

**ANÁLISIS DE  
DATOS MASIVOS**

# **Informe Práctica 2**



**Universidad  
de La Laguna**

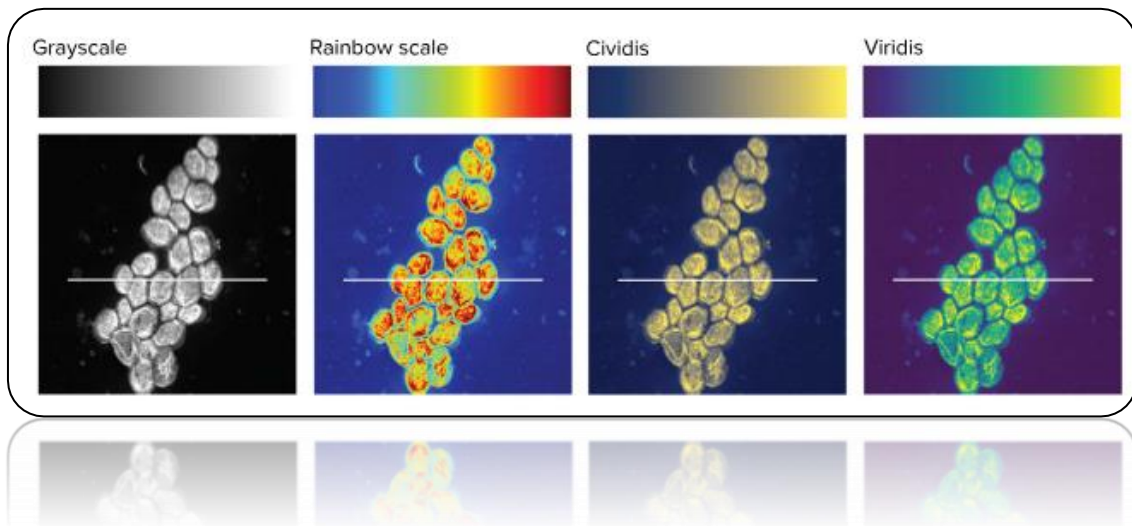
Realizado por:  
Omar Patricio Pérez Znakar.

## INDICE

1. Resumen texto propuesto por el profesor. ....	2
2. Framework de visualización. ....	3
2.1 Propuestas realizadas por el profesor. ....	3
2.1.1 Nueva capa de abstracción para cada tipo de archivo. ....	3
2.1.2 Ajuste dinámico de los ejes en función del conjunto de datos a representar. ....	4
2.1.3 Incorpora en el entorno anterior la posibilidad de comparar múltiples grupos. ....	5
2.2 Propuestas realizadas por el alumno. ....	6
2.2.1 Tipos de agrupamientos. ....	6
2.2.2 Nuevas gráficas. ....	7

## 1. Resumen texto propuesto por el profesor.

A pesar de que el color es una característica muy útil a la hora de representar datos, también es una forma de complicar el entendimiento de estos. Aunque parezca que es muy poco común, realmente es todo lo contrario. La percepción del ojo humano es mucho más complicada de lo que parece en primera instancia. Para poder representar datos existen diversas escalas, pero las más comunes son:



Como podemos ver en las escalas de la imagen anterior. El elemento de la izquierda es el elemento “menor” y el de la derecha el “mayor”. No obstante, ¿Por qué el azul es el “menor” y el rojo el “mayor”? A parte de la problemática anterior, hay diversas controversias más con respecto a este tema. Como por ejemplo, no se puede distinguir un color de la misma forma, si está rodeado de un color o de otro (contraste simultáneo). De igual forma, este tipo de gráfico puede provocar que se le desvíe la atención de lo que se quiere mostrar al lector.

Por todo lo comentado con anterioridad, opino que el autor de este documento tiene toda la razón, tanto a la hora de representar gráficas, como con el tema de los colores. En mi opinión, se debería de invertir más tiempo en el estudio de esta rama. Quien adquiera un nivel alto podría resaltar por encima de la competencia aún si su estudio no llega a estarlo. A veces, es más importante saber vender que el producto que se tiene. No obstante, no creo que se deba de focalizar en esta tarea, sino que, se llegue a un equilibrio entre la investigación y su representación.

## 2. Framework de visualización.

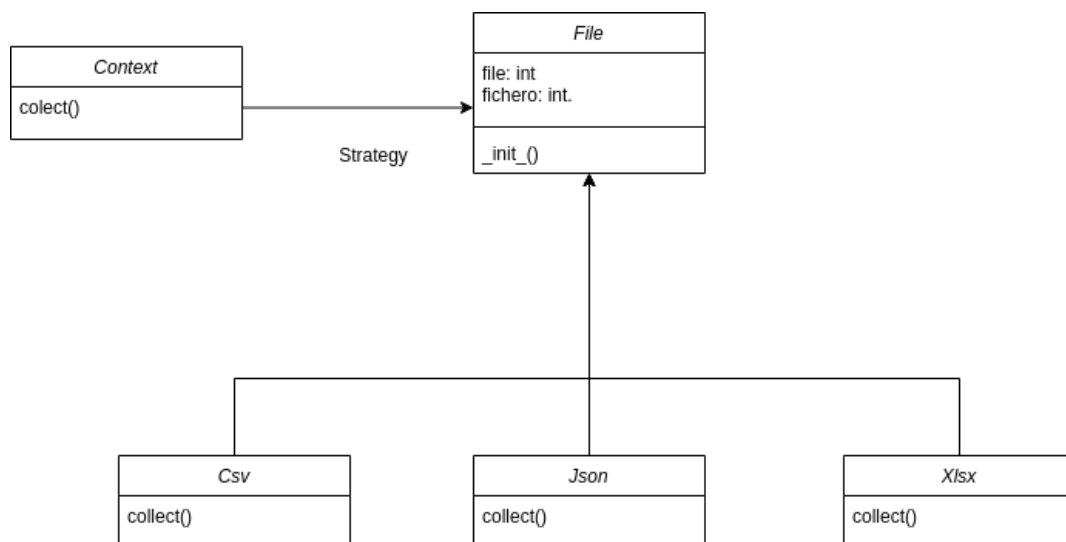
En este apartado, podremos ver cómo ha evolucionado el framework con respecto a la semana anterior. A continuación, podremos ver mejores propuestas tanto por el profesor como por el alumno.

### 2.1 Propuestas realizadas por el profesor.

En este apartado, veremos propuestas realizadas por el profesor en la corrección de la semana anterior.

#### 2.1.1 Nueva capa de abstracción para cada tipo de archivo.

La semana pasada el framework solo podría recibir archivos con formato “.csv”. Para esta semana, el profesor me propuso que pudiéramos pasar diversos formatos para darle a nuestro framework de mayor generalidad. De igual forma, me propuso que esta funcionalidad se realice a través de una nueva capa de abstracción para darle al código mayor simplicidad. Por ello, se ha realizado el siguiente esquema de clases:



Esta estructura se ha seguido a través de la filosofía de únicamente tratar los datos en formato “.xlsx”. Por tanto, para aquellos otros formatos, se ha convertido a este último a través de la librería “pandas”.

## 2.1.2 Ajuste dinámico de los ejes en función del conjunto de datos a representar.

Para este apartado, se ha propuesto poder elegir más de un elemento para un único eje. Por tanto, le daríamos a nuestro framework de mayor versatilidad a la hora de representar los datos. Para conseguir este propósito, se ha seguido los siguientes pasos:

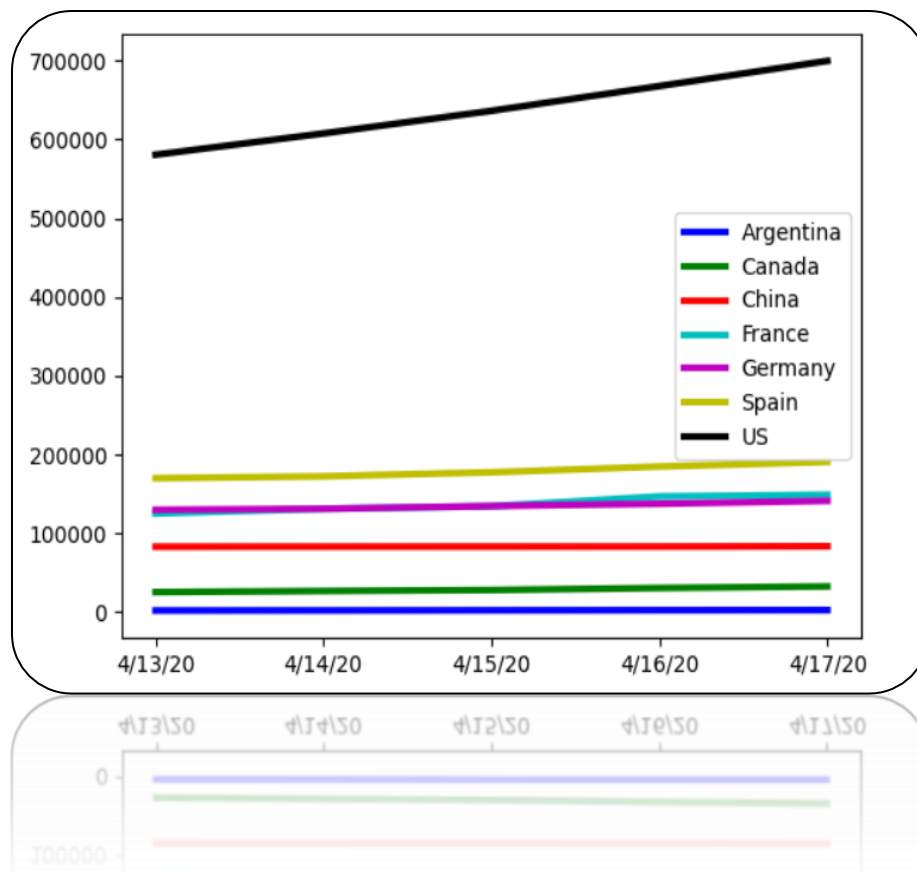
- Se le pedirá al usuario (ya sea por la terminal o por parámetro al ejecutar) un único elemento o varios separados por comas.
- Este string se pasará a un array a través de la función “split(',')” de Python.
- Se le pasará a la función de representación pertinente y este los representará.

Un ejemplo de esto, lo podremos ver a continuación:

### Terminal

```
Indique los valores del eje X a representar separados por comas.  
> 4/13/20,4/14/20,4/15/20,4/16/20,4/17/20
```

### Representación





### 2.1.3 Incorpora en el entorno anterior la posibilidad de comparar múltiples grupos.

Para este apartado, se ha propuesto la posibilidad de filtrar los elementos que se van a representar. Esto es debido a que en la mayoría de los archivos nos encontramos ante diversos elementos y estos no se diferencian en el eje X de forma clara. Para realizar esta funcionalidad, se han seguido los siguientes pasos:

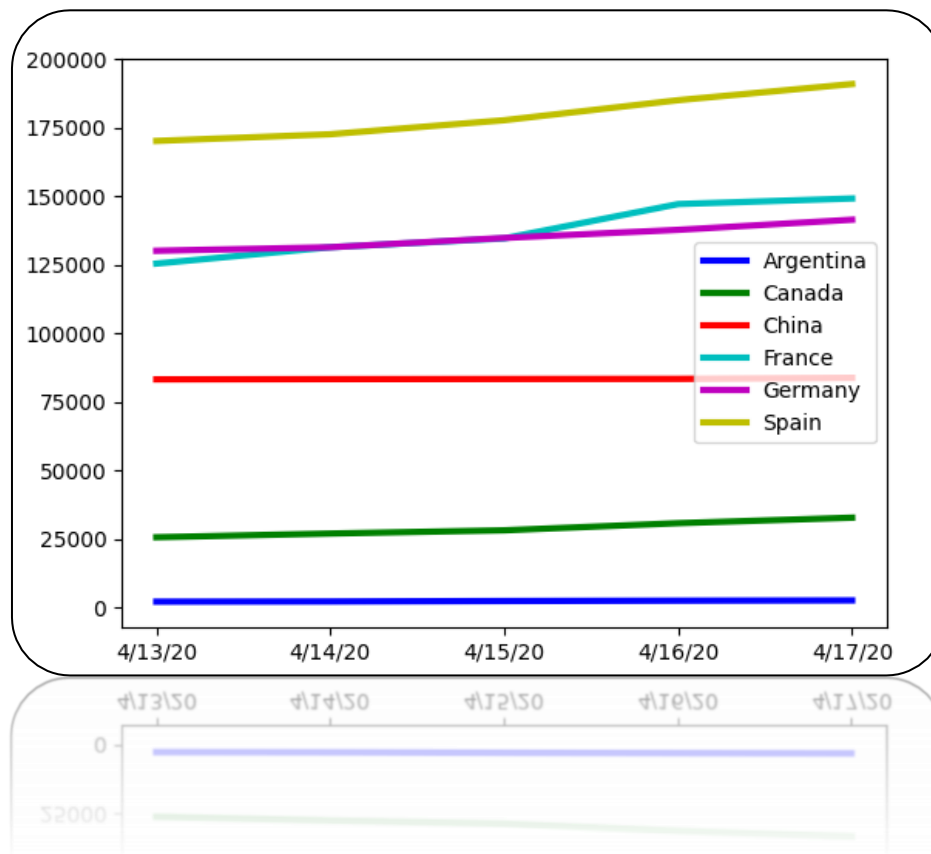
- Se le pedirá al usuario si quiere realizar un filtrado.
- Se le pedirá al usuario que columna es la que vamos a poder filtrar.
- Se le pedirá al usuario que elementos son los que se quieran representar.
- Dicha información se le pasará a las funciones de representación para que realice su función.

Un ejemplo de esto lo veremos a continuación:

#### Terminal

```
Indique los valores del eje X a representar separados por comas.  
> 4/13/20,4/14/20,4/15/20,4/16/20,4/17/20
```

#### Representación



## 2.2 Propuestas realizadas por el alumno.

### 2.2.1 Tipos de agrupamientos.

Para este apartado, se ha propuesto una mejora en la que se pueda agrupar (si el usuario quiere) elementos por un campo. Esta funcionalidad es interesante ya que en algunos casos no queremos la totalidad de estos. Sino que, queremos agrupar por un campo y en caso de colisión poder seleccionar una resolución del problema. Por ello, se han seguido los siguientes pasos:

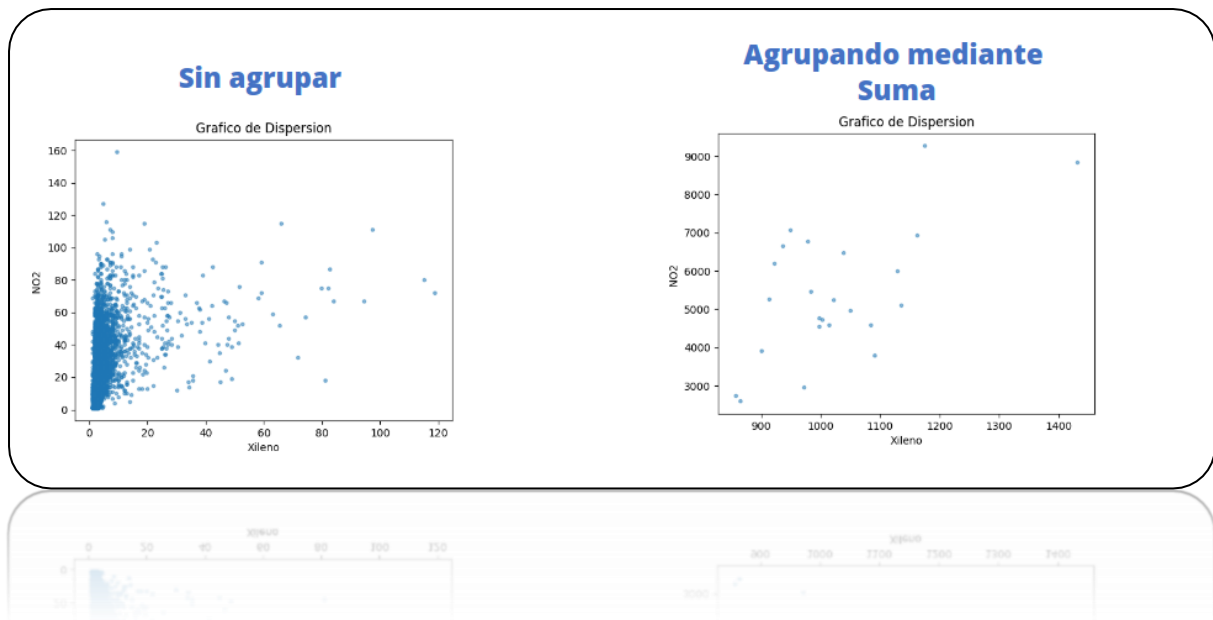
- Pedir al usuario si requiere un agrupamiento.
- En caso de que lo requiera, que dictamine el campo por el que se va a proceder a agrupar.
- Seleccionar la resolución al problema. Entre la gran cantidad de problemáticas se han seleccionado las siguientes:
  - Sumar: sumamos los elementos internos. Interesante si se tienen diversas tomas sobre una misma temática. Como por ejemplo, agrupar por país en el archivo de coronavirus por ciudad diario. En el podremos sumar todos campos para conseguir una cifra global de cada país.
  - Mínimo: cogeremos el valor más bajo en caso de colisión. Esta es una buena resolución en el caso de que se requiera coger por ejemplo el valor de temperatura más bajo de un mes.
  - Máximo: cogeremos el valor más alto en caso de colisión. Este es una buena resolución en caso de querer buscar por ejemplo la ciudad con más muertos en cada país.

Un ejemplo de esto lo podremos ver a continuación:

#### Terminal

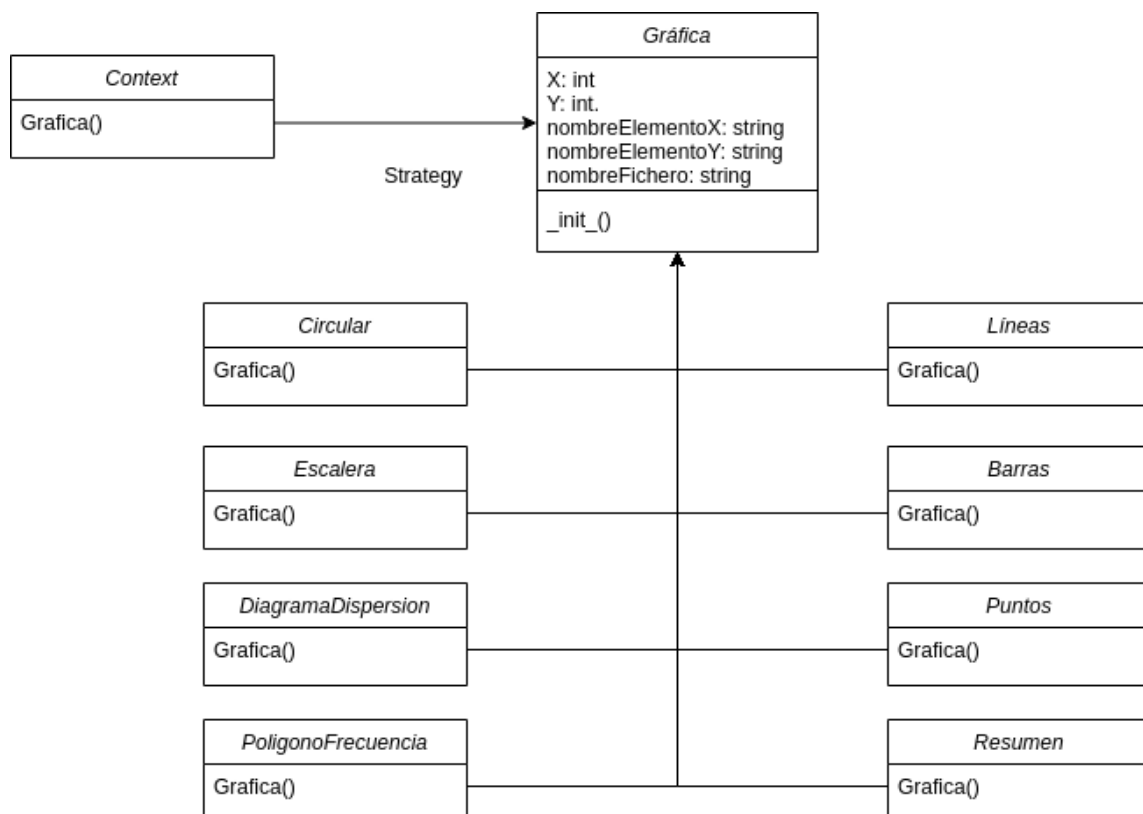
```
Indique el elemento por el que desea agrupar. Si no desea agrupar clicke enter
> 1
En caso de colisión de datos similares al agrupar ¿Que desea hacer?:
    1. Suma.
    2. Máximo.
    3. Mínimo.
    4. Ninguno.
> 1
> 1
```

## Representación



### 2.2.2 Nuevas gráficas.

En este apartado se han propuesto nuevos gráficos que nos permitirán buscar la mejor representación para nuestros datos. Por ello, se ha realizado el siguiente esquema de clases:





A continuación, podremos ver un ejemplo de cada tipo de gráfica.

Gráfica de Líneas

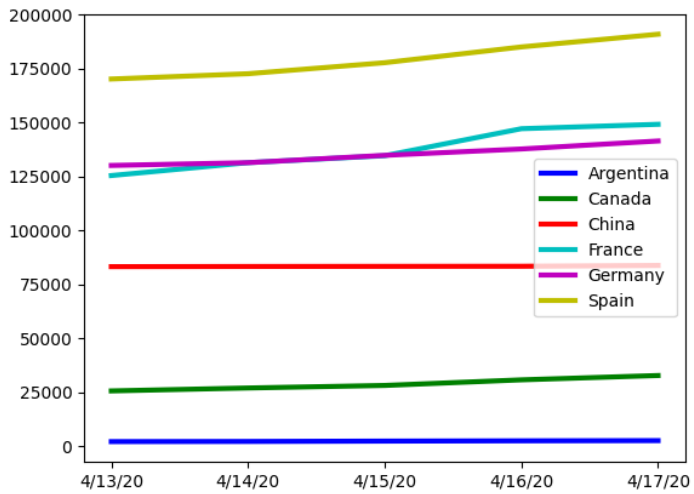


Diagrama de Barras

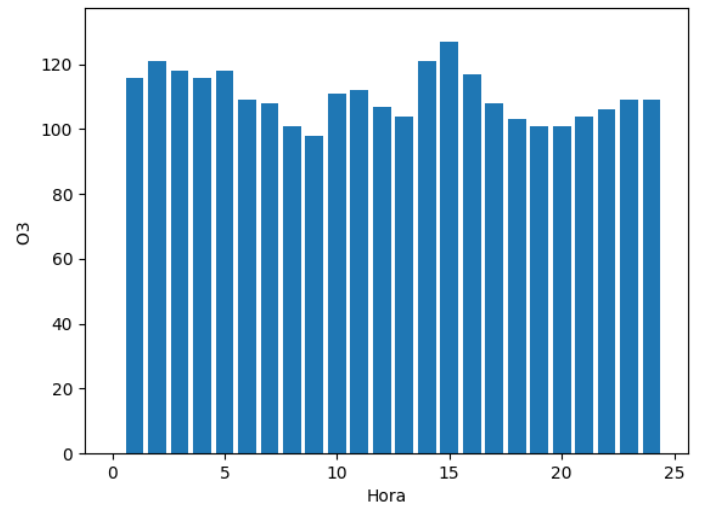


Grafico de Puntos

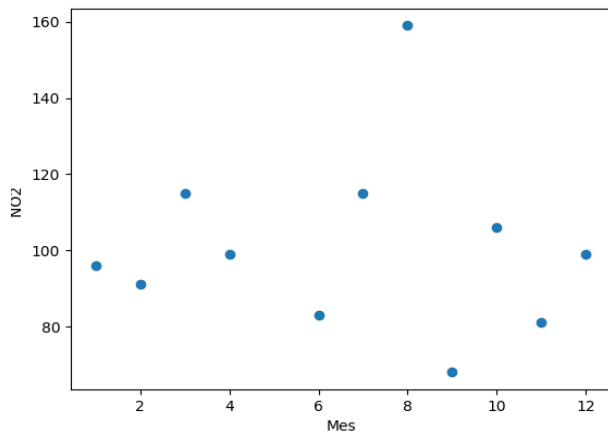


Grafico Circular

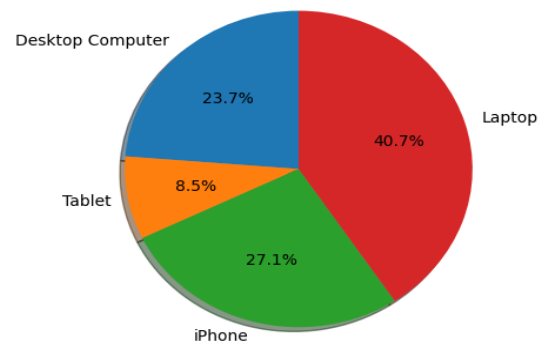


Grafico de Escaleras

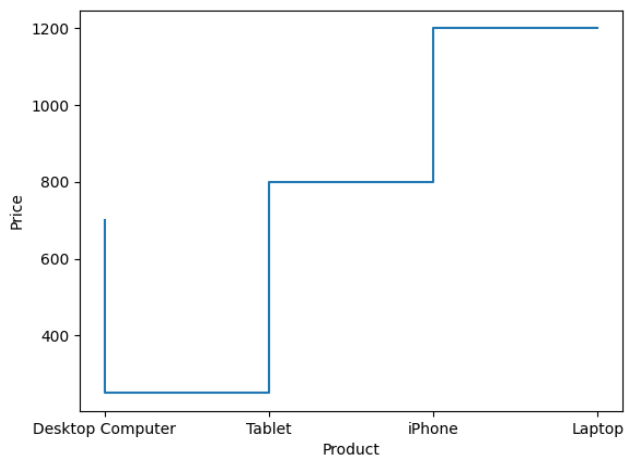
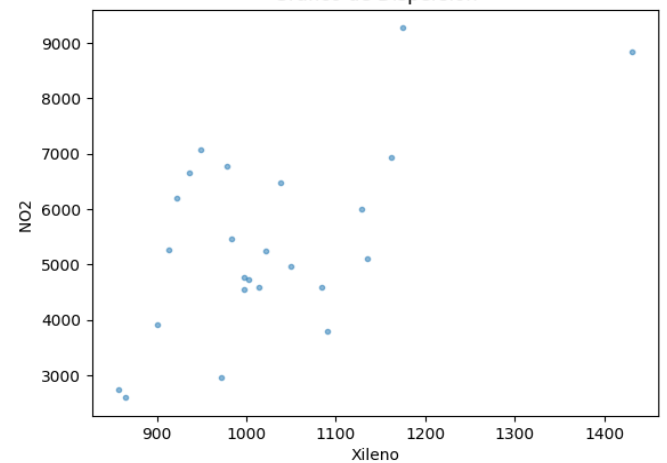
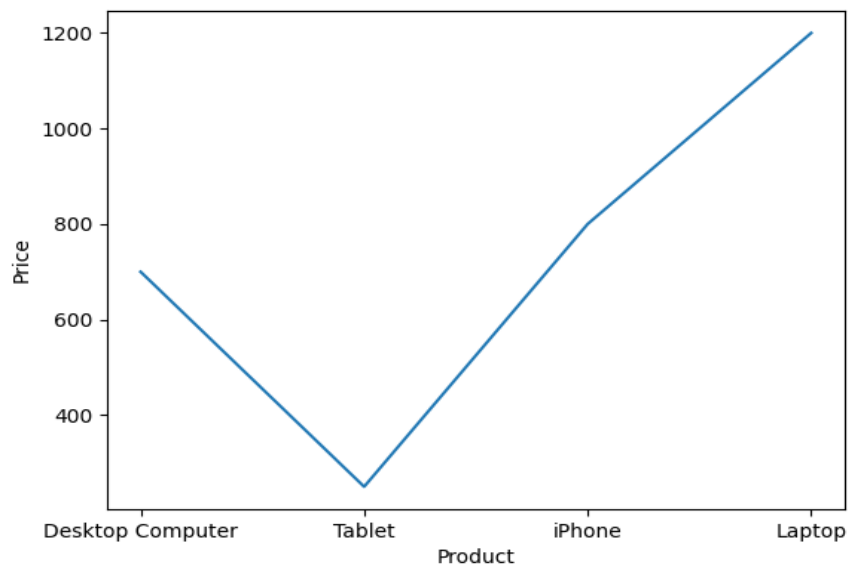


Grafico de Dispersion



Poligono de Frecuencia



Resumen

