**CASO PRÁCTICO**

**Máster Análisis de Datos con Python**

Iván Pinar Domínguez ©

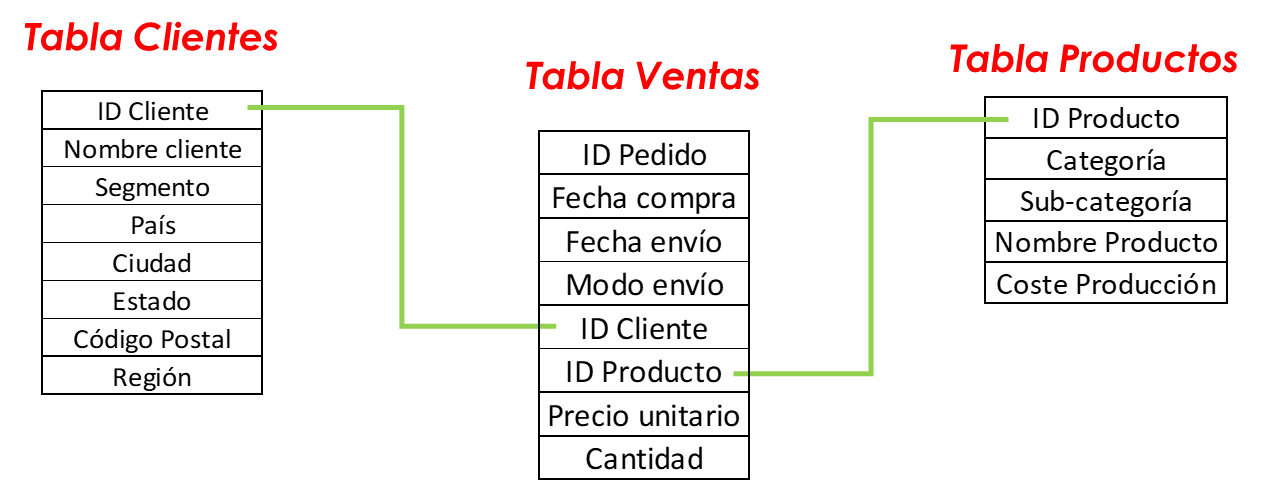
***Enunciado del Caso***

En este caso práctico usted es el Data Scientist de la empresa Mobpatec, dedicada a la fabricación de mobiliario de oficina y dispositivos tecnológicos.

Actualmente, su empresa dispone de una BBDD SQL donde registra lo siguiente:

* Tabla con las ventas de cada artículo.
* Tabla con la información de los productos que vende.
* Tabla con la información de sus clientes.

El modelo de datos de la BBDD relacional sería el siguiente:



Para facilidad de uso del caso práctico, dispone de estas 3 tablas en formato xlsx adjuntas, puesto que no conoce la IP del servidor de la BBDD SQL pero le han pasado el exportado con los datos.

Además, le han adelantado que va a tener que hacer una serie de estudios en función de la población del país de tus clientes, su PIB y la renta per cápita promedio, por tanto va a tener que buscar en la web estas tablas de datos:

* Población:

<https://es.wikipedia.org/wiki/Anexo:Pa%C3%ADses_y_territorios_dependientes_por_poblaci%C3%B3n>

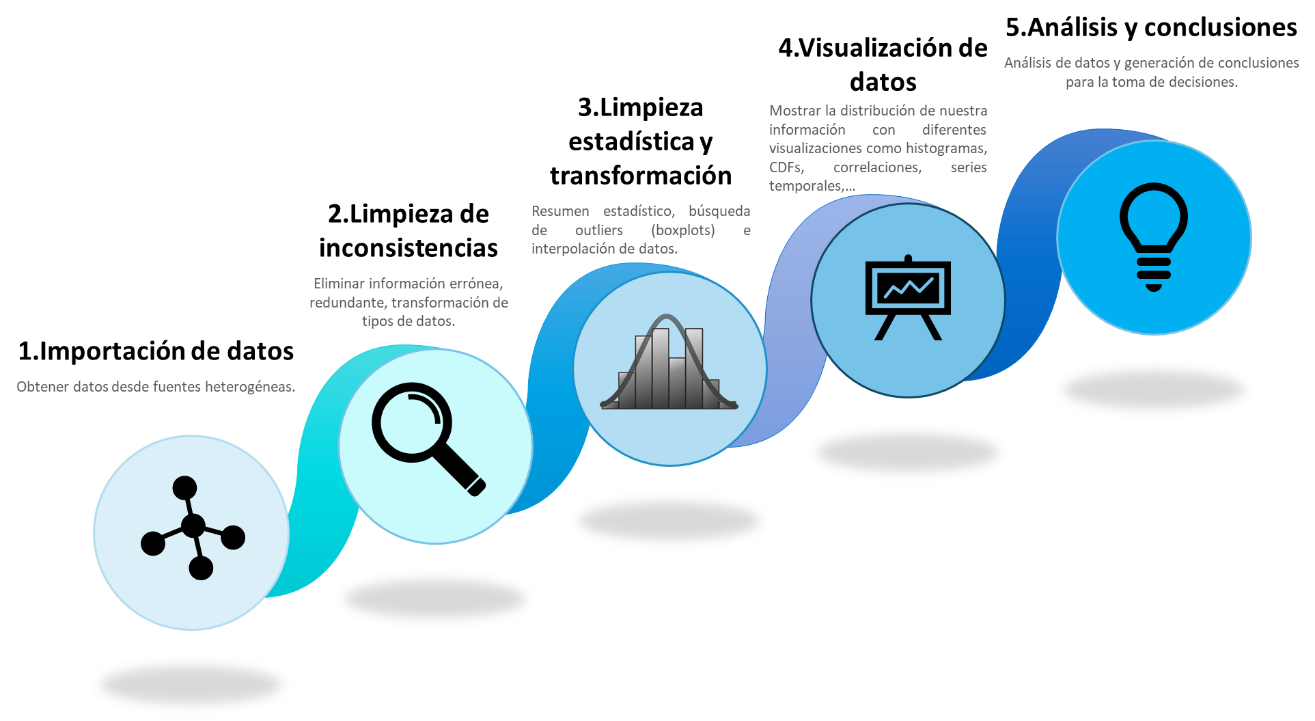
* PIB vs País:

<https://www.worldometers.info/gdp/gdp-by-country/>

* Renta per cápita:

<https://en.wikipedia.org/wiki/List_of_countries_by_average_wage>

En base a estas fuentes de información, como Data Scientist debe seguir el siguiente flujograma que ha aprendido en este curso y con ello dar respuesta a las preguntas que se plantean posteriormente:



**PASO 1: Importación de datos**

Debe importar las 6 fuentes de datos en diferentes dataframes nombrándolos como:

* Ventas 🡪 df\_ventas
* Clientes 🡪 df\_clientes
* Productos 🡪 df\_productos
* Población 🡪 df\_pob
* PIB 🡪 df\_pib
* Renta per cápita 🡪 df\_rpc

**PREGUNTA 1**: ¿Cuáles son las dimensiones (registros x columnas) de cada dataframe?

Resultado:

- El dataframe de ventas tiene 9994 filas y 8 columnas

- El dataframe de clientes tiene 793 filas y 8 columnas

- El dataframe de productos tiene 1864 filas y 5 columnas

- El dataframe de población tiene 246 filas y 12 columnas

- El dataframe del PIB tiene 177 filas y 8 columnas

- El dataframe de la renta per capita tiene 38 filas y 5 columnas

**PREGUNTA 2**: ¿Existen valores NaN en nuestras fuentes?

Resultado:

- En el dataframe de ventas hay 9 filas erroneas

- En el dataframe de clientas hay 455 filas erroneas

- En el dataframe de productos hay 2 filas erroneas

- En el dataframe de población hay 1 filas erroneas

- En el dataframe de PIB hay 0 filas erroneas

- En el dataframe de renta per capita hay 5 filas erroneas

**PASO 2: Limpieza de inconsistencias**

Ahora debe realizar la limpieza de aquéllos registros inconsistentes, como norma en su empresa se declara un registro como inconsistente si no existe una fecha de compra (está vacía).

Realice la limpieza en base a la suposición anterior (Ayuda: Puede aplicar el método isnull() sobre la columna de interés para filtrar la información y verificar si hay registros vacíos, para eliminarlos puede aplicar el método notnull() para quedarse con los registros no vacíos).

**PREGUNTA 3**: ¿Cuántos registros se han eliminado de df\_ventas?

Resultado:

- En el dataframe de ventas se han eliminado 9 filas

**PREGUNTA 4**: ¿Existen registros duplicados en nuestros datos? Si es así limpie la información de esta información redundante.

Resultado:

En el dataframe ventas hay un número de filas totales: 9985, de las cuales hay 1 filas duplicadas

En el dataframe clientas hay un número de filas totales: 338, de las cuales hay 0 filas duplicadas

En el dataframe productos hay un número de filas totales: 1862, de las cuales hay 0 filas duplicadas

En el dataframe población hay un número de filas totales: 245, de las cuales hay 0 filas duplicadas

En el dataframe PIB hay un número de filas totales: 177, de las cuales hay 0 filas duplicadas

En el dataframe renta per capita hay un número de filas totales: 33, de las cuales hay 0 filas duplicadas

Varios campos de nuestros dataframes contienen datos numéricos, pero en algunos casos, puede que se hayan interpretado como strings debido al separador decimal “,” en lugar de “.”, contienen algún espacio o bien contienen el símbolo “$”.

**PREGUNTA 5**: Transforme las columnas implicadas para que pueda modificar el tipo de datos de los campos que deben ser numéricos. ¿Qué campos ha tenido que modificar?

Resultado:

Dataframe Ventas -> Precio unitario

Dataframe PIB -> GDP (abbrev.)

**PASO 3: Limpieza estadística**

Es el momento de analizar nuestros datos desde el punto de vista estadístico, en primer lugar, visualice los diferentes dataframes para obtener los principales datos estadísticos (Recordatorio: método describe).

**PREGUNTA 6**: ¿Cuál es la mediana del valor de nuestras ventas?

Resultado:

La mediana de la cantidad vendida es 3.0.

La mediana del dinero ganado es 183.50400000000002 €

**PREGUNTA 7**: ¿Qué cantidad de unidades de productos no se superan en el 75% de los pedidos de sus clientes?

Resultado:

El 75% de los pedidos de los clientes no superan las 5.0 unidades.

Nuestra empresa ha decidido que la cantidad máxima de unidades de pedido que puede tramitar a un cliente es de 13 unidades, por tanto si en nuestros datos existen registros con número de unidades mayor, es un outlier y es un dato no válido para los análisis posteriores.

**PREGUNTA 8**: ¿Cuántos pedidos se han realizado con un número de unidades mayor que 13?

Resultado:

29 pedidos con más de 13 unidades.

**PREGUNTA 9**: Realiza la limpieza del dataframe df\_ventas de tal manera que descartemos los registros con unidades > 13. ¿Cuáles son las nuevas dimensiones del dataframe?

Resultado:

Tiene 9965 filas y 9 columnas.

**PREGUNTA 10**: En algunos pedidos, no está registrada la “Fecha de envío”, por tanto se deben completar esos registros como la “Fecha compra” + mediana de la diferencia entre “Fecha de envio” - “Fecha compra” del resto de pedidos.

Resultado:

**PASO 5: Análisis y conclusiones**

En base a la información consolidada de la que ya disponemos, como data scientist utilizando Pandas nos gustaría tener una tabla maestra que agregara las diferentes tablas y realizar sus análisis.

En primer lugar, se deben combinar los dataframes df\_ventas, df\_prod y df\_clientes para obtener un único dataframe global denominado df\_global (utilice el método inner).

**PREGUNTA 11**: ¿Qué país representa el mayor volumen de nuestras ventas?

Resultado:

**PREGUNTA 12**: ¿En qué año se han obtenido un mayor número de ventas?

Resultado:

**PREGUNTA 13**: Teniendo en cuenta el precio de venta unitario, las unidades vendidas y el coste unitario, nos solicitan obtener el beneficio obtenido en cada uno de los pedidos. Para ello debe crear una nueva columna “Beneficio”. ¿Cuál es la categoría de productos que supone un mayor beneficio?

Resultado:

**PREGUNTA 14**: ¿Existe correlacion entre el “Beneficio” total obtenido en un país y su renta per cápita?

Resultado:

**PREGUNTA 15**: ¿Existe correlacion entre el número de pedidos que se realizan en un país y su renta per cápita? (puede utilizar groupby y después el método “.to\_frame” para convertir un objeto grouped a dataframe, con este dataframe puede utilizar merge con df\_rpc y analizar la correlacion)

Resultado:

**PREGUNTA 16**: Intentando analizar a la población de los diferentes países, existe alguna correlación entre el PIB de un país, su población y la renta per cápita?

Resultado:

**PREGUNTA 17**: La empresa se está planteando abrir nuevas plantas de producción en aquéllas ubicaciones donde la diferencia entre “Fecha envío” y “Fecha compra” es más elevada para reducir los tiempos. ¿En qué países elegiría construir estas nuevas plantas de producción?

Resultado:

**PREGUNTA 18**: Como resultado final, el CEO de su compañía necesita disponer de un reporte en formato Excel con el beneficio de cada categoría en cada uno de los países. Exporta el resultado a un fichero con nombre “Reporte Beneficios” y pestaña “Beneficios\_cat\_pais”.

Resultado: