

# **Análisis gráficas tendencias TI**

**Realizado por: Mariana Charry Prada**

**Jesús Ariel González Bonilla**

**Servicio Nacional de Aprendizaje SENA**

**Centro De La Industria, La Empresa y Los Servicios**

**Tecnólogo En Análisis y Desarrollo De Software (ADSO)**

**Elaboración de Artículos Científicos en Actividades de Investigación**

**N ° de Ficha: 2694667**

**Neiva-Huila**

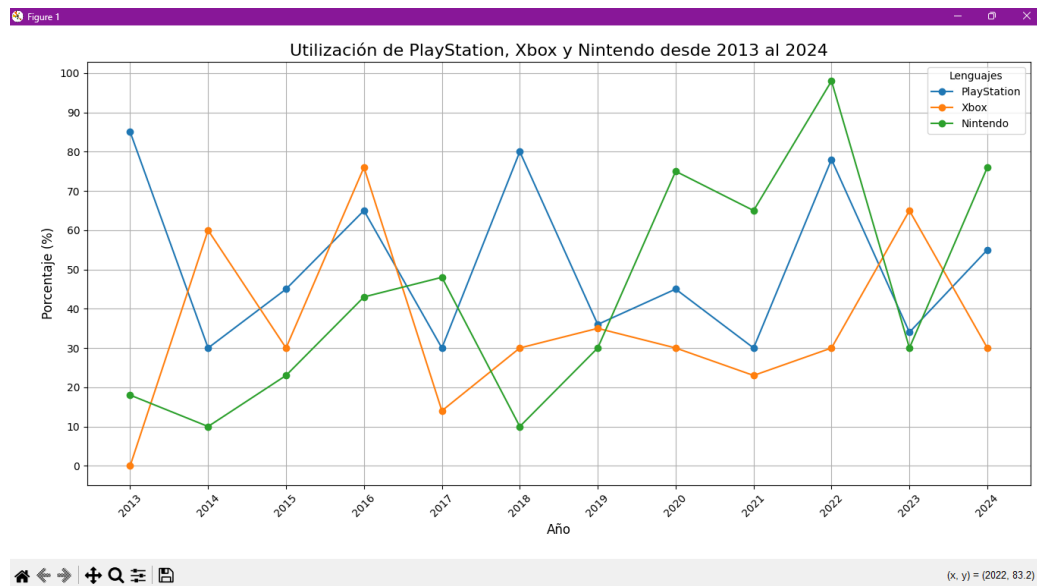
**2024**

## Página uno:

Excel con los datos a graficar (csv):

|             | A | B    | C    | D    | E    | F    | G    | H    | I    | J    | K    | L    | M    |
|-------------|---|------|------|------|------|------|------|------|------|------|------|------|------|
|             |   | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 |
| PlayStation |   | 85   | 30   | 45   | 65   | 30   | 80   | 36   | 45   | 30   | 78   | 34   | 55   |
| Xbox        |   | 0    | 60   | 30   | 76   | 14   | 30   | 35   | 30   | 23   | 30   | 65   | 30   |
| Nintendo    |   | 18   | 10   | 23   | 43   | 48   | 10   | 30   | 75   | 65   | 98   | 30   | 76   |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

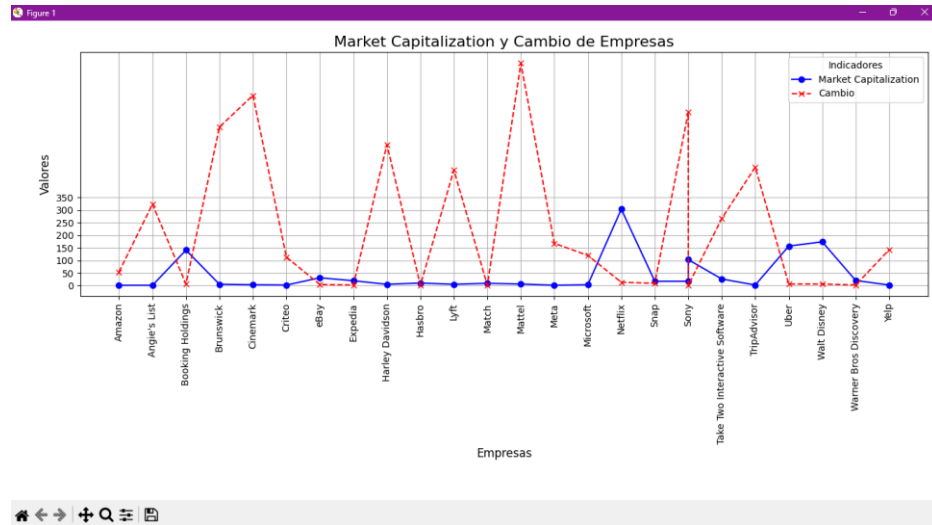
```
paginaUno.py > leer_csv
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9         # Mostrar la estructura del DataFrame para verificar
10        print("Estructura del DataFrame:")
11        print(df.head())
12
13        # Verificar que el número de filas es correcto
14        if df.shape[0] != 3:
15            raise ValueError(f"Se esperaban 4 filas, pero se encontraron {df.shape[0]} filas.")
16
17        # Asignar los nombres a las columnas, los años (del 2013 al 2024)
18        df.columns = ['Año', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022', '2023', '2024']
19
20        # Establecer la columna "Sistema Operativo" como índice
21        df.set_index('Año', inplace=True)
22
23        return df
24    except Exception as e:
25        print(f"Error al leer el archivo CSV: {e}")
26        return None
27
28 # Función para graficar los datos del CSV
29 def graficar_datos(df):
30     if df is not None:
31         # Configuración del gráfico
32         plt.figure(figsize=(10, 6))
33
34         # Graficar cada sistema operativo
35         for sistema in df.index:
36             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
37
38         # Agregar título y etiquetas
39         plt.title('Utilización de PlayStation, Xbox y Nintendo desde 2013 al 2024', fontsize=16)
40         plt.xlabel('Año', fontsize=12)
41         plt.ylabel('Porcentaje (%)', fontsize=12)
42         plt.xticks(df.columns, rotation=45)
43         plt.yticks(range(0, 101, 10))
44
45         # Agregar leyenda
46         plt.legend(title='Lenguajes')
47
48         # Mostrar gráfico
49         plt.grid(True)
50         plt.tight_layout()
51
```

## Página dos:

Enlace a la página:

<https://es.tradingeconomics.com/ttwo:us:market-capitalization>

Gráfica generada con Python:



Excel con los datos a graficar (csv):

|    | A                             | B                     | C      | D     |
|----|-------------------------------|-----------------------|--------|-------|
| 1  |                               | Market Capitalization | Cambio | Fecha |
| 2  | Amazon                        |                       | 1      | 54    |
| 3  | Angie's List                  |                       | 1      | 324   |
| 4  | Booking Holdings              |                       | 141    | 6     |
| 5  | Brunswick                     |                       | 5      | 631   |
| 6  | Cinemark                      |                       | 3      | 757   |
| 7  | Criteo                        |                       | 2      | 113   |
| 8  | eBay                          |                       | 31     | 4     |
| 9  | Expedia                       |                       | 19     | 2     |
| 10 | Harley Davidson               |                       | 5      | 561   |
| 11 | Hasbro                        |                       | 10     | 1     |
| 12 | Lyft                          |                       | 5      | 460   |
| 13 | Match                         |                       | 9      | 1     |
| 14 | Mattel                        |                       | 6      | 885   |
| 15 | Meta                          |                       | 1      | 168   |
| 16 | Microsoft                     |                       | 3      | 120   |
| 17 | Netflix                       |                       | 304    | 13    |
| 18 | Snap                          |                       | 17     | 9     |
| 19 | Sony                          |                       | 17     | 690   |
| 20 | Sony                          |                       | 103    | 1     |
| 21 | Take Two Interactive Software |                       | 26     | 267   |
| 22 | TripAdvisor                   |                       | 2      | 471   |
| 23 | Uber                          |                       | 157    | 6     |
| 24 | Walt Disney                   |                       | 174    | 6     |
| 25 | Warner Bros Discovery         |                       | 20     | 2     |
| 26 | Yelp                          |                       | 2      | 143   |

Código en Python para realizar la gráfica:

```
paginaDos.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 25:
16            raise ValueError(f"Se esperaban 25 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas
19        df.columns = ['Campos', 'Market Capitalization', 'Cambio', 'Fecha']
20
21        # Establecer la columna "Campos" como índice
22        df.set_index('Campos', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar Market Capitalization y Cambio de cada empresa
36         # Gráfico de Market Capitalization
37         plt.plot(df.index, df['Market Capitalization'], label='Market Capitalization', marker='o', linestyle='-', color='b')
38
39         # Gráfico de Cambio
40         plt.plot(df.index, df['Cambio'], label='Cambio', marker='x', linestyle='-', color='r')
41
42         # Agregar título y etiquetas
43         plt.title('Market Capitalization y Cambio de Empresas', fontsize=16)
44         plt.xlabel('Empresas', fontsize=12)
45         plt.ylabel('Valores', fontsize=12)
46         plt.xticks(rotation=45) # Rotar las etiquetas del eje X para que sean legibles
47         plt.yticks(range(0, 351, 50)) # Ajustar el rango de los valores en el eje Y
48
49         # Agregar leyenda
50         plt.legend(title='Indicadores')
51
```

## Página tres:

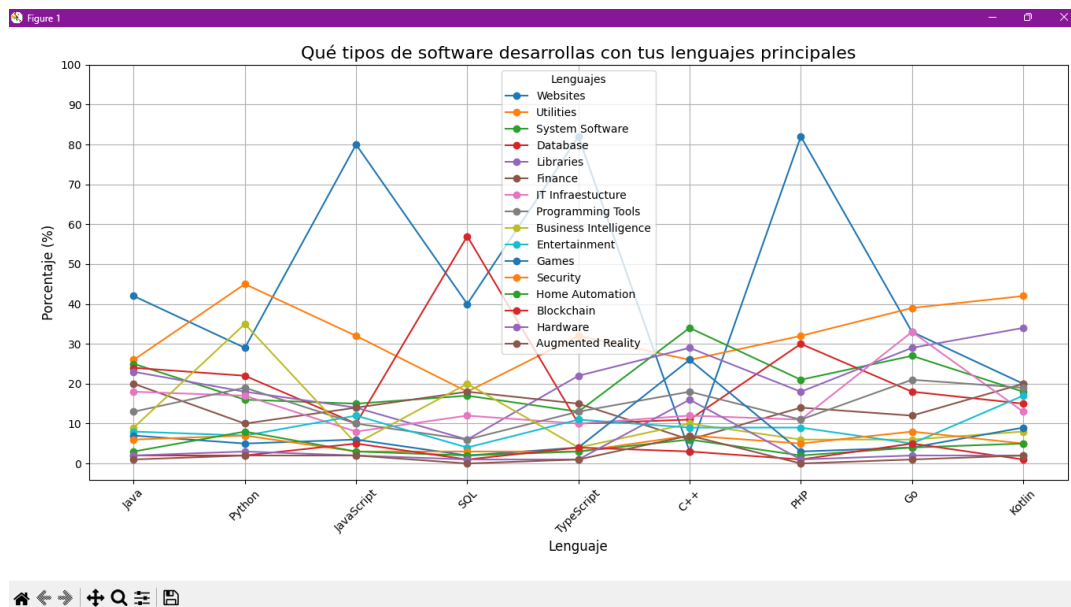
Enlace a la página:

<https://www.jetbrains.com/lp/devecosystem-2023/languages/>

Excel con los datos a graficar (csv):

|    | A                     | B    | C      | D          | E   | F          | G   | H   | I  | J      |    |
|----|-----------------------|------|--------|------------|-----|------------|-----|-----|----|--------|----|
| 1  |                       | Java | Python | JavaScript | SQL | TypeScript | C++ | PHP | Go | Kotlin |    |
| 2  | Websites              |      | 42     | 29         | 80  | 40         | 82  | 3   | 82 | 33     | 20 |
| 3  | Utilities             |      | 26     | 45         | 32  | 18         | 32  | 26  | 32 | 39     | 42 |
| 4  | System Software       |      | 25     | 16         | 15  | 17         | 13  | 34  | 21 | 27     | 18 |
| 5  | Database              |      | 24     | 22         | 10  | 57         | 10  | 11  | 30 | 18     | 15 |
| 6  | Libraries             |      | 23     | 18         | 14  | 6          | 22  | 29  | 18 | 29     | 34 |
| 7  | Finance               |      | 20     | 10         | 14  | 18         | 15  | 6   | 14 | 12     | 20 |
| 8  | IT Infraestructure    |      | 18     | 17         | 8   | 12         | 10  | 12  | 11 | 33     | 13 |
| 9  | Programming Tools     |      | 13     | 19         | 10  | 6          | 13  | 18  | 11 | 21     | 19 |
| 10 | Business Intelligence |      | 9      | 35         | 5   | 20         | 4   | 10  | 6  | 6      | 8  |
| 11 | Entertainment         |      | 8      | 7          | 12  | 4          | 11  | 9   | 9  | 5      | 17 |
| 12 | Games                 |      | 7      | 5          | 6   | 2          | 4   | 26  | 3  | 4      | 9  |
| 13 | Security              |      | 6      | 7          | 3   | 3          | 3   | 7   | 5  | 8      | 5  |
| 14 | Home Automation       |      | 3      | 8          | 3   | 2          | 3   | 6   | 2  | 4      | 5  |
| 15 | Blockchain            |      | 2      | 2          | 5   | 1          | 4   | 3   | 1  | 5      | 1  |
| 16 | Hardware              |      | 2      | 3          | 2   | 1          | 1   | 16  | 1  | 2      | 2  |
| 17 | Augmented Reality     |      | 1      | 2          | 2   | 0          | 1   | 7   | 0  | 1      | 2  |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

```
paginaTres.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 16:
16            raise ValueError(f"Se esperaban 4 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Languages', 'Java', 'Python', 'JavaScript', 'SQL', 'TypeScript', 'C++', 'PHP', 'Go', 'Kotlin']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Languages', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Qué tipos de software desarrollas con tus lenguajes principales', fontsize=16)
41         plt.xlabel('Lenguaje', fontsize=12)
42         plt.ylabel('Porcentaje (%)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Lenguajes')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()
```

## Página cuatro:

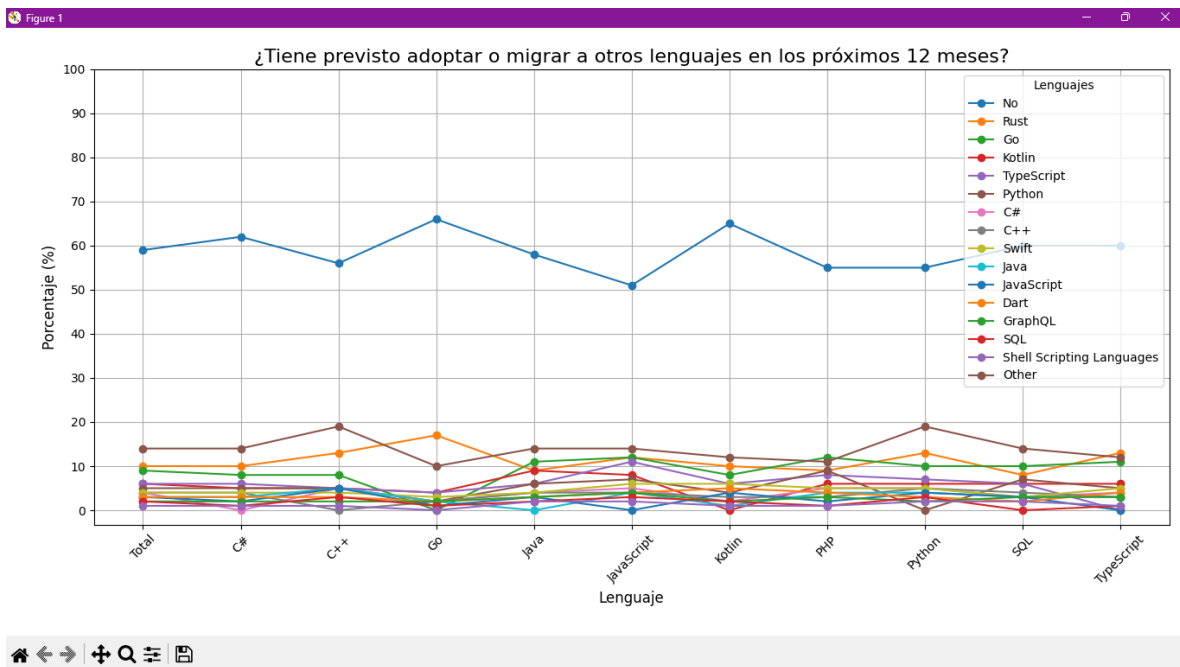
Enlace a la página:

<https://www.jetbrains.com/lp/devecosystem-2023/languages/>

Excel con los datos a graficar (csv):

|    | A                         | B     | C  | D   | E  | F    | G          | H      | I   | J      | K   | L          |
|----|---------------------------|-------|----|-----|----|------|------------|--------|-----|--------|-----|------------|
| 1  |                           | Total | C# | C++ | Go | Java | JavaScript | Kotlin | PHP | Python | SQL | TypeScript |
| 2  | No                        | 59    | 62 | 56  | 66 | 58   | 51         | 65     | 55  | 55     | 60  | 60         |
| 3  | Rust                      | 10    | 10 | 13  | 17 | 9    | 12         | 10     | 9   | 13     | 8   | 13         |
| 4  | Go                        | 9     | 8  | 8   | 0  | 11   | 12         | 8      | 12  | 10     | 10  | 11         |
| 5  | Kotlin                    | 6     | 5  | 5   | 4  | 9    | 8          | 0      | 6   | 6      | 6   | 6          |
| 6  | TypeScript                | 6     | 6  | 5   | 4  | 6    | 11         | 6      | 8   | 7      | 6   | 0          |
| 7  | Python                    | 5     | 5  | 5   | 2  | 6    | 7          | 4      | 9   | 0      | 7   | 5          |
| 8  | C#                        | 4     | 0  | 5   | 2  | 4    | 5          | 2      | 5   | 5      | 3   | 4          |
| 9  | C++                       | 4     | 4  | 0   | 2  | 4    | 4          | 3      | 3   | 5      | 4   | 3          |
| 10 | Swift                     | 4     | 4  | 4   | 3  | 4    | 6          | 6      | 5   | 5      | 3   | 5          |
| 11 | Java                      | 3     | 3  | 5   | 2  | 0    | 4          | 1      | 4   | 4      | 3   | 3          |
| 12 | JavaScript                | 3     | 2  | 5   | 1  | 3    | 0          | 4      | 2   | 4      | 3   | 0          |
| 13 | Dart                      | 3     | 3  | 3   | 2  | 3    | 4          | 5      | 4   | 3      | 2   | 4          |
| 14 | GraphQL                   | 2     | 2  | 2   | 2  | 3    | 4          | 2      | 3   | 2      | 3   | 3          |
| 15 | SQL                       | 2     | 1  | 3   | 1  | 2    | 3          | 2      | 1   | 3      | 0   | 1          |
| 16 | Shell Scripting Languages | 1     | 1  | 1   | 0  | 2    | 2          | 1      | 1   | 2      | 2   | 1          |
| 17 | Other                     | 14    | 14 | 19  | 10 | 14   | 14         | 12     | 11  | 19     | 14  | 12         |
| 18 |                           |       |    |     |    |      |            |        |     |        |     |            |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

```
paginaCuatro.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 16:
16            raise ValueError(f"Se esperaban 16 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Lenguajes', 'Total', 'C#', 'C++', 'Go', 'Java', 'JavaScript', 'Kotlin', 'PHP', 'Python', 'SQL', 'TypeScript']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Lenguajes', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('¿Tiene previsto adoptar o migrar a otros lenguajes en los próximos 12 meses?', fontsize=16)
41         plt.xlabel('Lenguaje', fontsize=12)
42         plt.ylabel('Porcentaje (%)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Lenguajes')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()
```

## Página cinco:

Enlace a la página:

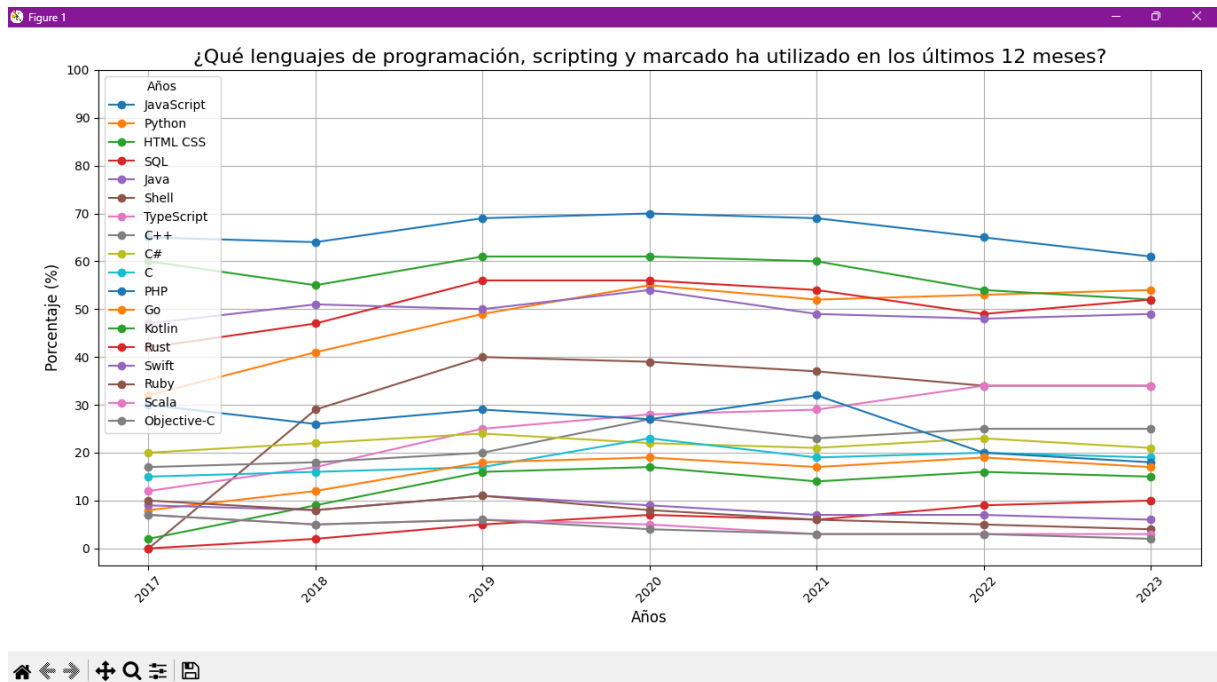
<https://www.jetbrains.com/lp/devecosystem-2023/languages/>

Excel con los datos a graficar (csv):

|    | A           | B    | C    | D    | E    | F    | G    | H    |
|----|-------------|------|------|------|------|------|------|------|
| 1  |             | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 |
| 2  | JavaScript  | 65   | 64   | 69   | 70   | 69   | 65   | 61   |
| 3  | Python      | 32   | 41   | 49   | 55   | 52   | 53   | 54   |
| 4  | HTML CSS    | 60   | 55   | 61   | 61   | 60   | 54   | 52   |
| 5  | SQL         | 42   | 47   | 56   | 56   | 54   | 49   | 52   |
| 6  | Java        | 47   | 51   | 50   | 54   | 49   | 48   | 49   |
| 7  | Shell       | 0    | 29   | 40   | 39   | 37   | 34   | 34   |
| 8  | TypeScript  | 12   | 17   | 25   | 28   | 29   | 34   | 34   |
| 9  | C++         | 17   | 18   | 20   | 27   | 23   | 25   | 25   |
| 10 | C#          | 20   | 22   | 24   | 22   | 21   | 23   | 21   |
| 11 | C           | 15   | 16   | 17   | 23   | 19   | 20   | 19   |
| 12 | PHP         | 30   | 26   | 29   | 27   | 32   | 20   | 18   |
| 13 | Go          | 8    | 12   | 18   | 19   | 17   | 19   | 17   |
| 14 | Kotlin      | 2    | 9    | 16   | 17   | 14   | 16   | 15   |
| 15 | Rust        | 0    | 2    | 5    | 7    | 6    | 9    | 10   |
| 16 | Swift       | 9    | 8    | 11   | 9    | 7    | 7    | 6    |
| 17 | Ruby        | 10   | 8    | 11   | 8    | 6    | 5    | 4    |
| 18 | Scala       | 7    | 5    | 6    | 5    | 3    | 3    | 3    |
| 19 | Objective-C | 7    | 5    | 6    | 4    | 3    | 3    | 2    |
| 20 |             |      |      |      |      |      |      |      |

Gráfica generada con Python:





Código en Python para realizar la gráfica:

```

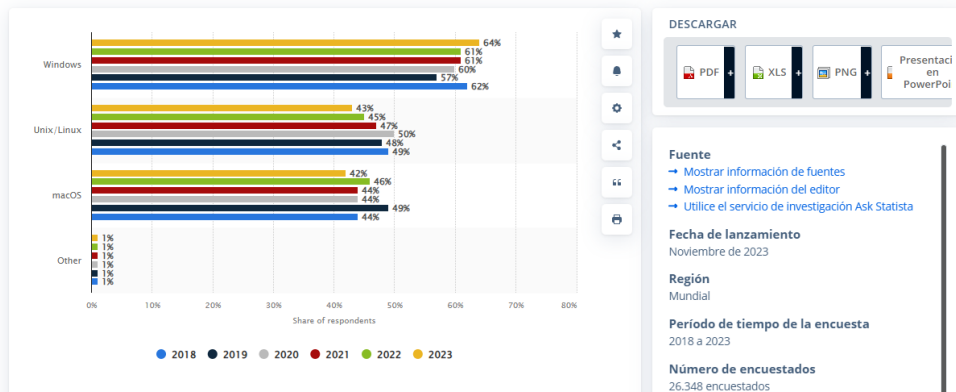
paginaCinco.py > graficar_datos
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Función para leer el archivo CSV y crear el DataFrame
5  def leer_csv(archivo_csv):
6      try:
7          # Leer el archivo CSV usando punto y coma como delimitador
8          df = pd.read_csv(archivo_csv, delimiter=";")
9
10         # Mostrar la estructura del DataFrame para verificar
11         print("Estructura del DataFrame:")
12         print(df.head())
13
14         # Verificar que el número de filas es correcto
15         if df.shape[0] != 18:
16             raise ValueError(f"Se esperaban 18 filas, pero se encontraron {df.shape[0]} filas.")
17
18         # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19         df.columns = ['Años', '2017', '2018', '2019', '2020', '2021', '2022', '2023']
20
21         # Establecer la columna "Sistema Operativo" como índice
22         df.set_index('Años', inplace=True)
23
24         return df
25     except Exception as e:
26         print(f"Error al leer el archivo CSV: {e}")
27         return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('¿Qué lenguajes de programación, scripting y marcado ha utilizado en los últimos 12 meses?', fontsize=16)
41         plt.xlabel('Años', fontsize=12)
42         plt.ylabel('Porcentaje (%)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Años')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()

```

**Página seis:**

Página consultada:

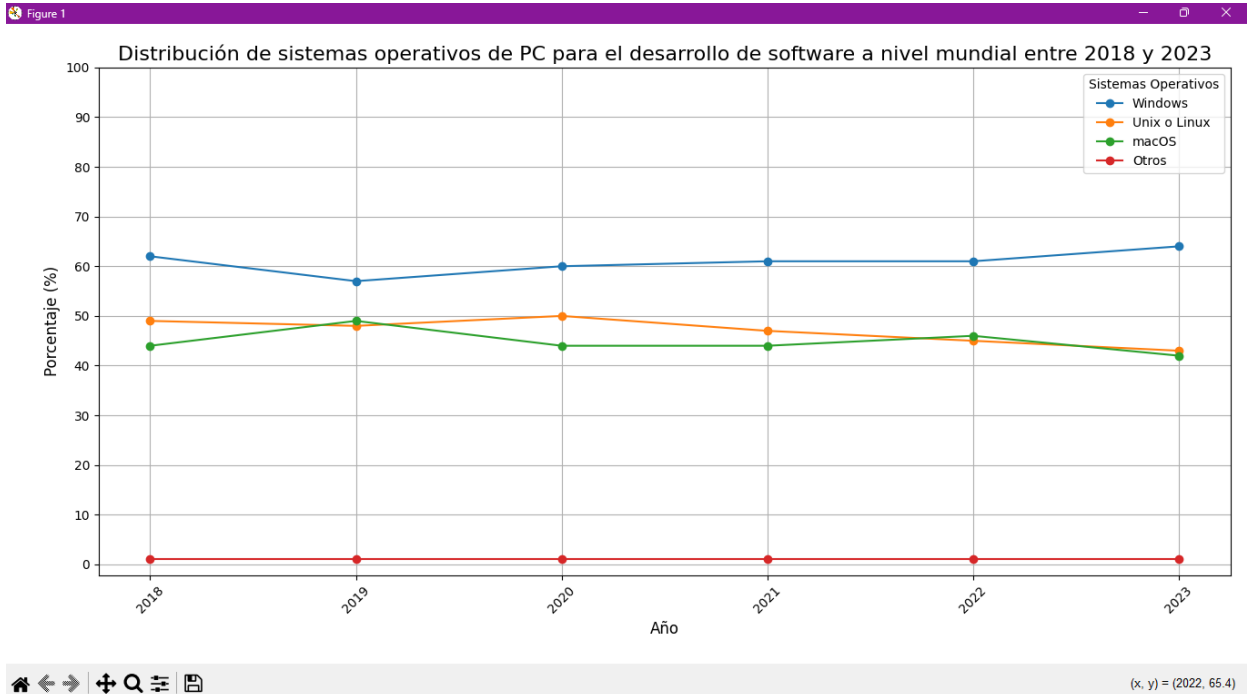
## Distribución de sistemas operativos de PC para el desarrollo de software a nivel mundial entre 2018 y 2023



Excel con los datos a graficar (csv):

|   | A            | B    | C    | D    | E    | F    | G    |
|---|--------------|------|------|------|------|------|------|
| 1 |              | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 |
| 2 | Windows      | 62   | 57   | 60   | 61   | 61   | 64   |
| 3 | Unix o Linux | 49   | 48   | 50   | 47   | 45   | 43   |
| 4 | macOS        | 44   | 49   | 44   | 44   | 46   | 42   |
| 5 | Otros        | 1    | 1    | 1    | 1    | 1    | 1    |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

```

paginaSeis.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 4:
16            raise ValueError(f"Se esperaban 4 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Sistema Operativo', '2018', '2019', '2020', '2021', '2022', '2023']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Sistema Operativo', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Distribución de sistemas operativos de PC para el desarrollo de software a nivel mundial entre 2018 y 2023', fontsize=16)
41         plt.xlabel('Año', fontsize=12)
42         plt.ylabel('Porcentaje (%)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Sistemas Operativos')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()

```

**Página siete:**

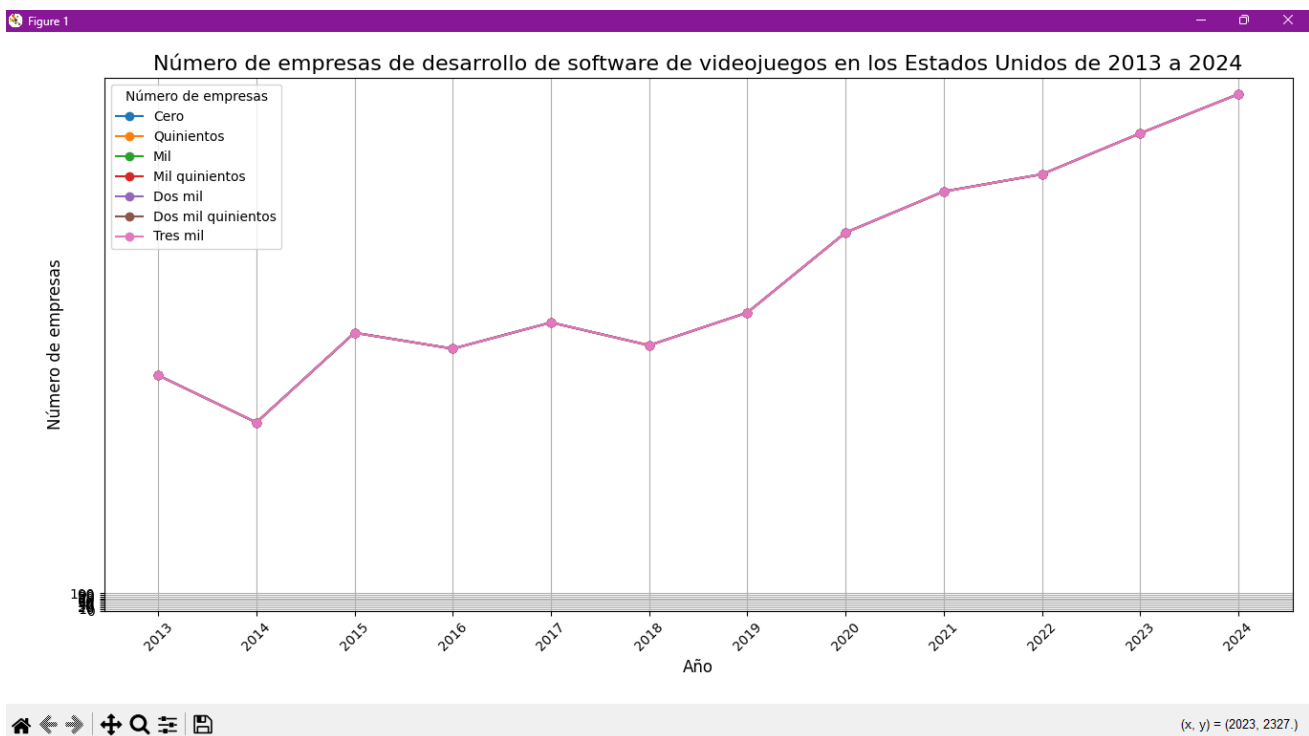
Enlace a la página:

<https://es.statista.com/estadisticas/1320523/numero-de-negocios-de-desarrollo-de-software-de-juegos-en-los-ee-uu/>

Excel con los datos a graficar (csv):

|   | A                  | B    | C    | D    | E    | F    | G    | H    | I    | J    | K    | L    | M    |
|---|--------------------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 |                    | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 |
| 2 | Cero               | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 3 | Quinientos         | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 4 | Mil                | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 5 | Mil quinientos     | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 6 | Dos mil            | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 7 | Dos mil quinientos | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |
| 8 | Tres mil           | 1328 | 1062 | 1568 | 1478 | 1626 | 1497 | 1682 | 2133 | 2365 | 2462 | 2693 | 2914 |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

```

paginaSiete.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 7:
16            raise ValueError(f"Se esperaban 7 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Número de empresas', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022', '2023', '2024']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Número de empresas', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Número de empresas de desarrollo de software de videojuegos en los Estados Unidos de 2013 a 2024', fontsize=16)
41         plt.xlabel('Año', fontsize=12)
42         plt.ylabel('Número de empresas', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Número de empresas')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()

```

## Página ocho:

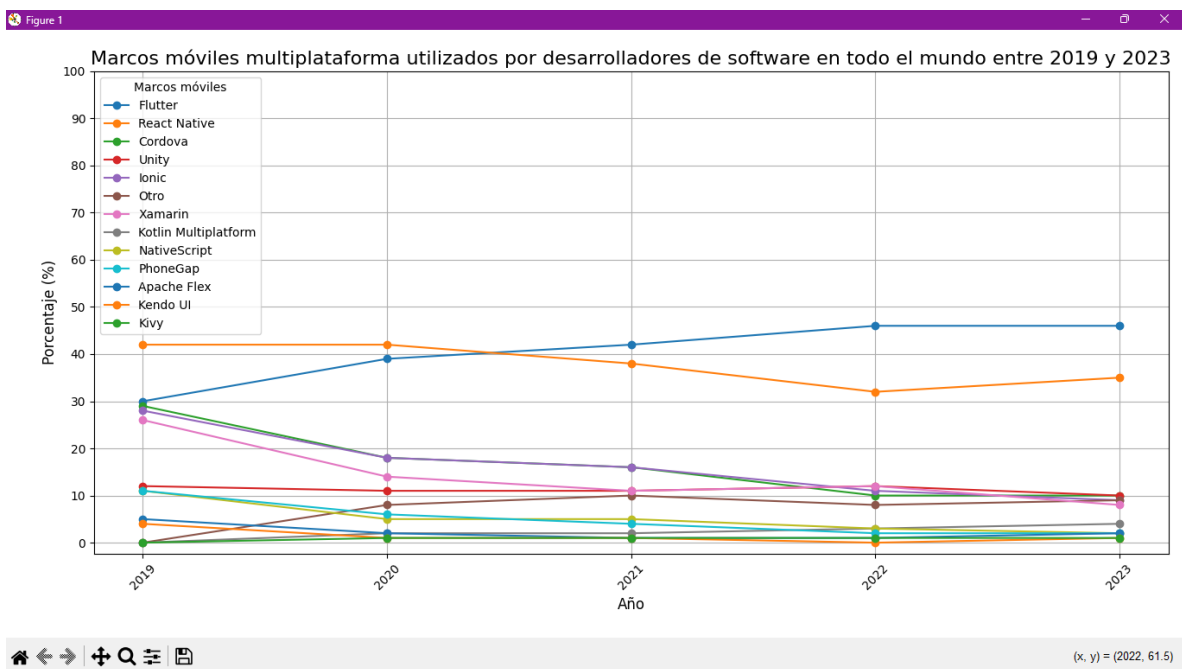
Enlace a la página:

<https://proxy.parisjc.edu:8293/statistics/869224/worldwide-software-developer-working-hours/>

Excel con los datos a graficar (csv):

|    | A                    | B    | C    | D    | E    | F    |
|----|----------------------|------|------|------|------|------|
| 1  |                      | 2019 | 2020 | 2021 | 2022 | 2023 |
| 2  | Flutter              | 30   | 39   | 42   | 46   | 46   |
| 3  | React Native         | 42   | 42   | 38   | 32   | 35   |
| 4  | Cordova              | 29   | 18   | 16   | 10   | 10   |
| 5  | Unity                | 12   | 11   | 11   | 12   | 10   |
| 6  | Ionic                | 28   | 18   | 16   | 11   | 9    |
| 7  | Otro                 | 0    | 8    | 10   | 8    | 9    |
| 8  | Xamarin              | 26   | 14   | 11   | 12   | 8    |
| 9  | Kotlin Multiplatform | 0    | 2    | 2    | 3    | 4    |
| 10 | NativeScript         | 11   | 5    | 5    | 3    | 2    |
| 11 | PhoneGap             | 11   | 6    | 4    | 2    | 2    |
| 12 | Apache Flex          | 5    | 2    | 1    | 1    | 2    |
| 13 | Kendo UI             | 4    | 1    | 1    | 0    | 1    |
| 14 | Kivy                 | 0    | 1    | 1    | 1    | 1    |
| 15 |                      |      |      |      |      |      |

Gráfica generada en Python:



Código en Python para realizar la gráfica:

```

paginaOcho.py > graficar_datos
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 13:
16            raise ValueError(f"Se esperaban 13 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Marcos móviles', '2019', '2020', '2021', '2022', '2023']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Marcos móviles', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Marcos móviles multiplataforma utilizados por desarrolladores de software en todo el mundo entre 2019 y 2023', fontsize=16)
41         plt.xlabel('Año', fontsize=12)
42         plt.ylabel('Porcentaje (%)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Marcos móviles')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()

```

**Página nueve:**

Enlace a la página:

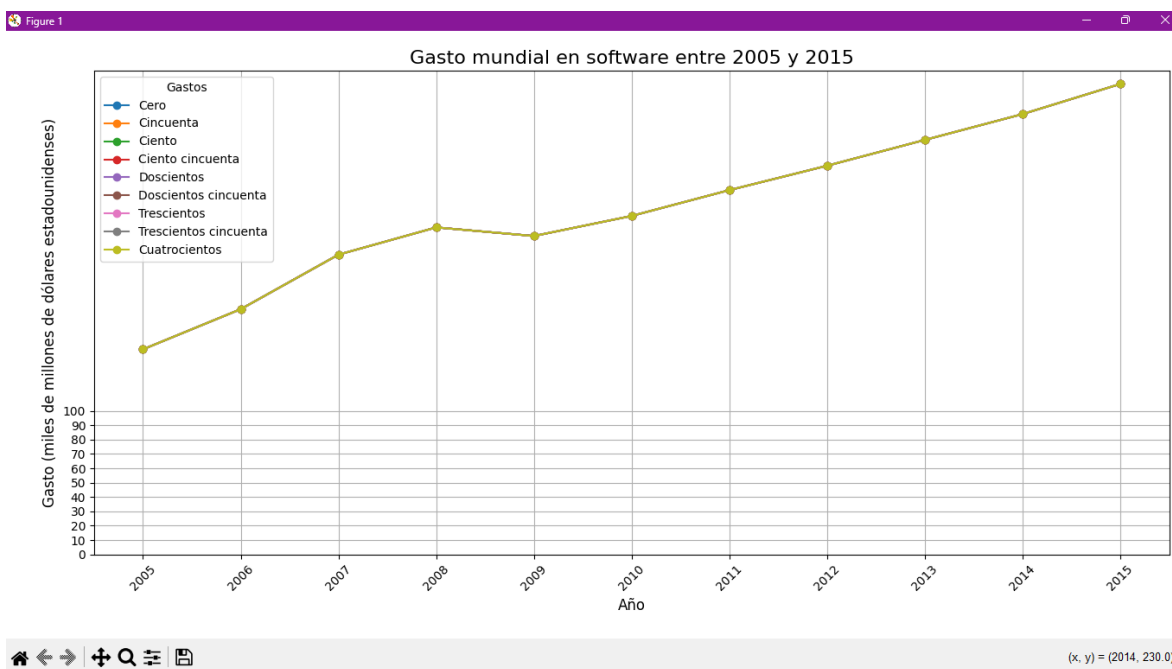


<https://www.statista.com/statistics/203964/global-software-spending-forecast/>

Excel con los datos a graficar (csv):

|    | A   | B   | C    | D    | E    | F    | G    | H    | I    | J    | K    | L    |      |
|----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|
| 1  |     |     | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
| 2  | 0   | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 3  | 50  | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 4  | 100 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 5  | 150 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 6  | 200 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 7  | 250 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 8  | 300 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 9  | 350 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 10 | 400 | 143 | 171  | 209  | 228  | 222  | 236  | 254  | 271  | 289  | 307  | 328  |      |
| 11 |     |     |      |      |      |      |      |      |      |      |      |      |      |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

```

paginaNueve.py > graficar_datos
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Función para leer el archivo CSV y crear el DataFrame
5  def leer_csv(archivo_csv):
6      try:
7          # Leer el archivo CSV usando punto y coma como delimitador
8          df = pd.read_csv(archivo_csv, delimiter=";")
9
10         # Mostrar la estructura del DataFrame para verificar
11         print("Estructura del DataFrame:")
12         print(df.head())
13
14         # Verificar que el número de filas es correcto
15         if df.shape[0] != 9:
16             raise ValueError(f"Se esperaban 9 filas, pero se encontraron {df.shape[0]} filas.")
17
18         # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19         df.columns = ['Gastos', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015']
20
21         # Establecer la columna "Sistema Operativo" como índice
22         df.set_index('Gastos', inplace=True)
23
24         return df
25     except Exception as e:
26         print(f"Error al leer el archivo CSV: {e}")
27         return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Gasto mundial en software entre 2005 y 2015', fontsize=16)
41         plt.xlabel('Año', fontsize=12)
42         plt.ylabel('Gasto (miles de millones de dólares estadounidenses)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Gastos')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()

```

**Página diez:**

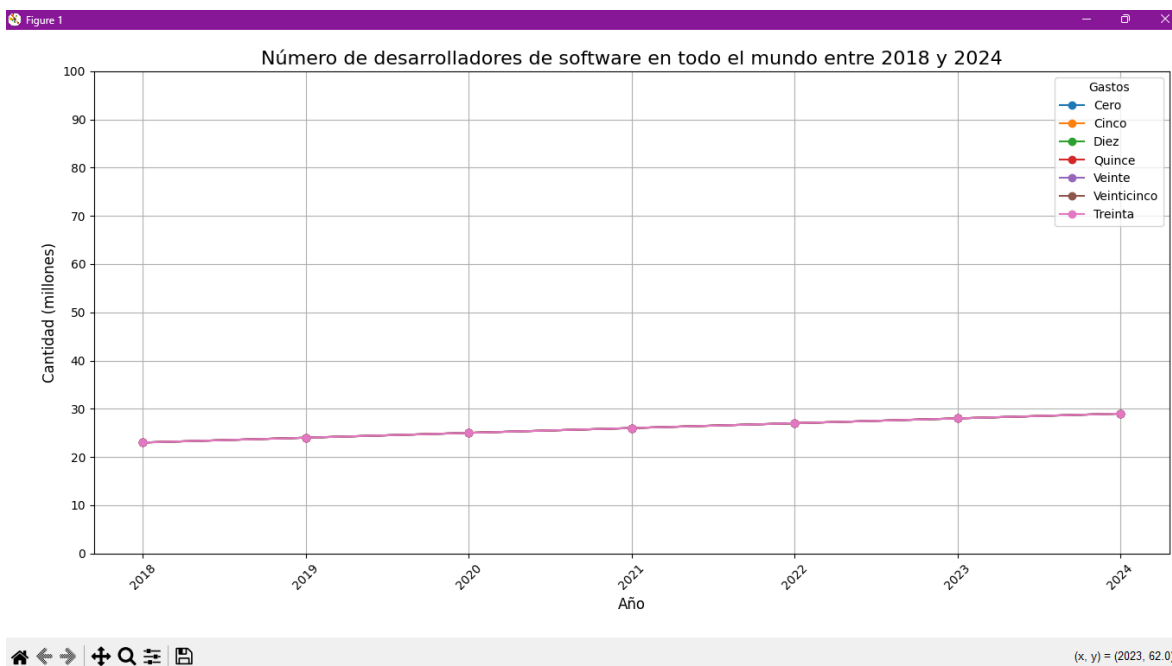
Enlace a la página:

<https://www.statista.com/statistics/627312/worldwide-developer-population/>

Excel con los datos a graficar (csv):

|   | A           | B    | C    | D    | E    | F    | G    | H    |
|---|-------------|------|------|------|------|------|------|------|
| 1 |             | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 |
| 2 | Cero        | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 3 | Cinco       | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 4 | Diez        | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 5 | Quince      | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 6 | Veinte      | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 7 | Veinticinco | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 8 | Treinta     | 23   | 24   | 25   | 26   | 27   | 28   | 29   |
| 9 |             |      |      |      |      |      |      |      |

Gráfica generada con Python:



Código en Python para realizar la gráfica:

paginaDiez.py > graficar\_datos

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Función para leer el archivo CSV y crear el DataFrame
5 def leer_csv(archivo_csv):
6     try:
7         # Leer el archivo CSV usando punto y coma como delimitador
8         df = pd.read_csv(archivo_csv, delimiter=";")
9
10        # Mostrar la estructura del DataFrame para verificar
11        print("Estructura del DataFrame:")
12        print(df.head())
13
14        # Verificar que el número de filas es correcto
15        if df.shape[0] != 7:
16            raise ValueError(f"Se esperaban 7 filas, pero se encontraron {df.shape[0]} filas.")
17
18        # Asignar los nombres a las columnas, los años (del 2018 al 2023)
19        df.columns = ['Cantidad', '2018', '2019', '2020', '2021', '2022', '2023', '2024']
20
21        # Establecer la columna "Sistema Operativo" como índice
22        df.set_index('Cantidad', inplace=True)
23
24        return df
25    except Exception as e:
26        print(f"Error al leer el archivo CSV: {e}")
27        return None
28
29 # Función para graficar los datos del CSV
30 def graficar_datos(df):
31     if df is not None:
32         # Configuración del gráfico
33         plt.figure(figsize=(10, 6))
34
35         # Graficar cada sistema operativo
36         for sistema in df.index:
37             plt.plot(df.columns, df.loc[sistema], label=sistema, marker='o')
38
39         # Agregar título y etiquetas
40         plt.title('Número de desarrolladores de software en todo el mundo entre 2018 y 2024', fontsize=16)
41         plt.xlabel('Año', fontsize=12)
42         plt.ylabel('Cantidad (millones)', fontsize=12)
43         plt.xticks(df.columns, rotation=45)
44         plt.yticks(range(0, 101, 10))
45
46         # Agregar leyenda
47         plt.legend(title='Gastos')
48
49         # Mostrar gráfico
50         plt.grid(True)
51         plt.tight_layout()
```