CIPFP www.fpmislata.com							
Actividad:	**Spark SQL 1**						
Ciclo:	IABD	**Modulo:**	SBD	**Curso:**	2022-23	**Agrupación:**	1
Alumno/a:						**Grupo:**	

Ejercicio 0

Vamos a generar nuestros propios datos de facturación para eso tenemos que usar el siguiente código

```
import csv
import random
import string


facturas = []
for i in range(100):
    factura = {
        'NumeroFactura': i+1,
        'Fecha': f'2023-03-{random.randint(1,28):02d}',
        'Cliente': ''.join(random.choices(string.ascii_uppercase + string.digits, k=10)),
        'Producto': ''.join(random.choices(['Producto A', 'Producto B', 'Producto C'],
weights=[0.4, 0.4, 0.2], k=1)),
        'Cantidad': random.randint(1, 10),
        'PrecioUnitario': round(random.uniform(10, 100), 2),
        'Total': 0
    }
    factura['Total'] = round(factura['Cantidad'] * factura['PrecioUnitario'], 2)
    facturas.append(factura)


with open('facturas.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['NumeroFactura', 'Fecha', 'Cliente', 'Producto', 'Cantidad', 'PrecioUnitario',
'Total'])
    for factura in facturas:
        writer.writerow([factura['NumeroFactura'], factura['Fecha'], factura['Cliente'],
factura['Producto'], factura['Cantidad'], factura['PrecioUnitario'], factura['Total']])
```

Ejercicio 1

Cargar el fichero CSV como un dataframe

Ayuda:
spark.read.csv

CIPFP www.fpmislata.com							
Actividad:	**Spark SQL 1**						
Ciclo:	IABD	**Modulo:**	SBD	**Curso:**	2022-23	**Agrupación:**	1
Alumno/a:						**Grupo:**	

Ejercicio 2
Calcular el total de ventas por producto
Ayuda:
Funciones: Groupby y agg

Ejercicio 3

Encontrar el cliente con mayor número de ventas
Ayuda:
Functiones: Groupby ,count y order by

Ejercicio 4

Calcular el promedio de ventas diarias

Ayuda:

Convertir la columna Fecha a un tipo de dato Date

df = df.withColumn("Fecha", df["Fecha"].cast("date")) así tenemos el tipo convertido a date

Ejercicio 5

Encontrar las facturas con un total de ventas superior a 500

Ayuda:

Función: Filter