

Proyecto Integrador

Equipo 1

Jennifer Avendaño Sánchez (A01656951)

Andrea Samantha Aguilar Ramírez (A01656200)

Andrés Eugenio Martínez Sánchez (A01656442)

Profesor: David Christopher Balderas Silva

Grupo: 106

Tecnológico de Monterrey

Escuela de Ingeniería y Ciencias

Pensamiento Computacional para Ingeniería

Lineamientos del proyecto:

Desarrollar en equipos de dos personas un programa que integre todos los elementos del contenido de la unidad de formación y que permita tomar datos de un dataset y generar resúmenes con estadísticas, reportes, y gráficas.

Cada uno de los integrantes del equipo deberá desarrollar:

- Una estadística que involucre dos o más columnas de un archivo o datos cruzados de dos archivos.
- Un reporte o resumen de datos.
- Una gráfica.

Hazlo más interesante agregando que sea configurable para el usuario al menos una de las opciones (una por cada integrante), es decir que al hacer una estadística, un reporte o una gráfica, el usuario pueda elegir entre algunas opciones. Por ejemplo, si estamos con un dataset acerca del Covid y la estadística que estás generando o resolviendo es el porcentaje de personas que tienen una determinada enfermedad y que han resultado positivo a Covid, pues que el usuario pueda elegir si quiere ver el porcentaje de personas con diabetes u obesidad o cardiovasculares o pulmonares, etc. (este es sólo un ejemplo sencillo para mostrar a qué se refiere que el usuario pueda elegir o configurar la estadística a mostrar).

El programa deberá contar con un menú de al menos 3 opciones: estadísticas, reportes y gráficas y cada uno con su submenú para acceder a las opciones desarrolladas por los integrantes.

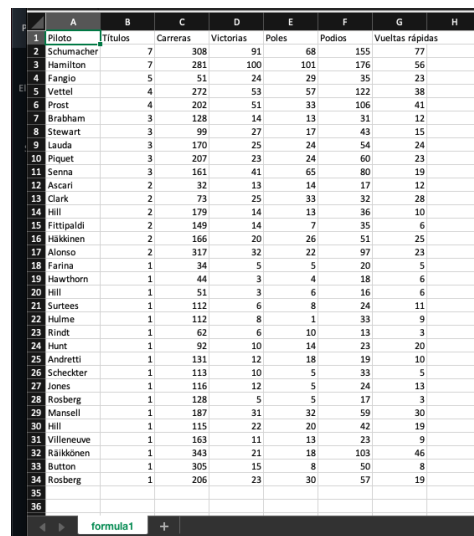
El programa deberá ser desarrollado en su totalidad en Python.

Introducción:

Dataset Fórmula 1

Nuestro equipo seleccionó este archivo csv ya que consideramos que tenía una mayor cantidad de datos íntegros que nos facilitan el desarrollo de las estadísticas. Además de que los datos contenidos nos parecían que eran de un gusto general.

Tuvimos en cuenta que eran datos fáciles de reconocer, asimilar y manipular, tales como: número de victorias, nombre, entre otros; siendo estas las principales.



The image shows a screenshot of a spreadsheet application displaying a table of Formula 1 driver statistics. The table has 8 columns: A (Piloto), B (Títulos), C (Carreras), D (Victorias), E (Poles), F (Podios), G (Vueltas rápidas), and H (empty). The rows list drivers from 1 to 36. The data is as follows:

	A	B	C	D	E	F	G	H
1	Piloto	Títulos	Carreras	Victorias	Poles	Podios	Vueltas rápidas	
2	Schumacher	7	308	91	68	155	77	
3	Hamilton	7	281	100	101	176	56	
4	Fangio	5	51	24	29	35	23	
5	Vettel	4	272	53	57	122	38	
6	Prost	4	202	51	33	106	41	
7	Brabham	3	128	14	13	31	12	
8	Stewart	3	99	27	17	43	15	
9	Lauda	3	170	25	24	54	24	
10	Piquet	3	207	23	24	60	23	
11	Senna	3	161	41	65	80	19	
12	Ascari	2	32	13	14	17	12	
13	Clark	2	73	25	33	32	28	
14	Hill	2	179	14	13	36	10	
15	Fittipaldi	2	149	14	7	35	6	
16	Häkkinen	2	166	20	26	51	25	
17	Alonso	2	317	32	22	97	23	
18	Farina	1	34	5	5	20	5	
19	Hawthorn	1	44	3	4	18	6	
20	Hill	1	51	3	6	16	6	
21	Surtees	1	112	6	8	24	11	
22	Hulme	1	112	8	1	33	9	
23	Rindt	1	62	6	10	13	3	
24	Hunt	1	92	10	14	23	20	
25	Andretti	1	131	12	18	19	10	
26	Scheckter	1	113	10	5	33	5	
27	Jones	1	116	12	5	24	13	
28	Rosberg	1	128	5	5	17	3	
29	Mansell	1	187	31	32	59	30	
30	Hill	1	115	22	20	42	19	
31	Villeneuve	1	163	11	13	23	9	
32	Räikkönen	1	343	21	18	103	46	
33	Button	1	305	15	8	50	8	
34	Rosberg	1	206	23	30	57	19	
35								
36								

Metodología:

Primero realizamos el cvs con base en los datos que obtuvimos de la Fórmula 1. Posterior a eso realizamos un programa con la librería matplotlib, la cual nos permite crear gráficas a partir de un archivo csv. Con esto en cuenta, creamos un diccionario que diera la opción al usuario de pedir solamente el dato que quisiera como: victorias, poles, etc...; y creamos una función para cada aspecto a evaluar.

```

def Victorias():
    corre = []
    var = []
    for row in reader:
        corre.append(row[0])
        var.append(float(row[3]))
    plt.bar(corre, var)
    plt.grid(True)
    plt.title('Número de victorias', fontsize=25)
    plt.xlabel('Pilotos', fontsize=10)
    plt.xticks(rotation=90)
    plt.ylabel('Número', fontsize=10)
    plt.show()

def Poles():
    corre = []
    var = []
    for row in reader:
        corre.append(row[0])
        var.append(float(row[4]))
    plt.plot(corre, var, 'y-')
    plt.grid(True)
    plt.title('Número de poles', fontsize=25)
    plt.xlabel('Pilotos', fontsize=10)
    plt.xticks(rotation=90)
    plt.ylabel('Número', fontsize=10)
    plt.show()

# Se abre el archivo csv
f = open("formula1.csv", encoding='ISO-8859-1')
# Se lee el archivo csv
reader = csv.reader(f)
# Se crea un diccionario para saber qué función usará
acciones = {1: 'Titulos', 2: 'Carreras', 3: 'Victorias', 4: 'Poles', 5: 'Podios', 6: 'Vueltas_rapidas'}
# SE interactua con el usuario para saber qué gráfica quiere
accion = int(input('Que accion deseas realizar?
1. Número de títulos
2. Número de carreras
3. Número de victorias
4. Número de poles
5. Número de podios
6. Número de vueltas rápidas
>>> '))

```

El usuario introduce el número correspondiente a la función que desea realizar y utilizando un método if, el programa da la instrucción de realizar la función correspondiente. Por ejemplo, si el usuario ingresa el numero 5, el código realizara la función correspondiente a Podios y lanzará la gráfica adecuada.

```

# El programa toma decisión sobre que condicional se cumple
if acciones[accion] == 'Titulos':
    Titulos()
elif acciones[accion] == 'Carreras':
    Carreras()
elif acciones[accion] == 'Victorias':
    Victorias()
elif acciones[accion] == 'Poles':
    Poles()
elif acciones[accion] == 'Podios':
    Podios()
else:
    Vueltas_rapidas()

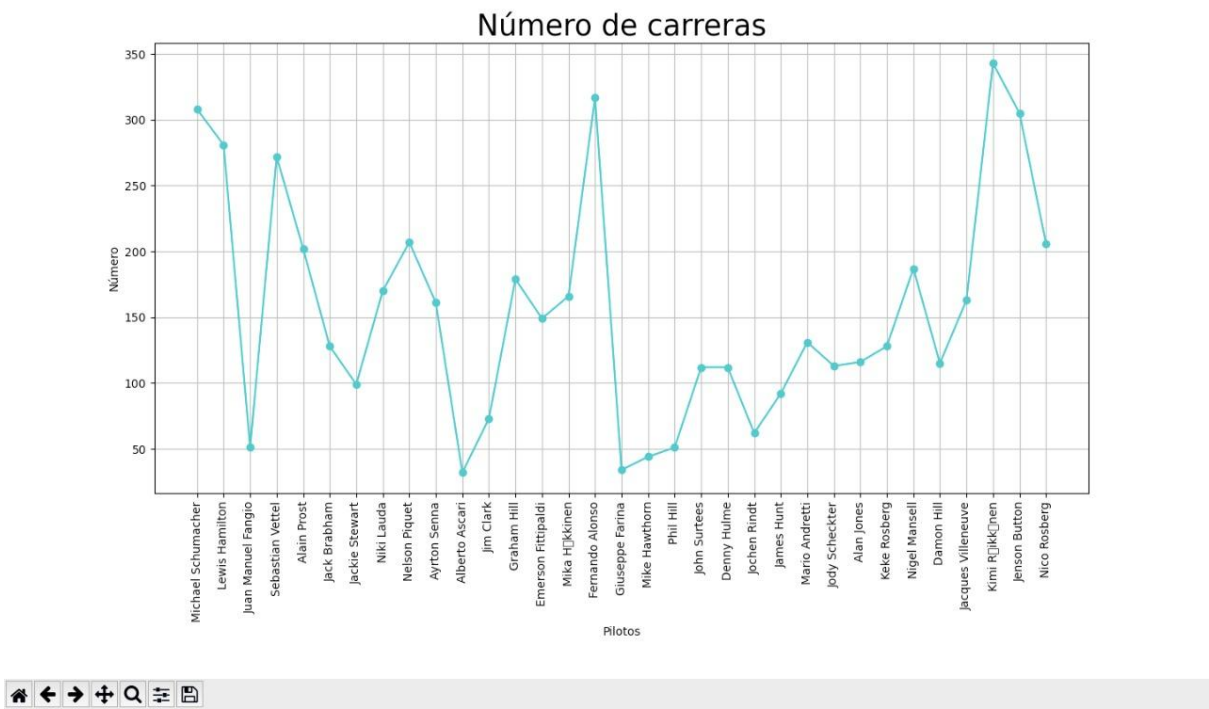
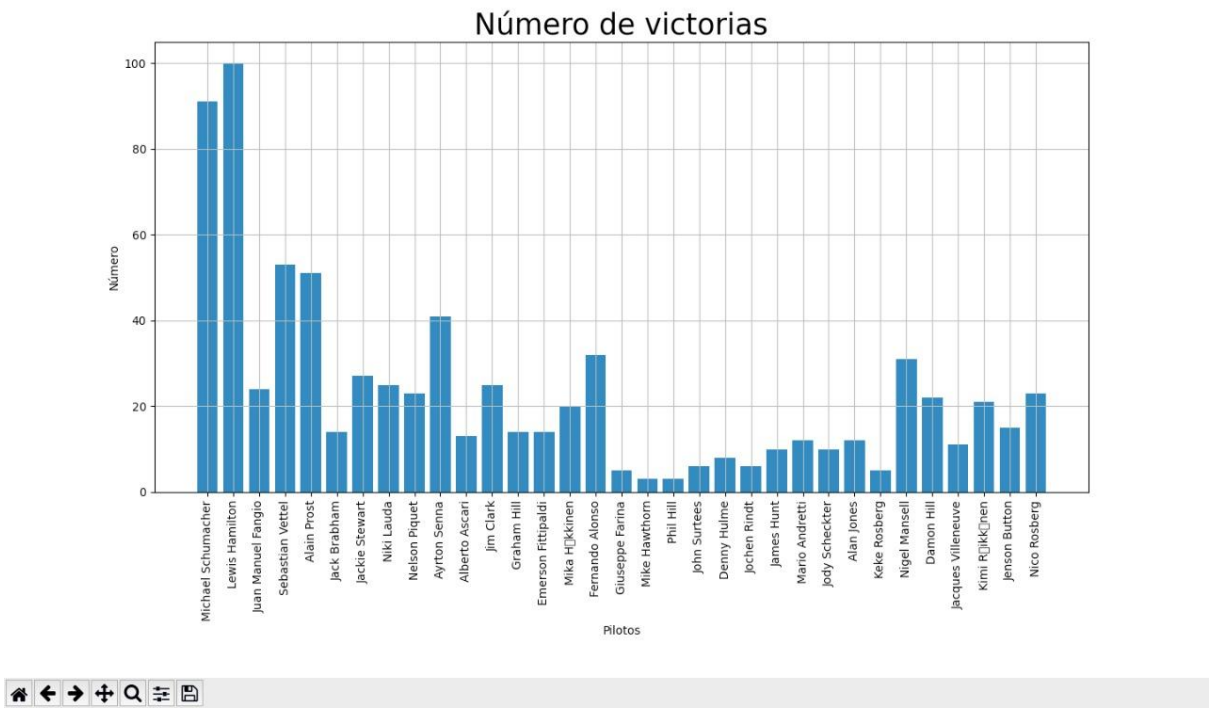
```

```

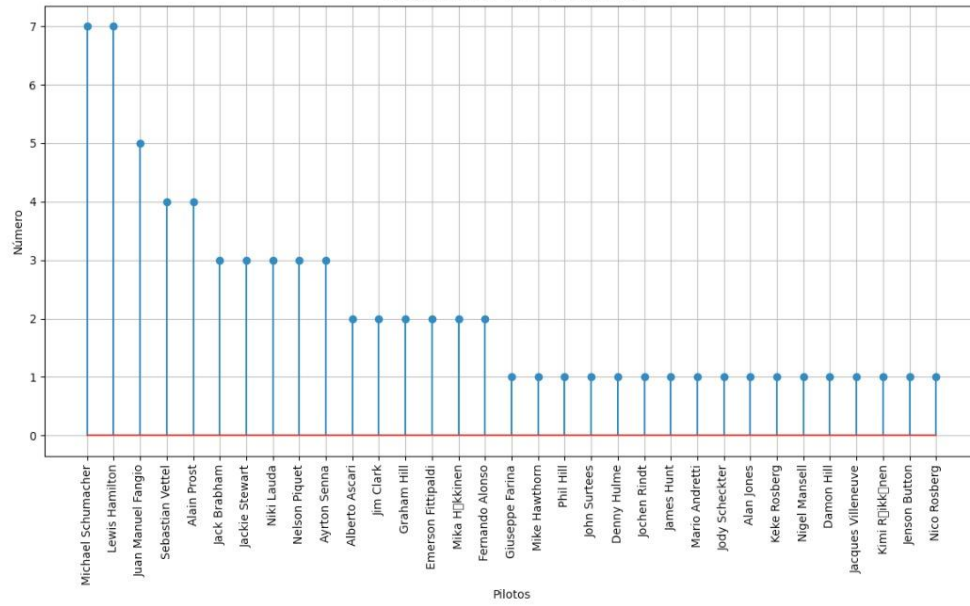
def Podios():
    corre = []
    var = []
    for row in reader:
        corre.append(row[0])
        var.append(float(row[5]))
    plt.scatter(corre, var)
    plt.grid(True)
    plt.title('Número de podios', fontsize=25)
    plt.xlabel('Pilotos', fontsize=10)
    plt.xticks(rotation=90)
    plt.ylabel('Número', fontsize=10)
    plt.show()

```

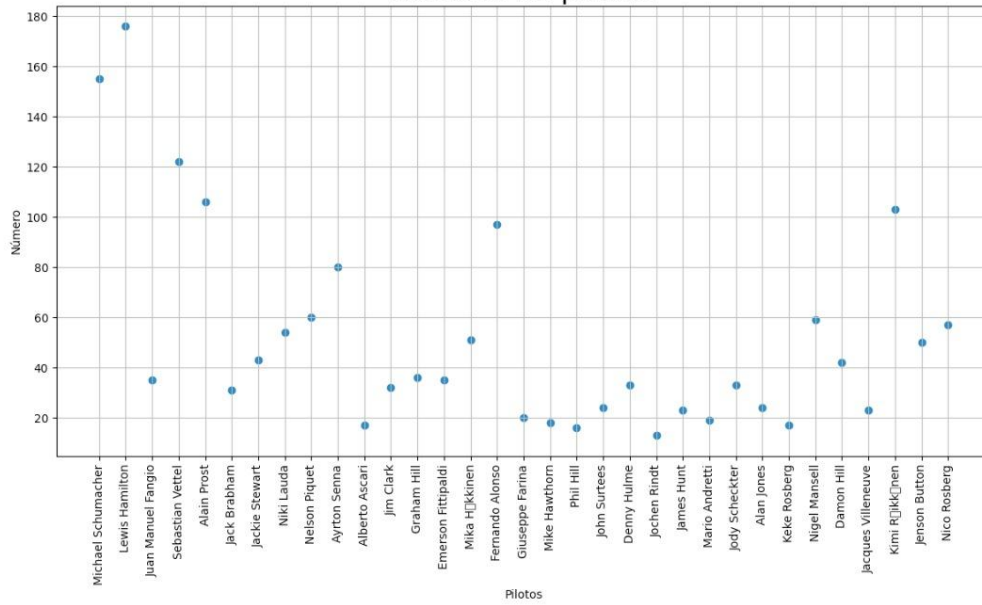
Gráficas:



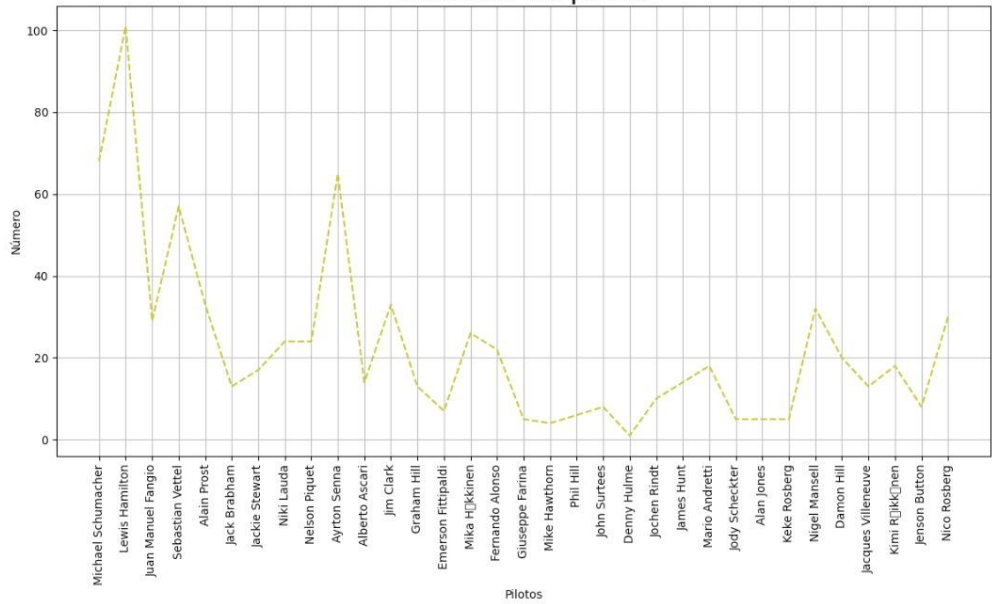
Número de títulos



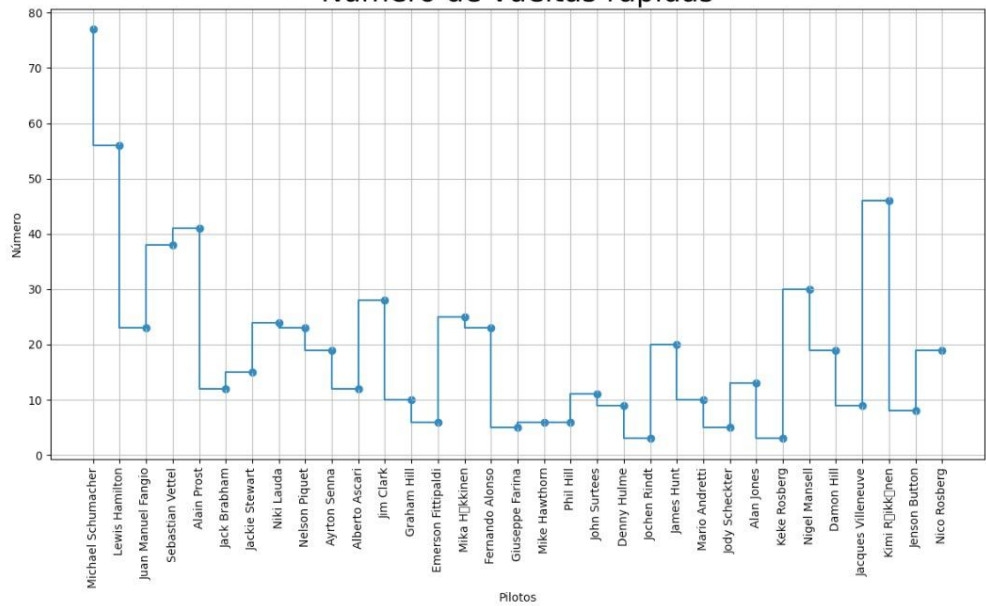
Número de podios



Número de poles



Número de vueltas rápidas



Casos prueba:

Para los casos pruebas encontramos mayor dificultad en los ejes de las estadísticas, ya que a pesar de que el código corriera correctamente, algunos valores en el eje de y se veían duplicados y sin ningún orden. Para el eje de las x debido a que se contaba con el nombre completo de los pilotos, al crear la gráfica estos se veían superpuestos unos con los otros, que dificultaba el correcto entendimiento de la visualización. Además de que un primer código se habían contemplado hacer todos los resultados con gráficas de barras.



Conclusiones:

Conclusiones de Implementación

A diferencia de otros recursos para manejar datos, como lo podría ser excel, utilizar python tiene la ventaja de ser completamente flexible a las necesidades del usuario y se pueden simplificar muchos procesos. Tiene aplicaciones a distintas áreas como negocios o

incluso medicina (por ejemplo si se hace un inventario de medicamentos o estadísticas de síntomas) por lo que es una herramienta muy útil en la vida cotidiana.

Reflexiones:

Jenny: Desde mi punto de vista es importante ver este tipo de comandos y programas porque para varias carreras es importante y en mi caso que quiero una carrera que tiene que ver con programación siento que es importante comprender cómo funciona y los límites de Python con otros programas. Creo que utilizar programas que nos ayudan a ampliar los conocimientos en Python es muy entretenido y asombroso porque realmente hacemos cosas que no creíamos posible de hacer. Cuando investigué lo de matplotlib me di cuenta que lo que hicimos era parte de algo muy grande porque hasta podemos configurarlas con solo colores.

Andrea: En general considero que todo este tipo de visualizaciones son importantes, como por ejemplo para la ciencia de datos, para presentar un informe ante una empresa o negocio del rendimiento económico o incluso puede ser útil en negocios pequeños para facilitar su administración económica y de archivos de datos de clientes. Creo que a veces día a día vivimos en trabajos escolares tener que ver y realizar gráficas y hoy en día estas se pueden realizar de una manera tan sencilla, que es difícil ver todos los procesos que se deben llevar a cabo para que las máquinas arrojen estas estadísticas de una manera perfecta y en cuestión de segundos, y que puedas añadir y quitar tantos datos quieras. O incluso cambiar el diseño o el estilo gráfico con un solo clic y que el programa automáticamente reconozca en qué orden colocar los datos. Además de lograr conocer todas las herramientas que ofrece Python y cómo aplicarlas en distintos contextos.

Andrés: Considero que el saber manipular, interpretar y visualizar información es muy importante en el mundo actual debido a que todo se dicta por la información, desde las redes sociales hasta la ropa que vestimos. Por eso es importante que utilicemos herramientas como

Python y en general lenguajes de programación para manipular archivos ya que nos da la flexibilidad de usar el formato que nosotros escojamos y de variar entre las opciones que predeterminamos. Básicamente nos ayuda a digerir información de manera más efectiva y descartamos todo aquello que no nos es particularmente útil. Esto puede aplicarse en diversos campos, desde alguien con una panadería hasta un estudiante haciendo una investigación. Por eso es importante aprender a manipular información.