

# Herramientas de visualización de datos

## Breve descripción:

Este componente formativo aborda aspectos generales y claves de las herramientas de visualización, mismas que han ayudado a interpretar los datos que pretenden guiar tendencias y que, realizando una serie de métodos, permiten sacarles el mejor provecho definiendo objetivos específicos. Estas herramientas se trasladan al lenguaje de programación preferido por la comunidad de desarrolladores (Python), gracias también a las librerías que permiten hacer análisis científico de datos.

---

Octubre 2023

## Tabla de contenido

Introducción .....	1
1. Herramientas a utilizar .....	3
2. Implementación con Matplotlib .....	10
2.1. Componentes de una figura .....	11
2.2. Definición de Pandas .....	20
2.3. Tipos de entradas para graficar funciones.....	24
3. Simulación de la herramienta mediante caso práctico .....	26
Síntesis .....	28
Material complementario .....	30
Glosario .....	31
Referencias bibliográficas .....	33
Créditos .....	35

## Introducción

A medida que la ciencia de datos se introduce cada vez más en la cotidianidad de las predicciones en las empresas, se torna indispensable para el futuro de estas, dándoles sentido a los millones de datos que circulan diariamente por la red, pero, para esto, es necesario el buen uso de las herramientas de visualización de datos.

A continuación, le invitamos a explorar el siguiente video, el cual introduce los temas de este componente formativo:

### Video 1. Herramientas de visualización de datos



[Enlace de reproducción del video](#)

#### Síntesis del video: Herramientas de visualización de datos

Las herramientas para la visualización de datos han ayudado a interpretar los datos que, en conjunto, pretenden guiar una tendencia y que, realizando una serie de métodos, permiten sacarles el mejor provecho para definir objetivos específicos.

Estas herramientas se trasladan al lenguaje de programación más fácil y preferido por la comunidad de desarrolladores del mundo (Python), gracias a las librerías que permiten el análisis científico de datos, que hace que hoy en día su interpretación sea posible.

Es difícil comprender hoy en día que los sectores productivos no se benefician de los datos y de sus predicciones para la toma asertiva de decisiones. Las habilidades de las personas tienden a cambiar para volcarse a la interpretación de los datos mediante la utilización de las herramientas que se han creado para ese fin.

Así no trabaje con datos todos los días, es probable que haya utilizado antes muchos tipos de visualizaciones de datos. No hay una única forma de crear una visualización de datos, aunque el proceso general de creación suele ser similar, siempre es mediado a través de alguna herramienta de visualización.

Recuerde que, así cada herramienta de visualización de datos sea diferente, la mayoría de estas comparten algunas características, y a través de este componente formativo, veremos las herramientas utilizadas por la comunidad Python que han demostrado ser las librerías más poderosas frente a los temas de interpretación de gráficas.

## 1. Herramientas a utilizar

La comunidad de Python es muy grande, por lo tanto, es posible encontrar un gran número de librerías para visualizar datos. Al tener tanta variedad de opciones, a veces se hace realmente difícil determinar cuándo utilizar cada una de ellas.

Por consiguiente, a continuación, se relacionan las más utilizadas por la comunidad Python y que han demostrado ser las librerías más poderosas frente a los temas de interpretación de gráficas:

- a. **Matplotlib.** Es la librería más antigua de todas y, asimismo, es la más potente; de hecho, de esta librería surgen muchas otras más. Se caracteriza por su gran posibilidad de hacer de todo con ella, pero de ahí viene también su complejidad y conseguir información de cómo hacer las cosas con esta librería no resulta tan fácil.
- b. **Bokeh.** Es una de las librerías más recientes o jóvenes, pero no por ello deja de ser una herramienta sofisticada y potente, ya que permite realizar gráficos interactivos al estilo de D3.js. Se caracteriza por su fácil trabajo con los navegadores actuales, ya que sus gráficas profesionales son congruentes con los grandes volúmenes de datos que puede manejar; es una gran herramienta para cualquier persona que quiera graficar de manera rápida y sencilla gráficos interactivos, como “dashboards”; sin embargo, esta librería es capaz de crear tanto gráficos estáticos como interactivos.
- c. **Seaborn.** Librería especializada en gráficos estadísticos. Como muchas otras, está basada en la librería madre de muchas, Matplotlib; con ella, se pueden elaborar visualizaciones complejas de manera sencilla. Una de las características de esta librería es el desarrollo de regresiones lineales,

distribuciones de una o varias variables aleatorias, entre otras características que le dan más importancia para que sea una buena opción a la hora de graficar.

- d. Folium.** Si se quiere realizar gráficos geográficos, esta librería es la indicada, ya que permite realizar gráficos de alta densidad e interactivos al estilo de leaflet.js., debido a que los gráficos de “folium” son interactivos, lo que hace que sea la herramienta ideal para realizar “dashboards”.

Cuando se empieza un proyecto de visualización, la duda que se viene a la cabeza es cuál herramienta utilizar para poder realizar un trabajo que satisfaga las necesidades del usuario final. En ese orden de ideas, es bueno tener en cuenta la siguiente guía, que ayudará a la escogencia de la herramienta o librería indicada; no es obligatorio seguirla al pie de la letra, pero será de gran valor a la hora de elegir:



- a. Dos variables (Relación).** En la categoría de relación, si tienes dos variables, podrás usar un gráfico de dispersión:



**b. Tres variables (Relación).** En la categoría de relación, si tienes tres variables, podrás usar un gráfico de burbujas:



**c. Variable única (Distribución).** En la categoría de distribución, podemos tener una variable única, pero que, a su vez, tenga pocos o muchos ítems; para este caso, se recomiendan los siguientes gráficos:



**d. Dos o tres variables (Distribución).** En la categoría de distribución, podemos tener dos o tres variables; para estos casos, se recomiendan los siguientes gráficos:

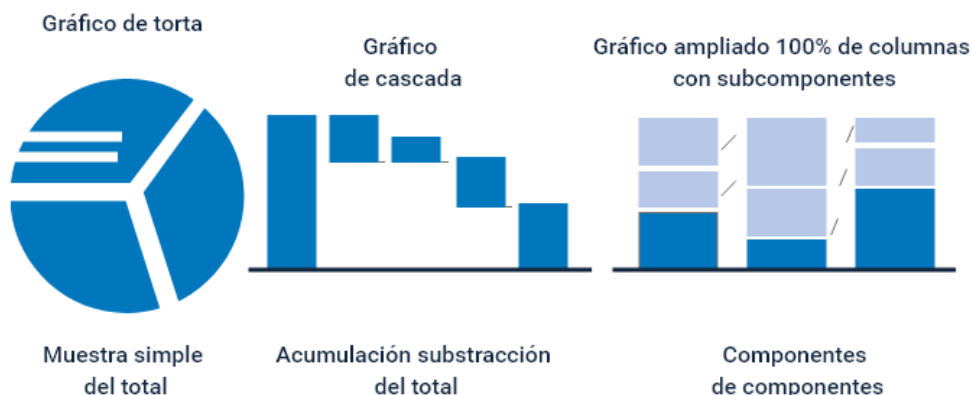


**e. Cambiantes en el tiempo (Composición).** En esta categoría de composición, si tenemos datos cambiantes en el tiempo, podemos manejar los siguientes gráficos, según corresponda:

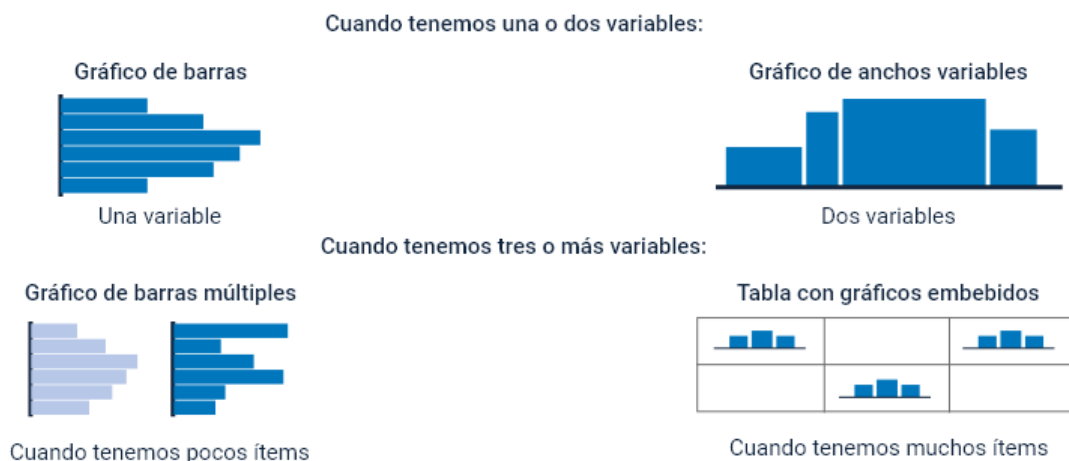


**f. Estático (Composición).** En la categoría de composición, si tenemos datos estáticos, podemos manejar los siguientes gráficos, según corresponda:

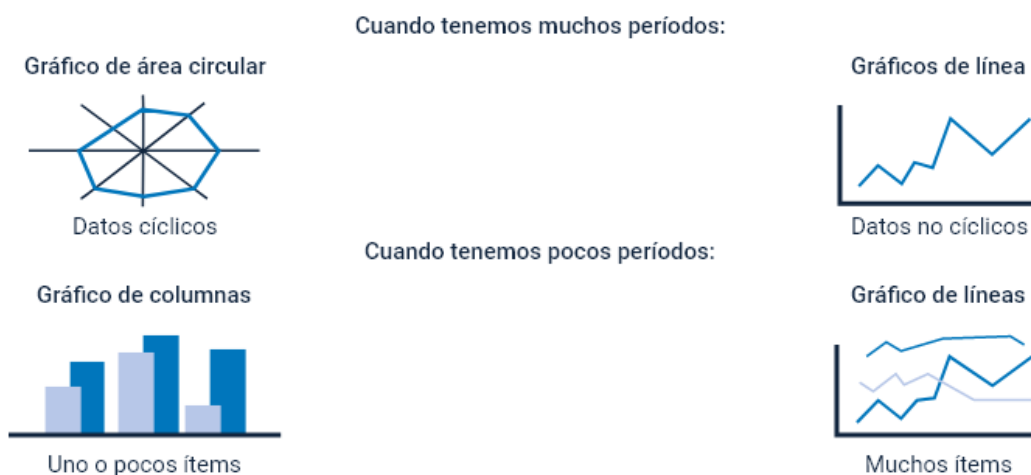




**g. Estático (Comparación).** En esta categoría de comparación, si tenemos datos estáticos, podemos usar los siguientes gráficos, según corresponda:



**h. Cambiantes en el tiempo (Comparación).** En esta categoría de comparación, si tenemos datos estáticos, podemos usar los siguientes gráficos, según corresponda:



Como se muestra, hay una división en cuatro partes principales y, estas, después, se subdividen en métodos de visualización, que representan las categorías iniciales, como se expone a continuación:

- **Distribuciones.** En esta parte, se intenta comprender la distribución de los datos, por lo tanto, se encuentran variables cuantitativas y categóricas; solo depende de las variables y la cantidad de ellas.
- **Comparaciones.** Se comparan valores a través de diferentes categorías; los gráficos que se identifican con esta etapa son las barras, cuando se realiza una comparación de elementos y categorías, y los diagramas de puntos y líneas, para las variables cuantitativas.
- **Relaciones.** Lo importante en este modelo es comprender la relación que hay entre las variables; su representación gráfica son las gráficas de dispersión.
- **Composiciones.** En este modelo, lo importante es comprender cómo está compuesta una variable y sus relaciones frente a otras, ya sea en una

proyección de tiempo o de manera estática. Las gráficas que las representan son las gráficas de barras y las tortas.

## 2. Implementación con Matplotlib

Esta librería se destaca por ser la más utilizada en la comunidad desarrolladora, particularmente, para los científicos de datos. Matplotlib, como ya se ha dicho, no es más que una biblioteca de Python para la visualización de gráficas dinámicas o estáticas. En este apartado, se tratará de explicar cómo es su implementación y cuáles son las gráficas resultantes.

Para instalar e implementar la biblioteca, se corre el comando “pip”, a través de los siguientes pasos:

- **Primer paso.** Correr el comando “pip” a través de Google Colab: `pip install matplotlib`.

Una vez realizada la activación de la librería, utilizamos los parámetros adecuados para realizar una gráfica, estos parámetros pueden estar dados por una empresa que requiera visualizar el comportamiento de determinado producto en el tiempo o simplemente con fines estadísticos o de geolocalización.

- **Segundo paso.** A continuación, veamos, a través de un ejemplo, la manera en que se utiliza la librería objeto de estudio:

Graficamos la siguiente función:  $f(x)=\sin^2(X-2)e^{-X^2}$  sobre el intervalo  $[0,2]$ , agregando la etiqueta de los ejes y un título apropiado, como vemos en la imagen.

- **Tercer paso.** Al mostrarse el “script”, se comentarán todos los ajustes correspondientes; los parámetros son esenciales para darle la forma y función al gráfico, pero puede cambiar y jugar con estos valores para estudiar sus cambios.

- **Cuarto paso.** Pero, a su vez, si lo necesitamos, al modificar los parámetros colocados, la gráfica debe cambiar.

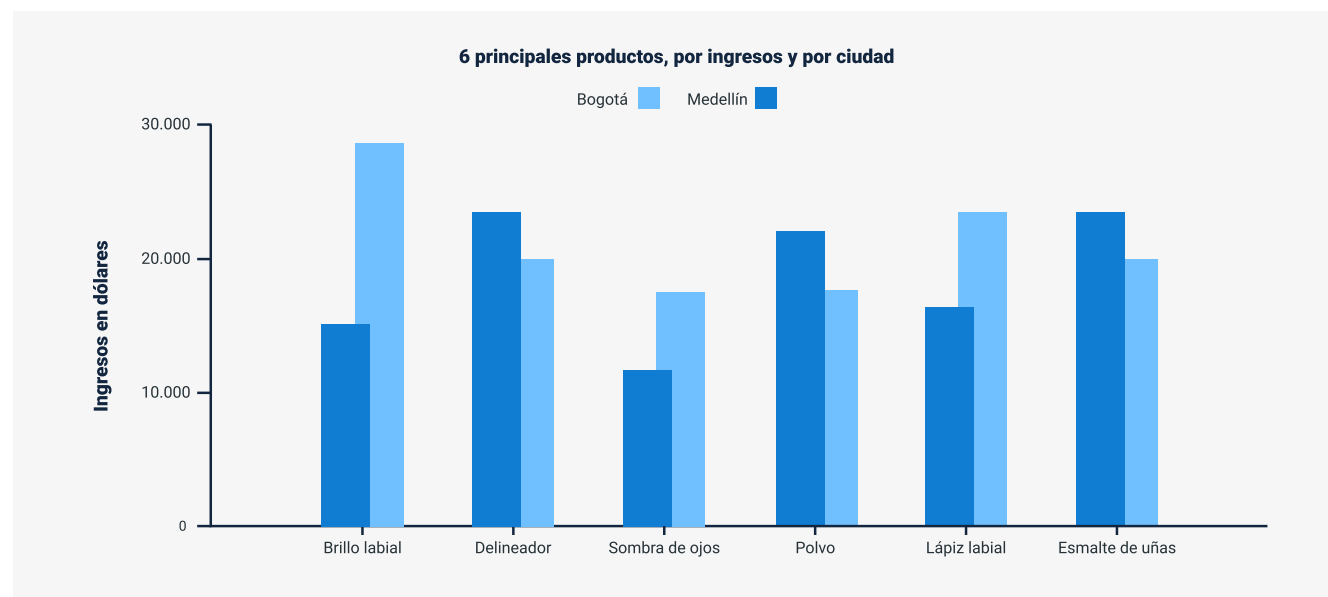
## 2.1. Componentes de una figura

La forma en que se puede realizar una gráfica puede variar según lo que se requiera representar, por lo tanto, existen diferentes formas con las que se puede representar una situación, siendo muy amplias y útiles, gracias a las librerías especializadas como Matplotlib.

Las gráficas que aportan estas librerías ofrecen información contundente frente a los datos ingresados, es posible categorizarlas bajo la premisa que se tenga para la organización de datos, tal y como se expone a continuación:

### Al comparar datos

**Figura 1.** Gráfica de comparación de datos



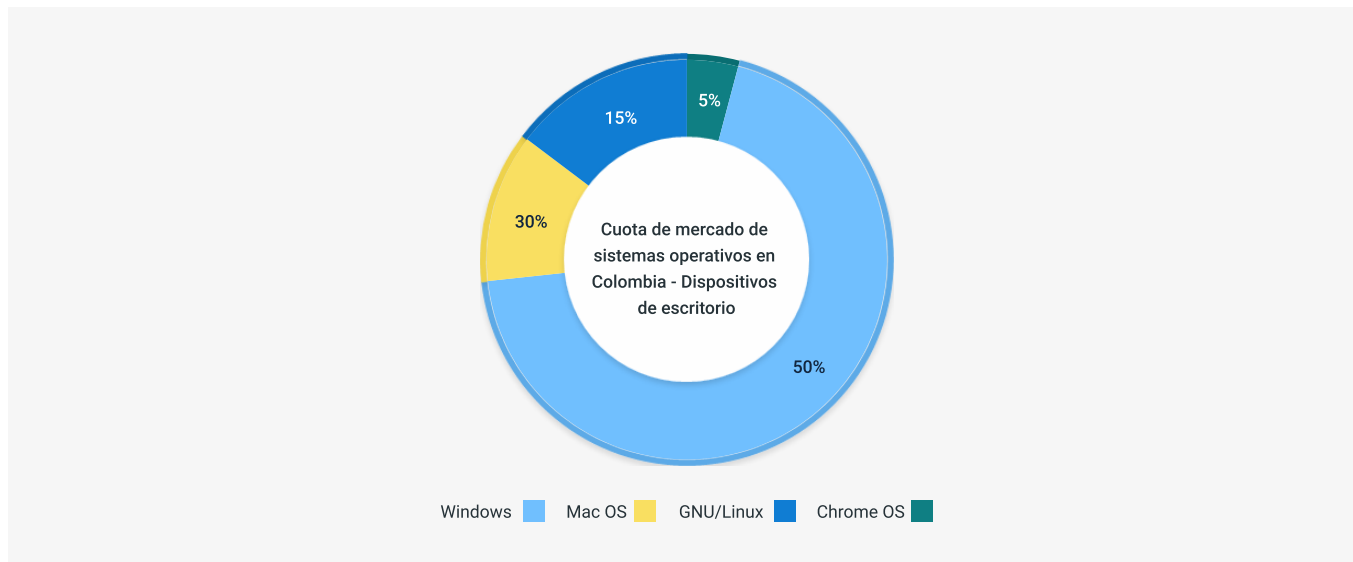
En una gráfica de comparación de datos se muestran, por ejemplo, en forma de barras, los datos existentes y, mediante el tamaño o extensión de las barras de la gráfica, es posible determinar la proporción de los datos. Es así como se pueden comparar datos de diferentes variables de un mismo “dataset” o datos de un “dataset” contra otro.

Los siguientes tipos de gráficos son muy útiles y suelen ser, también, frecuentemente usados:

- **Gráfico de barras.** Permite comparar, de forma directa, valores cuantitativos por variable o categoría.
- **Gráfico de pilas apiladas.** Permite analizar valores cuantitativos por categoría y, a la vez, compararlos con diferentes muestras. Por ejemplo, analizar el PBI de diferentes países, teniendo en cuenta el aporte porcentual de cada sector.
- **Gráfico de radar.** Es otra alternativa para comparar los valores cuantitativos de diferentes variables. Por ejemplo, mostrar el puntaje en diferentes habilidades de las destrezas de un jugador de fútbol, como conversión de goles, precisión en pases, etc.

## Explorar relaciones / “Part-To-Whole” (de la parte al todo)

**Figura 2.** Gráfica de exploración de relaciones



La cuota de mercado, en este caso, de sistemas operativos en Colombia es de:

- Windows 50 %
- Mac OS 30 %
- GNU/Linux 15 %
- Chrome OS 5 %

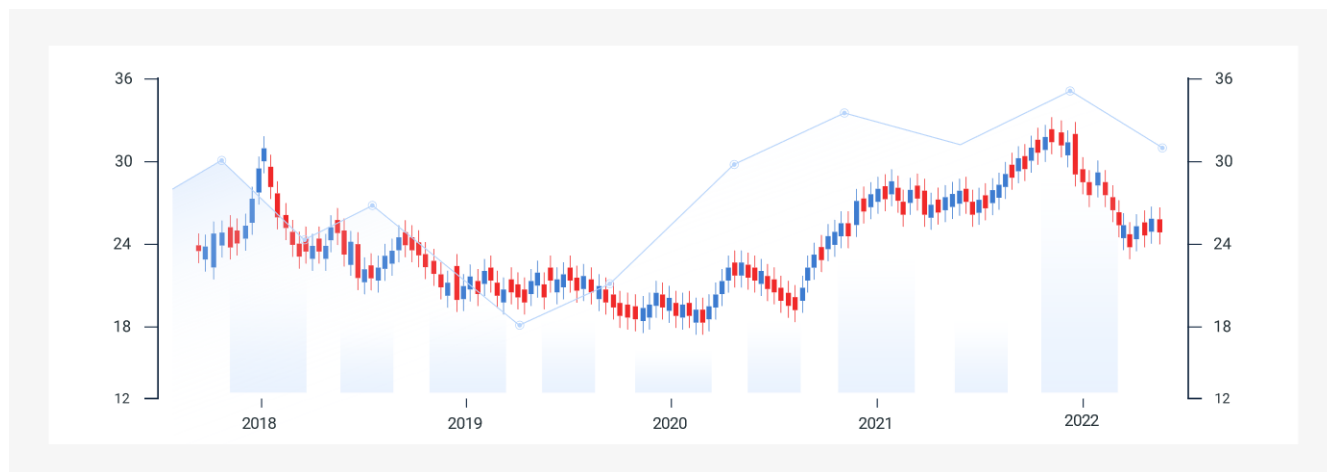
Cuando lo que se quiere es explorar la composición, por ejemplo, de una variable categórica y analizar relaciones del tipo “part-to-whole”, los siguientes tipos de visualizaciones son muy útiles:

- **Gráfico de torta.** Permite analizar de forma básica la composición porcentual de una variable categórica.
- **Gráfico piramidal.** Cuando los datos representan o responden a una "jerarquía", puede ser interesante utilizar este tipo de gráficos.

- **Mapa de árbol.** Permite representar relaciones jerárquicas complejas. Por ejemplo, la cantidad de ingresos anuales de una compañía agregados por unidades de negocio y para cada unidad de negocio por proyecto.
- **Gráfico de embudo.** Este tipo de gráfico se utiliza mucho para comparar variables cuantitativas que de alguna forma están relacionadas entre sí como parte de un proceso. Por ejemplo, se puede utilizar para analizar ventas de un sitio de “e-commerce”, ayudando a identificar los cuellos de botella del proceso.

## Análisis en el tiempo

**Figura 3.** Gráfica de análisis en el tiempo



Durante un determinado lapso de tiempo, año tras año, los estados o puntos de la variable en cuestión fue de:

- 12 (2018)
- 18 (2019)
- 24 (2020)
- 30 (2021)



- 36 (2022)

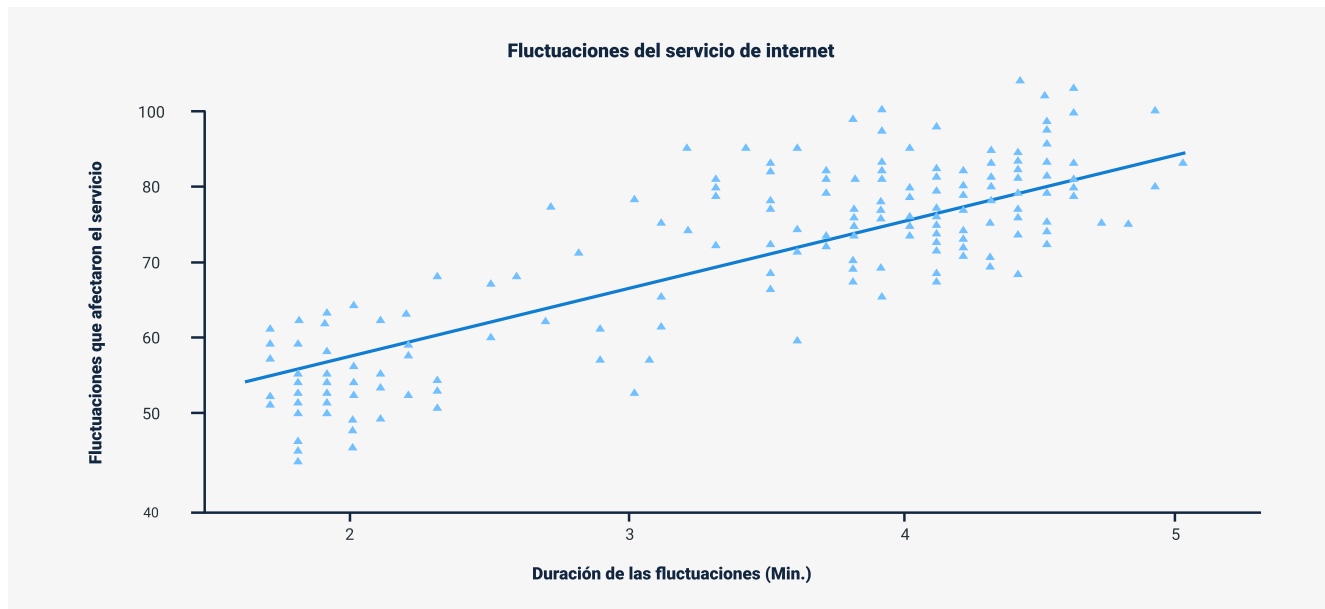
Lo que quiere decir que fue al alza, año tras año.

Cuando se busca analizar la relación entre una o más variables con un componente temporal como, por ejemplo, el analizar la evolución de los salarios de Uruguay, respecto del tiempo, los siguientes tipos de visualizaciones son los indicados:

- **Gráfico de línea / curva.** Es la representación gráfica más simple y utilizada para mostrar la evolución de una variable cuantitativa en función de otra variable.
- **Gráfico de área.** A diferencia de la anterior, utiliza áreas para representar una variable cuantitativa.
- **Gráfico de velas o candela japonesa.** Es muy útil cuando se quiere analizar una variable que puede tomar múltiples valores para un mismo momento, por ejemplo, analizar la evolución del precio de las acciones de una compañía frente a otras, tomando como referencia el precio de apertura y cierre de las acciones en la bolsa por día. Es una variante de las gráficas OHLC.
- **Gráfico “Sparkline”.** Consiste en construir versiones simplificadas y pequeñas de otros gráficos temporales, como “Line Charts” o “Area Charts”, eliminando, por ejemplo, los ejes del gráfico. Son muy útiles para utilizar en tablas donde, para cada fila, queremos hacer más visual una variable determinada. Por ejemplo, listar los montos totales de exportaciones de granos en Uruguay en un año entero, utilizando “Sparklines” para mostrar, para cada grano, la variación mensual del precio de este.

## Analizar distribución de un “dataset”

**Figura 4.** Representación gráfica de datos estructurados (“dataset”)



Esta gráfica representa, las fluctuaciones del servicio de internet, las fluctuaciones que afectaron el servicio y la duración de las fluctuaciones.

Para analizar la distribución de una variable en un determinado “dataset” y comparar, por ejemplo, con otros “datasets”, encontrando relaciones y correlaciones, los siguientes tipos de gráficos son muy útiles:

- **Histogramas.** Un histograma es una representación de la estimación de la distribución de valores de un “dataset” numérico.
- **Gráfico de dispersión de puntos.** Permite examinar las tendencias en distribución y correlación entre dos variables.
- **Gráfico de burbujas.** Cada punto se grafica como un disco, expresando dos de los atributos/dimensiones del dato a través de la ubicación del disco en el sistema de ejes cartesianos y el tercero, a través de su tamaño.

- **Gráfico “Box and Whisker”.** El diagrama de caja y bigotes es un tipo de gráfico que muestra un resumen de una gran cantidad de datos en cinco medidas descriptivas, además de intuir su morfología y simetría. Este tipo de gráficos nos permite identificar valores atípicos y comparar distribuciones.
- **Gráfico de errores.** Los gráficos de barras de error representan la variabilidad de los datos e indican el error (o incertidumbre) en una medida notificada.
- **Mapa de calor.** Es una representación visual bidimensional de los datos, donde los valores se codifican en colores.
- **Gráfico polar.** Los gráficos polares son útiles para mostrar datos científicos. Son gráficos circulares que usan valores y ángulos para mostrar información como coordenadas polares.

## Evaluar “performance” de un dato

**Figura 5.** Gráfica de “performance”



Según esta gráfica, la temperatura en la ciudad es de:

- 32 ° en el aire
- 28 ° en el agua

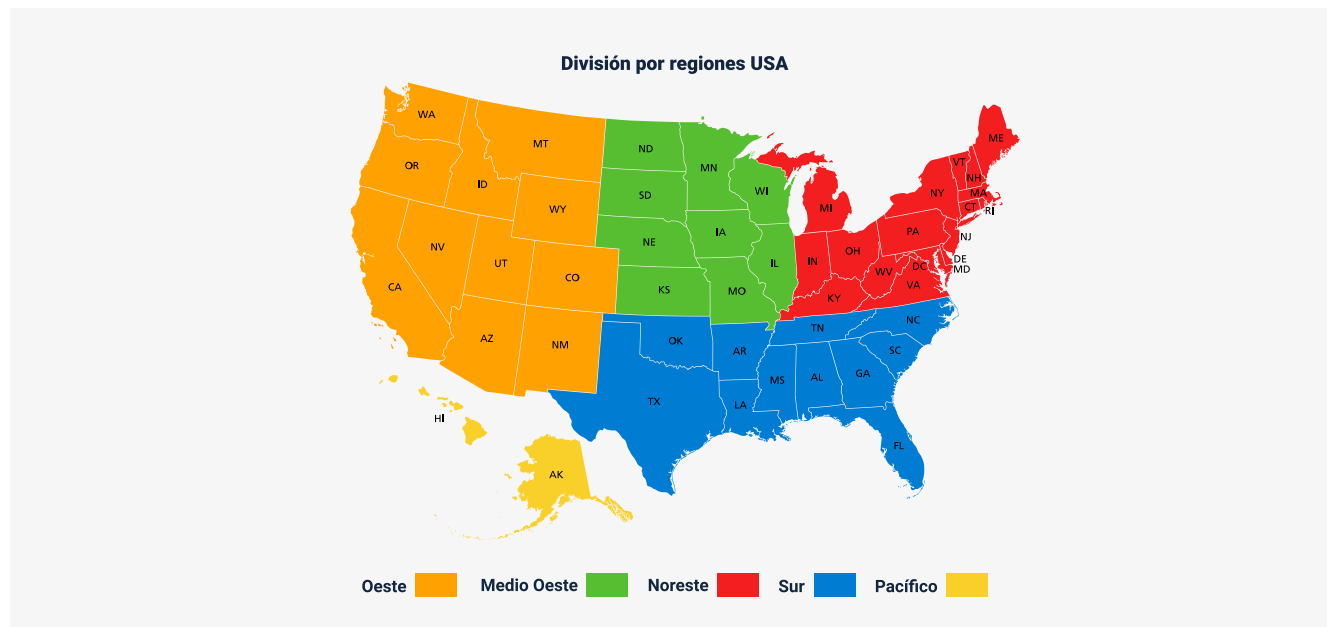
Teniendo un tiempo nublado, sin precipitaciones.

Cuando se quiere evaluar el rendimiento de una variable y comparar el valor real con valores KPI, se suele recurrir a los siguientes tipos de gráficos:

- **“Gauge” circular.** El indicador circular o indicador angular es un tipo de gráfico de indicadores, con una escala radial, como velocímetro, voltímetro, amperímetro y otros.
- **“Gauge” lineal.** “Linear Gauge” es una visualización de herramientas de medición con una escala lineal, como termómetros, tanques, indicadores de color (LED), etc. Los valores de los datos se muestran en el eje del indicador, con la ayuda de uno o varios punteros: marcadores, rangos, etc.
- **“Bullet graph”.** Los gráficos de viñetas están diseñados para superar los problemas fundamentales de los indicadores circulares y lineales, que a menudo están repletos de decoración inútil (y que distrae), mientras muestran muy poca información y requieren demasiado espacio. Los gráficos de viñetas pueden ser tanto verticales como horizontales.

## Datos geográficos

**Figura 6.** Gráfica de “performance” geográfico



En esta figura, se representan los estados en los que se dividen el país Estados Unidos de América.

Muchas veces, la mejor forma de darle sentido a datos geográficos (georreferenciados) es utilizando visualizaciones en un mapa, teniendo en cuenta:

- Mapa de colores (“choropleth map”).** permite identificar diferencias entre áreas geográficas.
- Mapa de puntos (“dot map”).** Proporciona una visualización accesible de la distribución geográfica de los fenómenos en las áreas geográficas en las que se encuentran.
- Mapa de burbujas (“bubble map”).** Similar a los gráficos de burbujas, permite agregar una tercera dimensión a la posición de un fenómeno, utilizando el tamaño de una burbuja. Son muy útiles para agregar la

información de un mapa de puntos cuando se cuenta con tantos puntos que se hace difícil entender la visualización.

- d. Mapa de conexiones (“connector map”).** Permite representar relaciones entre los distintos puntos del mapa mediante líneas que unen los puntos.
- e. Mapa de flujos (“flow map”).** Útiles para explorar cómo los objetos se mueven entre ubicaciones, en especial cuando interesa mostrar la dirección del flujo.

Hasta este punto, se han presentado diferentes maneras para realizar una gráfica, según la necesidad que se quiera representar, sabiendo que se tiene una gran cantidad de opciones, según el fin que se busque.

Algunos ejemplos de gráficas y representaciones de datos son:

- a. Gráfico de barras.** `bar(x, y)`: grafica una barra vertical, donde `x` es una lista con la posición de las barras en el eje X, y `y` es una lista con la altura de las barras en el eje Y.
- b. Histogramas.** `hist(x, bins)`: diagrama un histograma con las frecuencias que resultan de agrupar los datos de la lista `x` en las clases definidas por la lista `bins`.
- c. Diagrama de sectores.** Diagramar una imagen de sectores con frecuencias de la lista `X`.

## 2.2. Definición de Pandas

Según Google Colaboratory (2022), “Es una API de análisis de datos en columnas, ideal para manipular y analizar datos de entrada”. Además, muchos marcos de trabajo de AA admiten las estructuras de datos Pandas como entradas. Pandas es muy popular,

es una librería por preferencia dentro de la comunidad desarrolladora, es una de las más versátiles entre las de código abierto y más aún cuando se habla de temas como ciencia de datos y “machine learning” por su poderío en las estructuras que ofrece y la facilidad para manejar grandes cantidades de datos.

Sobre pandas, se puede afirmar que:

- Esta herramienta nace con la necesidad, de los analistas de datos, de unificar esfuerzos en una sola herramienta poderosa y facilitar la labor: analizar, depurar y seleccionar datos.
- Hay dos estructuras poderosas dentro de la librería de datos Pandas: Series y “DataFrame”. Las Series hacen referencia a un array unidimensional que puede almacenar cualquier tipo de dato.
- Los “DataFrame” son bidimensionales, con estructuras que pueden albergar cualquier tipo de dato; a su vez, se pueden interpretar como series.

## Datos geográficos

**Figura 7. Series y “DataFrames”**

Series			Series			Data frame		
apples			oranges			apples	oranges	
0	3	+	0	0	=	0	3	0
1	2		1	3		1	2	3
2	0		2	7		2	0	7
3	1		3	2		3	1	2

Como muestra la figura anterior, dos series (manzanas y naranjas), configurarán el “DataFrame”.

¿Qué se necesita para trabajar con Pandas a partir de ahora? Analice la información presentada en el siguiente video, la cual le indicará lo que necesitará:

## Video 2. Definición de Pandas



[Enlace de reproducción del video](#)

### Síntesis del video: Definición de pantas

Para poder trabajar con la librería de Pandas, es necesario activarla o descargarla mediante nuestro “notebook” de Google Colab, colocando el comando que se muestra en pantalla.

Realizamos lo mismo con la librería NumPy y procedemos a su activación o descarga.

Luego, para importar la librería Pandas, digitamos la línea de importación.

En la línea anterior a la escrita previamente, colocaremos como pd la librería Pandas.



Y en la siguiente línea, se importará la Api de Pandas, la cual nos mostrará su versión.

Existen varias formas de construir series, pero una de ellas es construir un objeto de series, tal y como se muestra en pantalla.

Los objetos de un “DataFrame” se pueden crear al enviar un “dict” para designar las columnas de las variables declaradas como “string” a sus series equivalentes; si estas series no coinciden con la longitud, los valores faltantes se autocompletarán con valores como NA (que es el valor no existe) o NaN (que es el que valor no está definido).

Normalmente, se cargan archivos completos a un “Dataframe”, que se muestra de la siguiente forma; estos se cargaron a través de un archivo con extensión .CSV, el cual es utilizado con fines ilustrativos. En ese sentido, se mostrarán los siguientes desarrollos partiendo de los datos que allí se encuentren, para lo cual se invita a descargar el mismo.

Del ejemplo descargado, se muestran datos interesantes al utilizar el comando “Dataframe.describe” y también al visualizar los primeros registros de ese documento con el comando “Dataframe.head”.

Otra de las muchas funciones de Pandas, y la cual es el objeto de este estudio, es el análisis por medio de gráficas con el comando “Dataframe.hist”, que estudia la distribución de valores en las columnas.

También se puede acceder a los datos del “Dataframe” por medio de comandos que Python permite, como el “dict” o el “list”.

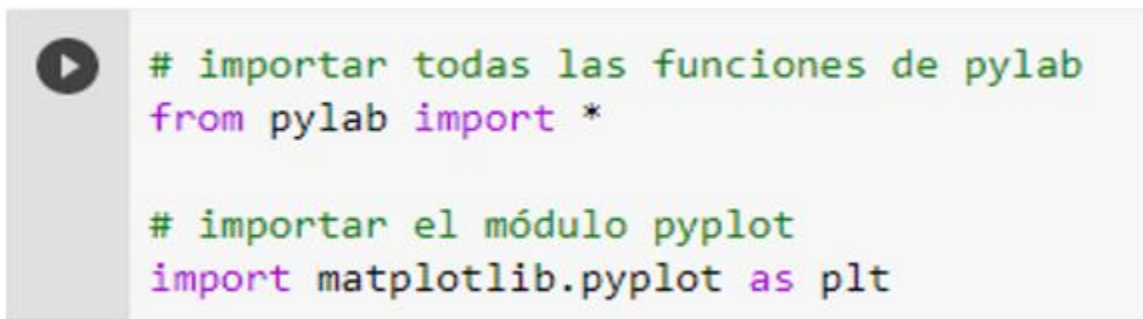
## 2.3. Tipos de entradas para graficar funciones

Como se ha venido estudiando en este componente formativo, existen diversas gráficas que pueden ayudar a la proyección estadística de algo que se quiera visualizar de la mejor manera, utilizando las herramientas correspondientes y las técnicas adecuadas para conseguir el objetivo propuesto; no obstante, las librerías que suministra Python son la mejor opción en el mercado de la ciencia de datos. Por ese motivo, la librería Matplotlib es la más utilizada en la comunidad del “Big Data”, se trata de la librería más grande y más completa de Python, que además de eso contiene dos módulos principales: pyplot y pylab, ambos con diferencias puntuales.

Pyplot es muy versátil y permite crear gráficas automatizando sus ejes cuando son creados. Por otro lado, pylab no se queda atrás con sus bondades, ya que trae algo de pyplot para la creación de gráficos con la versatilidad de Numpy y realizar cálculos con “arrays” con una etiqueta muy particular de Matlab.

A continuación, se ilustrará cómo es posible trabajar con estas funciones que marcan la pauta para una estructuración correcta de las gráficas de visualización:

**Figura 8.** Comenzando a graficar funciones



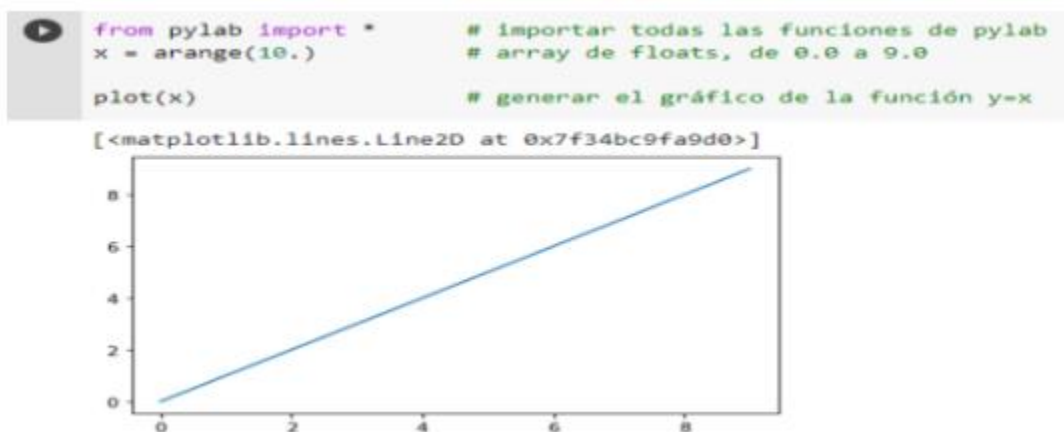
```
# importar todas las funciones de pylab
from pylab import *

# importar el módulo pyplot
import matplotlib.pyplot as plt
```

La imagen ilustra, en la primera línea de código, cómo se importa pylab y se tiene inmersa la librería NumPy como np.

Luego, se procede a crear un gráfico de la manera más sencilla posible:

**Figura 9.** Graficación



La anterior, es una gráfica que se ha creado con 10 puntos en el “array” y se ha visualizado con “show”().

### 3. Simulación de la herramienta mediante caso práctico

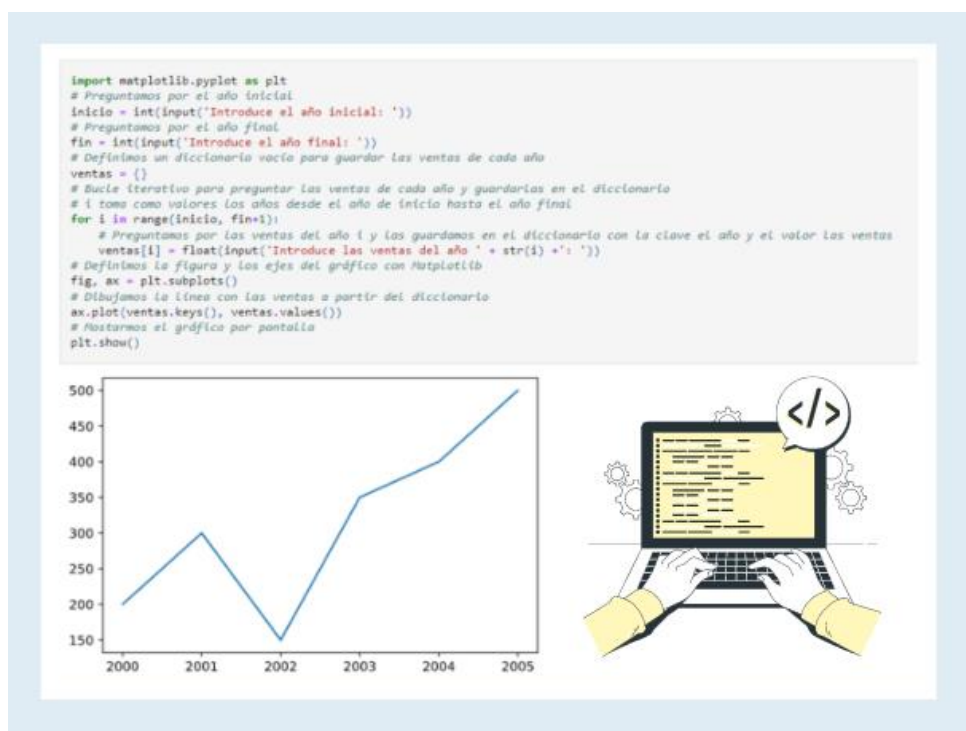
Para poder trabajar y poner en práctica los conceptos adquiridos, se verán dos ejemplos a continuación, para que, a través de la visualización de estos, pueda identificar lo estudiado hasta este punto y que sirva, a su vez, para su aplicación, teniendo en cuenta todo lo aprendido.

Las aplicaciones son muchas, pero aprópiase de las siguientes simulaciones donde es posible obtener el código utilizado y el resultado:

#### Ejemplo 1

Se digita un código donde se le pregunta a un usuario por las ventas en un intervalo de años y que muestra, mediante un diagrama, el incremento o decremento, dado el caso, de las ventas de la empresa:

**Figura 10.** Simulación de caso 1



## Ejemplo 2

Se digita un código para obtener un diccionario de notas con las materias de un salón de clase y, además, una cadena con la palabra de algún color y que retorne un diagrama de barras con las notas y también en el color que se indicó.

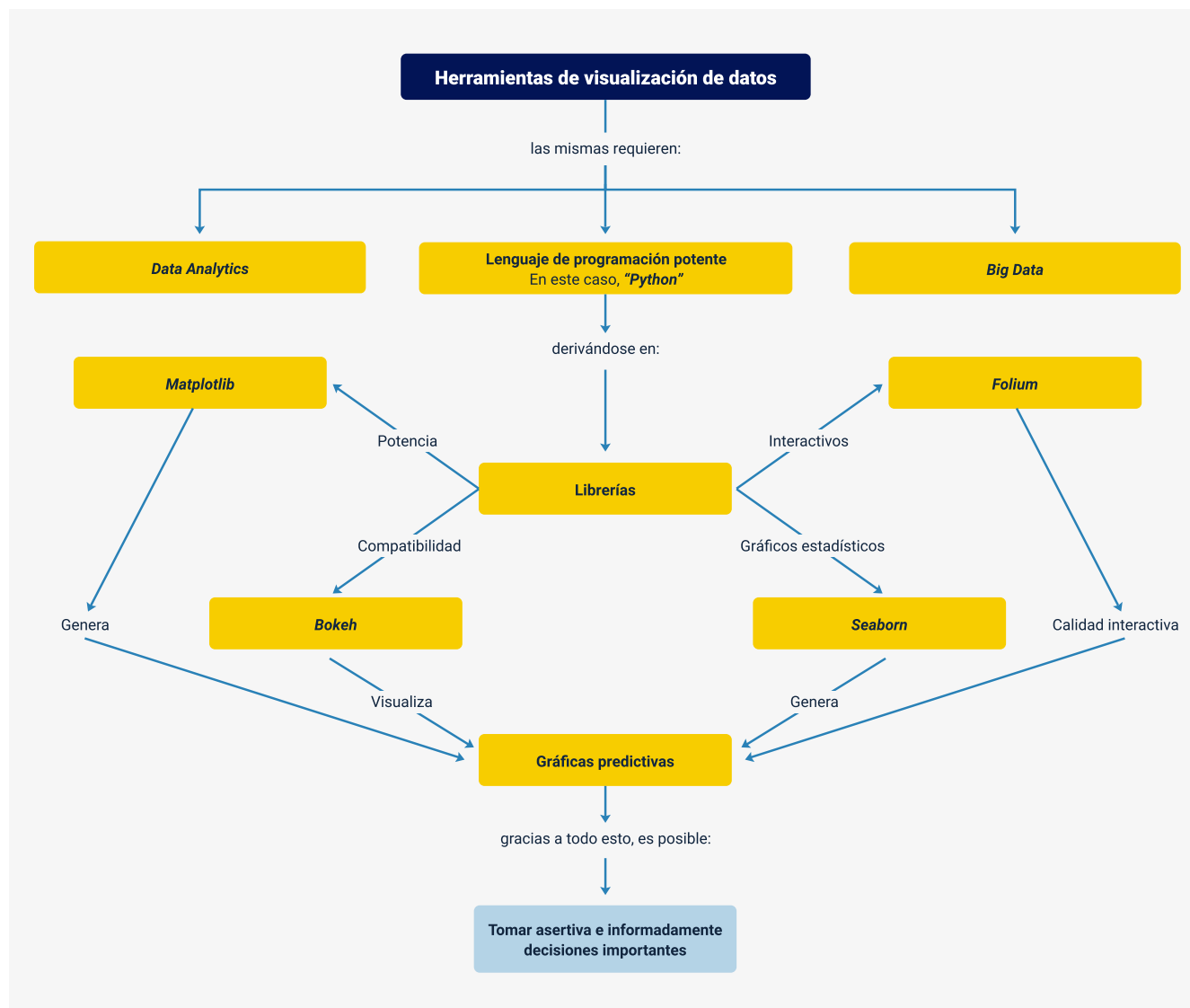
**Figura 11.** Simulación de caso 2



Los datos están cambiando la forma de ver los problemas y las decisiones, las empresas que están en la cúspide del mercado emplean sus habilidades para utilizar todo lo que esté a su alcance para la recolección de estos datos y así poder incursionar con nuevas tendencias en todos los campos; los datos están cambiando la base de las competencias.

## Síntesis

Aquí finaliza el estudio de los temas de este componente formativo; en este punto, analice el esquema que se muestra enseguida y realice su propia síntesis de los contenidos estudiados. ¡Adelante!



Este es el mapa de conceptos y temáticas desarrollados en este componente formativo. Su tema principal: Herramientas de visualización de datos.

Sus temas integradores:

- Herramientas a utilizar.
- Implementación con Matplotlib, lo cual vincula los componentes de una figura, definiciones de Pandas, tipos de entradas para graficar funciones.
- Simulación de la herramienta mediante caso práctico.

## Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
1. Herramientas a utilizar.	Credenciales Alternativas TEC. (2020). Las gráficas más populares para visualización de datos	Video	<a href="https://www.youtube.com/watch?v=OTtN0gu1-UA">https://www.youtube.com/watch?v=OTtN0gu1-UA</a>
1. Herramientas a utilizar.	Comunicación Numérica. (2021). Crea Reportes con Impacto, con visualización de datos.	Video	<a href="https://www.youtube.com/watch?v=TYp2IJhuE6A">https://www.youtube.com/watch?v=TYp2IJhuE6A</a>
1. Herramientas a utilizar.	Comunicación Numérica. (2020). Fundamentos del Análisis de Datos para Toma de Decisiones.	Video	<a href="https://www.youtube.com/watch?v=qvZxvMWMvDo">https://www.youtube.com/watch?v=qvZxvMWMvDo</a>



## Glosario

**Algoritmo:** son los algoritmos que mediante una serie de datos identifican distintas etiquetas. Por ejemplo, se pueden hacer dos grupos y diferenciar clientes que están cercanos a la compra en una página web de clientes que no lo están. Clientes muy rentables, poco rentables y no rentables.

**Centroides:** es la ubicación real o imaginaria que representa el centro del grupo. Cada punto de datos se asigna a cada uno de los grupos mediante la reducción de la suma de cuadrados en el grupo.

**“Dataset”:** no es más que un conjunto de datos tabulados en cualquier sistema de almacenamiento de datos estructurados.

**Google Colaboratory:** Colaboratory o "Colab" para abreviar, es un producto de Google Research. Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador. Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación.

**IPython:** es un “Shell” interactivo que añade funcionalidades extra al modo interactivo incluido con Python, como resaltado de líneas y errores mediante colores, una sintaxis adicional para el “Shell”, autocompletado mediante tabulador de variables, módulos y atributos.

**Jupyter:** jupyter Notebook es una aplicación cliente-servidor lanzada en 2015 por la organización sin ánimo de lucro Proyecto Jupyter. Permite crear y compartir documentos web en formato JSON que siguen un esquema versionado y una lista ordenada de celdas de entrada y de salida. Estas celdas albergan, entre otras cosas,

código, texto (en formato “Markdown”), fórmulas matemáticas y ecuaciones, o también contenido multimedia (“Rich Media”). El programa se ejecuta desde la aplicación web cliente que funciona en cualquier navegador estándar.

**“Machine Learning”:** “Machine Learning” es una forma de la IA que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. Sin embargo, “machine learning” no es un proceso sencillo. Conforme el algoritmo ingiere datos de entrenamiento, es posible producir modelos más precisos basados en datos. Un modelo de “machine learning” es la salida de información que se genera cuando entrena su algoritmo de “machine learning” con datos. Después del entrenamiento, al proporcionar un modelo con una entrada, se le dará una salida.

## Referencias bibliográficas

Aprende IA. (2020). ¿Qué es el Aprendizaje no Supervisado?.

<https://aprendeia.com/aprendizaje-no-supervisado-machine-learning/>

Chacón, J. (2021). Introducción a Pandas, la librería de Python para trabajar con datos. Profile. <https://profile.es/blog/pandas-python/>

De la Fuente, Ó. (2019). Google Colab: Python y Machine Learning en la nube. Adictos al trabajo. <https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/>

IDATHA. (s. f.).05 - DS - Visualizacion de Datos - Python.ipynb.

<https://colab.research.google.com/github/efviodo/idatha-data-science-course/blob/master/notebooks/05%20-%20DS%20-%20Visualizacion%20de%20Datos%20-%20Python.ipynb#scrollTo=RQftWD90aHmb>

Kumar, A. (2020). Tensor Explained with Python NumPy Examples. Data Analytics.

Mariños, J. (2015). Listas en Python. DevCode.

<https://devcode.la/tutoriales/listas-python/>

Na8. (2019). Análisis Exploratorio de Datos con Pandas en Python. Aprende Machine Learning. <https://www.aprendemachinelearning.com/analisis-exploratorio-de-datos-pandas-python/>

Python Software Foundation. (s. f.). 5. Estructuras de datos. Python.

<https://docs.python.org/es/3/tutorial/datastructures.html#more-on-lists>

Rodríguez, D. (2020). Formatos condicionales en Pandas. Analytics Lane.

<https://www.analyticslane.com/2020/04/17/formatos-condicionales-en-pandas/>

Tableau Software. (2019). Guía de visualización de datos para principiantes: definición, ejemplos y recursos de aprendizaje. Salesforce.

<https://www.tableau.com/es-mx/learn/articles/data-visualization>

## Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal	Responsable del Ecosistema	Dirección General
Rafael Neftalí Lizcano Reyes	Responsable de Línea de Producción	Centro Industrial del Diseño y la Manufactura - Regional Santander
José Luis Bastidas Pérez	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Zvi Daniel Grosman Landáez	Diseñador Instruccional	Centro de Gestión Industrial - Regional Distrito Capital
Andrés Felipe Velandia Espitia	Asesor Metodológico	Centro de Diseño y Metrología - Regional Distrito Capital
Fabián Leonardo Correa Díaz	Diseñadora Instruccional	Centro Industrial del Diseño y la Manufactura - Regional Santander
Darío González	Corrector de estilo	Centro de Diseño y Metrología - Regional Distrito Capital
Yerson Fabian Zarate	Diseñador de Contenidos Digitales	Centro Industrial del Diseño y la Manufactura - Regional Santander
Emilsen Alfonso Bautista	Desarrollador “Fullstack”	Centro Industrial del Diseño y la Manufactura - Regional Santander
Carmen Alicia Martínez Torres	Animador y Producción audiovisual	Centro Industrial del Diseño y la Manufactura - Regional Santander
Camilo Andrés Bolaño Rey	Locución	Centro Industrial del Diseño y la Manufactura - Regional Santander
Emilsen Alfonso Bautista	Actividad Didáctica	Centro Industrial del Diseño y la Manufactura - Regional Santander
Zuleidy María Ruiz Torres	Validador de Recursos Educativos Digitales	Centro Industrial del Diseño y la Manufactura - Regional Santander

Nombre	Cargo	Regional y Centro de Formación
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro Industrial del Diseño y la Manufactura - Regional Santander
Daniel Ricardo Mutis Gómez	Evaluador para contenidos inclusivos y accesibles	Centro Industrial del Diseño y la Manufactura - Regional Santander