



Datawheel

DRAFT

## MANUAL CMS

OCTUBRE 2020

## Contenido

<b>Introducción</b>	<b>4</b>
Figura 1: Estructura del CMS	4
<b>Elementos de CMS</b>	<b>5</b>
Generadores	5
Figura 2: Componentes del generador	6
Figura 3: Ejemplo API	6
Generated variables	7
Figura 4: Generador Atributos	8
Notas importantes (Generadores)	9
Materializadores	10
Figura 5: Editor del materializador	11
Figura 6: Creación de nuevas variables	12
Notas importantes (Materializadores)	12
Selectores	12
Figura 7: Editor del selector	13
Figura 8: Creación de nuevas variables	13
¿Cómo crear un selector simple? (2 ó 3 opciones)	14
¿Cómo crear un selector dinámico?	15
Figura 10: Creación selector dinámico	15
Figura 11: Selector dinámico	16
Notas importantes (Selectores)	16
Formateadores	17
Figura 12: Editor del formateador	17
Notas importantes (formateadores)	17
Secciones	18
Figura 13: Estructura de Secciones	18
Metadata	18
Figura 14: Estructura de Sección Metadata	19
Figura 15: Layout default	20
Figura 16: Layout single column	21

Figura 17: Layout multi column	21
Subtitle	22
Figura 18: Selector activation	22
Figura 19: Variable asociada al selector	23
Stats	23
Figura 20: Elementos de la sección Stats	24
Paragraphs	25
Visualization	27
Figura 21: Elementos de la sección Stats	27
<b>Caso práctico</b>	<b>28</b>
Figura 2.1 - Estructura de datos IED	29
Figura 2.2 - Dimensiones de datos sobre IED	29
Figura 2.3 - Jerarquía de datos en la dimensión sector - Industria	30
<b>Caso práctico : Crear una sección en la página web</b>	<b>34</b>

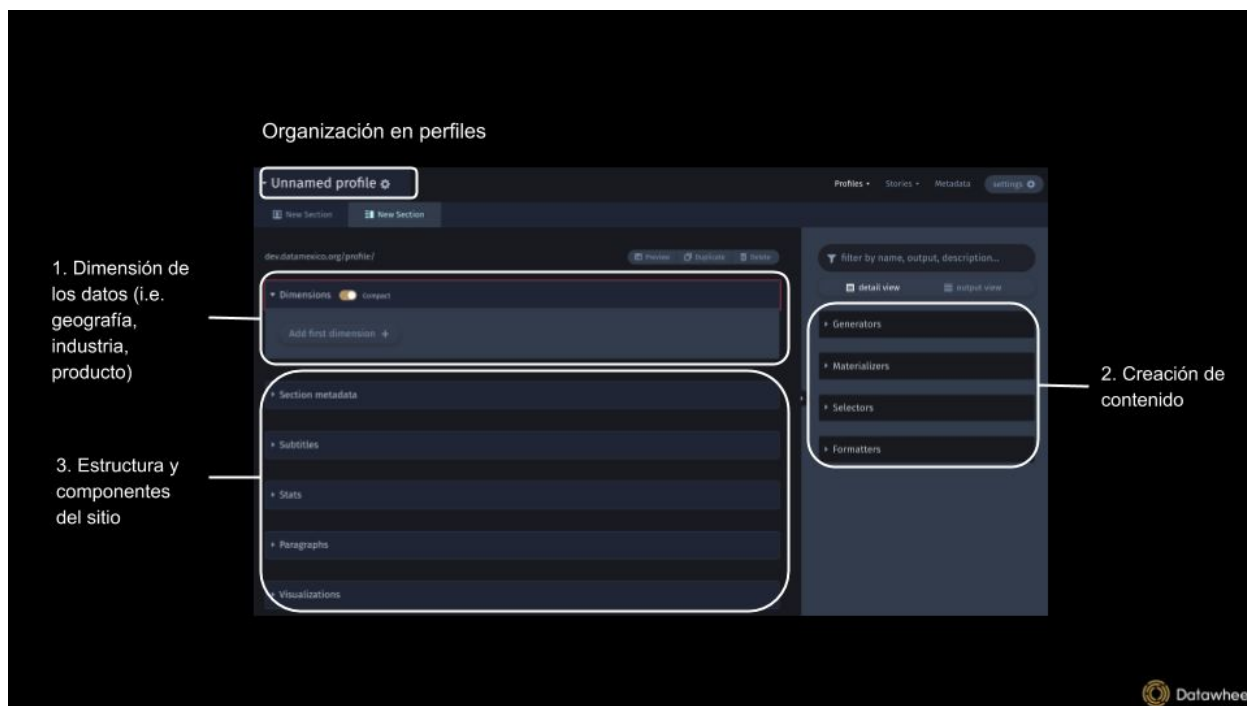
# Content Management System (CMS)

## Introducción

El Content Management System (CMS) es una tecnología creada en Datawheel para facilitar la distribución y visualización de datos. Al utilizar elementos visuales como tablas, gráficos y mapas, el CMS permite crear perfiles que facilitan transmitir información. Si bien el CMS busca ser intuitivo al usuario, permite utilizar Javascript para crear formas avanzadas de contenido.

El CMS se organiza en perfiles independientes que se crean a partir de dimensiones. El CMS está estructurado en tres secciones: (1) dimensiones (*dimensions*) de datos; (2) herramientas para la creación de contenido como generadores (*generators*), materializadores (*materializers*), selectores (*selectors*), y formateadores (*formatters*), y; (3) elementos para dar estructura al sitio: *metadata*, subtítulos (*subtitles*), título (*title*), párrafo (*paragraphs*), y visualizaciones (*visualizations*).

Figura 1: Estructura del CMS



El CMS está organizado en perfiles independientes. Esto permite la generación de contenido que a lo largo del perfil se mantiene por medio de dimensiones compartidas

por los datos y que son definidas por la naturaleza del proyecto. La dimensión de datos es un conjunto de atributos que pertenecen a algo de interés. Las dimensiones son cosas como "países", "productos", "universidades", "industrias", y "años". Para los usuarios del CMS, las dimensiones de datos son puntos de entrada a hechos numéricos (por ejemplo, exportaciones, producción, inversión extranjera, escolaridad, delitos, temperatura) que los usuarios del análisis desean explorar. Una dimensión en el CMS provee la estructura que categoriza los datos para permitir que los usuarios, a través de los perfiles, respondan a sus preguntas.

Cada perfil cuenta con generadores, materializadores, selectores y secciones independientes, por lo que la generación de contenido en cada uno de ellos no afecta el contenido del resto de los perfiles. Una excepción a esta regla corresponde a los formateadores (*formatters*), los cuales son módulos del CMS que permiten evitar tareas repetitivas. Por ejemplo, formato de dólar a valores numéricos (600 → US 600). Los formateadores pueden ser ocupados en distintas partes del CMS. Esto es, pueden aplicarse en los generadores, materializadores, contenido (texto, stats), y las visualizaciones.

Es importante comprender que el ciclo de carga del CMS a la hora de inicializar cada perfil, comienza con los generadores y luego los materializadores. De estos, se pueden cargar y construir los selectores. Se necesita comprender este ciclo para la documentación presentada a continuación.

## Elementos de CMS

Existen cuatro elementos en CMS: (1) generadores; (2) materializadores; (3) selectores, y; (4) formateadores.

### 1. Generadores

Los generadores permiten crear variables dinámicas para construir elementos dinámicos dentro del perfil (textos, visualizaciones, selectores, etc). Su construcción, comienza con la creación de un nuevo generador en la pestaña *Generators* disponible en el sidebar derecho dentro de un perfil en el CMS.

Un generador cuenta con 4 elementos, 2 obligatorios, y 2 opcionales:

- Name (**buena práctica**): identifica los elementos disponibles.
- Descripción (**opcional**): determina el uso del generador.

- API (**obligatorio**): dirección (link) o query proveniente de una api disponible (i.e. Tesseract UI); debe corresponder a un Object Array (figura 3).
- Generated variables (**obligatorio**): Permite el procesamiento de los datos obtenidos desde la api. Se puede construir de forma automática o personalizada. Para una asignación automática, es posible llamar a los datos mediante el botón Fetch data. Para un tratamiento personalizado, es posible habilitar un editor de texto de JavaScript mediante el switch UI mode.

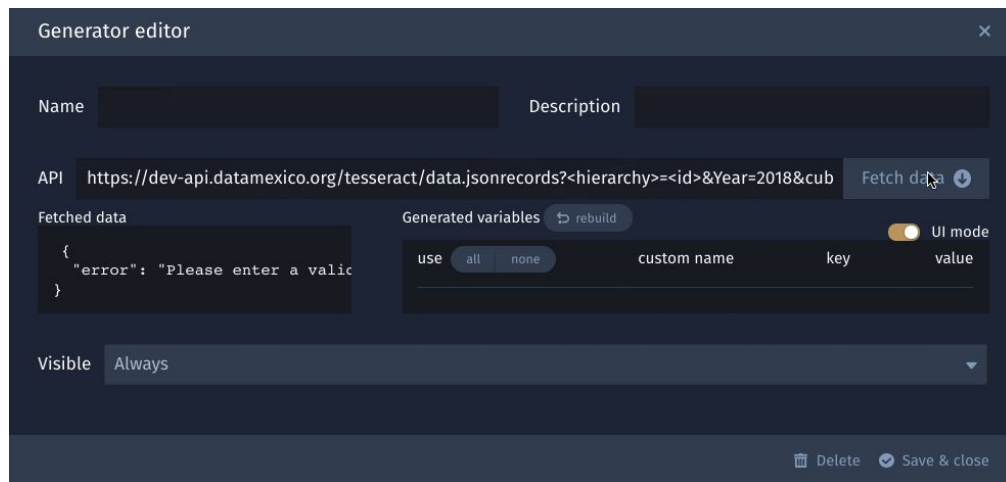
Figura 2: Componentes del generador

Figura 3: Ejemplo API

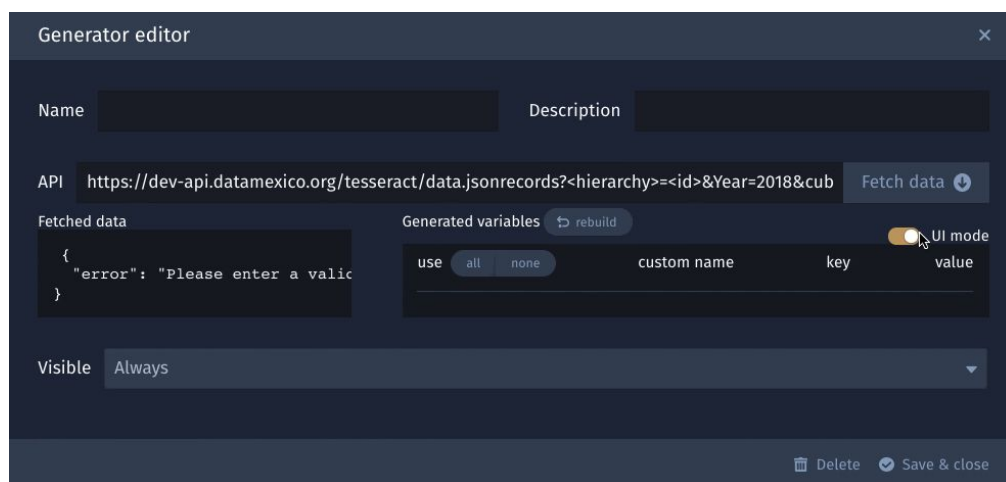
```
- data: [
  - {
    Nation ID: "mex",
    Nation: "México",
    Time ID: 20200113,
    Time: "2020-01-13",
    Daily Cases: 1
  },
  - {
    Nation ID: "mex",
    Nation: "México",
    Time ID: 20200114,
    Time: "2020-01-14",
    Daily Cases: 0
  },
  - {
    Nation ID: "mex",
    Nation: "México",
    Time ID: 20200115,
    Time: "2020-01-15",
    Daily Cases: 0
  }
],
```

## Generated variables

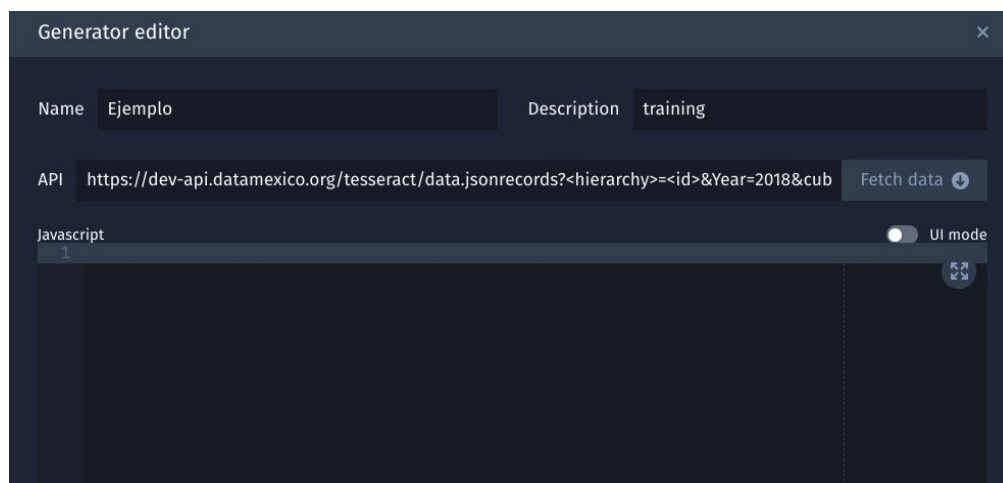
Variables retornadas desde la api call con para utilizar o procesar los datos. Su construcción permite realizar una asignación automática de las variables con respecto a los datos retornados, o una asignación personalizada. Para efectuar la asignación automática, luego de incluir la api se realiza una primera llamada a los datos mediante el botón Fetch data. Esta acción genera los valores correspondientes a partir de la query realizada.



En caso de requerir el tratamiento personalizado de los datos, hay que habilitar un editor de texto de JavaScript mediante el switch UI mode. Con este editor, es posible procesar los datos obtenidos de la api a través de los principales métodos disponibles para JavaScript (Math, filter, map, forEach, entre otros).

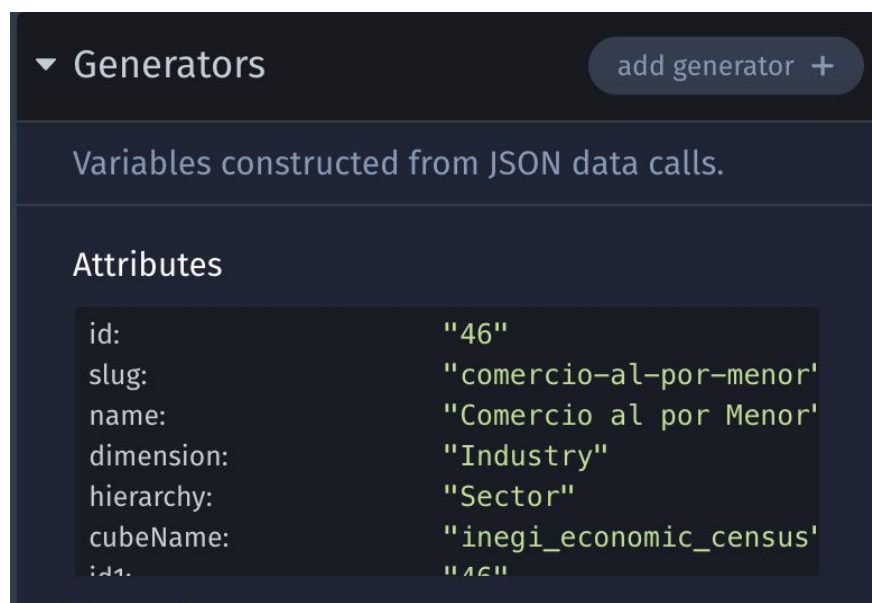


Para acceder a los datos, es necesario inicializar el editor de texto con la asignación de la respuesta correspondiente a una variable. Es importante que cada generador retorne un objeto que contenga las variables correspondientes.



Un generador permite desarrollar contenido dinámico dentro del perfil. Es importante recordar que los generadores de un perfil no están disponibles para otros perfiles, porque las variables predefinidas (en la carga de cada perfil) provienen de un generador fijo (primer generador) denominado Atributos (Attributes).

Figura 4: Generador Atributos





Las variables que están en la sección de Atributos corresponden a valores precargados disponibles para la generación de contenido. Estas variables pueden ser utilizadas para la construcción de la api de cada generador, o en el preprocesamiento de dichos datos. Los generadores pueden responder de forma dinámica a las variables creadas en Atributos. Para ello, es necesario incluir dichos valores entre angle brackets (<>):

```
/api/trade/data?<hierarchy>=<id>&Year=2018&cube=inegi_denue&drilldowns=Company+Size&measures=Companies
```

En caso de ser requeridos en el editor de código, es necesario inicializar dichas variables con:

```
const {hierarchy, id} = variables;
```

### Notas importantes (Generadores)

- Dada la estructura de carga de un generador, **NO** es posible utilizar variables retornadas en un generador en la construcción de otro generador. Esta nota aplica tanto a la generación del api link como a la generación de variables dentro del editor de código relacionado.
- Una llamada incorrecta de datos, aplicada a una autogeneración de variables, puede provocar una inestabilidad importante dentro del CMS, causando problemas en el navegador web del creador de contenido. Favor usar esta funcionalidad solo en el caso en que las variables retornadas sean un valor reducido.
- Dos variables retornadas por generadores **NO** pueden compartir un mismo nombre. Lo anterior generará un conflicto entre las variables generadas y por ende un conflicto en las secciones posteriores en donde estas son utilizadas. En caso de ocurrir una sobreasignación de variables a un mismo nombre, el CMS resaltará los elementos en conflicto y que requieren una revisión por parte del generador de contenido.
- Los generadores corresponden al primer punto de carga de todo perfil, por lo que es relevante que la llamada de datos asociada sea justa y necesaria y no realice una excesiva llamada de datos que puedan afectar el rendimiento del perfil.
- Si se utiliza el editor de código para la creación de variables en un generador, es importante estar **SEGUROS** de que el tratamiento relacionado dentro del editor de texto posea siempre un retorno válido y no genere conflictos durante el proceso. Un ejemplo recurrente, es retornar el primer elemento de los datos recibidos o `data[0]`, elemento que no existe si el retorno de la llamada de datos es un arreglo vacío.

- Un generador **NO** debe retornar un elevado número de variables de no ser necesario. Un elevado número de variables retornadas genera problemas de rendimiento dentro del CMS y del sitio en desarrollo. En casos excepcionales, retornos demasiado largos, pueden provocar una pérdida en la obtención del resto de los generadores, lo cual puede afectar el proyecto en desarrollo en su totalidad.
- De ser posible, es importante analizar los datos obtenidos de cada generador, de modo de evitar realizar múltiples llamadas a un mismo api link.

## Materializadores

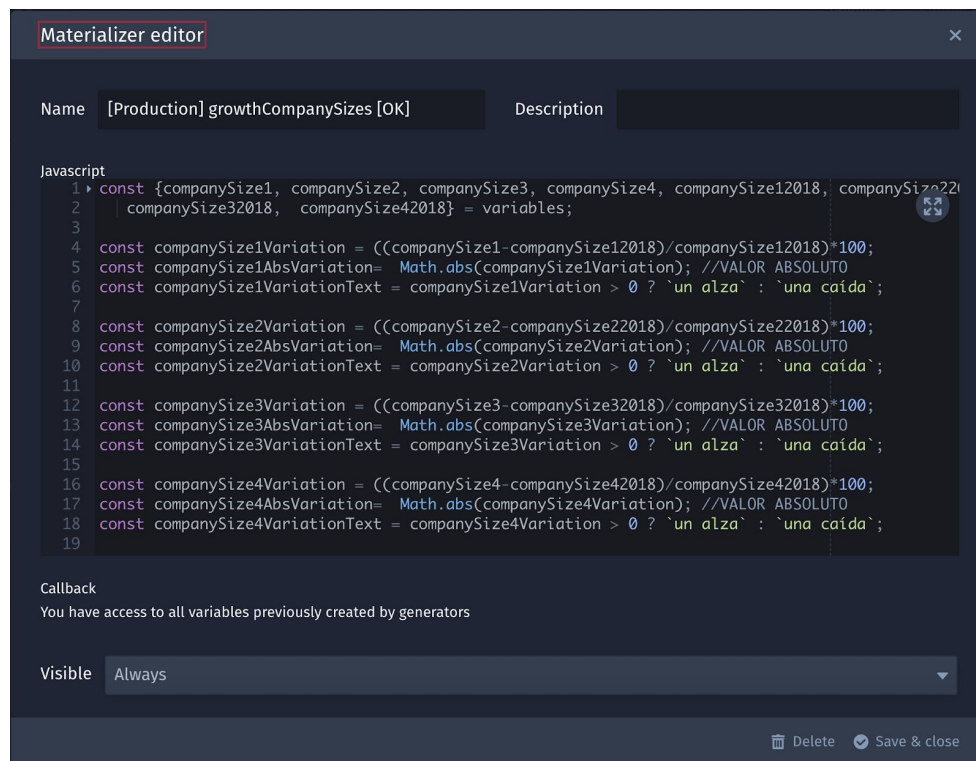
Los materializadores, a diferencia de los generadores, son elementos estáticos dentro del CMS, los cuales no constan/necesitan de una query o llamado de datos. Esto les permite cargar elementos que son válidos para cualquier nivel de jerarquía en un perfil.

Esta herramienta comparte 2 elementos en común con un generador:

1. *Name* (**buena práctica**): Nombre del materializador. Utilizado para identificar de forma preliminar los elementos disponibles dentro de cada materializador.
2. *Description* (**opcional**): Breve descripción del materializador para determinar con mayor profundidad el uso del materializador construido.

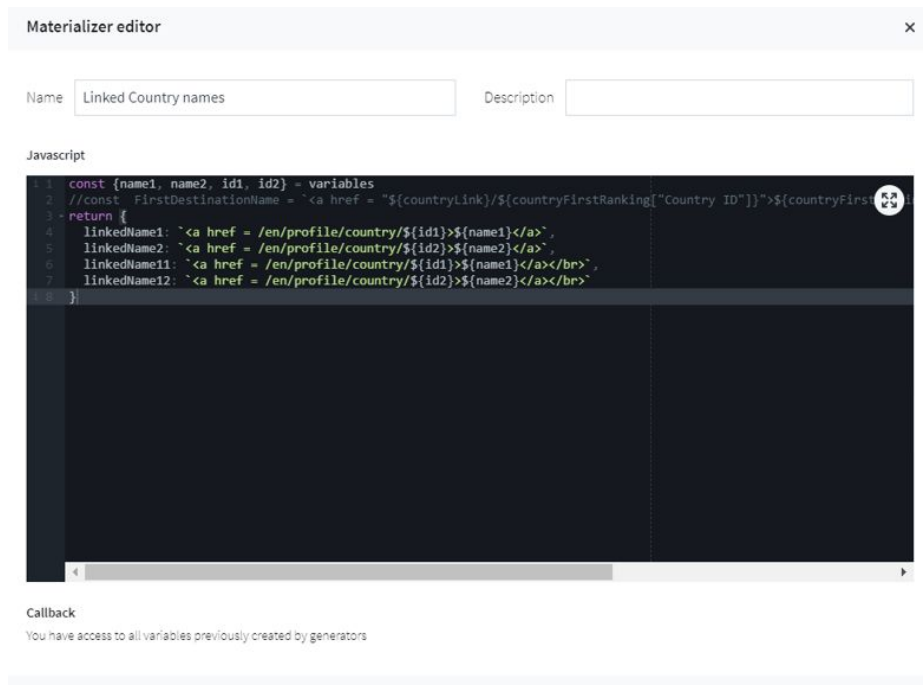
Además, por como el CMS carga los perfiles, es posible llamar a elementos creados en un generador en un materializado, pudiendo crear contenido “dinámico” respecto a lo disponible en la query realizada anteriormente.

Figura 5: Editor del materializador



Además, por la forma en la que el CMS carga los perfiles, es posible llamar a elementos creados en un generador a un materializador. Esto permite crear contenido “dinámico” respecto a los datos relacionados con la query realizada. En el ejemplo de la figura 6, se realiza una llamada a las variables {name1, name2, id1, id2} con el fin de crear nuevas variables relacionadas llamadas {linkedName}, las cuales quedan a disposición del usuario en el perfil.

Figura 6: Creación de nuevas variables



### Notas importantes (Materializadores)

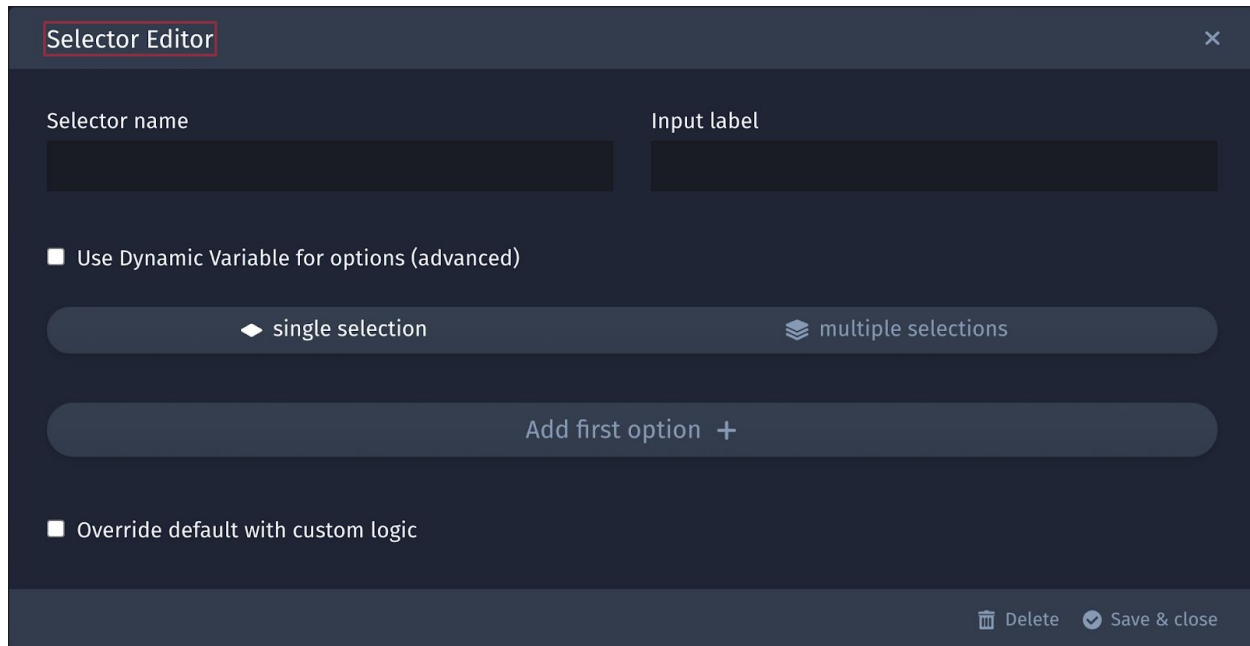
- Una **buena práctica** a la hora de crear un materializador con base en un generador, es que el nombre del materializador guarde una relación cercana al generador con los datos que están siendo utilizados, a modo de facilitar la validación del contenido al interior del CMS.
- **NO** puedes llamar a las variables creadas o a los resultados (return) de un materializador en otro materializador, porque crea problemas en la construcción de perfiles a la hora de cargar las páginas.

## Selectores

Una de las herramientas más útiles para crear contenido son los selectores. Dependiendo de sus opciones, pueden ser botones simples de 2 ó 3 opciones, o pueden ser selectores de múltiples opciones.

Los selectores permiten compactar múltiples visualizaciones en una sección que entrega más información, y permite al usuario jugar con las opciones disponibles, en lugar de descender por el perfil viendo construcciones similares de la misma data en distinto formato.

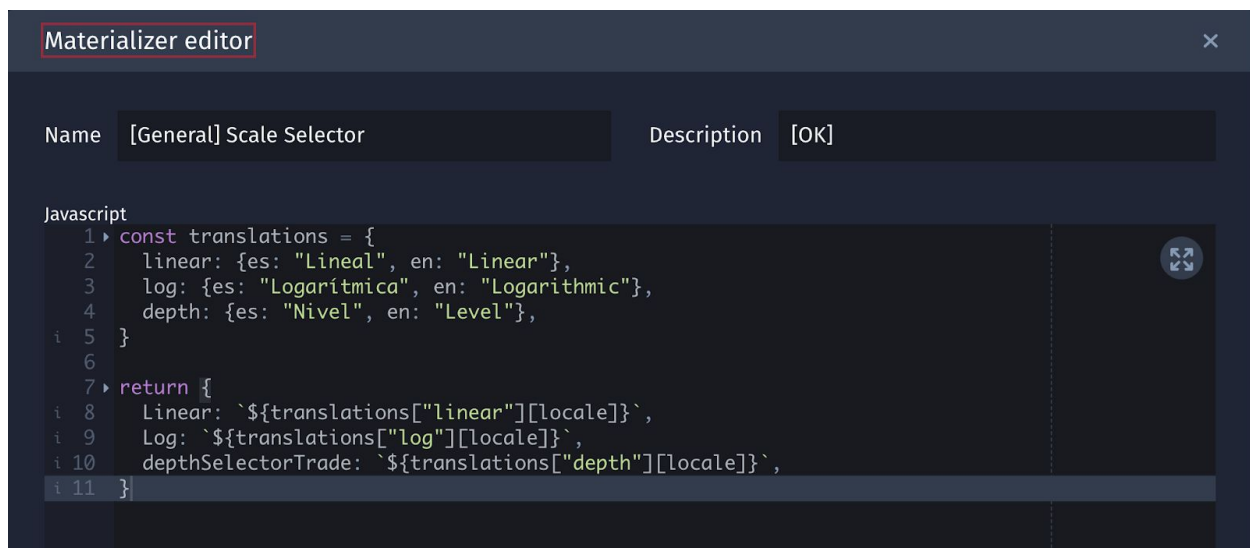
Figura 7: Editor del selector



The Selector Editor interface features a dark theme. At the top, the title 'Selector Editor' is highlighted with a red box. Below the title, there are two input fields: 'Selector name' and 'Input label'. A checkbox labeled 'Use Dynamic Variable for options (advanced)' is present. Below this, there are two buttons: 'single selection' (with a diamond icon) and 'multiple selections' (with a stack of squares icon). A large button labeled 'Add first option +' is centered below these. Another checkbox labeled 'Override default with custom logic' is at the bottom. At the very bottom right, there are 'Delete' and 'Save & close' buttons.

Estas opciones, en cualquiera de los 2 formatos deben ser creadas de antemano en un generador o materializador, como se muestra en la figura 8.

Figura 8: Creación de nuevas variables



The Materializer editor interface has a dark theme. The title 'Materializer editor' is highlighted with a red box. It has two input fields: 'Name' with the value '[General] Scale Selector' and 'Description' with the value '[OK]'. Below these is a 'Javascript' section with a code editor. The code is as follows:

```
1 ▶ const translations = {
2   linear: {es: "Lineal", en: "Linear"},
3   log: {es: "Logarítmica", en: "Logarithmic"},
4   depth: {es: "Nivel", en: "Level"},
5 }
6
7 ▶ return {
8   Linear: `${translations["linear"]}[locale]}`,
9   Log: `${translations["log"]}[locale]}`,
10  depthSelectorTrade: `${translations["depth"]}[locale]}`,
11 }
```

*Importante: una **buena práctica** es llamar al generador o materializador asociado al selector con un nombre que permita vincularse el uno con el otro para validación y en caso de realizar posteriores cambios.*

Dependiendo del soporte multilenguaje, o de lo que se trate de lograr en una visualización, las opciones de los selectores tienen dos partes:

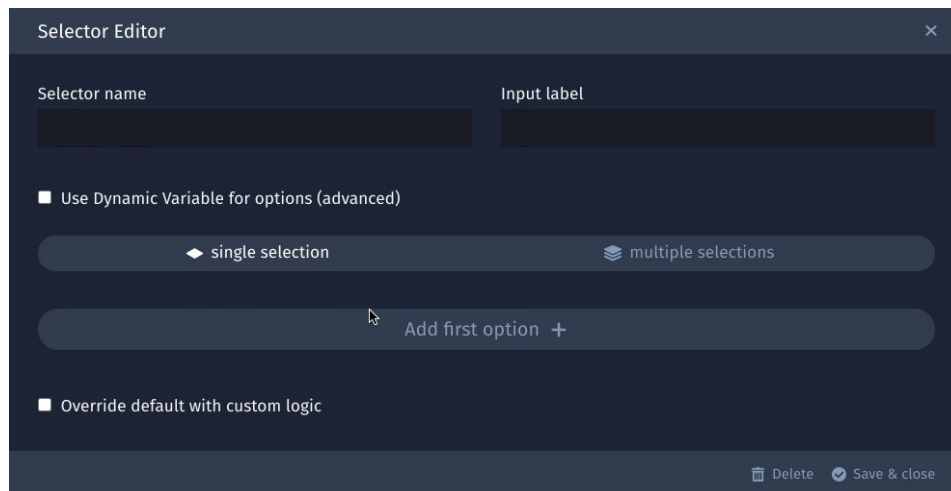
- *Label:* Nombre de la opción. Es el valor interno que tendrá la opción al interior del selector en el CMS que no verá el usuario, que se utiliza en la visualización para realizar los cambios que se desean en la misma.
- *Option:* Nombre de la opción. Es el valor externo que tendrá el selector para el usuario, asociado al atributo que está mostrando el selector. De esta forma, el usuario entiende por el nombre de la opción lo que obtendrá de la visualización, sin ver el trabajo interno y una variable con un nombre inentendible.

Dependiendo del soporte multilenguaje, puede ser necesario considerar más de un “*option*” a la hora de asignar valores para una alternativa en un selector.

Como se mencionó anteriormente, los selectores son cargados en los perfiles, después de los generadores y materializadores. Esto, debido a que dependen de los primeros para ser contruidos acordes al perfil en el que se encuentran. Los selectores pueden ser agrupados en 2 grupos, botones de 2 o 3 opciones, o selectores de más de 3 opciones.

#### ¿Cómo crear un selector simple? (2 ó 3 opciones)

Un selector de 2 ó 3 opciones, dada la estética del CMS, permite tomar un comportamiento de botón en un perfil, dado que son pocas opciones y el objetivo es que el usuario pueda interactuar libremente con los perfiles. Para crear un selector simple, primero se deben crear las variables que se usarán en el selector. Las variables pueden ser números enteros (por ejemplo, ids), fechas (por ejemplo, años), textos, (por ejemplo nombres y escalas). En la figura 9 se puede ver el ejemplo de selector.



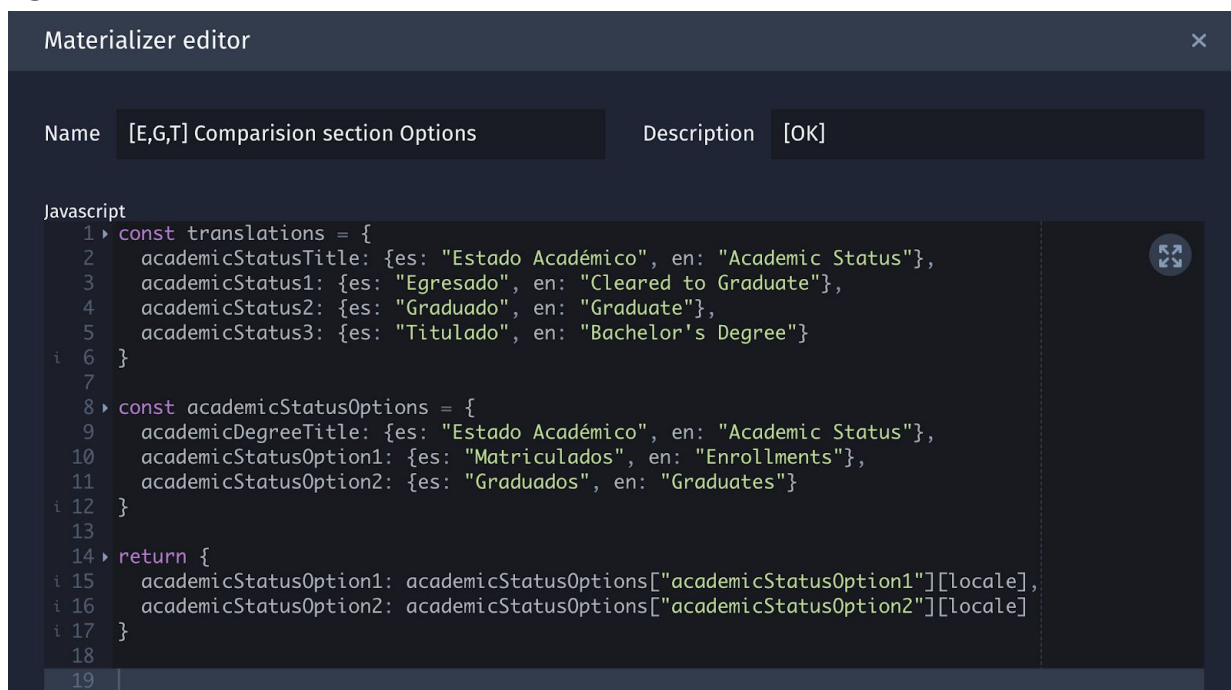
The Selector Editor interface includes the following elements:

- Selector name**: A text input field.
- Input label**: A text input field.
- Use Dynamic Variable for options (advanced)**: A checkbox.
- Selection type**: Two buttons, "single selection" (selected) and "multiple selections".
- Add first option**: A button with a plus icon.
- Override default with custom logic**: A checkbox.
- Footer**: "Delete" and "Save & close" buttons.

## ¿Cómo crear un selector dinámico?

A diferencia del selector de 2 ó 3 opciones, la estética del CMS muestra a los selectores con más de 3 opciones como un comportamiento dropdown de las opciones disponibles, las cuales se ocultan hasta que el usuario haga clic con el selector. Las variables que componen este tipo de selector son las mismas que las de 2 ó 3 opciones, pero dado su largo, pueden ser construidas de manera dinámica de ser necesario. En la figura 10 se puede visualizar la creación de un selector dinámico.

Figura 10: Creación selector dinámico



The Materializer editor interface shows the following details:

- Name**: [E,G,T] Comparision section Options
- Description**: [OK]
- Javascript**:
 

```

1 ▶ const translations = {
2   academicStatusTitle: {es: "Estado Académico", en: "Academic Status"},
3   academicStatus1: {es: "Egresado", en: "Cleared to Graduate"},
4   academicStatus2: {es: "Graduado", en: "Graduate"},
5   academicStatus3: {es: "Titulado", en: "Bachelor's Degree"}
6 }
7
8 ▶ const academicStatusOptions = {
9   academicDegreeTitle: {es: "Estado Académico", en: "Academic Status"},
10  academicStatusOption1: {es: "Matriculados", en: "Enrollments"},
11  academicStatusOption2: {es: "Graduados", en: "Graduates"}
12 }
13
14 ▶ return {
15   academicStatusOption1: academicStatusOptions["academicStatusOption1"][locale],
16   academicStatusOption2: academicStatusOptions["academicStatusOption2"][locale]
17 }
18
19
```

Para crear un selector dinámico (ver figura 11), se deben seguir los siguientes pasos:

1. Se debe crear una variable [ ] que encapsule las alternativas (options) que se usarán en el selector posterior.
2. Cada alternativa debe llevar 2 elementos definidos: option y label. Allowed, como se muestra más adelante, hace alusión a la visibilidad de las opciones del selector, y de no ser definido, se presume como siempre visible (always).
3. El return del selector o materializador debe retornar la variable que encapsula las alternativas con { }, para poder entregarlo al selector en su opción avanzada de desarrollo.

Figura 11: Selector dinámico

Default	Option	Visible	Actions
<input checked="" type="radio"/>	1: Entidad federativa	always	
<input type="radio"/>	2: Municipio	always	

### Notas importantes (Selectores)

- Selector Name (**buena práctica**): Nombre interno que tendrá el selector para que las visualizaciones llamen a la variable que está seleccionada en el selector.
- Input Label (**opcional**): Nombre “secundario” del selector. Es el nombre externo que tendrá el selector para el usuario (“fecha”, “año”, “escala”, etc.)
- Options: Opciones disponibles. Existen 2 formas de cargar las opciones: (a) por medio de la opción “Add options”; (b) por medio de entregar un Objeto al selector en la opción avanzada del CMS con el botón de Dynamic Variable.



- Visible: (opcional): Condicional de visibilidad para perfiles con multinivel o con jerarquías (i.e. Geografía: País, Región, Ciudad), donde ciertos elementos del selector no son adecuados, por lo que se puede condicionar su aparición de acuerdo a variables de jerarquía (**isNation: true** ó **isState:false**).

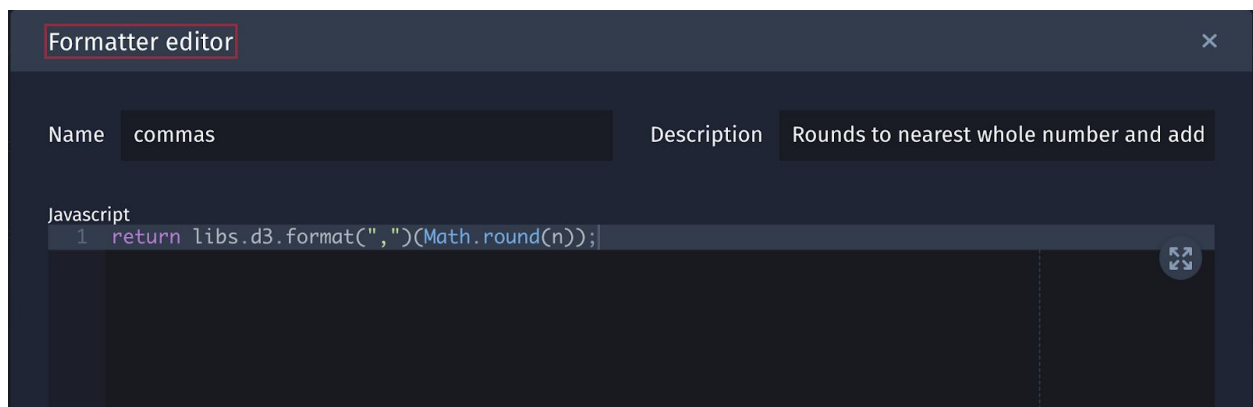
## Formateadores

A diferencia de los elementos anteriormente mencionados, los formateadores son fórmulas/funciones que pueden ser llamadas para que realicen cambios a una variable de una forma deseada.

Los elementos que componen un formateador se puede ver en la figura 12 y se describen a continuación:

1. *Name* (**buena práctica**): Nombre del formateador. Es el nombre por el cual se puede llamar el formateador al interior del CMS para que realice los cambios esperados.
2. *Description* (**buena práctica** - **opcional**): Descripción del formateador. Utilizado para comentar brevemente qué operación realiza en concreto el formateador, de manera que los usuarios no deban entrar al código para averiguarlo.

Figura 12: Editor del formateador



### Notas importantes (formateadores)

- Los formateadores, a diferencia de los otros elementos mencionados para la construcción de contenido del CMS, son cargados con anterioridad a los generadores, materializadores y selectores, por lo cual pueden ser llamados desde

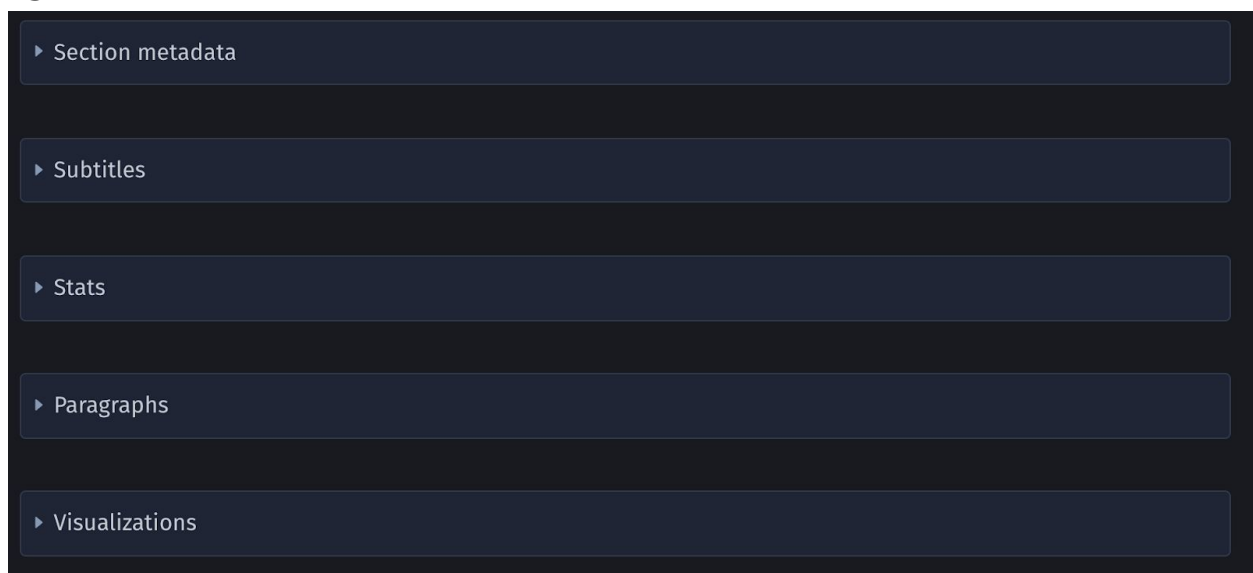
cualquier elemento que permita texto al interior del CMS. Esto es, desde un mismo generador, como un texto, título o incluso una estadística.

- Dado que los formateadores son funciones, retornan valores asociados para lo que fueron contruidos. Por ello, usarlos en variables inapropiadas puede tener como consecuencia generar N/A en textos donde se esperaban valores, o propiciar textos mal escritos por la sintaxis misma de lo escrito.
- Un formateador puede ser llamado directamente al elemento deseado por medio de: `variables.nombreFormatter(n)`, siendo *n* el elemento que se quiere transformar.

## Secciones

Las secciones permiten trabajar el contenido que es visible en los sitios. En cada sección se puede agregar título (Metadata) y subtítulo, estadísticas, párrafos y visualizaciones (ver figura 13). Además, se puede elegir entre diferentes configuraciones en el sitio para mostrar la información.

Figura 13: Estructura de Secciones



### Metadata

En Sección Metadata, hay 5 elementos que destacar:

Figura 14: Estructura de Sección Metadata

1. Title

2. Slug

3. Visible

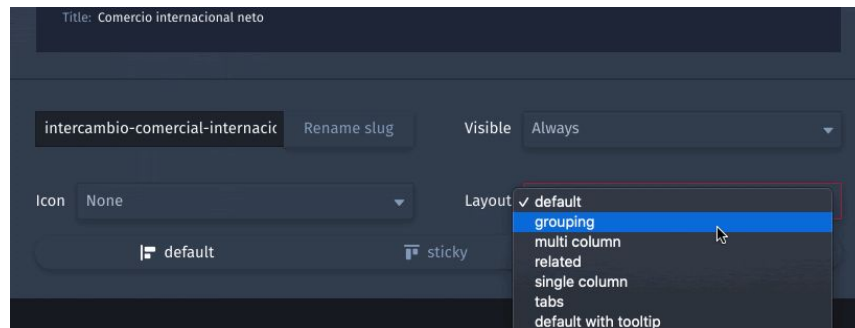
4. Icon

5. Layout

1. **Title:** Aquí se define el nombre de la sección. Una **buena práctica** es usar títulos cortos y explicativos de lo que mostrará la visualización, y escribir la primera letra de la primera palabra en mayúscula y todo el resto en minúscula, excepto cuando se trate de siglas. Por ejemplo, “Fuerza laboral y salarios según municipio”.
2. **Slug:** Es una identificación de referencia para la sección que es visible en la URL del sitio. Una **buena práctica** es no usar conectores, y separar las palabras con un guión medio. Además, cada sección debe tener un slug único.

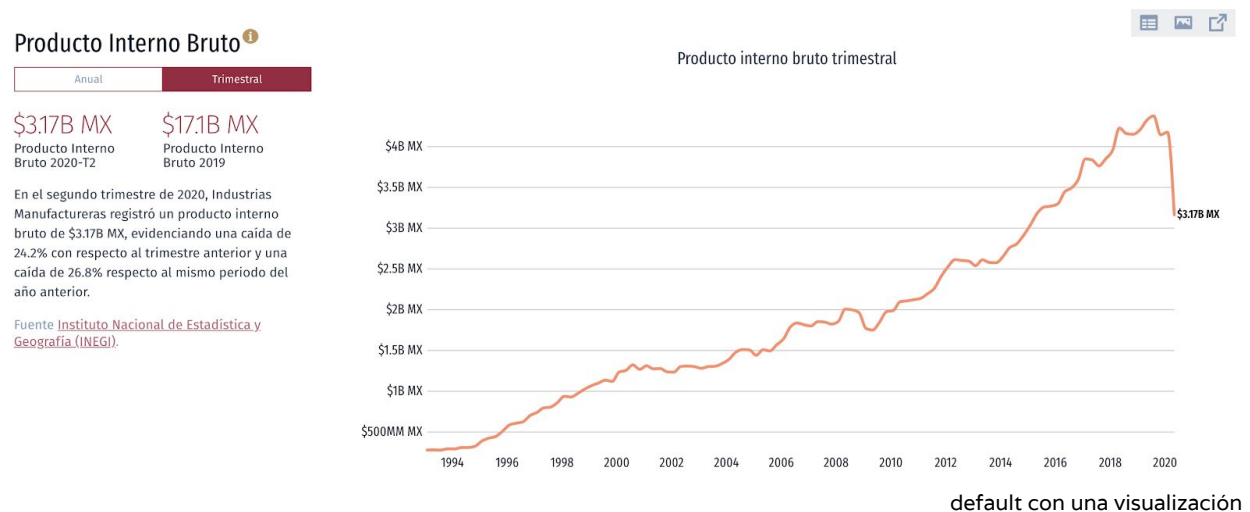
<https://dev.datamexico.org/es/profile/product/partes-y-accesorios-de-vehiculos-automotores#mercado-global>

3. **Visible:** Permite configurar una variable para que sea visible sólo cuando se cumple una condición. La visibilidad de facto es *Always* (la sección siempre es visible). Cuando se espera que una sección sea visible solamente si se cumple una condición determinada, es conveniente crear una variable en un materializador que se denomine *false* cuando se cumple la condición para la cual la sección no debe ser visible. Una **buena práctica** es crear una variable `hidden: false` en un materializador para ocultar una sección rápidamente en caso de detectar problemas y la sección no deba ser visible para el público.
4. **Icon:** Da la opción de agregar un ícono en el título. Generalmente no se utiliza.
5. **Layout:** Esta opción permite cambiar el diseño y disposición de la sección. Existen diferentes opciones de layout, las más utilizadas son:



- a. *Default*. Es el layout más utilizado (ver figura 15). La visualización ocupa la mayor parte de la pantalla, y el espacio lateral izquierdo es reservado para el título, selectores, estadísticas y párrafos.

Figura 15: Layout *default*

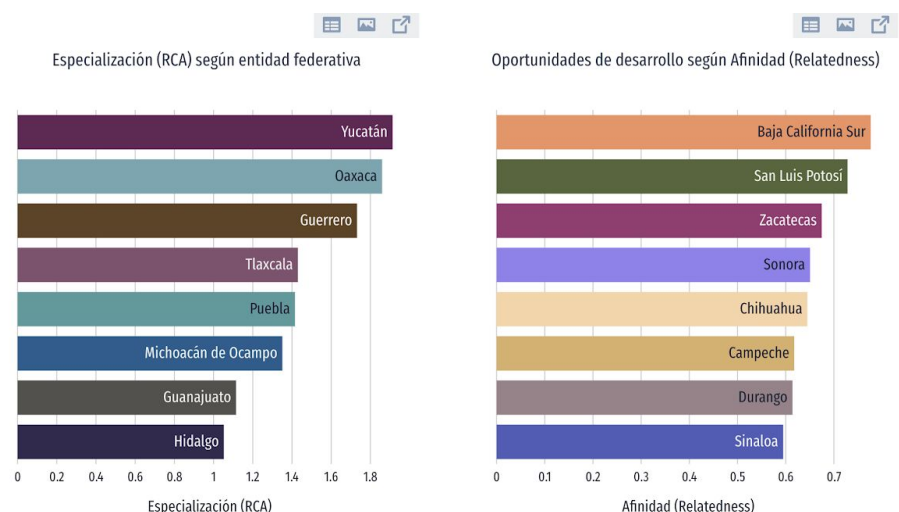


## Especialización y Oportunidad

En 2014, las entidades federativas con mayor grado de especialización en Industrias Manufactureras fueron [Yucatán](#) (1.91), [Oaxaca](#) (1.86) y [Guerrero](#) (1.73). El grado de especialización es medido usando el índice RCA, el cual representa la razón entre los establecimientos observados y esperados de cada entidad federativa para cada industria.

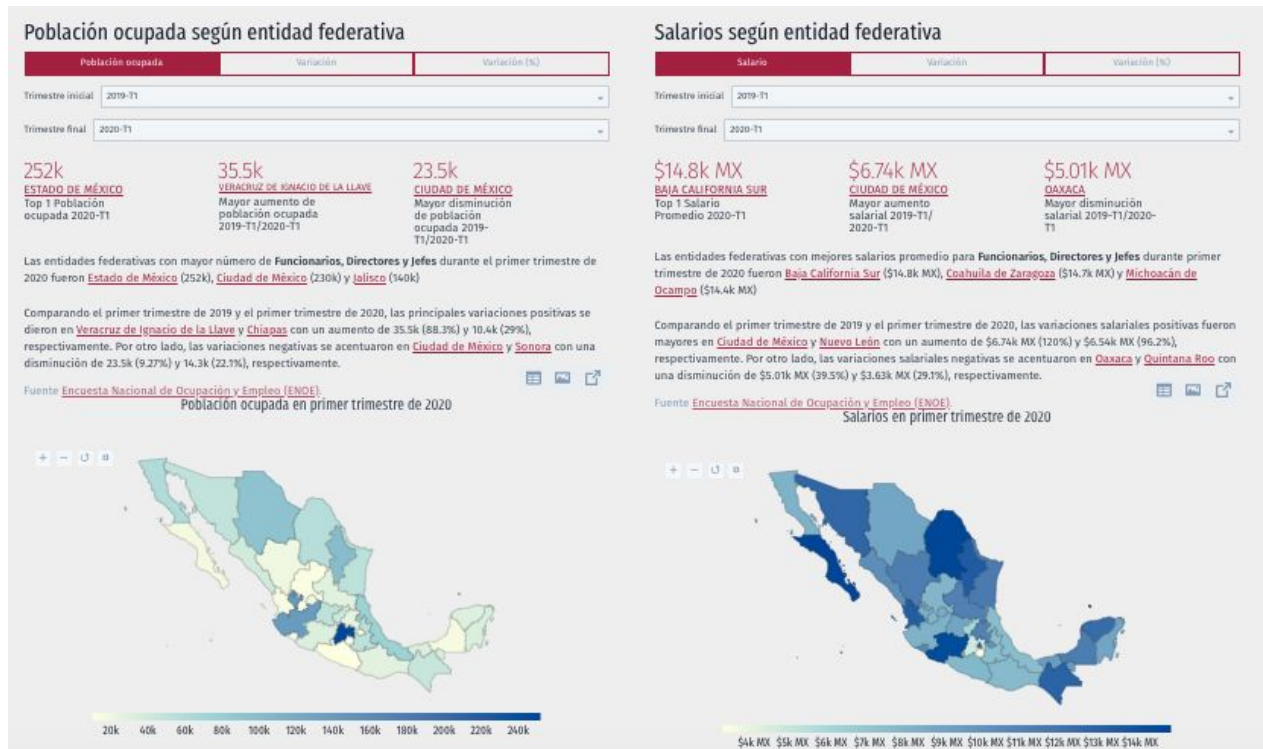
En 2014, las entidades federativas con mayor grado de oportunidad de desarrollo en Industrias Manufactureras según el índice de Afinidad (Relatedness) fueron [Baja California Sur](#) (0.78), [San Luis Potosí](#) (0.73) y [Zacatecas](#) (0.67). El índice de Afinidad mide la distancia entre una industria y la composición industrial actual en cada entidad federativa. Las entidades federativas presentadas registran un nivel de especialización menor a la unidad (RCA < 1).

Fuente [DataMéxico](#).



- b. *Single column*: Con este layout sólo hay una columna de contenido. La columna comienza con el título en la parte superior, seguido de las estadísticas, texto y visualización. Si hay más de una visualización se verá una sección al lado de la otra (ver figura 16).

Figura 16: Layout *single column*



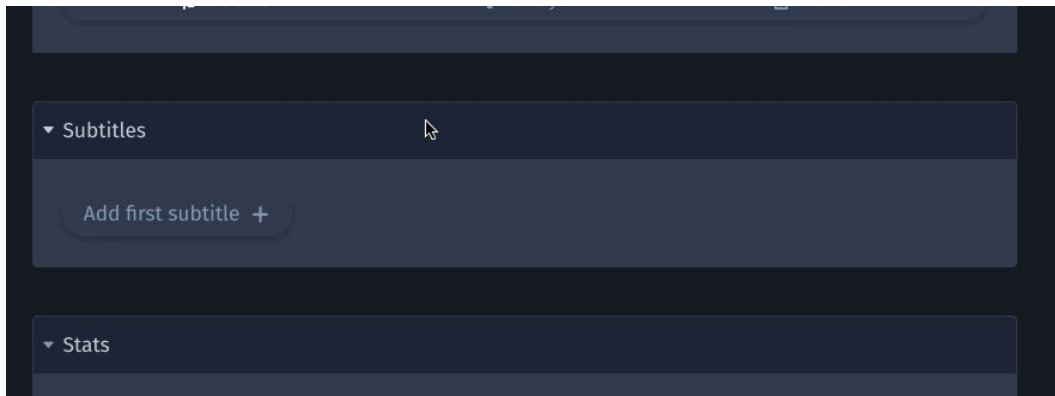
- c. *Layout multi column*: Este diseño permite equilibrar el contenido en columnas. Generalmente se utiliza en la sección de About o Acerca de, tomando los párrafos y dividiéndolos en columnas (ver figura 17).

Figura 17: Layout *multi column*



## Subtitle

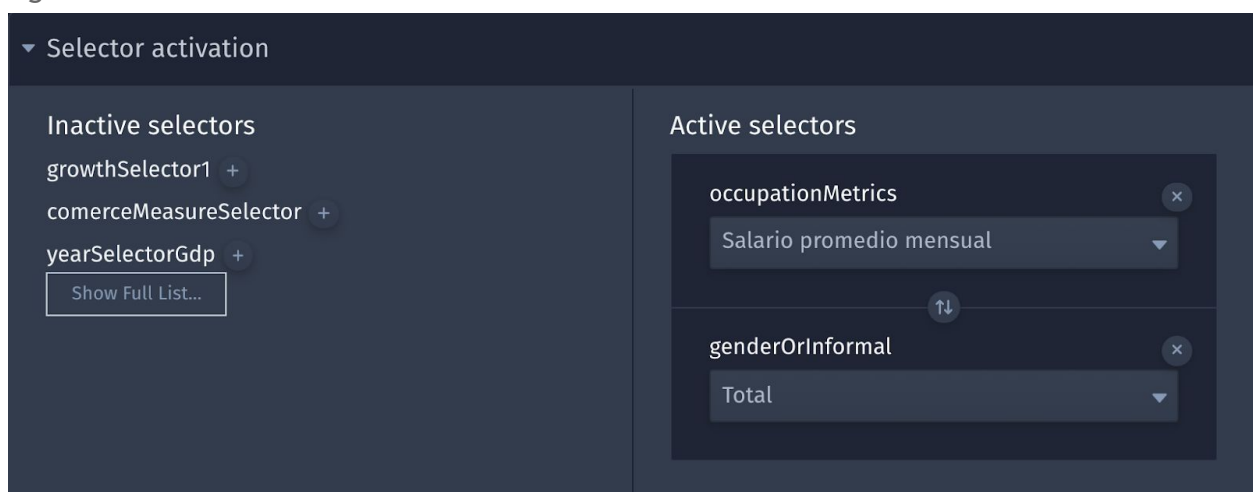
Subtítulos permite agregar un subtítulo a la sección. Para ello, solo es necesario hacer *click* en *Add first subtitle* + y escribir el subtítulo. Además, tiene la opción *Visible* que puede ser configurada de acuerdo con determinadas condiciones de visibilidad. Generalmente no se utilizan subtítulos en las secciones, salvo determinados casos en que se justifica agregarlo.



## Selector activation

Esta opción permite agregar selectores a la visualización que han sido creados con anterioridad en un selector o materializador (ver la sección sobre crear un selector). En el lado izquierdo (Inactive selectors) aparecerá el listado de selectores disponibles, basta con hacer clic en + para que el selector sea agregado. En el lado derecho (Active selectors) se mostrará el listado de selectores activos (ver figura 18). Generalmente una visualización tiene de 1 a 3 selectores, dependiendo de la complejidad de esta misma.

Figura 18: Selector activation



Una vez activado el(los) selector(es), es necesario llamar a la variable asociada al selector, como se muestra en la figura 19.

Figura 19: Variable asociada al selector

```
const {hierarchy, id, hierarchy_name, name} = variables;
const {mean} = libs.d3;
const {lowercase, quarterShort} = formatters;
const selector = variables["[[employSelector1]]"];
const selector2 = variables["[[typeJob1]]"];
const yMeasure = selector === "Población ocupada" ? "Workforce" : "Monthly Wage";
const dd = "Classification+of+Formal+and+Informal+Jobs+of+the+First+Activity";
```

En este caso, el selector interactúa con la visualización de manera que elige entre las medidas de “Población ocupada” y “Salario”. Al comparar estas alternativas en “yMeasure”, se cambia la variable que se está graficando en el eje Y.

**Importante:** Una buena práctica es usar un `console.log()` de la variable que guarda el contenido del selector, esto para verificar que se le está pasando desde la llamada del mismo.

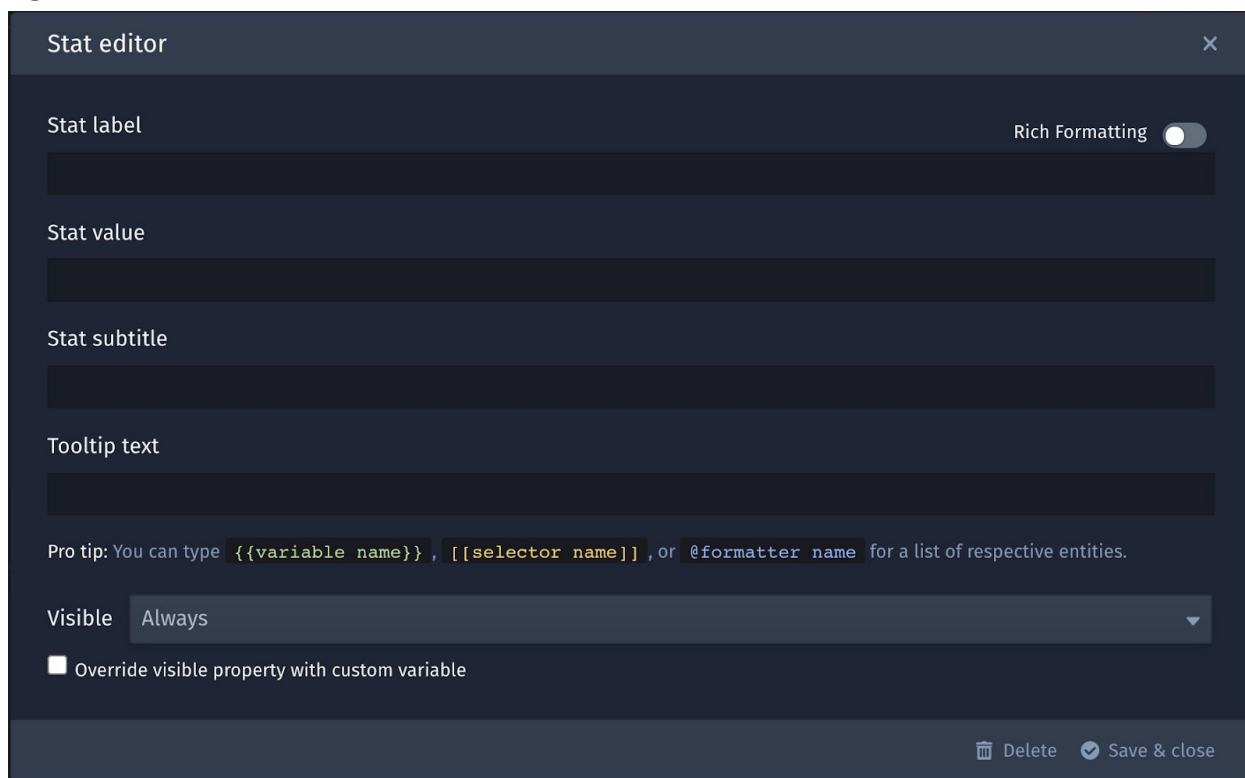
## Stats

Esta opción permite agregar estadísticas a la sección. Se trata básicamente de un número que se quiere enfatizar con una etiqueta y un subtítulo. Basta con presionar add stat + para abrir la pestaña que permite configurar la sección de estadísticas.





Figura 20: Elementos de la sección Stats



The screenshot shows the 'Stat editor' window with the following fields and controls:

- Stat label:** A text input field with a 'Rich Formatting' toggle switch to its right.
- Stat value:** A text input field.
- Stat subtitle:** A text input field.
- Tooltip text:** A text input field.
- Pro tip:** A text block stating: 'You can type `{{variable name}}`, `[[selector name]]`, or `@formatter name` for a list of respective entities.'
- Visible:** A dropdown menu currently set to 'Always'.
- Override visible property with custom variable:** A checkbox that is currently unchecked.
- Footer:** 'Delete' and 'Save & close' buttons.

Los 4 elementos principales de la sección de estadísticas son:

1. *Stat label*: Es una descripción del número que compone a la estadística. Es importante considerar que varias estadísticas con la misma descripción se agruparán en columnas y su etiqueta solo se mostrará una vez.
2. *Stat value*: Es el valor sobre el cual se quiere hacer énfasis. Este valor debe ser calculado con anterioridad en un generador o materializador y se accede a él mediante doble llave `{{valor}}`. Adicionalmente se le puede aplicar formatos, por ejemplo agregar el símbolo de porcentaje (30%), abreviar el número (16,5k), agregar separador de miles (1,990,980), incluir el signo de moneda (\$1.5M), entre otros. El formato debe ser definido con anterioridad (ver sección crear un formateador). Es una **buena práctica** que el valor que se asigne a *stat value* sea un número y no una palabra para resguardar el diseño del sitio.
3. *Stat subtitle*: Es un espacio para añadir cualquier aclaración a la estadística. Generalmente se utiliza para indicar el período, fecha, o denominación correspondiente a la estadística.
4. *Visible*: La opción de visibilidad permite ocultar o hacer visible la estadística. Se recomienda que al configurar en la visibilidad la misma variable que se llamó en



*Stat value*, la estadística será visible solo si tiene un valor y se evitará mostrar N/A en caso de que la variable no tenga un valor por falta de datos.

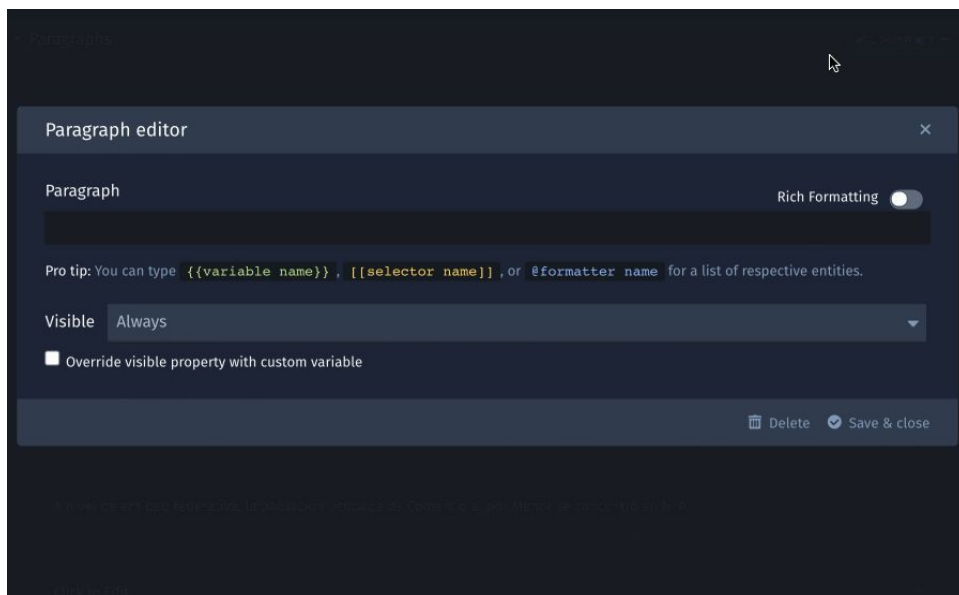
## Paragraphs

Los párrafos son explicaciones de la sección en los que enfatizan algunos valores rápidamente visibles en el gráfico. Es importante que la redacción sea precisa, evitar el uso innecesario de conectores, y no crear estadísticas difíciles de obtener con una simple inspección del gráfico. Para agregar un párrafo tienes que presionar add paragraph +.

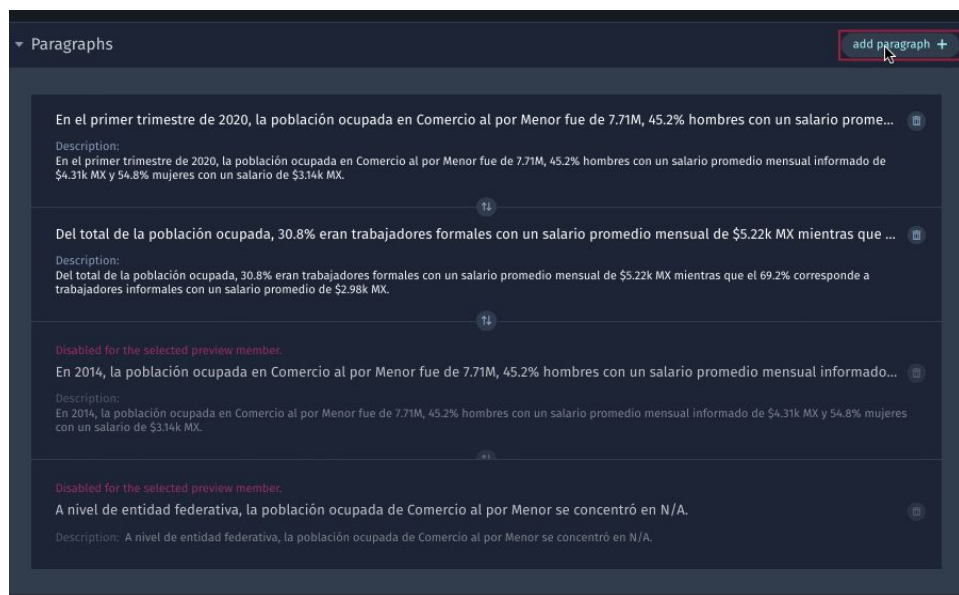


**Importante:** es necesario revisar que los valores que se presentan en el texto correspondan con la visualización. Además, cuando la visualización tenga varios selectores, el texto debe coincidir con la opción marcada por defecto en los selectores. Es una **buena práctica** tener 2 párrafos por cada sección para mantener las proporciones con la(s) visualización(es).

Para incluir un párrafo se escribe el texto en el espacio disponible bajo *Paragraph*. Para redactar un párrafo se pueden llamar variables creadas con anterioridad en un generador o materializador haciendo uso de doble llave `{{valor}}` y aplicar formatters de la misma manera que se explicó para las *Stats*.



La sección de párrafo cuenta con la opción Rich Formatting en la esquina superior derecha. Al activarlo se puede dar formato al texto: aplicar negritas, cursivas, agregar links, listas, entre otros.

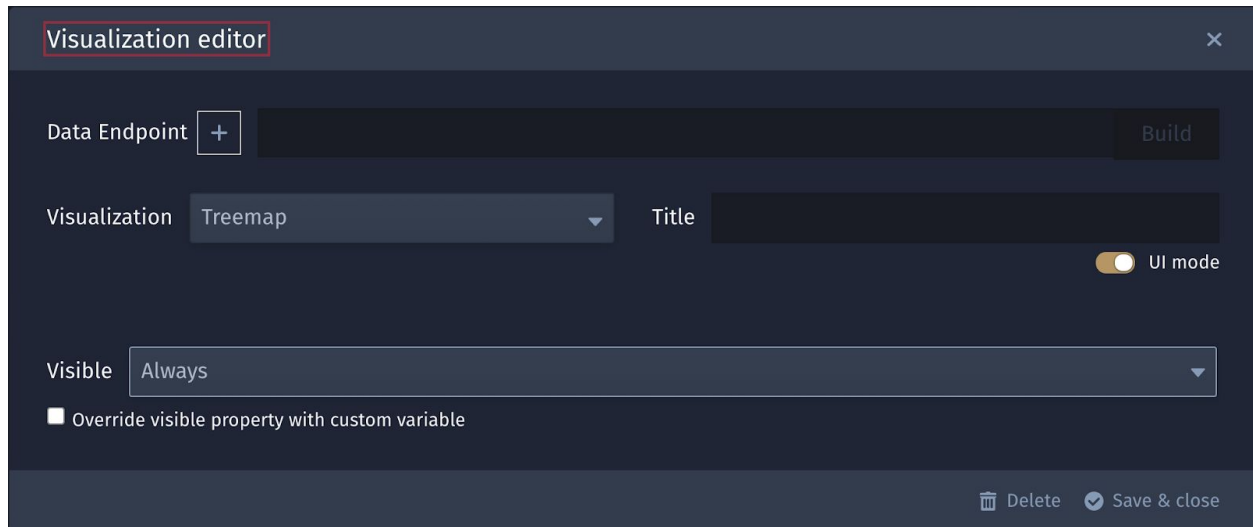


En caso de querer incorporar código html al párrafo, el texto junto al código html debe ser escrito en un materializador y posteriormente llamarlo como variable en Paragraphs.

## Visualization

En esta sección se escribe el código para crear visualizaciones (gráficas, tablas, imágenes) a partir de un conjunto de datos. Para comenzar a crear una visualización basta con presionar *Add visualization* + para abrir la pestaña que permite hacer las configuraciones para crear la visualización.

Figura 21: Elementos de la sección Stats

The image shows a 'Visualization editor' window with a dark theme. At the top, there's a title bar with 'Visualization editor' and a close button. Below it, the 'Data Endpoint' field is empty, with a '+' icon to its left and a 'Build' button to its right. The 'Visualization' dropdown menu is set to 'Treemap'. The 'Title' field is empty. There's a 'UI mode' toggle switch that is currently turned on. The 'Visible' dropdown menu is set to 'Always'. Below this, there's a checkbox labeled 'Override visible property with custom variable' which is unchecked. At the bottom right, there are 'Delete' and 'Save & close' buttons.

CMS permite crear algunas gráficas sencillas en pocos pasos en la pestaña de configuración. Lo primero es definir el conjunto de datos en Data Endpoint. Al estar trabajando con datos de Tesseract UI, un ejemplo de data endpoint es:

```
api.datamexico.org/tesseract/data?<hierarchy>=<id>Quarter=20201&cube=inegi_enoe  
&drilldowns=Sex,Age+Range&measures=Workforce&parents=false&sparse=false
```

Luego, en *Visualization* se escoge de las opciones disponibles la gráfica que se quiere implementar y se agrega el título en Title. Finalmente se configura la opción de visibilidad como se mencionó en secciones anteriores.

Las opciones incluidas son limitadas y no suficientemente flexibles para mostrar la información de la mejor manera. Para crear visualizaciones dinámicas, con diversos conjuntos de datos, interactivas con los selectores, y configuraciones personalizadas, es necesario habilitar la opción UI mode y escribir código Javascript. Las visualizaciones del CMS utilizan [d3plus](#).

## Caso práctico

### La Inversión Extranjera Directa (IED) en México por industria

La Inversión Extranjera Directa (IED) se divide en tres categorías: Nuevas Inversiones (inversiones iniciales), Cuentas entre Compañías (préstamos o adelantos de pagos de las matrices a sus filiales en México), y Reinversión de Utilidades (utilidades que no se distribuyen como dividendos). La IED se registra a nivel de Nación (México) y Estados (32 estados). Los datos se clasifican de acuerdo a sectores (18 sectores), subsectores (89 subsectores), e industrias (283 industrias). Finalmente, el registro incluye el país de origen de la IED (149 países). Los datos se actualizan cada trimestre.

#### Proceso de creación de contenido

1. Establecer la dimensión de los datos
  - a. La dimensión es la perspectiva que vamos a adoptar al realizar el análisis. Para analizar La Inversión Extranjera Directa (IED) en México por industria, la dimensión que estamos seleccionando es ver la IED desde la perspectiva de la **Industria**.
2. Identificar la jerarquía de los datos.
  - a. Los datos de IED están clasificados por inversión genérica (nuevas inversiones, cuentas entre compañías, y reinversión de utilidades), geografía (país, estado), industria (sectores, subsectores, grupo industrial), fecha (año, trimestre), y origen (país, continente).
  - b. La jerarquía se expresa como un campo de datos que contiene un solo hecho o atributo de una dimensión, y es un punto de partida para ver qué información existe en los datos y cuál es su estructura. Una forma de entender los datos es como componentes más pequeños o más grandes en una jerarquía.
  - c. Para la dimensión **Industria**, la jerarquía de los datos es: **Sectores**, **Subsectores** y **Grupo Industrial**.

Figura 2.1 - Estructura de datos IED

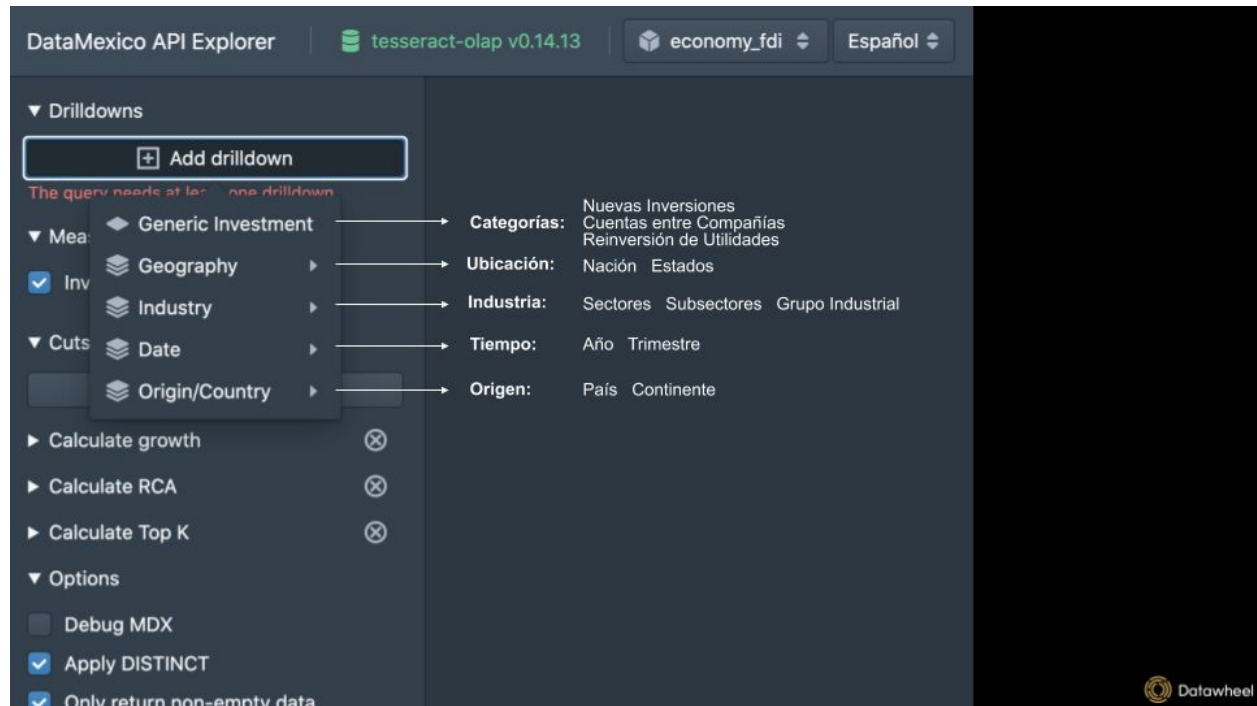


Figura 2.2 - Dimensiones de datos sobre IED

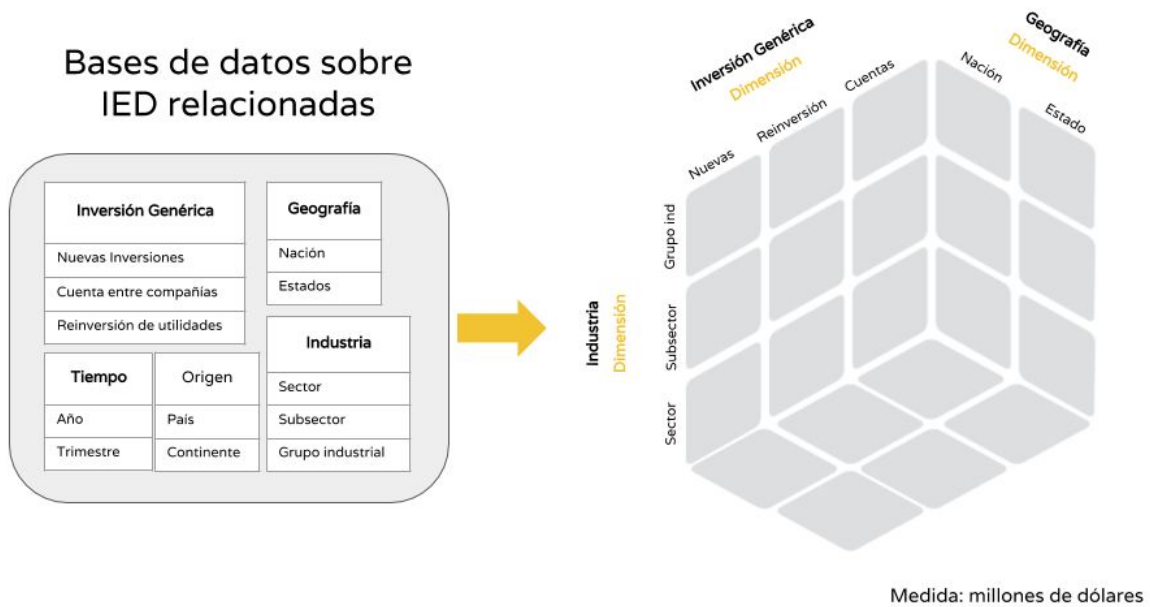
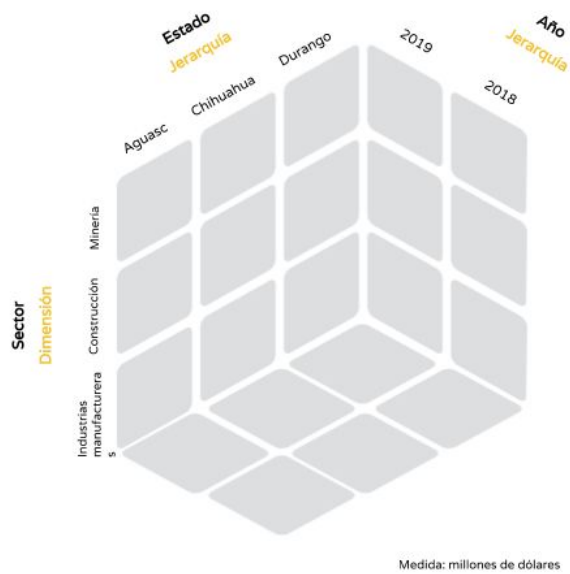


Figura 2.3 - Jerarquía de datos en la dimensión sector - **Industria**



### 3. Crear una dimensión de datos

**Dimensión:** Industria (4 dígitos)

**Medida:** Inversión (millones de dólares)

### 4. Crear contenido: generadores

#### a. API:

[https://api.datamexico.org/tesseract/data.jsonrecords?Sector=22&cube=economy\\_fdi&drilldowns=Generic+Investment&measures=Investment+%28Million%29](https://api.datamexico.org/tesseract/data.jsonrecords?Sector=22&cube=economy_fdi&drilldowns=Generic+Investment&measures=Investment+%28Million%29)



Visible: siempre

Stat editor

Stat label

Rich Formatting

IED|

Stat value

Stat subtitle

Tooltip text

Pro tip: You can type `{{variable name}}`, `[[selector name]]`, or `@formatter name` for a list of respective entities.

Visible

Always

☐

Override visible property with custom variable

Delete

Save & close

## 6. Crear contenido: párrafo

En **year19**, México captó **dollaried19** mdd de IED en **name**.

Paragraph editor

Paragraph

Rich Formatting

Pro tip: You can type `{{variable name}}`, `[[selector name]]`, or `@formatter name` for a list of respective entities.

Visible

Always

☐

Override visible property with custom variable

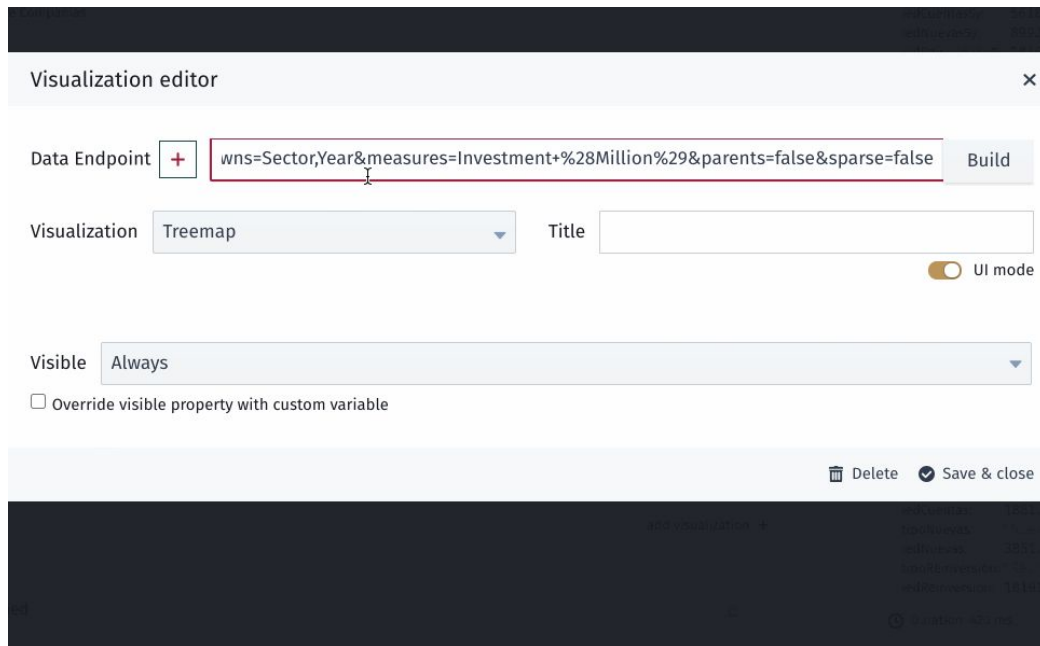
Delete

Save & close



## 7. Crear una visualización

### a. Opción automática :



The screenshot shows the 'Visualization editor' window. At the top, there's a title bar with a close button. Below it, the 'Data Endpoint' field is highlighted with a red border and contains the URL 'wns=Sector,Year&measures=Investment+%28Million%29&parents=false&sparse=false'. To the right of this field is a 'Build' button. Below the 'Data Endpoint' field, there's a 'Visualization' dropdown menu set to 'Treemap' and a 'Title' text input field. To the right of the 'Title' field is a toggle switch for 'UI mode'. Below these fields, there's a 'Visible' dropdown menu set to 'Always' and a checkbox labeled 'Override visible property with custom variable'. At the bottom right of the editor, there are 'Delete' and 'Save & close' buttons. The background of the editor is dark, and there's a faint 'Datawheel' logo in the top right corner.

Visualization editor

Data Endpoint  Build

Visualization Treemap Title

Visible Always

☐ Override visible property with custom variable

Delete Save & close

### b. Opción UI mode:

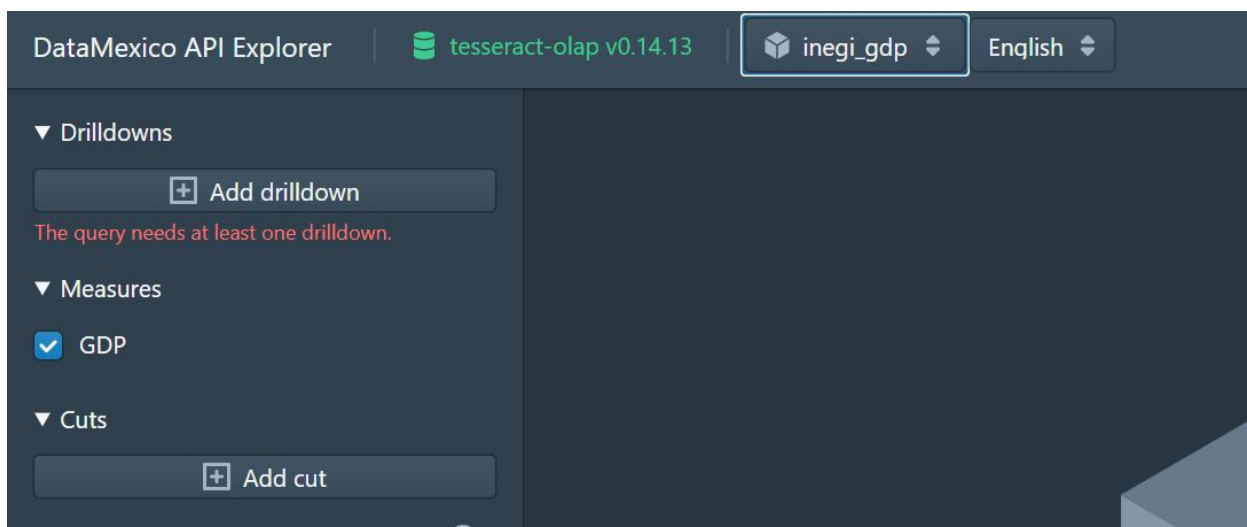
## Caso práctico : Crear una sección en la página web

### Tesseract UI



“Primer paso: Elección de datos para crear una sección”




A continuación se describen las instrucciones para la elección de datos en Tesseract UI (base de datos) para crear una sección nueva en la página web. En este ejemplo se utilizará la información del producto interno bruto de DataMéxico para crear una nueva sección que visualice el PIB anual y trimestral.

- (1) El primer paso es ver donde se encuentra la información del producto interno bruto, es por esto que es importante leer el glosario de cubos del proyecto ([Glosario Cubos DM](#)). Luego de saber en qué cubo se encuentra la información, se analizan los datos en la aplicación Tesseract UI. En la figura X se muestra el cubo correspondiente al producto interno bruto.



- (2) Este cubo contiene dos drilldowns (variables), las cuales son “date” que indica el tiempo en que se requiere ver los datos ( Anual y trimestral) y “sector” que es un nivel industrial perteneciente a los perfiles de industrias (Sector-Subsector-Rama industrial). Además, contiene una medida que es el producto interno bruto.

DataMexico API Explorer |  tesseract-olap v0.14.13 |  inegi\_gdp | English

▼ Drilldowns	Quarter ID	Quarter	GDP
<input type="button" value="+ Add drilldown"/>	1	1993-1	1433532
<small>The query needs at least one drilldown.</small>	2	1993-2	1485623
▼ Measures	3	1993-3	1488501
<input checked="" type="checkbox"/> GDP	4	1993-4	1558416
▼ Cuts	5	1994-1	1587348
<input type="button" value="+ Add cut"/>	6	1994-2	1696746
► Calculate growth 	7	1994-3	1696403
► Calculate RCA 	8	1994-4	1804692
► Calculate Top K 	9	1995-1	1931907
▼ Options	10	1995-2	2157747
<input type="checkbox"/> Debug MDX	11	1995-3	2262436
<input checked="" type="checkbox"/> Apply DISTINCT	12	1995-4	2510014
<input checked="" type="checkbox"/> Only return non-empty data	13	1996-1	2736930
	14	1996-2	2913310
	15	1996-3	3012024
	16	1996-4	3330123
	17	1997-1	3439085
	18	1997-2	3741351
	19	1997-3	3821508
	20	1997-4	4136215

- (3) En este caso para visualizar cómo varía el PIB anual por sector industrial, la variable (drilldown) elegida fue date (year) y el corte se hizo en la variable sector (para que solo se vea la información de un sector), en la figura X se muestra la información del producto interno bruto del sector “Agricultura” entre los años 1993-2004.

DataMexico API Explorer

tesseract-olap v0.14.13

inegi\_gdp

English

▼ Drilldowns

◆ Date/Year

+ Add drilldown

▼ Measures

☒ GDP

▼ Cuts

☰ Sector (Agriculture, Animal Pro...

+ Add cut

	Year	GDP
1	1993	318289
2	1994	328981
3	1995	409115
4	1996	608200
5	1997	705373
6	1998	800153
7	1999	844944
8	2000	889744
9	2001	962357
10	2002	977642
11	2003	1066219
12	2004	1179181

## Content Management System

### “Segundo paso: Crear contenido en la página web”

Una vez que están seleccionados los datos, se entra al CMS para crear una sección nueva, elegimos el perfil correspondiente, en este caso como es un ejemplo ocupamos un perfil “Unnamed profile”, si fuera un caso real, ocupamos el perfil industria, ya que, los datos están por sector industrial. Luego, procedemos a crear una sección en el perfil, como se mencionó anteriormente en cada sección se le puede agregar Título y distintas características, en este caso a la sección se le pondrá el nombre de “Producción” y luego a la visualización “Producto interno bruto”.

▼ Unnamed profile ⚙️

New Section Producción Producto interno bruto

dev.datamexico.org/profile/ Preview Duplicate Delete

▼ Dimensions ☒ Compact

Add first dimension +

▼ Section metadata

Title

Producto interno bruto  
Title: Producto interno bruto

Rename slug Visible Always

Icon None Layout default

default sticky modal

Una vez creadas las nuevas secciones Producción y Producto interno bruto, como el “Producto interno bruto” será una visualización de la sección “Producción”, en las opciones de layout, se elige la opción “Grouping” en Producción, así se incorpora “Producto interno bruto” a la sección “Producción” como se muestra en la figura X.

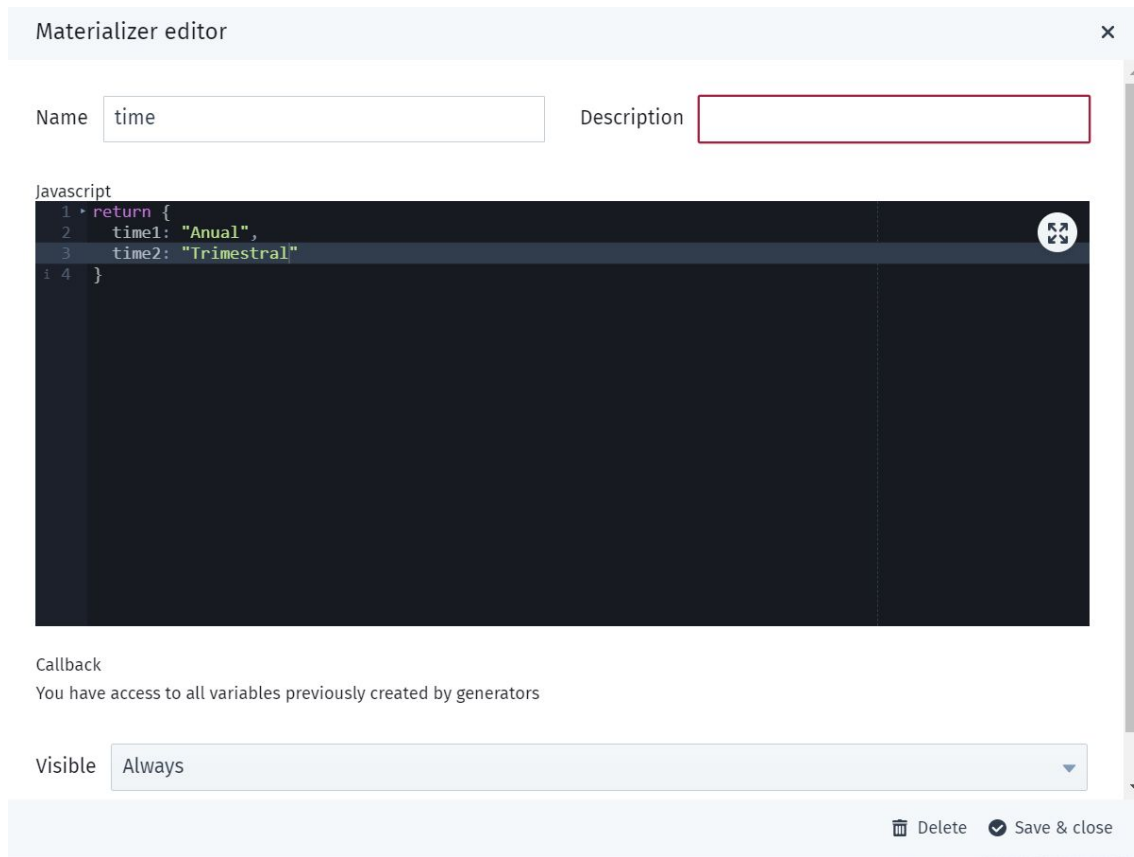
The screenshot shows the Datawheel interface for configuring a new section. At the top, there's a header 'Unnamed profile' with a settings gear icon. Below it, a 'New Section' button is visible, and a dropdown menu is open showing 'Producción' selected. Underneath, a list of sections includes 'Producto interno bruto'. The URL 'dev.datamexico.org/profile/' is shown, along with 'Preview', 'Duplicate', and 'Delete' buttons. The 'Dimensions' section has a 'Compact' toggle and an 'Add first dimension +' button. The 'Section metadata' section contains a 'Title' field with 'Producción' and 'Title: Producción' below it. There's a 'Rename slug' button next to an empty slug field. The 'Visible' dropdown is set to 'Always'. The 'Icon' dropdown is set to 'None'. The 'Layout' dropdown is set to 'grouping' and is highlighted with a yellow box.

Después de haber creado la sección, hay diversas opciones de cómo seguir agregando contenido, en este ejemplo se procederá a crear contenido en orden por sección (Selector, estadísticas, párrafos y visualización).

### Sección Selector

En primer lugar, se crea un selector, en este caso crearemos un selector estático, es por esto que ocuparemos un materializador. Se presiona agregar un materializador y luego se le da contenido a la consola, las opciones serán “Anual” y “Trimestral” ya que, las

opciones en Tesseract UI para ver el producto interno bruto en el tiempo son anuales o trimestrales. En la figura X se muestra el código en JavaScript que se utiliza para crear las opciones, una vez generado se guarda y cierra la consola.



The screenshot shows the 'Materializer editor' window. At the top, there are two input fields: 'Name' with the value 'time' and 'Description' which is empty. Below these is a 'Javascript' code editor with the following code:

```
1 • return {  
2   time1: "Anual",  
3   time2: "Trimestral"  
4 }
```

Below the code editor, there is a 'Callback' section with the text 'You have access to all variables previously created by generators'. At the bottom, there is a 'Visible' dropdown menu set to 'Always'. The bottom right corner of the window contains 'Delete' and 'Save & close' buttons.

Después de estar creadas las opciones se procede a crear el selector, con las opciones correspondientes. No olvidar agregar una opción por default.

Selector Editor

×

Selector name

Selector GDP

Input label

☐ Use Dynamic Variable for options (advanced)

◆ single selection

≡ multiple selections

Default	Option	Visible	Actions
<input type="radio"/>	timeOption0: Anual	Always	
<input checked="" type="radio"/>	timeOption1: Trimestral	Always	

Add option +

☐ Override default with custom logic

Delete

Save & close

Finalmente en la sección de selectores, se agrega el selector creado y queda disponible en la página web (figura X).

▼ Selector activation

Inactive selectors

growthSelector1 +

comerceMeasureSelector +

yearSelectorGdp +

Show Full List...

Active selectors

Selector GDP

Anual


×

## Sección Stats

En esta sección se crean estadísticas, en este ejemplo se utilizará una estadística con el dato más actualizado del producto interno bruto anual. Para esto se requiere crear un generador que retorne esa variable. El primer paso para crear el generador es buscar la API que traiga el dato más actual del PIB, para obtener esa API se entra a la tecnología Tesseract UI y se modifican las opciones para poder extraer este dato, en este ejemplo se sigue utilizando el cubo inegi\_gdp mencionado anteriormente y se agregan las opciones drilldown: Year y Cut: Sector, una vez cargado el cubo con estas opciones se procede a





copiar la URL que trae los datos en la sección de Debug parameters, específicamente en Tesseract LogicLayer API URL.

 Debug parameters ×



## Download dataset

CSV file | JSON Tidy file | JSON Arrays file

## Tesseract Aggregate API URL

 [https://dev-api.datamexico.org/tesseract/cubes/inegi\\_gdp/aggregate.jsonrec](https://dev-api.datamexico.org/tesseract/cubes/inegi_gdp/aggregate.jsonrec)  Open

## Tesseract LogicLayer API URL

 [https://dev-api.datamexico.org/tesseract/data.jsonrecords?Sector=11&cube=inegi\\_gdp&drilldowns=Year&measures=GDP&parents=false&sparse=false](https://dev-api.datamexico.org/tesseract/data.jsonrecords?Sector=11&cube=inegi_gdp&drilldowns=Year&measures=GDP&parents=false&sparse=false)  Open


olap-client


[https://dev-api.datamexico.org/tesseract/data.jsonrecords?Sector=11&cube=inegi\\_gdp&drilldowns=Year&measures=GDP&parents=false&sparse=false](https://dev-api.datamexico.org/tesseract/data.jsonrecords?Sector=11&cube=inegi_gdp&drilldowns=Year&measures=GDP&parents=false&sparse=false)

Una vez obtenida la URL, se pega en la opción API (figura X) y se selecciona la opción “Fetch data” para ver la información que trae la API, en la figura X se ejemplifica.

Generator editor ×

Name  Description



API  

JavaScript 

```
1 const data = resp.data;
2
3 const dataFilter = data.filter(d => d["Year"] === 2020)
4
5 const testGDP = dataFilter[0]["GDP"]
6
7 return {
8   testGDP
9 }
```

Callback

You have access to the variable **resp**, which represents the response to the above API call.  
Be sure to return an **object** with the variables you want stored as keys.

 Delete  Save & close

Name test11

API [https://dev-api.datamexico.org/tesseract/Sector=11&cube=inegi\\_gdp&drilldowns=Year&measures=GDP](https://dev-api.datamexico.org/tesseract/Sector=11&cube=inegi_gdp&drilldowns=Year&measures=GDP)Compiled URL: [https://dev-api.datamexico.org/tesseract/Sector=11&cube=inegi\\_gdp&drilldowns=Year&measures=GDP](https://dev-api.datamexico.org/tesseract/Sector=11&cube=inegi_gdp&drilldowns=Year&measures=GDP)

Fetched data

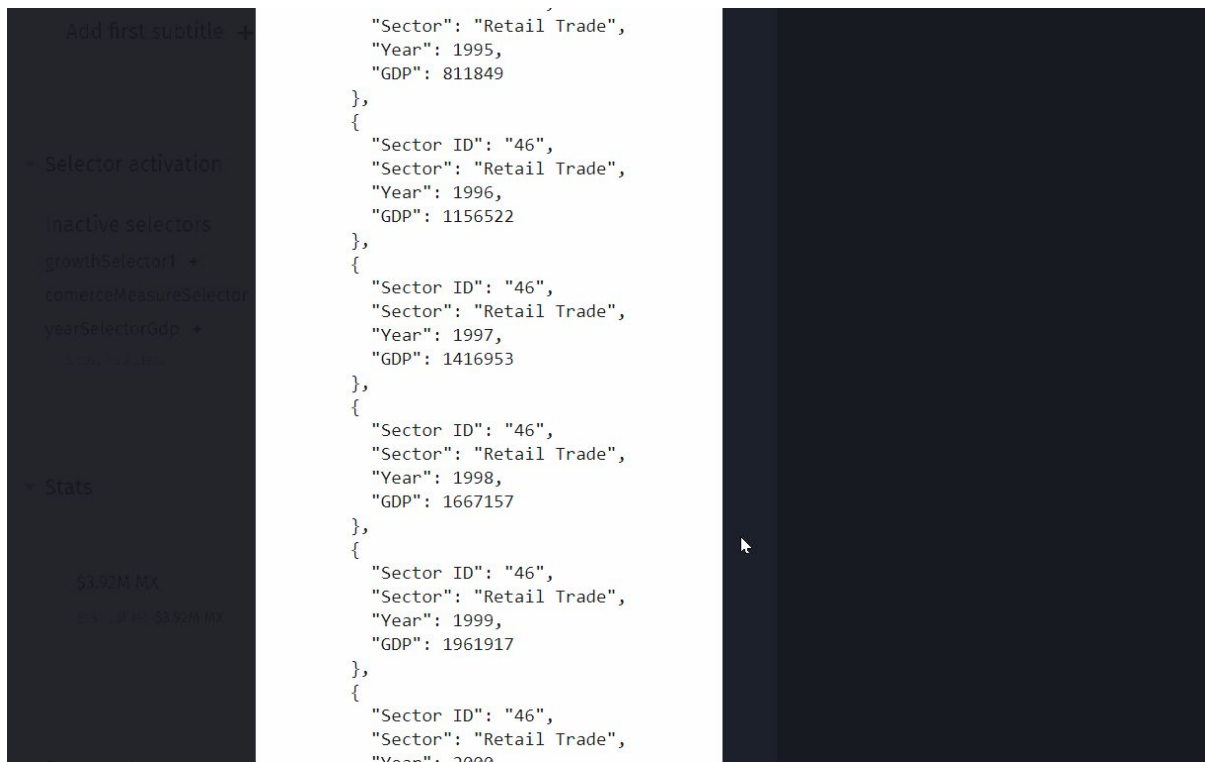
```
{
  "data": [
    {
      "Year": 1993,
      "GDP": 318289
    },
    {
      "Year": 1994,
      "GDP": 328981
    },
    {
      "Year": 1995,
      "GDP": 409115
    },
    {
      "Year": 1996,
      "GDP": 608200
    },
    {
      "Year": 1997,
      "GDP": 705373
    }
  ]
}
```

Si se analizan los datos, se puede ver que solo trae información de un sector, y si se utiliza esta URL para crear contenido en los perfiles, solo se mostrará la estadística perteneciente al sector elegido, en este caso al sector 11 ( ver URL), para que no ocurra esto y que cada perfil muestre la información correspondiente, se ocupa el customAttribute “hierarchy”, el cual entrega la información correspondiente a cada nivel seleccionado y se cambia [Sector=11](#) por “<hierarchy>=<id>”.

URL a utilizar:

[https://dev-api.datamexico.org/tesseract/data.jsonrecords?<hierarchy>=<id>&cube=inegi\\_gdp&drilldowns=Sector%2CYear&measures=GDP&parents=false&sparse=false](https://dev-api.datamexico.org/tesseract/data.jsonrecords?<hierarchy>=<id>&cube=inegi_gdp&drilldowns=Sector%2CYear&measures=GDP&parents=false&sparse=false)

Como se puede ver en la figura X, los datos extraídos de la API muestran la información perteneciente a cada sector (nivel industrial).



Al tener la información correspondiente, se hace el código en la consola utilizando JS, cabe destacar que este código puede ser de múltiples formas dependiendo del programador. En la figura X se muestra el código utilizado para generar una variable que retorne el último valor anual del PIB, el cual se utilizara como estadística en la sección "Producto interno bruto".

```
JavaScript UI mode  
1 const data = resp.data;  
2  
3 const dataFilter = data.filter(d => d["Year"] === 2020)  
4  
5 const testGDP = dataFilter[0]["GDP"]  
6  
7 return {  
8   testGDP  
9 }  
10
```

Como se puede ver en la figura X la variable retornada es “testGDP”, esta variable es la que entrega el último valor del PIB anual(el más actualizado). Para asegurar que la variable es retornada, se sugiere revisar el generador y ver si se retorna el valor, ejemplificado en la figura X. Una vez creado el generador, se procede a ocuparlo en la sección Stats, donde se tendrá que “llamar” a la variable “testGdp”.



Finalmente, el último paso para crear la Estadística, es presionar el botón de agregar Stats y completar la descripción, en donde se llamará a la variable creada como “{{testGDP}}” y se le aplica el fomatter pesoMX, el cual le da formato de pesos a la variable.

Stat editor

Stat label

PIB 2020

Stat value

pesoMX{{testGDP}}

Stat subtitle

Tooltip text

Pro tip: You can type `{{variable name}}` , `[[selector name]]` , or `@formatter name` for a list of respective entities.

Visible

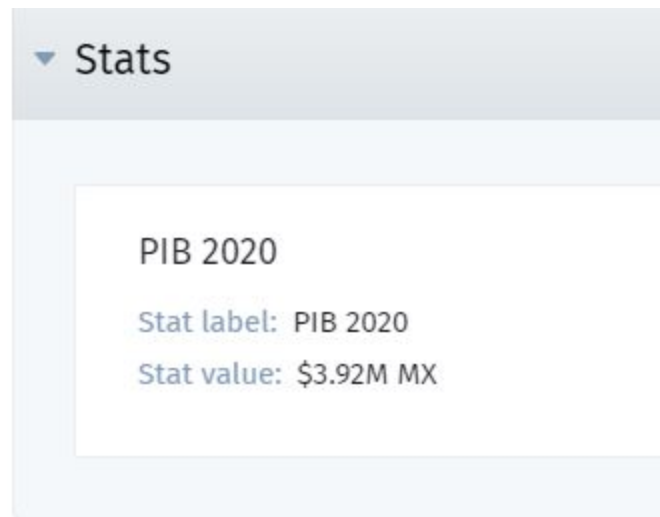
Always

☐ Override visible property with custom variable

Delete

Save & close

Finalmente la estadística se ve reflejado en la figura X.



### Sección Párrafos

Para crear un párrafo, solo se requiere presionar el boton “add paragraph” y escribir el contenido, en este caso agregaremos una pequeña oración y llamaremos a la variable “testGDP”.

Paragraph editor

Paragraph

Rich Formatting ☐

El GDP fue pesoMX{{testGDP}}

Pro tip: You can type `{{variable name}}` , `[[selector name]]` , or `@formatter name` for a list of respective entities.

Visible 

Always

☐ Override visible property with custom variable

Delete

Save & close

En la figura X se ejemplifica el resultado.



### Sección visualización

Para crear una visualización se presiona el botón “add visualization”, y aparecerá el editor de visualizaciones. El primer paso es agregar una API en el “Data endpoint”, para esto buscaremos la API en Tesseract UI, copiaremos la URL desde “Debug parameters”. Como el objetivo es visualizar el producto interno bruto anual por sector industrial, elegiremos el cubo inegi\_gdp, drilldown: Year y Cut: sector (explicado anteriormente), donde luego se agregara el customHierarchy como se mostró anteriormente.

Api:

[https://dev-api.datamexico.org/tesseract/data.jsonrecords?<hierarchy>=<id>&cube=inegi\\_gdp&drilldowns=Sector%2CYear&measures=GDP&parents=false&sparse=false](https://dev-api.datamexico.org/tesseract/data.jsonrecords?<hierarchy>=<id>&cube=inegi_gdp&drilldowns=Sector%2CYear&measures=GDP&parents=false&sparse=false)

Luego de obtener la API, se elige el tipo de visualización, en este caso será un Lineplot, que se usa para visualizar variaciones en el tiempo, además se elige la opción grouping por sector, para que así la visualización se muestre en cada sector industrial. Por último, se definen las variables X: tiempo e Y : producto interno bruto.

Visualization editor ×

Data Endpoint + https://dev-api.datamexico.org/tesseract/data.jsonrecords?<hierarchy>=<id>&cube Rebuild

Visualization Line Plot ▼

Title

UI mode

Grouping + Sector ▼

No formatter ▼

X-Axis Year ▼

No formatter ▼

Y-Axis GDP ▼

No formatter ▼

X-axis label Year

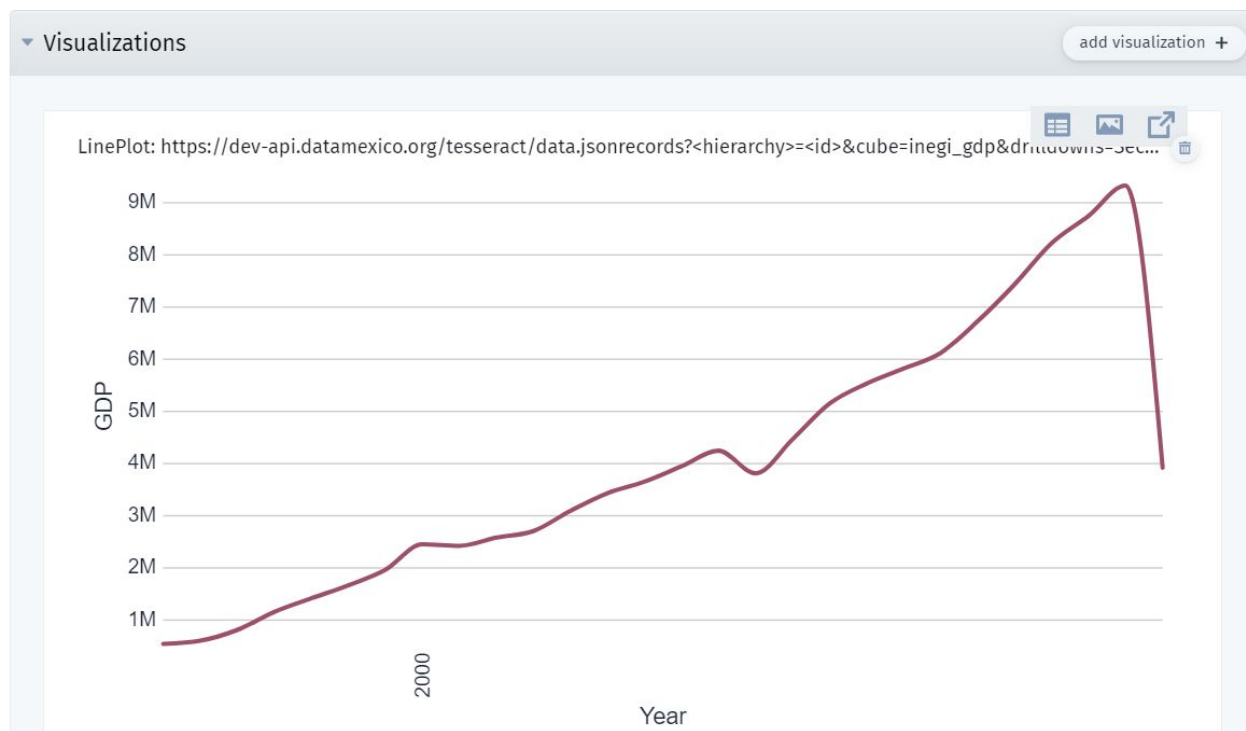
Y-axis label GDP

Visible Always ▼

☐ Override visible property with custom variable

Delete Save & close

Una vez rellenado las opciones del editor de la visualización , se genera la visualización de la siguiente forma (ver figura X).



Finalmente, la sección completa en la página web, con los elementos creados anteriormente (selector, estadística, párrafo y visualización) se muestran en la figura X. Donde el producto interno bruto corresponde al Título, el selector tiene las opciones de mostrar la información de forma anual y trimestral, además se creó la estadística que refleja el PIB del año 2020, también está el párrafo y por último la visualización que muestra cómo varía el PIB en los años 1993-2020 en el sector Z.

## Producto interno bruto

Anual

Trimestral

\$1.4M MX

PIB 2020

El GDP fue \$1.4M MX

Fuente [Instituto Nacional de Estadística y Geografía \(INEGI\)](#).

