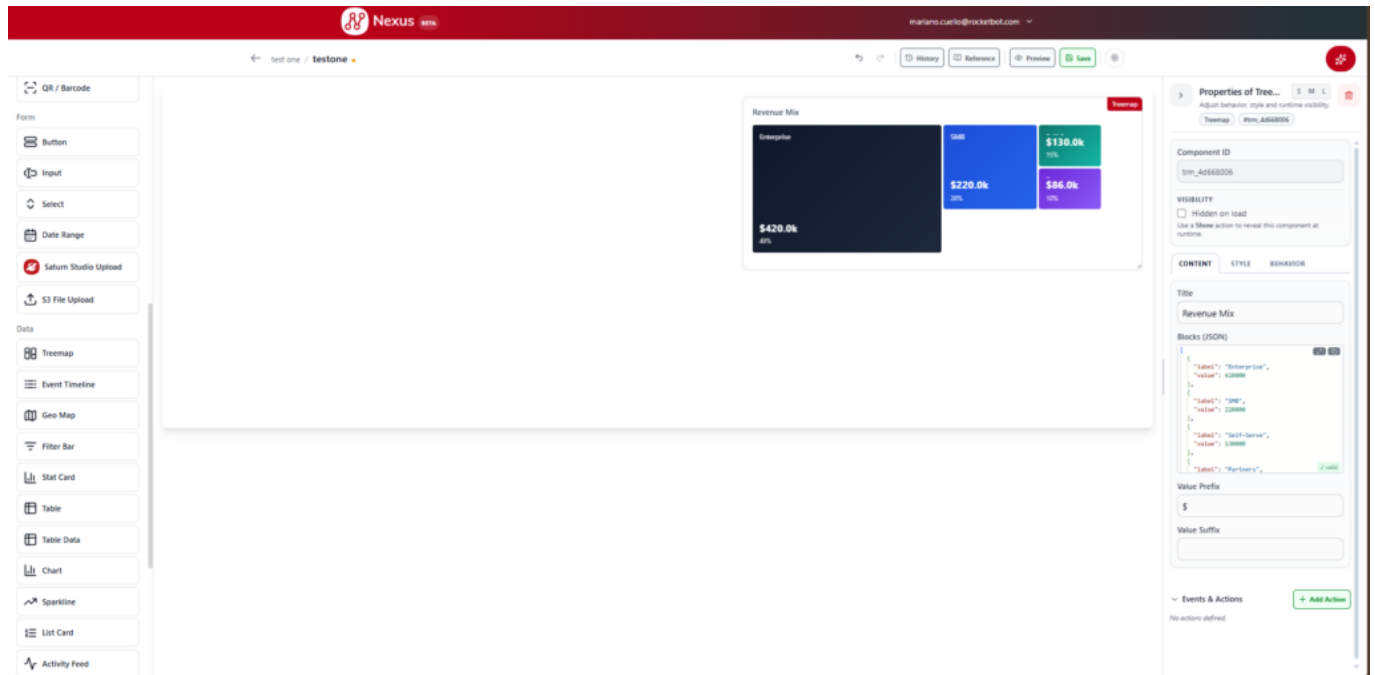
- **Component ID**: identificador único del componente, autogenerado (ej. trm_4d668006).
- **Visibility → Hidden on load**: casilla para que el componente esté oculto al cargar la pantalla. Puede mostrarse luego mediante una acción de tipo **Show** (ver Events & Actions).

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:

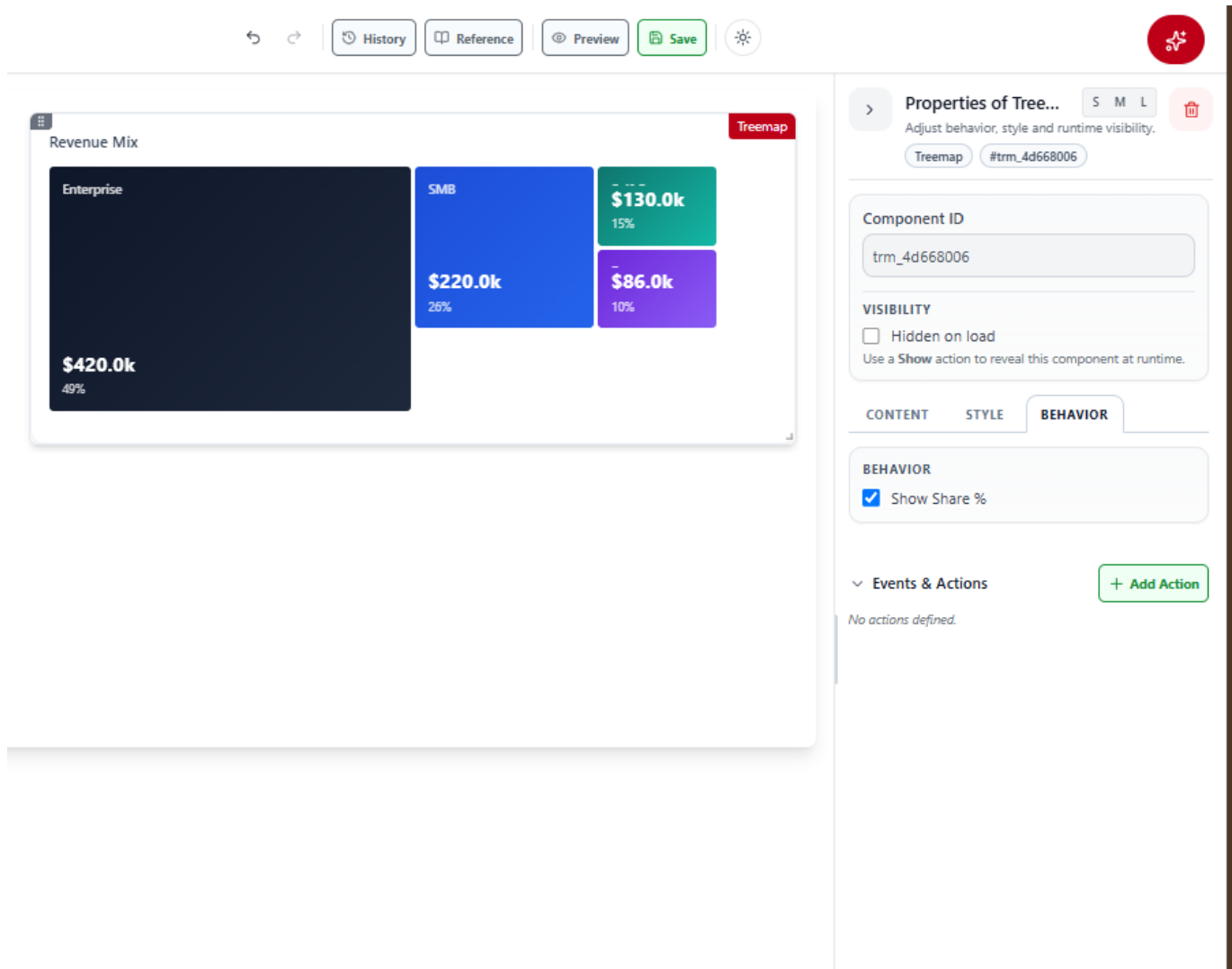
- **Custom Class:** clase CSS personalizada.
- **Custom Style:** CSS personalizado aplicado directamente a este componente.

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 visualization titled "Revenue Mix". The treemap consists of four colored rectangles: a large dark blue rectangle labeled "Enterprise" with a value of "\$420.0k" and a share of "49%", a medium blue rectangle labeled "SMB" with a value of "\$220.0k" and a share of "26%", a green rectangle with a value of "\$130.0k" and a share of "15%", and a purple rectangle with a value of "\$86.0k" and a share of "10%". A red "Treemap" button is located in the top right corner of the visualization area.

On the right side, there is a "Properties of Tree..." panel. It includes a title bar with a collapse icon, a title "Properties of Tree...", and three tabs: "S", "M", and "L". Below the title bar, there is a subtitle "Adjust behavior, style and runtime visibility." and two buttons: "Treemap" and "#trm_4d668006". The panel is divided into three sections: "Component ID" with a text input field containing "trm_4d668006", "VISIBILITY" with a checkbox labeled "Hidden on load" and a note "Use a Show action to reveal this component at runtime.", and "STYLE" with a "Custom Class" text input field and a "Custom Style" text input field with a "CSS" button. Below the "STYLE" section is an "Events & Actions" section with a dropdown arrow and a note "No actions defined.", and a green "+ Add Action" button.

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.

