TRABAJO DE FIN DE GRADO

Grado en Estadística

Título: Construcción del IPC con datos alternativos

Autor: Miriam Ortuño Solvas

Director: Salvador Torra Porras

Departamento: Econometría, Estadística y Economía Española

Convocatoria: Junio 2023-24

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

AGRADECIMIENTOS

Me gustaría mostrar mi más sentido agradecimiento:

A mi tutor, Salvador Torra Porras, por su ayuda, paciencia y disposición durante el proceso de desarrollo de este TFG. Gracias por tu paciencia en la enseñanza y ayúdame a defender este proyecto.

A mis padres y mi hermano, que me han enseñado todo lo que se. Siempre han creído en mí y a quienes les debo todo. Gracias por haberlo dado todo por mí, vuestro apoyo y amor incondicional. Todo lo que soy es gracias a vosotros. Os quiero.

A mis abuelos, José Antonio y Manuel que no podrán ver los logros que he logrado, pero me acuerdo de vosotros en cada paso que doy, cada meta que alcanzo. Se que estaríais orgullosos de mí. Gracias por cuidarme desde arriba.

A mis abuelas, Arsenia y José Fina, por acompañarme a lo largo de mi vida y darme un cariño y admiración diario. Gracias por hacerme la niña más feliz.

A mi pareja, Nicolás, gracias por haber aguantado mis momentos más duros y haberme dado comprensión, amor y motivación para seguir siempre adelante. Gracias por estar siempre el uno para el otro y hacernos tan felices. Me siento orgullosa de quienes somos juntos.

RESUMEN DEL TRABAJO

En este trabajo se presenta la nueva forma de calcular el IPC denominada scanner data. Además, se explica el cálculo del IPC tradicional, permitiendo así la comparación entre ambos métodos y mostrando sus ventajas y desventajas. Además, se investiga otra metodología llamada web scraping, donde previamente se han obtenido los conocimientos sobre el significado de las API, la legalidad de esto, cómo funciona, las herramientas imprescindibles para llevarlo a cabo, y el proceso del cálculo del IPC que corresponde a esta técnica.

Este estudio tiene como objetivo final la extracción de datos de varios supermercados previamente seleccionados. Una vez obtenida la información, se procede a su tratamiento y procesamiento de manera óptima y eficiente. Finalmente, se calcula el IPC general y el correspondiente a cada uno de los supermercados.

SUMMARY OF THE WORK

This paper presents a new methodology for calculating the CPI called scanner data. The calculation of the traditional CPI is also explained, doing a comparison between the two methods, and highlighting their respective advantages and disadvantages. Additionally, another methodology called web scraping is explored. This includes an investigation about the meaning of APIs, the legality of data extraction, how it works, the essential tools to required, and the process of calculating the CPI using web scraping.

The final objective of this study is to extract data from several preselected supermarkets. Once the data has been obtained, it is processed and processed in an optimal and efficient manner. Finally, the general CPI and the CPI corresponding to each of the supermarkets is calculated.

PALABRAS CLAVE

Scanner data, IPC, Web scraping, APIs, Python, Legalidad de la web, Extracción de datos, Análisis de datos.

KEYWORDS

Scanner data, IPC, Web scraping, APIs, Python, Web legality, Data extraction, Data analysis.

CLASIFICACIÓ AMS

62-XX STATISTICS

62-02 Research exposition

62-07 Data analysis

68N19 Other programming techniques (object-oriented, sequential, concurrent, automatic, etc.)

ÍNDICE

1. INTRODUCCIÓN

En este trabajo se analiza la nueva metodología de obtención de datos del Instituto Nacional de Estadística (INE). Específicamente, se investigará el scanner data, un nuevo método de recopilación de datos que permite la elaboración de índices oficiales de manera más eficaz. Para ello, se comparará el Índice de Precios al Consumidor (IPC) tradicional con esta nueva metodología, identificando las ventajas y desventajas que ofrece.

Además, se estudiarán los supermercados españoles que ya se han adherido a este nuevo proyecto, con el fin de replicar el proceso de obtención de datos. Por otro lado, se explorará otro método empleado por el INE conocido como web scraping. Esto implicará investigar qué son las API, la legalidad de este método y el funcionamiento de la web.

Para replicar la extracción automática de datos de algunos supermercados mediante web scraping, se examinarán las herramientas más útiles para realizarlo, como Python y R. Antes de proceder con la extracción, es necesario adquirir un conocimiento detallado de estas herramientas y realizar cursos para alcanzar un nivel avanzado.

El objetivo final es comprender en detalle las dos nuevas metodologías, scanner data y web scraping, y evaluar sus ventajas e inconvenientes. También se buscará realizar web scraping en dos o tres de los supermercados más importantes de España, obteniendo la información necesaria para calcular el IPC.

En cuanto a la estructura del trabajo, este se divide en tres bloques: scanner data, web scraping y obtención del IPC. El primer bloque es teórico y ofrece una descripción detallada del scanner data, incluyendo sus ventajas e inconvenientes, así como la implementación de esta metodología en otros países.

El segundo bloque combina teoría y práctica. En la parte teórica, se explica qué es el web scraping, su marco legal, qué son las APIs y cómo se implementan en esta nueva metodología. Además, se describen los pasos iniciales para llevar a cabo el web scraping. En la parte práctica, se realiza el scraping de datos de tres supermercados, detallando cada paso del proceso.

Finalmente, el tercer bloque es práctico y se centra en el cálculo del IPC. Se lleva a cabo un preproceso de los datos y se calcula el IPC utilizando PowerQuery. Los resultados se visualizan en PowerBI.

2. METODOLOGÍA

Para la elaboración de este proyecto, es esencial detallar las diversas herramientas, técnicas y recursos utilizados en su desarrollo. En la parte teórica del trabajo, se ha investigado a fondo las dos nuevas metodologías: scanner data y web scraping. Para ello, se ha recopilado información de fuentes web y documentos de instituciones oficiales como el INE y Eustat. Posteriormente, esta información se ha comprendido y sintetizado de manera clara y precisa, con el fin de facilitar la comprensión de los dos grandes bloques del estudio.

Para la parte práctica del proyecto, primero fue necesario aprender el funcionamiento de la web, las diversas formas de extraer información de ella y las herramientas disponibles para realizarlo. En lugar del software estadístico R, se utilizó Python para la extracción de datos. La elección de usar Python se debe a que es uno de los softwares más utilizados a nivel mundial y cuenta con una amplia documentación en internet sobre cómo realizar web scraping. Por este motivo, se aprendió desde cero a usar Python para poder llevar a cabo el scrapeo de datos.

Una vez adquiridos estos conocimientos, se procedió a la extracción de datos de tres supermercados. Se realizó un estudio de los candidatos para seleccionar los tres supermercados más adecuados. El primero fue Mercadona, ya que es el supermercado más importante a nivel nacional, aunque presentaba el inconveniente de requerir registro para acceder a los datos de la web. El segundo supermercado elegido fue Carrefour, debido a su amplia variedad de productos y su relevancia como el segundo supermercado más importante en España. El tercer supermercado seleccionado fue DIA, no por ser el tercero más importante, sino porque su web es la más estable y presenta menos cambios, lo cual reduce la necesidad de posteriores ajustes al algoritmo de web scraping. También, se llevó a cabo una exhaustiva investigación de códigos existentes, con el objetivo de utilizar algunos de ellos para la extracción de datos de los tres supermercados de manera simultánea. Es importante mencionar que el código final desarrollado es completamente automático, priorizando en todo momento la automatización y descartando cualquier modificación que requiriera intervención manual.

Una vez completada la extracción de datos, se procedió al cálculo del IPC. Primero, se realizó una limpieza de los datos utilizando los conocimientos adquiridos a lo largo del grado, mediante una nueva herramienta Power Query. Luego, se calculó el IPC aplicando el conocimiento teórico obtenido de la investigación previa.

Finalmente, los resultados se presentaron en un *Dashboard* utilizando PowerBI, una de las herramientas más potentes para la visualización de datos. La elección de PowerBI se basó en su capacidad para crear visualizaciones claras y efectivas. Afortunadamente, he tenido la oportunidad de realizar prácticas que me han permitido aprender a un nivel muy avanzado el uso tanto de Power Query como de PowerBI, lo que facilitó significativamente esta parte del proyecto.

3. SCANNER DATA

3.1. ¿Qué es el scanner data?

Actualmente, y mediante el Big Data el Instituto Nacional de Estadística (INE) está trabajando en varios métodos para optimizar la recopilación de datos y la elaboración de sus índices oficiales de manera más eficaz. Estos métodos aprovechan los avances tecnológicos para lograr una mayor actualización de los datos y reducir los costos asociados. Uno de los métodos innovadores que se están utilizando es el scanner data, que mejora la precisión en el cálculo del Índice de Precios de Consumo (IPC) y el Índice de Precios de Consumo Armonizado (IPCA). Esta mejora se debe a la frecuencia más alta de actualización de los datos, lo que conduce a una mayor precisión en las mediciones.

El scanner data representa una alternativa en la adquisición de información, la cual se emplea para calcular el Índice de Precios al Consumidor (IPC) de manera más eficaz y económica. Este enfoque implica aprovechar los datos de las bases de datos de las empresas, que contienen información detallada sobre todas las transacciones que se realizan en la línea de caja. Esta práctica resulta un ahorro significativo en costos y tiempo en comparación con los métodos de recopilación de datos convencionales. La información recabada de las empresas abarca todos los productos adquiridos en el punto de venta, lo que proporciona un registro completo de los productos consumidos por las familias. Esta recopilación exhaustiva permite una actualización continua sobre los productos que tienen una mayor demanda y aquellos que experimentan un menor consumo.

Como resultado, esta nueva metodología de recolección de datos proporciona información en tiempo real sobre las ventas y las tendencias del mercado, lo que permite mantener una cesta de productos representativa de los hábitos de consumo de las familias. Por contra, no tendremos información sobre los consumidores, es decir, no conocemos que perfil de clientes está comprando.

Las bases de datos recogidas almacenan la información del producto, su código, las unidades vendidas, el precio y la fecha de la venta. El código asignado a los productos puede variar dependiendo de la clasificación interna de cada empresa. Esto significa que la información será recopilada por los supermercados individuales que pertenecen a esa empresa. Para obtener estas bases de datos, será necesario contar con el consentimiento de las empresas en cuestión. Por lo tanto, surge un desafío, ya que será necesario implementar un proceso diferente para calcular el índice en cada supermercado.

El uso del scanner data en los cálculos del Índice de Precios al Consumidor (IPC) e Índice de Precios al Consumidor Armonizado (IPCA) está en práctica en varios países de la Unión Europea[1] y esta promueve su funcionamiento para garantizar la armonización de las estadísticas entre los estados miembros.

[1] Alemania, Francia, España, Italia, Reino Unido, Países Bajos, Suecia, Bélgica, Portugal, Austria, Finlandia, Dinamarca, Irlanda, Grecia, Polonia, República Checa, Eslovaquia, Hungría, Rumania, Bulgaria, Lituania, Letonia, Estonia, Croacia, Eslovenia, Chipre, Malta y Luxemburgo.

En 2014, el Instituto Nacional de Estadística (INE) inició un proyecto basado en el diseño metodológico adoptado por otros países de la UE. El objetivo era establecer cómo recopilar y analizar las variables necesarias para calcular el IPC e IPCA. Posteriormente, en enero de 2020, inició el proyecto para obtener los datos de los supermercados que quisieran unirse[2] a la iniciativa. Esto permitió aprovechar la información detallada sobre las transacciones comerciales y mejorar la precisión en los cálculos de los índices de precios.

Las bases de datos proporcionadas por los supermercados contienen información sobre una amplia gama de productos de consumo masivo, que incluyen bebidas, productos envasados, artículos de limpieza, alimentos, productos de cuidado personal, alimentos para mascotas y artículos de parafarmacia. Con la implementación del scanner data, se recopilarán datos de las compañías que participen en el proyecto. Estas bases de datos requieren un preprocesamiento, ya que la información recibida por cada compañía se presenta en formatos distintos, a pesar de contener la misma información. Es crucial que el uso del scanner data no implique costos adicionales para la empresa, por lo que será necesario adaptar el preprocesamiento a cada base de datos de manera individual. Además, una vez procesados, se necesitará un nuevo método para calcular el Índice de Precios al Consumidor (IPC), dado que la información obtenida a través del scanner data difiere significativamente del método tradicional. Finalmente, una vez que el scanner data esté implementado de manera efectiva y su confiabilidad esté comprobada, se eliminará la recopilación tradicional de datos de aquellas compañías que proporcionen sus bases de datos para este fin.

La implementación del scanner data supone un gran cambio en la metodología y cálculo del IPC en base a 2016[3]. Se utilizará la información que las empresas envíen, sin suponer ningún coste adicional para ellas. Se deben de tener en cuenta dos cuestiones importantes:

- La disposición de las empresas en enviar información recurrentemente, ya que sin las bases de datos no se puede obtener ningún estadístico.
- El tratamiento de la información recibida, así como una nueva metodología de cálculo para el IPC[4]:
    - Diferencias entre concepto de precio y valor unitario.
    - Diferencias entre producto y gama completa de variedades.
    - Volumen de la información.

3.1.1. Supermercados adheridos al scanner data en España

El scanner data es un proyecto relativamente nuevo, por tanto, no todos los supermercados se han adherido aún a él para proporcionar sus datos. El INE busca que todos los

[2] Consultar apartado 3.1.1.

[3] Consultar apartado 3.2.1.

[4] Explicado con más detalle en el apartado 3.2.2.

supermercados españoles se adhieran a este proyecto para obtener las bases de datos y poder medir la inflación de manera más eficiente y rápida. Los primeros supermercados en adherirse en España han sido el Corte Inglés, Auchan (Alcampo y Sabeco) y el Carrefour.

Los supermercados españoles con una mayor cuota de mercado son: Mercadona (26.2%), Carrefour (9.9%), Lidl (6.4%), Eroski (4.4%), DIA (4.1%), Consum (3.4%), Alcampo (3.1%), Aldi (1.5%) y el grupo IFA (9.6%, Condis, Bon Preu, Dino Sol, …). Mercadona destaca como el líder con una participación predominante, con un incremento del 0.6% en el año 2023. Carrefour tiene el segundo puesto, donde también tiene un crecimiento año tras año, pero más moderado un 0.2%, ya que tienen una amplia gama de promociones y un programa de fidelidad. Lidl y Aldi están subiendo su participación porque han abierto nuevos supermercados e invierten en medios. Eroski sigue siendo la cuarta, ya que tiene una gran distribución. Por último, DIA es el único que ha bajado su porcentaje en la cuota de mercado un 0.5% respecto el 2022, debido a la venta de los supermercados Alcampo. Para el 2024 se prevé que la cuota de mercado sea aproximadamente la misma mientras la inflación se vaya controlando.

En consecuencia, el INE está solicitando la cooperación de todos los supermercados, con un enfoque particular en aquellos que tienen una mayor presencia en el mercado. En la actualidad, se encuentra buscando la colaboración de Mercadona y Lidl, ya que ambos cuentan con una notable participación en el mercado, destacando especialmente Mercadona como líder indiscutible.

La cuota de mercado se determina al comparar las ventas de cada supermercado con el total de ventas en el mercado. Por tanto, los supermercados compiten para ofrecer los mejores precios y así atraer a más consumidores, lo que les permite obtener un margen comercial. En resumen, su objetivo es captar la mayor parte posible de los consumidores. Durante el 2023 los precios aumentaron y, por tanto, los consumidores compraban más en el pequeño comercio, por su contra, los supermercados realizaron más promociones y ofertas para atraerlos. Por tanto, la estrategia para el 2024 es seguir lanzando ofertas y promociones para que los clientes se mantengan fieles y aumentar la cuota de mercado. Es relevante destacar que no todos los supermercados ofrecen la misma variedad de productos. Al calcular la cuota de mercado, se consideran las ventas totales de cada supermercado, sin tener en cuenta específicamente qué productos se están vendiendo. Esto permitirá obtener una medición más precisa de la participación de cada supermercado en el mercado en general.

3.2. IPC

En este apartado se explica el IPC “tradicional” i la nueva forma de calcular el IPC a través del scanner data, además se explican las diferencias entre estos dos métodos.

3.2.1. IPC “tradicional”

Empezamos por entender que es el IPC, Índice de Precios de Consumo, este es un índice utilizado para medir la inflación y se calcula de forma mensual. Es uno de los índices más importantes, ya que nos muestra la evolución de los precios de los servicios y bienes, de los

hogares españoles. Además, tiene otras aplicaciones como la revisión de los contratos de arrendamiento de inmuebles, fijación de pensiones, otros tipos de contrato y como actualización de las primas de los seguros. El INE (Instituto Nacional de Estadística) es el encargado de actualizar este índice mensualmente, que se publica entre el 10 y 15 de cada mes. Se publica el IPC mensualmente, anualmente y para el acumulado del año. Hay dos tipos de IPC importantes a conocer:

- **IPCA (Índice de Precios de Consumo Armonizado):** es un índice que se calcula para que sea comparable con los diferentes miembros de la UE, este está revisado por el Eurostat, que es la agencia estadística de la Unión Europea y su revisión se realiza anualmente. La diferencia con el IPC es que la cesta es igual para todos los países de la UE.

- **IPC Subyacente:** para su cálculo no se tienen en cuenta el precio de los productos energéticos y productos alimenticios no elaborados, estos son los que tienen los precios más volátiles y, por tanto, podremos ver un índice más estable.

Nos centraremos en el cálculo del IPC, para calcularlo se basa en una cesta de la compra, que son los productos más consumidos por los españoles. Aunque esta cesta va variando y, por tanto, se realizan actualizaciones recurrentemente para quitar o añadir algunos productos. Hay 12 grupos y cada uno de ellos con una ponderación según el peso en las familias, esta ponderación también se actualiza. Dentro de los 12 grupos hay una subclasificación donde encontramos más de 400 artículos. Los grupos y ponderaciones para este 2024 son:

- Grupo 1: alimentación y bebidas no alcohólicas (19,2%)
- Grupo 2: bebidas alcohólicas y tabaco (3,8%)
- Grupo 3: artículos de vestir y calzado (3,9%)
- Grupo 4: vivienda, agua, electricidad, gas y otros combustibles (12%)
- Grupo 5: menaje (mobiliario, equipamiento del hogar y gastos corrientes de conservación de la vivienda) (5,3%)
- Grupo 6: salud (5,8%)
- Grupo 7: transportes (14,4%)
- Grupo 8: comunicaciones (3,3%)
- Grupo 9: ocio y cultura (8,6%)
- Grupo 10: enseñanza (1,9%)
- Grupo 11: hoteles, cafés y restaurantes (13,9%)
- Grupo 12: otros bienes y servicios (7,8%)

Toda esta información sobre los precios se recoge a través de la EPF (Encuesta de Presupuestos Familiares[5]), se realiza anualmente aproximadamente en 24.000 viviendas seleccionadas aleatoriamente, estas deben de estar en todo el territorio nacional. De esta forma se obtienen los productos más consumidos en los hogares españoles y sus ponderaciones, es decir, nos permite conocer el consumo de los españoles. Cabe comentar que estos artículos siguen la clasificación internacional de consumo, COICOP[6] (Eustat).

Antes de realizar la recogida de datos se debe de realizar el diseño de la muestra, este es un diseño no probabilístico[7]. Para que el índice sea de calidad, significativo y representativo se desagrega el diseño de la muestra en: selección de municipios, selección de zonas comerciales y establecimientos y selección de artículos. La selección de municipios hay que tener en cuenta que estos estén repartidos por toda la provincia, asegurando la representatividad poblacional, incluyendo aquellos municipios más pequeños y todos los municipios deben tener todos los artículos especificados en la cesta. Por otro lado, la selección de zonas comerciales y establecimientos se realiza un estudio de los diferentes tipos y características de los establecimientos, teniendo en cuenta otras fuentes de información: Encuesta de Comercio[8] (INE) y el Ministerio de Agricultura, Pesca y Alimentación. La recogida de precios de aquellos artículos con una mayor ponderación se debe de seleccionar un mayor número de establecimientos. Los establecimientos seleccionados deben de ser los más visitados en la localidad o con mayor número de ventas. Por último, para la selección de artículos se utiliza la EPF con la que se obtiene la cesta de la compra. Estos artículos deben de cumplir que la evolución de los precios de estos sea similar a los de su mismo subgrupo, consumidos frecuentemente por la población, precios fáciles de recoger y que tengan garantías de permanecer en el mercado.

Una vez se ha diseñado la muestra, se procede a recopilar datos de más de 400 artículos en aproximadamente 29.000 establecimientos, ubicados en 177 municipios en todo el territorio nacional, todo conforme al diseño previamente establecido de la muestra. Para facilitar la recogida, cada uno de los artículos viene acompañado con su respectiva descripción. La recogida de estos datos se realiza de la forma tradicional: visita personal a los supermercados, por teléfono, catálogo, fax, internet o por correo electrónico. Esta recogida se realiza mediante un cuestionario aproximadamente del 1 al 22 de cada mes, los dos incluidos, aunque en algunos artículos la recogida puede durar hasta final de mes. Cada establecimiento se intenta visitar el mismo día cada mes para recoger de forma más exacta la variación mensual. La recogida de precios se realiza 2 o 3 veces cada mes por cada artículo, para tener las fluctuaciones de este, aproximadamente se recogen más de 220.000 precios. También

[5] https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176806&menu=ultiDatos&idp=1254735976608

[6] COICOP: Classification Of Individual Consumption by Purpose.

[7] El diseño no probabilístico es un diseño en que los sujetos de estudio no son seleccionados al azar. Los participantes se seleccionan de manera que no garantiza que todos tengan la misma probabilidad de ser seleccionados.

[8] https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736174702&menu=ultiDatos&idp=1254735576778

quedan reflejadas las ofertas y/o rebajas que puedan tener cada uno de los artículos, siempre y cuando los descuentos sean las rebajas de temporada recogidas en los períodos oficiales y ofertas de cualquier tipo, siempre que no sean liquidaciones o saldos. Esta recogida se realiza mediante un cuestionario generado para cada establecimiento, donde se recogen los precios y las incidencias. La lleva a cabo un entrevistador, excepto en las grandes superficies, donde hay varios entrevistadores.

La recogida de precios del alquiler de las viviendas es la más diferente y se realiza a través de una muestra aleatoria simple[9] repartida por todo el territorio nacional y se actualiza a través de la EPA (Encuesta de Población Activa[10]). Estos precios se recogen una vez cada trimestre y cada muestra se divide en tres para poder obtener el índice mensual.

El IPC debe tener dos características principales: representatividad y comparabilidad temporal. Garantizamos la representatividad al definir claramente una cesta de la compra, visitando los establecimientos más visitados y calculando con precisión las ponderaciones de cada categoría de productos. Por otro lado, la comparabilidad temporal se consigue recogiendo los datos mensualmente, ya que un índice sin comparabilidad en el tiempo no tiene significado alguno. Con esto conseguimos que los cambios en el índice sean debidos a cambios en el precio y no a cualquier cambio metodológico de este.

Como se mencionó previamente, se aplica un cuestionario y, una vez que se ha completado, se procede a la grabación de los datos y su depuración, utilizando aplicaciones informáticas. Seguidamente, se hace un análisis para aquellos valores atípicos que sean superiores al ±10% en productos de alimentación y ±5% para el resto de los artículos. Además de realizar el tratamiento de aquellos artículos que no tengan un precio. También se realizan ajustes de calidad, estos ajustes hacen referencia a que la diferencia entre el precio del artículo y el sustituto es muy diferente y viene dado por una diferencia de calidad entre estos dos artículos. Ejemplos de precios de cambio de calidad son, aparatos electrónicos como televisores y lavadoras, artículos de alimentación que estos son más difíciles de medir el cambio de calidad[11]. Finalmente, una vez se obtiene la base de datos depurada se pasa al cálculo del IPC.

Para el cálculo del IPC se usa la fórmula de Laspeyres encadenado, que se empezó a utilizar en el IPC, base 2016. Una fórmula muy sencilla para entender el cálculo es: IPC = (Precios nuevos*cantidades nuevas) / (Precios anteriores*cantidades anteriores) * 100. O bien una forma más desglosada:

[9] Esta muestra inicial consta de 130.000 individuos, lo que equivale a 55.000 familias, y ha sido extraída de la Encuesta de Población Activa (EPA).

[10] https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176918&menu=ultiDatos&idp=1254735976595

[11] Estos ajustes pueden ocurrir cuando se introducen mejoras en un producto, como una actualización tecnológica o una mejora en los materiales utilizados, lo que podría justificar un aumento o disminución en el precio.

Cambio porcentual de un artículo = (Precio año actual – Precio año anterior) / Precio año anterior

Promedio ponderado = Σ (Cambio porcentual de artículo * Ponderación del artículo)

➔ IPC = Promedio ponderado * 100

El cálculo del IPC no es tan simple, ya que implica el uso de una fórmula de Laspeyres encadenada, lo cual involucra un proceso de cálculo complejo[12]. El índice general se expresa de la siguiente manera:

$$Índice_0^t = \prod_{k=1}^{t} \frac{\sum_i Precio_i^k Cantidad_i^{k-1}}{\sum_i Precio_i^{k-1} Cantidad_i^{k-1}} \qquad (3.2.1.1)$$

Es el índice en el año t en base al año 0, es decir establece comparaciones entre el período corriente t y el período base 0, considerando los períodos intermedios k. Por ejemplo: el cálculo del IPC en base a 2016 para el mes m y año t se calcula:

$$Índice\ General_{16}^{mt} = Índice\ General_{16}^{diciembre(t-1)} * (\frac{ÍndiceGeneral_{diciembre(t-1)}^{mt}}{100}) \qquad (3.2.1.2)$$

Se deben tener en cuenta (INE, Índice de Precios de Consumo. Base 2016, 2017):

- Calcular cada uno de los **índices elementales**: calcular cada uno de ellos por artículo y provincia. Se calcula mediante el precio medio del artículo (i), en el mes (m) y año (t), entre el de referencia en diciembre del año (t-1).

$$Índice\ Elemental_{dic(t-1)}^{mt} = \frac{\bar{P}_i^{mt}}{\bar{P}_i^{dic(t-1)}} * 100 \qquad (3.2.1.3)$$

La media del precio del artículo se calcula con la media geométrica simple:

$$\bar{P}_i^{mt} = n_i^{mt} \sqrt{\prod_{j=1}^{n_i^{mt}} P_{i,j}^{mt}} \qquad (3.2.1.4)$$

Siendo, n = número de precios procesados del artículo (i), en el mes (m) y año (t).

[12] Es un proceso complejo debido a la necesidad de datos detallados, ajustes por pesos, encadenamiento de índices y riesgos de sesgo.

- Determinar las **ponderaciones individuales** para cada artículo, las cuales están definidas por la encuesta EPF.

$$W_i = \frac{Gasto\ realizado\ en\ las\ parcelas\ representadas\ por\ el\ artículo\ i}{Gasto\ total} \qquad (3.2.1.5)$$

- Cálculo de los **índices agregados**: en este caso es el índice referido a diciembre t-1, artículo i, en la provincia p, en el mes m y año t:

$$Índice\ Agregado^{mt}_{dic(t-1)} = \sum_{i \in A} Índice\ Elemental^{mt}_{dic(t-1)} * W_{i,p,dic(t-1)} \qquad (3.2.1.6)$$

Siendo, A = dentro de la agregación A.

De la misma forma se puede calcular el índice por una agregación geográfica R, sustituyendo la A por la R.

- Se deben calcular las **tasas de variación**, es el índice del mes actual (m) entre el índice del mes anterior (m-1) y obtenemos las tasas de variación mensual del mes m y año t:

$$Variación\ Mensual^{\frac{mt}{(m-1)t}} = (\frac{Índice\ Elemental^{mt}_{dic(t-1)}}{Índice\ Elemental^{(m-1)t}_{dic(t-1)}} - 1) * 100 \qquad (3.2.1.7)$$

También podemos obtener la tasa de variación anual, pero en este caso será el índice del mes corriente (m) entre el índice de diciembre del año anterior (t):

$$Variación\ Anual^{\frac{mt}{dic(t-1)}} = (\frac{Índice\ Elemental^{mt}_{dic(t-1)}}{Índice\ Elemental^{dic(t-1)}_{dic(t-1)}} - 1) * 100 \qquad (3.2.1.8)$$

- Por último, se deben calcular las **repercusiones**, se calcula de cada artículo que ha variado el precio, cuál es la repercusión si todos los precios de los demás artículos hubiesen sido estables. La repercusión mensual de un artículo (i), en el mes (m) y el año (t):

$$Repercusión_i^{\frac{mt}{(m-1)t}} = \frac{Índice\ Elemental^{mt}_{i,dic(t-1)} * Índice\ Elemental^{(m-1)t}_{i,dic(t-1)}}{Índice\ General^{(m-1)t}_{dic(t-1)}} * W_{i,dic(t-1)} * 100 \qquad (3.2.1.9)$$

3.2.2. Proceso de cálculo del "nuevo" IPC

El scanner data se beneficia del uso de las bases de datos existentes en las empresas, lo que lleva a una reducción notable en los costos relacionados con la recolección de datos. Sin

embargo, la incorporación de nuevas bases de datos requiere procesarlas y aplicar un enfoque diferente para calcular el Índice de Precios al Consumidor (IPC).

En primer lugar, se recopila la información de las empresas, lo que implica utilizar las bases de datos existentes sin añadirles ninguna carga adicional de trabajo. Estas bases de datos deben contener: ingresos/valores unitarios, cantidades, nombre del producto, descripción, código EAN[13] (si no existe, el que use la empresa) y la clasificación interna. Las empresas proporcionan estas bases de datos en el formato que consideren conveniente, sin añadir ningún sobrecargo. Por lo tanto, para cada empresa se tendrá un proceso diferente para el tratamiento de estas bases de datos.

Hay dos diferencias conceptuales respecto el método tradicional, ya que ahora no se realiza una recogida de precios, sino que se recogen datos sobre las ventas de las empresas. Por tanto, hay dos diferencias conceptuales muy importantes:

- **Concepto de precio y valor unitario.** El IPC, tal y como hemos definido antes, nos muestra la evolución de los precios de los servicios y bienes, de los hogares españoles. Estos precios se recogen a través de un cuestionario en cada uno de los establecimientos. Ahora registramos el código EAN de cada producto y calculamos el total de ingresos dividido por el total de unidades vendidas. Este proceso se lleva a cabo durante un período de tiempo determinado. Por lo tanto, en lugar de obtener el precio directamente, ahora capturamos su valor unitario.

- **Producto y gama completa de variedades.** Antes, por cada uno de los artículos se recogía únicamente una variedad de este producto. Ahora con el scanner data conseguimos recoger todas las variedades que existan de ese producto en cada supermercado y afinar mejor la variación de los precios.

Así es, con este nuevo método nos encontramos con un gran volumen de datos que requerirán nuevos tratamientos y metodologías para su preparación y validación. Es importante poder procesar y validar adecuadamente estos datos para garantizar su calidad y confiabilidad en el cálculo del Índice de Precios al Consumidor (IPC).

La pregunta clave es cómo gestionar toda esta información para calcular un índice similar al Índice de Precios al Consumidor (IPC). Para lograrlo, se deben establecer las relaciones entre los artículos proporcionados por cada empresa y la clasificación de artículos de COICOP. Esto nos permite clasificar los artículos y relacionar todas sus variedades.

Luego, es necesario seleccionar los artículos que formarán parte de cada grupo, es decir, crear la "cesta de la compra". Anteriormente, la Encuesta de Presupuestos Familiares definía la cesta de la compra, pero ahora cada supermercado tendrá su propia cesta, basada en los ingresos del año anterior y excluyendo los artículos no representativos. Esta cesta se actualiza

[13] EAN: European Article Number.

anualmente, agregando o eliminando productos según los cambios en los hábitos de consumo.

Una vez seleccionados los artículos, se eligen las variedades que se incluirán. No se incluyen todas las variedades, ya que muchas no son representativas y complicarían el cálculo del índice. Se seleccionan solo las variedades cuyos ingresos tienen un tamaño significativo. Esto se puede hacerse de forma estática, anualmente, o de forma dinámica, mensualmente.

Con el enfoque estático, se estiman los productos que ya no se consumen. Con el enfoque dinámico, se reflejan mensualmente todos los productos representativos, ya que esta actualización se realiza mensualmente.

La selección de artículos es específica para cada supermercado y provincia, basada en sus respectivos ingresos.

A continuación, se detallará el proceso de cálculo del IPC utilizando el scanner data. En este punto, las bases de datos ya están preprocesadas, es decir, se ha realizado la selección de artículos y variedades, lo que nos proporciona todas las cestas de compra de cada supermercado. Este procedimiento se llevará a cabo para cada variedad de producto, formato de establecimiento (supermercado, hipermercado, grandes superficies, etc.) y provincia.

Primeramente, se seleccionan las variedades comunes entre el mes m y m-1. Es decir, después de las tres primeras semanas de recogida de datos, se denominarán a estas tres semanas mes m y el anterior m-1.

Ahora se explica paso a paso el cálculo del índice equivalente al IPC mediante scanner data (todo el proceso de cálculo sacado del INE: (INE, Metodología de cálculo del scanner data en el IPC y IPCA, 2020)):

1) Calculamos los porcentajes de peso de cada variedad de los artículos.

   Para cada **variedad (i)** del **artículo (j)**, se calcula el peso por cada variedad de **establecimiento (f)**, cada **provincia (p)**, **mes (m)** y **año (t)**. La nomenclatura será la misma para todas las fórmulas.

$$Peso\ Variedad_{i,f,p}^{m,t} = \frac{Ingreso\ Variedad_{i,f,p}^{m,t}}{\sum_{i \in j} Ingreso\ Variedad_{i,f,p}^{m,t}} * 100 \qquad (3.2.2.1)$$

2) Selección de las variedades que se usaran para el cálculo.

   Seleccionamos por cada artículo aquellas variedades representativas, entre aquellas comunes entre los meses m y m-1.

$$\frac{Peso\ Variedad_{i,f,p}^{m-1,t} + Peso\ Variedad_{i,f,p}^{m,t}}{2} > \frac{1}{Número\ Variedades\ Artículo_{f,p} * \delta} * 100 \qquad (3.2.2.2)$$

Siendo, δ=1.25, permite seleccionar aquellas variedades con un porcentaje de ventas superior al 80%.

Además, se añadirán todas las variedades que formaron parte del mes anterior (m-1), pero en este no han estado incluidas por falta de ventas y no superar el porcentaje. Esto se mantendrá tres meses, a partir de los tres meses, si no supera el umbral, la variedad, será eliminada.

3) Calculamos el valor unitario de cada unidad.

En este caso, para cada variedad (i) del artículo, se calcula el valor unitario de establecimiento (f), cada provincia (p), mes (m) y año (t). Este valor unitario no tiene que ser el mismo que el precio del producto, ya que para calcularlo se divide todos los ingresos de la variedad entre las unidades vendidas de la variedad.

$$Valor\ Unitario\ Variedad_{i,f,p}^{m,t} = \frac{Ingreso\ Variedad_{i,f,p}^{m,t}}{Unidades\ Vendidas\ Variedad_{i,f,p}^{m,t}} \qquad (3.2.2.3)$$

4) Cálculo de la variación mensual del artículo.

Calculamos la media geométrica de los valores unitarios del mes m y m-1, para cada año (t), para cada artículo (j), establecimiento (f) y cada provincia (p):

$$VU\ Actual_{j,f,p}^{m,t} = \sqrt[n_{j,f,p}^{m,t}]{\prod_{i=1}^{n_{j,f,p}^{m,t}} Valor\ Unitario\ Variedad_{i,f,p}^{m,t}} \qquad (3.2.2.4)$$

$$VU\ Anterior_{j,f,p}^{(m-1),t} = \sqrt[n_{j,f,p}^{m,t}]{\prod_{i=1}^{n_{j,f,p}^{m,t}} Valor\ Unitario\ Variedad_{i,f,p}^{(m-1),t}} \qquad (3.2.2.5)$$

Siendo, n = número de variedades del artículo.

Con estas dos fórmulas quedará calculada la variación de los precios y se calcula la variación mensual (m), del año (t), del artículo (j), establecimiento (f) y cada provincia (p), como:

$$\Delta_{j,f,p}^{m,t} = \left(\frac{VU\ Actual_{j,f,p}^{m,t}}{VU\ Anterior_{j,f,p}^{(m-1),t}} - 1 \right) * 100 \qquad (3.2.2.6)$$

5) Cálculo de los índices elementales.

Estos índices son la expresión más básica a la que se puede llegar. En el IPC "tradicional" se calcula un índice elemental por cada artículo en cada una de las provincias. Con el scanner data será lo mismo, pero se calculará por cada artículo, formato y provincia.

Para el cálculo de estos índices se realiza con base en diciembre del año anterior (t-1), para cada mes (m), año (t), del artículo (j), establecimiento (f) y cada provincia (p):

$$Diciembre_{(t-1)} I_{j,f,p}^{m,t} = Diciembre_{(t-1)} I_{j,f,p}^{(m-1),t} * \left(1 + \frac{\Delta_{j,f,p}^{m,t}}{100} \right) \qquad (3.2.2.7)$$

6) Cálculo de las ponderaciones.

Las ponderaciones son la relación cada artículo entre el gasto de este y el gasto total de todos los artículos vendidos por el establecimiento. Por tanto, se obtendrá una ponderación para cada establecimiento:

$$W_j^t = \frac{Ingresos\ del\ artículo\ j\ en\ el\ año\ (t-1)}{Ingresos\ totales\ en\ el\ año\ (t-1)} \qquad (3.2.2.8)$$

7) Cálculo del índice agregado funcional dentro de una provincia.

Obtenemos los índices de cualquier artículo del scanner data, juntando los diferentes formatos de establecimiento de las cadenas. Este índice es calculado con base en diciembre del año anterior (t-1), para cada mes (m), año (t), de la subclase (S) y cada provincia (p):

$$Diciembre_{(t-1)} I_{S,p}^{m,t} = \sum_{j \in S} \sum_{f=1}^{\varepsilon} Diciembre_{(t-1)} I_{j,f,p}^{m,t} * W_{j,f,p}^{t} \qquad (3.2.2.9)$$

S = subclases de ECOICOP[14], la nueva clasificación europea de consumo.

También se pueden calcular los índices para cada subclase ECOICOP y agregados geográficamente de una agregación funcional.

Por último, se debe de saber calcular la integración del IPC "tradicional" con el IPC del scanner data. Se realizará a nivel de subclase y provincia, ya que es lo que hay en común entre las dos cestas, la del "tradicional" y el scanner data.

1) Cálculo de las ponderaciones.

   Se calculan las ponderaciones del IPC "tradicional" más el scanner data y se repartirán proporcionalmente. Se tiene en cuenta que el peso será el mismo para todas las subclases del scanner data y en sus agregados.

   Para cada artículo j de la cesta del scanner data, dentro de la **subclase (S)**, las ponderaciones serán:

$$Scanner\ Data\ WW_{j,f,p}^{t} = \ Recogida\ tradicional W_{S,p}^{t} * Coef_{p}^{C} * \frac{Scanner\ Data\ W_{j,f,p}^{t}}{\sum_{j \in S} \sum_{f} Scanner\ Data\ W_{j,f,p}^{t}} \qquad (3.2.2.10)$$

   $Coef_{p}^{C}$= el peso de la **cadena (C)** en la recogida tradicional, en la provincia (p).

   Para cada artículo **k** de la cesta tradicional, dentro de la **subclase (S)**, las ponderaciones serán:

$$Recogida\ tradicional\ WW_{k,p}^{t} = \ Recogida\ tradicional W_{k,p}^{t} * (1 - Coef_{p}^{C}) \qquad (3.2.2.11)$$

   Ponderación de la subclase S en la provincia p será:

$$WW_{i,p}^{t} = \sum_{j \in S} \sum_{f} Scanner\ Data\ WW_{j,f,p}^{t} + \sum_{k \in S} Recogida\ tradicional\ WW_{k,p}^{t} \qquad (3.2.2.12)$$

2) Cálculos de índices agregados (diciembre de (t-1)).

   Con estas ponderaciones y con los índices elementales del scanner data (calculados en el paso 5) y del IPC "tradicional", se obtienen los índices con una agregación funcional **A**, referido a diciembre del año anterior:

[14] ECOICOP: European Classification of Individual Consumption by Purpose.

$$Diciembre_{(t-1)}I_{A,p}^{m,t} = \sum_{j\in A}\sum_{f} Diciembre_{(t-1)}I_{j,f,p}^{m,t} * Scanner\ Data\ WW_{j,f,p}^{t} + \sum_{k\in A} Diciembre_{(t-1)}I_{k,p}^{m,t} * Recogida\ tradicional\ WW_{k,p}^{t} \quad (3.2.2.13)$$

3) Cálculo de los índices en base 2016:

$$16I_{A,p}^{m,t} = 16I_{A,p}^{Diciembre(t-1)} * \frac{Diciembre_{(t-1)}I_{A,p}^{m,t}}{100} \quad (3.2.2.14)$$

3.2.3. Diferencias entre el IPC "tradicional" y el obtenido mediante scanner data

En este apartado se tratan las principales diferencias que hay entre los dos métodos el scanner data y el IPC.

La diferencia principal está en el volumen de datos entre el scanner data y el método tradicional, siendo mucho mayor en el scanner data que en la recogida tradicional. Esto es debido a que la información se obtiene automáticamente de la línea de caja de las empresas. Estas únicamente tienen que enviar las bases de datos en los períodos de tiempo acordados, sin introducir ninguna modificación en ellas, únicamente con las variables necesarias acordadas al inicio del proyecto, que son: ingresos/valores unitarios, cantidades, nombre del producto, descripción, código EAN y la clasificación interna.

El web scraping[15] puede ser una herramienta útil para obtener datos de una página web si la empresa no proporciona la base de datos necesaria, pero es importante hacerlo de forma ética y legal, respetando las normas de protección de datos y los derechos de privacidad. La recopilación tradicional, por otra parte, requiere que un gran número de personas que realicen la recopilación de datos mediante: visitas personales a los supermercados, por teléfono, catálogos, faxes, Internet o correo electrónico. Además, esta recopilación de datos se produce entre el 1 y el 22 de cada mes, pero solo se recopila de 2 a 3 veces por artículo.

Por otro lado, con el scanner data[16], los datos se recopilan cada vez que un artículo pasa por la caja, por lo que el tiempo para recopilar datos lleva mucho más tiempo que con los métodos tradicionales. Otra cuestión a tener en cuenta es que los datos del scanner data recopilan todas las variedades existentes de cada artículo, mientras que los métodos tradicionales sólo utilizan una. En consecuencia, los datos del scanner data son mucho más detallados que los

[15] Es una técnica de extracción de datos automatizada que consiste en recopilar información de páginas web de manera estructurada para su posterior análisis.

[16] Es una técnica de recopilación de datos automatizada de las transacciones de venta de los supermercados.

métodos tradicionales porque se centran en una muestra específica de artículos y no capturan todas las transacciones de compra. Al mismo tiempo, el scanner data nos proporciona una mayor granularidad de los datos[17].

Existen diferencias en la composición de la cesta de la compra cuando se emplean ambos métodos. Para el scanner data esta cesta se compone de aquellos artículos que pasan por la línea de caja y todas sus variedades, posteriormente se calculan si son significativos para incluirlos en la cesta. Por otro lado, en el método tradicional, la selección de la cesta de la compra se realiza de manera más minuciosa a través de la Encuesta de Presupuestos Familiares (EPF). En contraste, en el scanner data, la cesta de la compra puede actualizarse mensual o anualmente según las tendencias de compra, mientras que en el método tradicional se realiza con una menor frecuencia.

Otra de las diferencias muy importantes, son las diferencias conceptuales, ya que ahora no se realiza una recogida de precios sino datos sobre las ventas de las empresas. Por ello, se deben de tener claras las diferencias de concepto de precio y de valor unitario, producto y gama completa de variedades. Entonces, es importante considerar que criterios usar para el cálculo de los valores unitarios y como integrar los datos sobre los precios que se usan en el IPC tradicional.

A diferencia del cálculo del IPC tradicional, con el scanner data se recoge una mayor cantidad de precios. Por lo consiguiente, se captan las fluctuaciones de los precios con una mayor precisión, pero a veces estás, son irreales, ya que han tenido alguna oferta o descuento, por ello se debe de usar fórmulas de índices multilaterales[18], es decir, un índice que combina diversas variables o indicadores.

3.3. Ventajas y desventajas del scanner data

Esta nueva metodología de cálculo del IPC ofrece muchas ventajas y a su vez algunas desventajas o retos que se tendrán que abordar para su correcta implementación. Las ventajas superan con creces a las desventajas, ya que la mayoría de estas últimas representan desafíos que requieren una inversión considerable de tiempo al principio. Sin embargo, una vez se llega a la fase final y solo se requiere un seguimiento, por lo que proporciona numerosas ventajas.

En primer lugar, se tratarán las ventajas del scanner data:

1) La más importante es que se obtendrán las bases de datos con mayor rapidez al mismo tiempo que la grabación de los datos. Por tanto, se eliminará el proceso de recogida de datos tradicional y todos sus costes. Al mismo tiempo, se eliminarán los errores de muestreo de realizar las encuestas manualmente. En consecuencia, se

[17] Implica que se dispone de información más detallada y específica de los datos.

[18] Los índices multilaterales son herramientas que comparan la variación de precios entre diferentes países o regiones, permitiendo entender las diferencias en los niveles de precios y la inflación.

tendrá una información más optimizada porque tendrá una mejor estructura, detalle y calidad.

2) La recogida de datos es más rápida, por ello, se pueden tomar muchos más datos que con la recogida tradicional. Es decir, tendremos una mayor cobertura de los productos consumidos. Se recogerán mejor las fluctuaciones de los precios de los productos, al mismo tiempo que se recogen una mayor gama de artículos y sus variedades.

3) Mejorar la representatividad del IPC, reflejando mejor las tendencias de los precios y los patrones de gastos de las familias. Es decir, se obtendrá una mayor precisión y actualización de los datos. Como resultado, el IPC medirá mejor la inflación.

4) Al mismo tiempo tendremos una mayor transparencia en el índice, ya que las fuentes de datos son más objetivas y con una mayor facilidad para verificarlas.

5) Mayores controles de calidad de los datos, ya que, al tener la información estructurada, se podrá dedicar un mayor tiempo a la supervisión y fiabilidad de esta información, por tanto, la muestra será de una mayor calidad. Por lo consiguiente, si hay fuertes variaciones de precios y cantidades en alguno de los artículos, se podrá detectar con una mayor facilidad.

6) Mejor actualización de la cesta de bienes, gracias a la información directa de la línea de caja, lo que permite añadir nuevos productos representativos y la eliminación de aquellos que ya no lo son. Por tanto, se detectarán mejor los cambios en el mercado.

7) La obtención de las bases de datos no supone ningún coste adicional para las empresas, directamente mandan la información que se les pide, pero sin modificar su contenido ni estructura.

8) Las bases de datos suponen una gran ventaja, ya que se obtiene una gran información dentro de ellas, contienen información de cada producto que pasa por la línea de caja: artículo, su código, las unidades vendidas, el precio y la fecha de la venta. Por tanto, se obtiene una mayor información de todos los artículos vendidos.

9) Unificación de las estadísticas a nivel europeo. Actualmente existe el IPCA que es un índice que es comparable con los diferentes miembros de la UE. Pero con el scanner data al necesitar un nuevo proceso de cálculo se podría hacer un proceso único para todos los estados miembro de la UE. De esta forma, se prescindiría de un estadístico simplificando la información.

También se tienen en cuenta las desventajas/retos de este nuevo método:

1) Gran volumen de datos, todos estos datos son recogidos por cada uno de los supermercados. Por ello, se debe de tener un proceso diferente para tratar cada una de las bases de datos de los diferentes supermercados. Esto al inicio supone un gran costo computacional y técnico, a su vez una supervisión de cada uno de estos

procesos. Es decir, realizar un nuevo diseño para el tratamiento y procesamiento de estos nuevos datos.

2) Como se obtienen tantas bases de datos como diferentes supermercados, hay que compilarlos y esto supone un gran costo de tiempo y armonización de estas, para obtener una base de datos estructurada y con las mismas variables.

3) Las empresas deben de comprometerse a enviar recurrentemente y con constancia los datos al INE. Además, el inicio del proyecto sí que habrá una mayor carga para los informantes porque se debe de decidir los campos necesarios en la base de datos, el proceso de envío y la frecuencia del envío.

4) Falta de cobertura, aunque el scanner data aborda una mayor cantidad de información que el método tradicional, este podría no acabar de representar bien toda la gama de productos consumidos en los hogares españoles, por ello, podríamos tener una menor precisión.

5) Errores en la obtención de los datos, como los datos provienen de la línea de caja, podrían tener errores como un mal escaneo de los códigos o de las cantidades de los productos y en consecuencia sesgarnos la muestra.

6) Gran dependencia a la tecnología, si las líneas de caja fallan y no almacenan bien los productos podría haber problemas con la representatividad de los datos.

7) La recopilación de los datos con el scanner data podría suponer un reto ante la privacidad y seguridad de los datos, ya que los clientes no dan el consentimiento para su uso o si se utilizan para otro tipo de fines.

3.4. Implementación scanner data en otros países

Actualmente, ya son muchos países que han implementado o están desarrollando la metodología del scanner data. La Unión Europea promueve esta nueva metodología para poder sacar estadísticos de mayor calidad, aunque alrededor de todo el mundo también se está desarrollando. El scanner data tiene presencia: en Francia, Holanda (lleva 15 años en desarrollo), Luxemburgo, Italia, Bélgica, Noruega, Suecia, Estados Unidos, Chile, Brasil, Australia, Japón, Corea de Sur...

CEPAL, es la Comisión Económica de América Latina y el Caribe, es la encargada de realizar investigaciones, análisis y asistencia técnica en muchas áreas, una de ellas el desarrollo de estadísticos, que incluyen indicadores económicos como el PIB, la inflación y la tasa de desempleo, así como indicadores sociales, ambientales, demográficos y comerciales. También la BSL, Oficina de Estadísticas Laborales de Estados Unidos. La ONS, Instituto de Estadísticas Nacionales del Reino Unido. La ABS, Oficina Australiana de Estadística. La INSEE en Francia, el Instituto Nacional de Estadística y Estudios Económicos. Todas estas instituciones y oficinas, entre otras, están en desarrollo de esta nueva metodología, se pueden encontrar en diferentes etapas y entre algunos países la implementación puede variar. Pero

el hecho que el scanner data este en presencia de tantos países no es una casualidad, es debido a que es un muy buen método para el nuevo cálculo del IPC.

4. WEB SCRAPING

El scanner data no es el único proyecto que está llevando a cabo el INE, también están trabajando en web scraping, es decir, es otra fuente de extracción de datos. En este caso, consiste en la obtención automatizada de la información sobre precios de las páginas web de los supermercados. El problema es que las páginas web están protegidas, por tanto, el INE necesita pactar acuerdos para poder acceder a ellas para extraer la información necesaria y realizar el cálculo del IPC. Por otro lado, en el scanner data, los propios supermercados nos proporcionaban las bases de datos, sin necesidad de extraerlas de la web, y luego se realizaba el cálculo del IPC. Sin embargo, esta nueva fuente, al igual que el scanner data, nos acorta los plazos de publicación y nos proporciona estadísticos con una mayor precisión.

4.1. ¿Qué es el web scraping?

El web scraping es el proceso el cual se recoge información de forma automática a través de un programa de múltiples páginas web. Por tanto, el objetivo de un scraper será obtener grandes volúmenes de datos de manera automática. Para realizarlo, es necesario programar diversas aplicaciones o códigos que extraigan la información que se desea obtener. Los datos pueden incluir: texto, imágenes, archivos PDF, tablas, entre otros. Posteriormente, los datos son almacenados en una base de datos para el uso que se desee dar. Es decir, es una herramienta muy útil para poder sacar datos de la web, pero mucha de esta información puede no estar disponible por la confidencialidad de los datos.

Se pueden scrapear múltiples datos en la web, por ejemplo: las webs gubernamentales, las redes sociales, webs de búsqueda, empresas… Pero no siempre todos estos datos están abiertos, depende del sitio web que se intente acceder se deberá aplicar algunas herramientas o trucos para poder extraer la información.

La forma más sencilla es acceder a través de una API[19] (interfaz de programación de aplicaciones) del sitio web al que se intenta acceder. Si la web proporciona su API, se podrá extraer la información de forma sencilla y en el formato conveniente. Twitter, Facebook, LinkedIn y Google, por ejemplo, proporcionan este tipo de herramientas. No obstante, muchas veces esta API no tiene los datos que queremos o en el formato adecuado. Por tanto, cuando se quieren extraer datos y no tenemos la API del sitio web, los datos no están en el formato deseado, no se permite la integración de los datos del mismo web, el volumen de datos es insuficiente o la velocidad de descarga, el web scraping es necesario. Este proporciona un acceso prácticamente ilimitado a todos los datos.

[19] API: Application Programming Interface.

Aunque esta metodología tiene algunos inconvenientes a tratar, sobre todo la legalidad de este. Existe la Ley de Protección de Datos o Privacidad de Datos[20], las cuales nos restringen el uso de datos, sobre todo cuando son datos personales se podría incurrir un delito, aunque los datos sean públicos no da el derecho de scrapearlos directamente.

El web scraping tiene múltiples aplicaciones, como:

- Estudio de mercado: las empresas lo usan para extraer información de sus competidores y les ayuda ajustar sus estrategias comerciales.
- Reforzar estrategias de comercio electrónico.
- Automatización del negocio: cuando se necesitan sacar datos de diferentes sitios web para un fin, se automatiza.
- Generación de clientes: crear listas de clientes potenciales (dirección de correo electrónico y números de teléfono).
- Seguimiento de precios, es una de las aplicaciones más comunes ya que tiene múltiples usos, como el control de precios.
- Tratamiento de redes sociales, como extraer comentarios de las redes sociales y poder crear un modelo predictivo para detectar a los haters. Para realizar análisis de sentimientos o identificar tendencias.
- Generar mapas de restaurantes o casas accediendo a través de Google Maps.
- Visualizar noticias recientes para controlar la bolsa y actuar en consecuencia. También para mantenerse informado sobre las tendencias, temas populares o noticias falsas.
- Análisis financiero: los inversiones y analistas financieros pueden recopilar datos sobre acciones y otras fuentes.
- Inmobiliaria, para comparar precios o características de las casas, apartamentos o pisos.
- En general, nos abre nuevas fronteras para la creación de conocimiento y nos permite realizar pronósticos

Este nuevo método proporciona grandes ventajas. Entre ellas la automatización de recopilación de datos de forma eficiente, que permite ahorrar un gran costo de tiempo y monetario si se compara con la recopilación manual. Acceso a una gran cantidad de datos, además estos pueden estar estructurados o no, al mismo tiempo que se tiene una actualización continua de estos.

[20] https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673

En el web scraping, hay desventajas importantes que deben tenerse en cuenta, que son:

- Legalidad y ética: siempre es fundamental revisar las condiciones legales del sitio web del cual se desean extraer datos y asegurarse de no hacerlo de manera abusiva.

- Mantenimiento del Código: Internet es muy cambiante, por lo que las páginas web pueden cambiar o incluso desaparecer, lo que requiere un constante mantenimiento del código.

- Dificultad Técnica: el desarrollo de scrapers requiere un sólido conocimiento de programación, HTML y otras tecnologías.

- Bloqueo por parte de los sitios web: los sitios web pueden bloquear la dirección IP si detectan actividades inusuales, lo que puede afectar a la extracción de datos.

- Responsabilidad en el uso de los datos: existe una gran responsabilidad legal y ética en el uso de los datos extraídos, los cuales deben ser utilizados de manera responsable y en conformidad con las leyes vigentes.

4.1.1. Limitaciones de las API

Una API es una herramienta proporcionada por el administrador de la web que permite obtener sus datos, es decir, dan acceso directo a los datos que se desea extraer.

Idealmente acceder a una API, es la mejor forma de extraer datos de la web. El proveedor de esta nos garantiza a que los datos tengan un formato estándar y bien estructurados. Por tanto, primeramente, antes de realizar web scraping se comprobará que el sitio web tiene las API. En el caso de la existencia de API, las utilizaremos, pero en ocasiones, aunque tengamos acceso a ellas, pueden surgir algunos inconvenientes:

- Existe la API, pero esta no es gratuita y el acceso web sí lo es. Por ejemplo, la API de Twitter y Google Maps, el acceso a la web es gratuito para los usuarios, pero el acceso a la API requiere un plan de pago.

- La API limita el número de accesos y peticiones (por día, segundo, hora…). La mayoría de las webs limitan el acceso. Por ejemplo, Twitter impone el límite para las cuentas de usuario gratuitas de realizar 900 solicitudes en 15 segundos. El mismo ejemplo para GitHub, para los usuarios autenticados el límite es 5.000 solicitudes por hora, aquellos no autenticados el límite son 60 solicitudes por hora.

- La API no permite recolectar toda aquella información de interés que se muestra en la web. Por ejemplo, en LinkedIn la API no muestra toda la información de los perfiles de los usuarios, en cambio, dentro de la web si que muestra esta información.

En estos tres casos, o en el caso de no tener una API se realizará web scraping.

4.1.2. Legalidad del web scraping

Cuando se realiza web scraping debe de tenerse en consideración una serie de aspectos legales y éticos. El uso de este es legal, cuando un sitio web publica sus datos puedes acceder a ellos, pero no todos los datos son legales para scrapear. Por ejemplo, datos personales o datos de propiedad intelectual (protegidos por leyes), pueden provocar sanciones. Es decir, que los datos sean públicos no quiere decir que el administrados haya dado permiso para scrapearlos.

En primer lugar, debe de tenerse en cuenta que las personas que realizan web scraping deben de conocer las leyes vigentes y la seguridad cibernética aplicada a esta actividad. Por consecuencia, es responsabilidad de cada persona cumplir con los términos de uso especificados por el sitio web que se quiere extraer los datos. En España, esta actividad es legal, el problema principal es el uso que se le dan a los diferentes datos.

Antes de realizar web scraping debemos de tener en cuenta que suele estar permitido si los datos que extraemos están publicados públicamente y no se debe de realizar ningún inicio de sesión, además hay que seguir las siguientes peticiones:

- Obedecer al fichero robots.txt, es un archivo que se encuentra en los servidores de cada sitio web y se utiliza para controlar el acceso de los robots a determinadas partes de la web o acceso total.

- No afectar al rendimiento del servidor, responsabilidad del usuario asegurarse no sobrecargarlo e influir en el funcionamiento de la web.

- Cumple todos los términos y condiciones de uso del web scraping, la mayoría de las webs incluyen términos y condiciones de uso, normalmente se encuentra en la parte inferior de la web, donde nos lleva a un enlace o directamente se visualizan las condiciones de esa web. Palabras clave a buscar dentro de estos archivos: robot, automatizar y extraer. Al entrar a una web se pide aceptar estos términos y al ser aceptados se deben de cumplir para no tener problemas legales o sanciones.

- Respetar el copyright.

- No infringir los derechos de autor o marca registrada: hay información con uso limitado protegido por los derechos de autor y esa información no se puede usar sin el permiso explícito del titular de los derechos.

- Seguir la ley de propiedad intelectual y datos personales.

4.1.2.1. Robots.txt

El archivo robots.txt es un fichero que se encuentra dentro de las webs que nos indica como interactuar con la web. En este se especifica que se puede hacer con un mecanismo automatizado de extracción de datos, es decir, las restricciones para tener en cuenta cuando se pretende rastrearlas. No todas las webs lo incluyen, a veces están incluidos dentro de los

términos y condiciones del uso del sitio web. Estas restricciones no son obligatorias, pero si una sugerencia para reducir las posibilidades de ser rastreado, bloqueado y evitar problemas legales futuros.

El archivo robots.txt se utiliza para guiar a los robots, especialmente durante la recopilación de datos de gran tamaño. Para acceder a él usamos: *http://www.nombredelaweb/robots.txt*. Esta dirección nos llevará a una página donde se encuentra el siguiente contenido:

- User-agent: *

  Indica a quien se aplican las condiciones. Si aparece un * se aplican a todos los robots. A veces, aparece el nombre de algún robot en concreto.

- Allow: /

  Indica las rutas de acceso de las que se puede extraer los datos. Si aparece / es a todas.

- Disallow: /*/alt-*.html

  Especifica todas aquellas rutas dentro del sitio web a las cuales no da autorización para extraer datos de forma automatizada. En este caso, todas aquellas que incluya alt-.

- Crawl-Delay: 3

  Indica cuanto tiempo hay que dejar entre las peticiones que se hagan al sitio web. Esto permite que el funcionamiento de la web sea el correcto y no se vea afectado por el raspado.

Ejemplos:

- Permiso de acceso completo a todos los robots:

  User-agent: *
  Disallow:

- Permiso de acceso a un solo robot:

  User-agent: Carrefour
  Disallow:
  User-agent: *
  Disallow: /

- Exclusión de páginas completas:

  User-agent: *
  Disallow: /browse/*

Disallow: /HuchaPromocion*
Disallow: /e-commerce/www/*

4.1.3. Adaptación del web scraping al IPC

Al igual que en el scanner data, se debe de pasar por un proceso de adaptación para poder realizar el cálculo del IPC. En esta nueva metodología extraemos la información directamente de las webs de los supermercados, por tanto, se necesitará un proceso diferente para cada web que se extraigan datos. Además, igual que en el scanner data, se necesitará un proceso de tratado de datos para cada uno de los supermercados.

El objetivo es extraer los datos del formato HTML y convertirlos en información que pueda ser almacenada en hojas de cálculo y bases de datos, facilitando así su posterior análisis.

En primer lugar, se realizará la extracción datos, es decir, se creará un programa para cada una de las páginas web que quieran scrapear. Serán múltiples programas informáticos que se adecúen a cada una de las páginas web. Una vez se tengan creados se realizará una extracción de información semanal. Estos programas están diseñados principalmente para extraer información sobre los precios de cada empresa en el formato requerido. Se pueden llevar a cabo dos tipos de extracción de datos:

- Extracción completa: extraer toda la información de la web, es decir, de todos los productos, sus respectivos precios y características.

- Extracción selectiva: se extraen aquellos productos incluidos a la muestra previamente seleccionada para el cálculo del IPC.

Ahora se realiza una adaptación de la clasificación a la ECOICOP, es decir, se cogen cada uno de los productos y se clasifican según las diversas parcelas de la ECOICOP. Además, se debe de tener un constante control de calidad de la información. En este punto, sería el procesamiento de las bases de datos, es decir, mirar si hay productos comunes, comprobación de precios de los productos, productos nuevos… Una vez se obtienen todas las bases de datos, se pasa a la fase de cálculo del índice. En este caso, a diferencia del scanner data, se calcula un índice para cada página web. Se calculan los precios medios de cada producto, sus índices y sus agregaciones.

Por tanto, el web scraping es otra herramienta muy potente que puede sustituir a la captación de precios de forma manual.

4.2. Pasos iniciales del web scraping

El web scraping tiene dos pasos muy importantes, el primero navegar automáticamente por la web que desees scrapear y el segundo extraer los datos que desees. Estos dos pasos se realizan con *scrapers* y *crawlers*. Los *crawlers*, también conocidos como “arañas” navegan por el web buscando la información, estos son los que guían a los *scrapers*. Los *scrapers* son los encargados de obtener la información deseada del sitio web.

Web scraping tiene múltiples lenguajes de programación que pueden ser usados para llevarlo a cabo. Pero independientemente del lenguaje de programación usado, inicialmente se deben de seguir una serie de pasos:

Para los pasos siguientes se ha elegido la web https://www.dia.es/ para realizar web scraping como ejemplo.

1) Revisar el archivo robots.txt.

   Inicialmente, se debe revisar este archivo, ya que es una sugerencia de como interactuar con el sitio web (explicado en el apartado 4.1.2.1.). En este caso, se revisará el archivo robots.txt del DIA (https://www.dia.es/robots.txt).

Figura 4.1: Robotx.txt DIA I

```
# For all robots
User-agent: *

# Allow search crawlers to discover the sitemap
Sitemap: https://www.dia.es/sitemap.xml

# Disallow
Disallow: /pt/*
Disallow: /en/*
Disallow: /ca/*

# Pagination
Disallow: */pag-*
Allow: */pag-1/*
Allow: */pag-2/*
Allow: */pag-3/*
Allow: */pag-4/*
Allow: */pag-5/*

# Block access to specific groups of pages
Disallow: /cart
Disallow: /checkout
Disallow: /my-account
Disallow: /thank-you
Disallow: /*/reviewhtml/
Disallow: /products/
Disallow: /productes/
Disallow: /prueba-compra-online
Disallow: /l/politica-cookies
Disallow: /l/aviso-legal
```

Fuente: DIA robotx.txt

En esta primera parte del archivo, figura 4.1, se dirige para todos los agentes (*user-agent:**). Primeramente, en el *Sitemap* indica donde se puede encontrar el archivo XML, dentro de este se encuentran todas las URL del sitio web que el propietario deja scrapear y nos aporta información adicional sobre cada una de ellas como: la URL, la frecuencia de actualización y la importancia en la web. En las primeras líneas con la directiva *Disallow* nos indica que los scrapers no deben rastrear las URL que comiencen por */pt/*, /en/** y */ca/**, seguramente no se puedan rastrear el contendido en portugués, inglés y catalán. Las siguientes directivas de *Disallow* bloquean el acceso a ciertas páginas del sitio web: */cart* bloquea el acceso a la página del carrito de compras, */checkout* bloquea al acceso a la página de proceso de pago, */my-account* bloquea el acceso a la página de mi cuenta… Por último, para la parte de paginación se bloquea el acceso a cualquier URL que contenga */pag-* en su ruta, pero con la directiva *Allow* se permite el acceso desde las páginas 1 a la 5.

Figura 4.2: Robotx.txt DIA II

```
# Block CazoodleBot as it does not present correct accept content headers
User-agent: CazoodleBot
Disallow: /

# Block MJ12bot as it is just noise
User-agent: MJ12bot
Disallow: /

# Block dotbot as it cannot parse base urls properly
User-agent: dotbot/1.0
Disallow: /

# Block Gigabot
User-agent: Gigabot
Disallow: /

# Amazon's user agent
User-agent: Amazonbot
Disallow: /
```

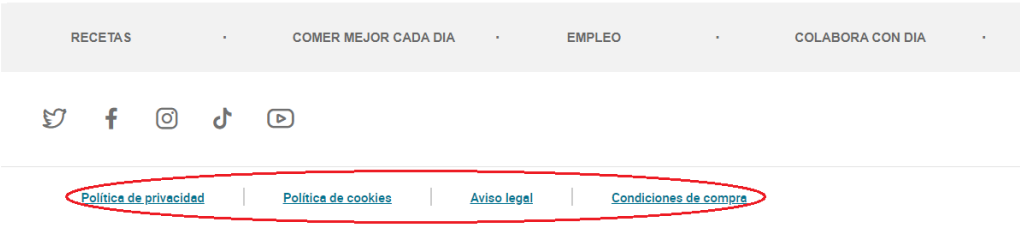
Fuente: DIA robotx.txt

En este caso, se bloquea el acceso a ciertos robots de búsqueda al sitio web. El primer robot bloqueado es *CazoodleBot*, este no presenta correctamente los encabezados de aceptación del contenido. El siguiente, MJ12bot, se considera que es ruido, es decir, su actividad no es relevante para el sitio. Al *dotbot/1.0*, especifica que no puede analizar correctamente las URL base. Al *Gigabot* y *Amazonbot* se les bloquea el acceso completo al sitio web. Estas restricciones o bloqueos pueden ser debido a problemas técnicos, falta de relevancia o cualquier otro motivo que el administrador haya considerado.

2) Revisar los términos y condiciones de la web.

   Se encuentran buscando directamente el nombre de la web y seguidamente términos y condiciones, nos aparecerá el enlace a la web o PDF (DIA, 2023). Por otro lado, se puede ir al final de todo de la web y allí, normalmente, se especifica toda la política de uso de la web.

Figura 4.6: Donde encontrar TyC de DIA



Fuente: dia.es

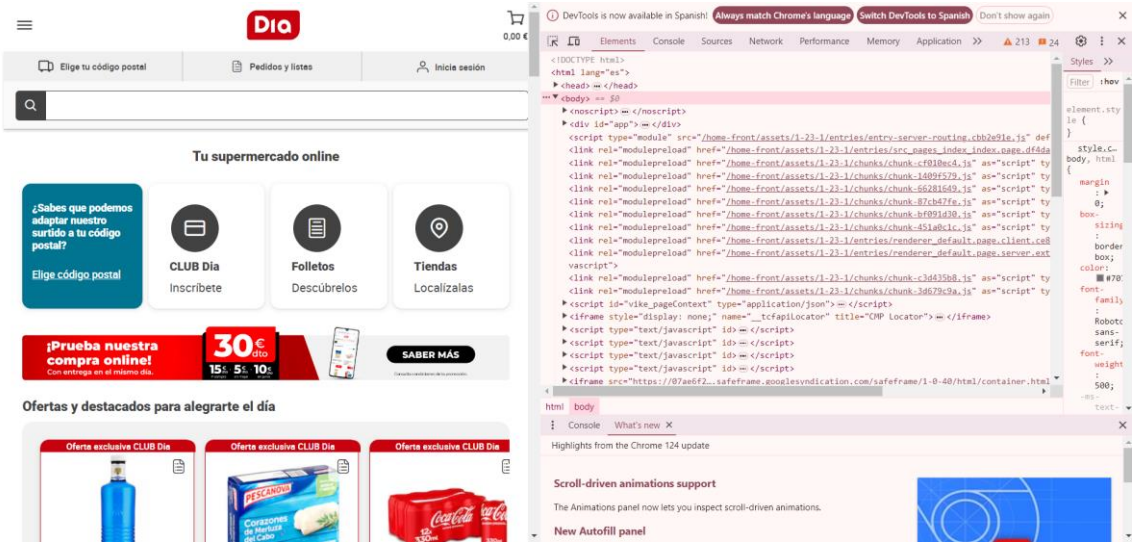
4.2.1. Funcionamiento de la web

Una página web funciona de la siguiente forma, hay clientes que hacen sus peticiones y un servidor manda las respuestas de las peticiones demandadas. Los clientes son los dispositivos de los usuarios conectados a Internet y el software utilizado (Chrome, Firefox…). Los servidores son sistemas informáticos que dan servicios y recursos a otras computadoras dentro de una web, estos son robustos, confiables y diseñados para manejar cargas de trabajo de gran volumen.

Además, tenemos otros componentes como la IP (Protocolo de Internet), es un número único asignado a cada dispositivo conectado a Internet, sirve para comunicar cómo deben viajar los datos. El DNS (Sistema de Nombres de Dominio) es un sistema que traduce los nombres de dominio leíbles por humanos en direcciones IP numéricas para los ordenadores, así enviar las direcciones HTTP[21] correctamente. HTTP es el idioma para que los clientes y servidores se puedan comunicar entre ellos, facilitando la solicitud y entrega de recursos. Cuando se busca algo por Internet, el navegador va al servidor DNS y encuentra la dirección real del servidor del sitio web. A través de tu dirección IP, se te envían todos los datos a través de la petición HTTP al servidor. Una vez el servidor envíe la información, el navegador junta toda la información para que se pueda visualizar el sitio web.

Aunque lo interesante para realizar web scraping es conocer el código de HTML. Una web es un archivo plano, este archivo tiene un código que corresponde a todos los elementos que hay dentro de ella, esto hace que se visualice de forma correcta. Para acceder a las herramientas de desarrollo de Chrome usaremos el comando Ctrl+Shift+i.

Figura 4.7: Código HTML del DIA



Fuente: dia.es

Tenemos mucho contenido dentro de este código:

- <h1>…</h1>: nos muestra el título más grande
- <h2>…</h2>: título un poco más pequeño
- … : el párrafo
- …: una lista ordenada y cada elemento viene representado por …

Para que todo no tenga la misma tipografía se pueden crear las clases como <h1 class="nombre-de-la-clase">…</h1>, luego se podrá definir las características de cada clase en una hoja de estilo. También existen los “id”, estos solo se pueden usar una vez en todo el documento y son muy útiles para webs con interacción, por ejemplo, cuando se desliza el

[21] HTTP: Hypertext Transfer Protocol.

cursor por encima de algún elemento y aparezca un enlace (normalmente, se usa con JavaScript).

Si se amplía el código HTML del DIA observaremos múltiples elementos:

Figura 4.8: Código HTML del DIA

Fuente: dia.es

En la figura 4.8, se muestra un fragmento de código HTML:

- <html>: representa la raíz de todo el código, todos los elementos descienden de esta.
- <body>: solo puede aparecer una vez y es el contenido del documento HTML.
- : etiqueta obligatoria en el contenido de texto, para organizar bloques o secciones.
- <noscript>: sirve para crear contenido dinámico y aplicaciones web, son lenguajes de scripting y el más común es JavaScript.
- <script>: igual que el anterior

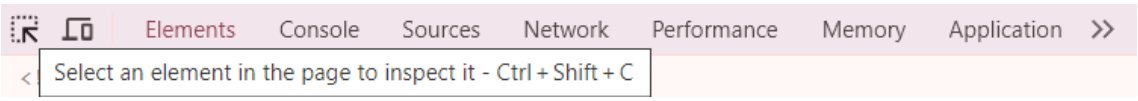
Dentro de cada elemento mencionado hay unas flechas desplegables donde aparecen más líneas de código con más elementos, no se definirán todos los elementos, pero para la comprensión del código se ha usado una guía[22].

Todo el código de la web es muy importante conocerlo para poder sacar los elementos que necesitamos al realizar web scraping y no sacar todo el contenido de la web, ya que habría mucha información no útil y por ello sería ineficiente. Aunque muchas veces si no conseguimos identificar un elemento, podemos hacerlo al revés. Seleccionamos algún elemento de la página web con el cursor y nos mostrará en qué parte del código podemos encontrar ese elemento. Si ya tenemos abierto el código vamos a la parte superior y hacemos

[22] Véase el enlace: https://perma.cc/95F8-PRLD

clic en el botón de Inspeccionar, si no tenemos el navegador abierto usamos el comando Ctrl+Shift+C, para activar esta opción.

Figura 4.9: Comando inspect



Fuente: dia.es

4.2.2. Herramientas para el web scraping

Hay múltiples softwares y herramientas para realizar web scraping, se han seleccionado aquellas más frecuentes:

- BeautifulSoup[23]

  Biblioteca de Python de código abierto diseñada para extraer datos de archivos HTML y XML. Muy útil para analizar y extraer información de páginas web, pero requiere alto conocimiento de programación. Librería muy buena para principiantes.

- Selenium[24]

  Herramienta de automatización de navegadores web, se puede usar para hacer un rastreo de datos en aquellas webs que tienen interacciones dinámicas o JavaScript. Es decir, imita a la navegación que puede realizar un humano interactuando con la web. Esta puede ser usada en diversos lenguajes: Java, Python …

- Scrapy[25]

  Framework de Python, código abierto para la extracción de datos eficiente de sitios web y rastreo de ella. Se puede personalizar y es muy potente para proyectos de gran escala.

- Octoparse[26]

  Para aquellas personas sin conocimiento de programación y quieren realizar un raspado de datos en un sitio web, herramienta visual. Tiene muchas ventajas como la rotación de la IP y los servicios de almacenamiento en la nube.

- ParseHub[27]

  Otra herramienta de scraping visual muy parecida a Octoparse, permite extraer datos sin necesidad de tener conocimientos de programación.

[23] https://pypi.org/project/beautifulsoup4/

[24] https://www.selenium.dev/

[25] https://scrapy.org/

[26] https://scrapy.org/

[27] https://www.parsehub.com/

- R[28]

  Por último, cabe mencionar que para realizar web scraping con R es posible usando la librería rvest, esta está diseñada específicamente para extraer datos de páginas web. Además, para realizar una verificación del archivo de robots.txt podemos usar la librería *polite*. Otras bibliotecas para tener en cuenta son: *xml2* (trabajar documentos XML y HTML), *httr* (realizar solicitudes en HTTP) y *RSelenium* (automatizar un navegador web).

4.2.3. Consejos para Navegar por la Web de Forma Eficiente y Ética

Para proteger los sitios web, muchos administradores implementan medidas para evitar la extracción no autorizada de datos. Utilizan diversos métodos para prevenir el acceso de programas de extracción de información no eficiente y ética:

- Bloqueo de las direcciones IP: cada dispositivo tiene asignada una IP, si una web detecta que una IP está realizando peticiones masivas al servidor, la web puede bloquear esa IP para que no tenga acceso a la web. Para no ser detectado podemos ir cambiando nuestra dirección IP a través de un proxy o una VPN (redes privadas virtuales). La mayoría de proxies son gratuitas y a su vez menos seguras y las VPN suelen ser de pago, pero más seguras. Esto se debe a que las proxies ocultan tu identidad al sitio web que visitas, pero no para el propio servidor proxy, por tanto, te arriesgas a tener una conexión menos segura.
- Trampas *honeypots*: hay que ir con cuidado, ya que son enlaces que a la vista humana no son detectables, pero los bots y scrapers las visitaran y directamente se bloqueará la dirección IP o agentes de usuario. Son como páginas en blanco que sirven de trampa para detectar a los scrapers y bots.
- Limitar la velocidad de las solicitudes: es importante verificar en el archivo robots.txt si hay algún tiempo establecido entre solicitudes y respetarlo. Si no lo hay, es importante simular el comportamiento humano y establecer un tiempo entre cada solicitud.
- Imitar el comportamiento humano: el web scraping siempre sigue el mismo patrón de rastreo, el que hayamos programado, en cambio, los humanos visitamos el sitio web de forma aleatoria sin ningún patrón. Por tanto, se debe de cambiar el patrón de raspado de vez en cuando, además de realizar un uso aleatorio de la web.
- Usar *user-agent* apropiados: cambiar el encabezado de usuario para que sea igual al de un navegador web común o simular diferentes dispositivos. Leer atentamente los términos y condiciones, ya que puede estar prohibido modificar el *user-agent*. Este es una cadena de texto que está dentro de las solicitudes HTTP enviadas por un cliente al servidor de la web.
- Respetar el archivo robots.txt.

Todas estas técnicas no deberían preocupar al raspador si utiliza la web de manera responsable y extrae información de manera ética. Sin embargo, es posible que

[28] https://www.r-project.org/

ocasionalmente encuentre algunas de estas limitaciones y es legamente aceptable seguir los consejos proporcionados. Siempre es importante hacer un uso adecuado de la información extraída de la web.

4.3. ¿Como realizar web scraping?

En este apartado se detalla la implementación del web scraping. Empezando por aprender a usar el lenguaje de programación elegido, además de estudiar las diversas técnicas existentes para extraer información de las páginas webs de manera eficiente y precisa.

Para este proyecto se ha decidido realizar el scraping con Python porque es un lenguaje de programación utilizado a nivel mundial que cada vez está cogiendo una mayor importancia. Además, tiene numerosas librerías para realizar web scraping, entre ellas se encuentran: BeautifulSoup, Selenium, Pandas, Tqdm y Requests.

El primer desafío fue aprender a usar Python, ya que nunca lo había empleado[29].

Antes de empezar, se instaló Python y las librerías necesarias para realizar web scraping (instrucciones en el anexo 1). Además, un editor de código para poder crear y ejecutar los diferentes programas. Este nos da una mayor facilidad para crear programas más complejos y detectar los errores de forma más eficiente. Se ha elegido Visual Studio Code[30], ya que es uno de los más usados a nivel mundial, aunque hay múltiples programas de editor de código como: Sublime Text, Atom, Notepad++, Eclipse…

El objetivo es poder scrapear los datos de tres grandes supermercados: Mercadona, Carrefour y DIA, por su cuota de mercado y su accesibilidad a los datos. Además, todos ellos proporcionan la información básica que se desea extraer, es decir, el precio, el nombre y el ID para cada uno de los productos.

4.3.1. Código de buenas prácticas

Primeramente, se deben revisar el archivo robots.txt y los Términos y Condiciones de la web (revisado en el apartado 4.2) y respetarlo al detalle. Además de tener especial cuidado con los consejos dados en el apartado 4.2.3. para no ser bloqueados como scrapers. No hay que bombardear a los servidores con peticiones innecesarias, tampoco raspar datos personales sin consentimiento y respetar todos los datos obtenidos con el scraping, es decir, usarlos de forma responsable y legal.

También, hay que tener en cuenta que podemos tener bloqueos de IP, para ello se han implementado diferentes proxys. Estas son un intermediario entre el usuario y el servidor al que se desea acceder. Usando diferentes proxies ayuda a proteger la dirección IP original y a evitar ser bloqueado por el sitio web al que se intenta acceder. Sin embargo, para realizar

[29] Para aprender este nuevo lenguaje de programación contaba con una muy buena base de código de R, por tanto, no fue tan difícil aprenderlo. Para ello, se siguieron una serie de cursos y documentación. El curso de Brais Moure de inicialización a Python que podemos encontrar en YouTube (Moure, Curso de PYTHON desde CERO para PRINCIPIANTES, 2022) y el tutorial oficial de Python (Documentation, s.f.). Además, como el objetivo final era realizar web scraping, también se han visualizado otros vídeos para poder enfocar el aprendizaje de Python al objetivo final. Se visualizaron la serie de vídeos del canal “FRIKIdelTO” (FRIKIdelTO, 2021) y se utilizó el *playbook* "*Python Web Scraping Playbook*" (ScrapeOps, s.f.) como fuente de información complementaria. Con todas estas guías más muchas horas de práctica, se consiguió tener un nivel intermedio/avanzado de Python.

[30] https://code.visualstudio.com/

web scraping no es estrictamente necesario usarlas mientras se haga un uso adecuado, es decir, con moderación y respetando las políticas de uso del sitio web. Se cogió un listado de proxies gratuitas para probar su funcionalidad, pero tiene sus limitaciones: la calidad y la confiabilidad de las proxies, que los sitios webs bloqueen directamente el acceso de esas proxies por ser usadas desde varios dispositivos, además al tratarse de proxies gratuitas pueden ser inseguras. También, si se cogen proxies de otros países pueden devolver resultados erróneos, por ejemplo, al tomar el precio, si la proxy tiene en cuenta la localización geográfica posiblemente nos lo dé erróneo porque está diseñada para tratar en $ y recoge los precios de forma equívoca. Por ello, se decidió no implementar el uso de proxies, ya que sin ellas se ha podido llevar a cabo el raspado de la web sin problemas. Cabe mencionar que para este proyecto se han explorado las proxies gratuitas, se desconoce el uso de las proxies de pago.

Figura 4.10: seleccionar diferentes proxies

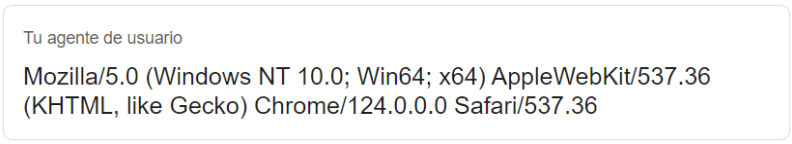
```python
# Importamos las lubrerias necesarias
import requests
from bs4 import BeautifulSoup
import random

# Lista de proxies gratuitos, por ejemplo: https://es.proxyscrape.com/lista-proxy-gratuita
proxies = [
    # Agrega las proxies aquí
]

# Web a la que se quiere realizar scraping:
url = 'https://ejemplo.com'

# Seleccionar un proxy aleatorio de la lista
proxy = {'http': random.choice(proxies)}
```

Fuente: repositorio de GitHub

Aunque no se hayan empleado proxies, en caso de desear utilizarlas, se puede utilizar el código proporcionado, este nos muestra como implementar diferentes proxies gratuitas en nuestro raspado de datos. En primer lugar, importamos las librerías necesarias. Luego, creamos una lista de diferentes proxies gratuitas que pueden ser utilizadas, los cuales pueden obtenerse de la página web especificada en el código. Luego, se selecciona una proxy aleatoria y se procese a realizar el raspado de datos.

También se tuvo en cuenta los user-agent, es decir cambiar el encabezado de usuario para poder realizar el raspado web. Al realizar scraping hay una variable que se puede modificar llamada *headers*, ya que si no al realizar el raspado la web detectará que accedemos desde Python y nos saldrá el error 403, esto indica que el acceso a la página está restringido. Para mirar cual es nuestro user-agent buscamos en Google: “¿Cuál es mi user-agent?” y en respuesta nos lo proporcionará.

Figura 4.11: Mi user-agent



Fuente: Google

También podemos obtener diferentes user-agents e ir rotándolos aleatoriamente como las proxies, de esta forma protegemos nuestra IP.

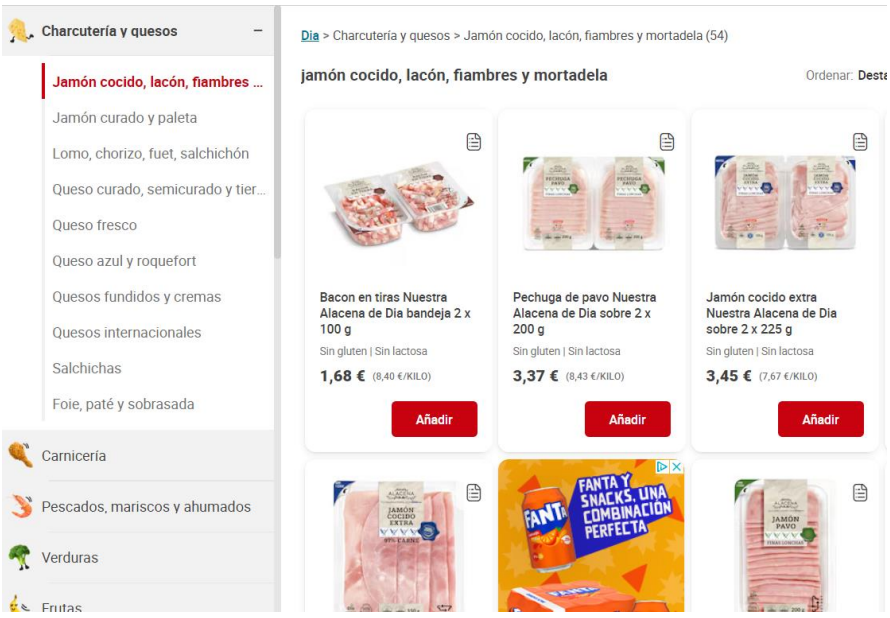
4.3.2. Uso de las APIs ocultas

Antes de comenzar con el web scraping, lo primero que tenemos que hacer es comprobar si el sitio web proporciona una API pública, es decir, acceso directo a la información que contiene la web, con una documentación de ella. Si no hay ninguna API pública disponible, abandonaremos esta opción. Sin embargo, también podemos buscar APIs ocultas que carecen de documentación oficial pero que se pueden encontrar en línea si sabes dónde buscar. Estas API son completamente legales, aunque carecen de documentación. Si existen, siempre deberían de ser nuestra primera opción.

Para encontrar estas APIs ocultas hay que seguir unos pasos muy sencillos:

1. Acceder al sitio web deseado: https://www.dia.es/
2. Revisar que se desea extraer de la web, en este caso la información de cada producto, a ser posible; el id del producto, el precio, la categoría, el nombre y si existe algún tipo de descuento.

Figura 4.12: Categorías dentro de la web



Fuente: dia.es

En la imagen se observa que DIA está estructurada por diferentes categorías, entre ellas encontramos: Charcutería y quesos, Carnicería, Pescados, maricos y ahumados… Si entramos en una de ellas, como por ejemplo Charcutería y quesos,

vemos que se vuelve a dividir en más categorías. Entramos en una de estas subcategorías y miramos qué información queremos extraer de cada producto. En este caso, se desea extraer: el nombre del producto, el id, la categoría a la que pertenece, la subcategoría y el precio. Además, si se entra en otras categorías podemos ver que hay productos con descuentos:

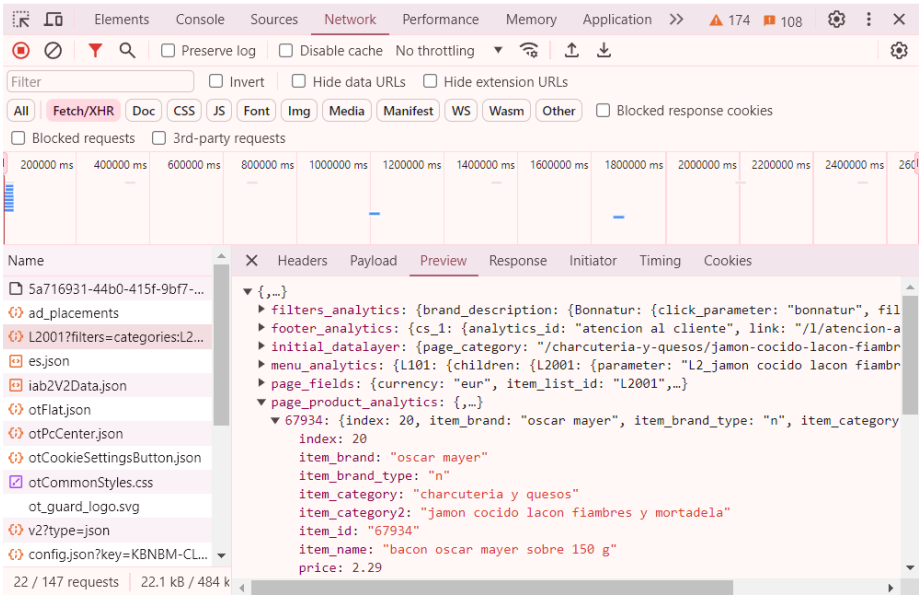
Figura 4.12: Productos con descuentos

Fuente: dia.es

Por tanto, también se desea extraer si hay algún descuento y con qué condiciones, es decir, el nombre del cupón. Una vez tenemos clara la información que deseamos extraer podemos pasar al siguiente paso, ya que es muy importante sacar solo aquella información que queremos para realizar un raspado eficiente.

3. El siguiente paso, es entrar en el código fuente con clic derecho e Inspeccionar o bien con Ctrl+Shift+C. Una vez dentro vamos al apartado de *Network* y tecleamos Ctrl+R, nos aparecerá la siguiente pantalla:

Figura 4.13: Código fuente de la web

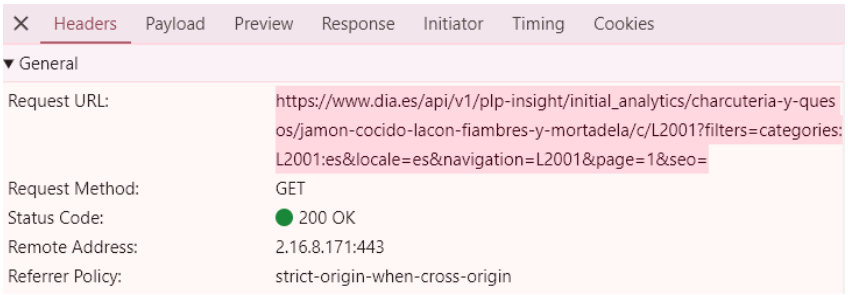


Fuente: dia.es

En la pestaña de *Network* vemos todas las peticiones que realiza la página a los recursos que necesita de la nube. Dentro de este apartado, tenemos que ir al *Fetch/XHR* y vemos a ver todas las solicitudes que realiza. En la parte izquierda, en *Name,* debemos de revisar todas estas solicitudes una a una y mirar si contienen la información que nos interesa. Para ello, se accede al apartado *Preview*, donde podemos ver una previsualización de la respuesta que se obtiene en la web. En este caso, ya se ha marcado la solicitud que nos interesa y se ha filtrado dentro del código anidado, donde se observa la información que nos interesa sacar de la web. Podemos visualizar que, dentro del primer producto, tenemos la marca del producto (‘*item_brand’*), la categoría (‘*item_category’*), la subcategoría (‘*item_category2’*), el id (‘*item_id’*), el nombre (‘*item_name’*) y el precio (‘*item_price’*). Por tanto, dentro de este código se obtiene la información que queremos extraer de la web.

4. Dentro del mismo editor de código vamos a *Headers* donde verificaremos que el código de la respuesta sea 200 OK, quiere decir que la solicitud ha tenido éxito. Dentro de este apartado podremos verificar mucha información como el tipo de petición, el user-agent… Lo que nos interesa es el *Request URL*, copiaremos este enlace dentro de un archivo CSV. Este enlace son las llamadas APIs ocultas, ya que realmente están en la web, pero hay que saberlas buscar.

Figura 4.14: Enlace de la API oculta



Fuente: dia.es

Se deben copiar todas las URL de las subcategorías que hay dentro de la web porque DIA no dispone de un apartado dentro de su web donde se puedan visualizar todos los productos a la vez. Hay que ir filtrando según la categoría del producto. Todas estas URL hay que almacenarlas en un archivo .csv, en este caso, *urls.csv*.

Una vez ya se tienen todas las URL ya se puede empezar a programar el código para extraer la información deseada[31]. En este caso el código es el archivo con el nombre: *dia.py*.

[31] Para poder acceder a todo el código y sus archivos he creado un repositorio de GitHub donde se va a almacenar todos los archivos de código y sus respectivas salidas (https://github.com/miruu02/TFG-Web-Scraping).

Figura 4.15: Primera parte del código para la extracción de datos de DIA

```python
import csv
import requests

def scrape_product_data(url):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    response = requests.get(url, headers=headers)

    if response.status_code == 200:
        data = response.json()['page_product_analytics']

        with open('productos_dia.csv', mode='a', newline='', encoding='utf-8') as csvfile:
            writer = csv.DictWriter(csvfile, fieldnames=['item_brand', 'item_category','it

            # Escribir las cabeceras solo si el archivo está vacío
            if csvfile.tell() == 0:
                writer.writeheader()

            # Escribir los datos de cada producto
            for product in data.values():
                writer.writerow({
                    'item_brand': product.get('item_brand', ''),
                    'item_category': product.get('item_category', ''),
                    'item_category2': product.get('item_category2', ''),
                    'item_id': product.get('item_id', ''),
                    'item_name': product.get('item_name', ''),
                    'price': product.get('price', ''),
                    'discount': product.get('discount', ''),
                    'item_coupon': product.get('item_coupon', '')
                    })
    else:
        print(f'Error {response.status_code}')
```

Fuente: Repositorio de GitHub

En esta primera parte del código primero se cargan las librerías necesarias, en este caso, csv ya que los datos los exportaremos a Excel, es decir la necesitamos para escribir los datos en un archivo nuevo. También se cargará la librería Requests, esta nos sirve para interactuar con recursos web a través de HTTP, ofrece una interfaz simple, pero poderosa para realizar solicitudes y gestionar respuestas de manera eficiente. Esta se usa para extraer información de las APIs y realizar web scraping, entre otras. Además, hay otra librería muy poderosa para realizar web scraping llamada Selenium, esta nos permite para interactuar con la web y simular acciones de usuario en las páginas web. Esta librería por el momento no es necesaria, ya que extraeremos datos desde las APIs ocultas.

El código incluye la definición de los encabezados (*headers*) con el objetivo de evitar bloqueos y simular una navegación más realista, esto ayuda a evitar la detección de bots. De este modo, se aumenta la probabilidad de recoger la información deseada de manera exitosa y evitando bloqueos del sitio web.

Se crea la función '*scrape_product_data'*, donde si la respuesta tiene un código de estado HTTP 200, se procede a analizar el contenido de cada URL, buscando los elementos especificados. Primero dentro del código accede al apartado 'page_product_analytics', donde se encuentran todos los productos y de cada uno de ellos extrae: 'item_brand', 'item_category', 'item_category2', 'item_id', 'item_name', 'price', 'discount' y 'item_coupon'. Si algún producto no tiene alguna de estas categorías las deja en blanco. Además, nos conserva las cabeceras con los nombres especificados anteriormente.

Además, si el código de estado HTTP no es 200, se mostraría un mensaje indicando que el código no funcionó, junto con el motivo, es decir, el código de respuesta obtenido para identificar el error.

Figura 4.16: Segunda parte del código para la extracción de datos de DIA

```
# Leer las URLs desde el archivo CSV
with open('urls.csv', mode='r', newline='', encoding='utf-8') as csvfile:
    reader = csv.DictReader(csvfile)
    urls = [row['url'] for row in reader]

# Procesar cada URL
for url in urls:
    scrape_product_data(url)
```

Fuente: Repositorio de GitHub

En esta segunda parte del código se lee el archivo *urls.csv*, creado previamente, en este se encuentran todas las APIs ocultas. Luego se llama a la función creada y se obtienen los datos deseados de la web del DIA. El resultado es un archivo csv con el siguiente aspecto:

Figura 4.17: Resultado de la extracción de datos DIA

item_brand	item_category	item_category2	item_id	item_name	price	discount	item_coupon
elpozo	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	146710	jamon cocido extra elpozo 1954 sobre 150 g	2		
elpozo	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	178002	fiambre de pavo sandwich elpozo bandeja 270 g	2		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273731	pechuga de pavo bajo en sal nuestra alacena de dia sobre 200 g	1.48	0.49	CLUB_bajadadeprecio
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273733	jamon de pavo nuestra alacena de dia sobre 200 g	1.48		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273736	jamon cocido extra bajo en sal nuestra alacena de dia sobre 200 g	1.99		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273737	jamon cocido extra 97 carne nuestra alacena de dia sobre 150 g	1.53		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273740	fiambre sandwich nuestra alacena de dia sobre 250 g	1.54		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273742	fiambre maxi york nuestra alacena de dia sobre 400 g	1.9		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273745	lacon ahumado nuestra alacena de dia sobre 200 g	2.29		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273749	bacon nuestra alacena de dia sobre 200 g	1.99		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273750	bacon en tiras nuestra alacena de dia bandeja 2 x 100 g	1.68		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273810	pechuga de pavo nuestra alacena de dia sobre 2 x 200 g	3.37		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273957	jamon cocido extra nuestra alacena de dia sobre 2 x 225 g	3.45		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273959	chopped pork nuestra alacena de dia sobre 250 g	1.39		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273960	mortadela nuestra alacena de dia sobre 250 g	1.25		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273961	mortadela con aceitunas nuestra alacena de dia sobre 250 g	1.25		
nuestra alacena de dia	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	273965	pechuga de pavo 90 carne nuestra alacena de dia sobre 120 g	2.19		
serrano	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	296598	pavo trufado con pistachos serrano sobre 150 g	1.89		
elpozo	charcuteria y quesos	jamon cocido lacon fiambres y mortadela	298369	jamon cocido elpozo sobre 80 g	1		

Fuente: Resultado del código dia.py

A día, 24/04/2024 se ha obtenido un total de 2804 productos del supermercado DIA, con la información deseada y lista para ser tratada.

En este caso, se observa que no es necesario el raspado de datos, ya que tenemos a disposición las APIs ocultas, por tanto, el proceso es algo más sencillo. Pero si no tuviéramos estas APIs ocultas se debe de realizar un raspado de la web.

4.3.3. Obtención de los datos a través del web scraping

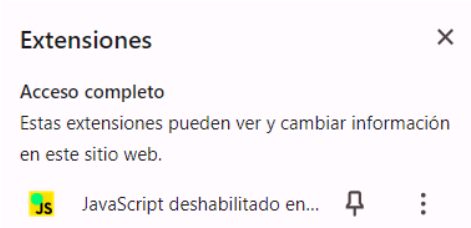
Una vez hemos entrado en la web de la cual queremos extraer información y dentro de esta no existes ni APIs públicas ni ocultas, la siguiente opción es realizar web scraping. Es decir, realizaremos la petición a la web según su tipología, esta puede ser:

- Estática: todo el contenido HTML de la página es fijo, por tanto, el creador de la página si quiere modificar el contenido tiene que modificar directamente el código HTML. Con estas páginas, con la librería *Requests* funciona perfectamente y usaremos la librería *BeautifulSoup* para estructurar los datos del código HTML. Pero, estas librerías no pueden ejecutar código JavaScript.
- Dinámica: es una página donde hay una plantilla en la que se encuentra el esqueleto de la página web y algunos datos fijos, pero hay otros datos que se completan mediante la ejecución de programas. Estos datos se pueden completar de dos formas. La primera, mediante ejecución del programa que actualiza los datos en el servidor (*back-end*), es decir, el cliente realiza una petición y el servidor completa la plantilla

con los datos necesarios HTML. Seguidamente, envía al cliente el código HTML relleno, en este caso la librería *Requests* nos funciona, pero en la segunda opción no. En esta segunda opción, la actualización de los datos se lleva a cabo en el cliente (*front-end*) y se realiza mediante código *JavaScript*. En este caso, se usará la librería *Selenium*, ya que si usamos la *Requests* nos devolverá el código HTML vacío.

Por tanto, antes de empezar a escribir código, es importante verificar si el sitio usa *JavaScript*, ya que esto influirá en la elección de que librería usar. Para ello instalaremos en Chrome la extensión que deshabilita el *JavaScript* (Faizulinartem, 2024):

Figura 4.18: Extensión JavaScript



Fuente: Chrome

Con esta extensión accedemos a la web que se desea raspar y simplemente hacemos clic en ella para desactivar *JavaScript*, si vemos que la web no carga bien los datos o incluso nos sale una advertencia que no tenemos *JavaScript*, deberemos usar *Selenium*, en caso contrario, la librería *Requests*.

También hay páginas que no usan *JavaScript*, pero al intentar scrapearlas con la librería *Requests* nos bloquean y eso es porque usan los servicios de CloudFlare, uno de los cuales es el sistema de detección de bots, por tanto, esto se resuelve con la librería *Selenium*.

Selenium es una librería diseñada para testear páginas web, automatizando interacciones con la página web, por tanto, no está diseñada para realizar web scraping, pero sus funciones son muy útiles para ello.

En el caso de usar *Selenium* en Python lo más recomendable es usar la librería '*undetected-chromedriver'*, esta permite no ser detectado por los servicios antibot (incluido el CloudFlare). La documentación oficial nos da una información muy detallada de esta librería (Undetected-chromedriver, 2024). Esta la usamos ya que mediante el uso de JavaScript los servidores pueden obtener muchos datos de nuestro equipo, del navegador y de cómo interactuamos con su web, por tanto, esta librería nos ayudará a evitar esto y no ser baneados.

Para realizar el "raspado" de las webs, primero se obtuvo un gran nivel para saber scrapear diversas webs más pequeñas y luego se realizó una búsqueda exhaustiva de proyectos relacionados con el web scraping. Hay muchísimos scrapers en Internet para realizar el scrapeo de los supermercados, pero se encontró un proyecto muy interesante donde realiza el scrapeo de los tres supermercados a la vez, es decir, del Carrefour, DIA y Mercadona[32].

[32] Me puse en contacto con el creador del código para poder usarlo y así fue. Este proyecto está almacenado en una carpeta de GitHub (Luam, 2024). Este ha estado actualizado el 22/04/2024, ya que cuando contacté con el creador del código este no funcionaba, eso fue debido al uso de Cookies de cada supermercado y se incorporó al código.

Para conocer todo el desarrollo y funcionamiento del código consultar el anexo 2.

4.3.3.1. Resultados de la función

Al usar la función para scrapear los tres supermercados a la vez, obtenemos cuatro Excels, uno para cada supermercado y uno que nos une todos los datos. Para cada supermercado el día 18/04/2024 se han obtenido:

- Carrefour: 6790 productos
- DIA: 2809 productos
- Mercadona: 5118 productos

Luego, estos tres Excels se anexan y se obtiene el siguiente:

Figura 4.27: Resultado ejecución de la función

Id	Nombre	Precio	Precio Pack	Formato	Categoria	Supermercado	Url	Url_imagen	Favorito
4240	Aceite de (	8.95	8.95	l	112	Mercadona	https://tie	https://prod-	FALS
4717	Aceite de (	29.55	9.85	l	112	Mercadona	https://tie	https://prod-	FALS
4740	Aceite de (	9.90	9.90	l	112	Mercadona	https://tie	https://prod-	FALS
4706	Aceite de (	8.10	10.80	l	112	Mercadona	https://tie	https://prod-	FALS
4640	Aceite de (	8.95	8.95	l	112	Mercadona	https://tie	https://prod-	FALS
4711	Aceite de (	28.50	9.50	l	112	Mercadona	https://tie	https://prod-	FALS
4749	Aceite de (	9.55	9.55	l	112	Mercadona	https://tie	https://prod-	CERT
4204	Aceite de (	26.70	8.90	l	112	Mercadona	https://tie	https://prod-	FALS
4609	Aceite de (	26.70	8.90	l	112	Mercadona	https://tie	https://prod-	FALS
4718	Aceite de (	3.30	16.50	l	112	Mercadona	https://tie	https://prod-	FALS
4850	Aceite de (	7.85	15.70	l	112	Mercadona	https://tie	https://prod-	FALS
4041	Aceite de ع	6.75	1.35	l	112	Mercadona	https://tie	https://prod-	FALS
4040	Aceite de ع	1.45	1.45	l	112	Mercadona	https://tie	https://prod-	FALS
4193	Aceite de (	4.40	9.78	l	112	Mercadona	https://tie	https://prod-	CERT
4940	Vinagre de	0.75	0.75	l	112	Mercadona	https://tie	https://prod-	FALS

Fuente: Elaboración propia. Resultado de Python.

Para cada producto se ha extraído su ID, el nombre, el precio, el precio por unidad, el formato de las unidades, la categoría dentro de cada supermercado, el supermercado a la que pertenecen, la URL, la URL de la imagen y si el producto estaba en favorito o no. Esto último, es irrelevante para este proyecto, ya que la lista de favoritos se marca al inicio manualmente.

Si comparamos los resultados obtenidos con esta función con el código DIA del apartado 4.3.2. veremos que para el supermercado DIA solo difieren 5 productos, es decir, en esta función obtenemos 5 productos más que para la otra, así que tomaremos los resultados como válidos.

Como se observa en el código, se han extraído los datos de APIs “ocultas” dentro de la web, consiguiendo los URL, por tanto, no se ha realizado web scraping en sí, ya que se ha accedido al código HTML público, ya que, si disponemos de este, esta debe de ser nuestra primera opción.

5. OBTENCIÓN DEL IPC

Una vez que se obtuvieron las tres bases de datos, se procedió a procesarlas para calcular el Índice de Precios al Consumidor (IPC) a partir de ellas. Para este propósito, se decidió utilizar Power BI, una herramienta muy potente para visualización y análisis de datos (para consultar su instalación ir al anexo 3).

Dentro de Power BI, se utilizó Power Query para el tratamiento y transformación de las bases de datos. Power Query es una funcionalidad robusta que permite conectarse a diversas fuentes de datos, realizar limpiezas, transformar y combinar datos de manera eficiente. Con Power Query, se realizaron las transformaciones necesarias para que los datos estuvieran en el formato adecuado para el análisis.

Además, se empleó el lenguaje de fórmulas DAX (*Data Analysis Expressions*) dentro de Power BI. DAX es una colección de funciones, operadores y constantes que se pueden utilizar en fórmulas y expresiones para calcular y devolver valores. Gracias a DAX, se pudieron crear métricas precisas para calcular el IPC.

En primer lugar, se realizó la depuración de la base de datos utilizando Power Query, un paso crucial para garantizar un análisis preciso de los datos. Se seleccionó la base de datos que incluía información de los tres supermercados simultáneamente y se eligieron las columnas necesarias: ID del producto, nombre, precio, precio por pack, formato, categoría y supermercado.

A continuación, se ajustó la columna de formato, convirtiendo todas las letras a minúsculas para evitar duplicados en las categorías. Este proceso de estandarización es fundamental para asegurar la coherencia y precisión en los datos antes de proceder con el análisis. La limpieza de datos es esencial para garantizar la integridad y confiabilidad de la información, de esta manera aumenta la confiabilidad de los datos y se obtendrán unos resultados más eficientes, precisos y efectivos.

La columna de categoría contenía datos de la categoría, subcategoría e información adicional que no era necesaria. Por esta razón, se tuvo que tratar esta columna de manera diferente para cada supermercado, ya que su estructura variaba entre ellos. En Power Query, se realizaron las siguientes transformaciones para cada supermercado:

- Dia

  Ejemplo de formato de la columna *Categoría*: *"/charcuteria-y-quesos/jamon-cocido-lacon-fiambres-y-mortadela/c/L2001"*

  Se cogieron las categorías de Dia de referencia para los demás supermercados ya que son las que venían mejor estructuradas y con una mayor integridad. Por tanto, formatear este supermercado fue tan fácil como dividir la columna en dos por los delimitadores / y la información adicional se eliminó directamente.

  Ejemplo de resultado del formateo:

  Categoría: *charcuteria-y-quesos*

  Subcategoría: *jamon-cocido-lacon-fiambres-y-mortadela*

- Mercadona

  Ejemplo de formato de la columna *Categoría*: 112

  Para este supermercado, por cada categoría se tenía un número diferente, por ello, se realizó un mapeo para cada categoría:

Figura 5.1: Mapeo de las categoría y subcategorías de Mercadona.

Id_Categoria	Categoria	Subcategoria
112	aceites-salsas-y-especias	aceites
115	aceites-salsas-y-especias	sal-y-especias
116	aceites-salsas-y-especias	mayonesa-ketchup-y-otras-salsas
117	aceites-salsas-y-especias	tomate
156	agua-refrescos-y-zumos	agua
163	agua-refrescos-y-zumos	bebidas-energeticas
158	agua-refrescos-y-zumos	cola
159	agua-refrescos-y-zumos	limon-lima-limon
161	agua-refrescos-y-zumos	tonicas
162	agua-refrescos-y-zumos	tes-frios-cafes-frios
135	patatas-fritas-encurtidos-y-frutos-secos	aceitunas-y-encurtidos
133	patatas-fritas-encurtidos-y-frutos-secos	frutos-secos
132	patatas-fritas-encurtidos-y-frutos-secos	patatas-fritas-y-aperitivos
118	arroz-pastas-y-legumbres	arroz
121	arroz-pastas-y-legumbres	lentejas
120	arroz-pastas-y-legumbres	pastas
89	azucar-chocolates-y-caramelos	azucar-y-edulcorantes
95	azucar-chocolates-y-caramelos	caramelos-chicles-y-golosinas
92	azucar-chocolates-y-caramelos	chocolates-y-bombones
97	azucar-chocolates-y-caramelos	caramelos-chicles-y-golosinas

Fuente: Elaboración propia.

A través de este mapeo se pudo categorizar las dos columnas, categoría y subcategoría.

Ejemplo de resultado del formateo:

Categoría: *aceites-salsas-y-especias*

Subcategoría: *aceites*

- Carrefour

Ejemplo de formato de la columna *Categoría*: */congelados-promocion/F-10flZ13rjo/c*

Para este supermercado, la categorización fue más complicada porque, aunque las categorías principales eran aproximadamente iguales a las de los otros supermercados, las subcategorías no lo eran. En primer lugar, se corrigieron las categorías que no coincidían exactamente con las de los demás supermercados. Para las subcategorías, solo se pudieron clasificar algunas. Debido a esta dificultad, algunos productos no pudieron ser categorizados adecuadamente y se perdieron en el proceso. Específicamente, de un total de 6790 productos, se lograron categorizar 6727.

Ejemplo de resultado del formateo:

Categoría: *congelados*

Subcategoría: *croquetas-y-rebozados*

Seguidamente, se estandarizó la columna formato, es decir, se cogieron aquellas unidades que, sí que se les podía aplicar una conversión de unidades, para obtener las unidades de medida lo más estándares posibles. Para ello se realizó es siguiente mapeo:

Figura 5.2.: Mapeo de la normalización de las unidades.

Unidades	Conversión a kilos
100g	0,1
100ml	0,1
docena	0,75
kg	1
l	1
lavado	1
m	0,08
ud	1

Fuente: Elaboración propia.

Se decidió pasar las unidades a kilos y se aplicaron los siguientes factores de conversión:

- 100g: 100 gramos equivalen a 0.1 kilos.
- 100ml: para convertir mililitros a kilogramos, es crucial considerar la densidad de los líquidos. En este contexto, dado que la mayoría de los productos son agua y refrescos, se emplea la densidad del agua, aproximadamente 1 kg/L, lo que equivale a 1 kg por cada 1000 ml. Por lo tanto, 100 mililitros equivalen a 0.1 kilogramos.
  Tanto los refrescos como la cerveza tienen una densidad muy similar a la del agua, dado que principalmente están compuestos de agua. Por otro lado, en el caso de los productos alcohólicos, su densidad varía aproximadamente entre 0.8 y 0.95 kg/L. Para esta conversión, se ha considerado una pequeña variación, aunque insignificante.
- Docena: En este caso, dado que se refiere a una docena de huevos de gallina, y sabiendo que un huevo de gallina pesa entre 60 y 65 gramos en promedio, podemos calcular el peso total de la docena. Un huevo de gallina en media pesa 62.5 gramos, 0.0625 kg. Es decir, una docena serán 0.75 kilos.
- Kg y l: permanecen iguales. Se ha aplicado la densidad del agua tanto para la conversión de litros como para la de mililitros.
- Lavado: se ha decidió usar 1 ya que en este caso no tenemos ningún factor de conversión posible.
- m: para el metro se han considerado los productos que la utilizaban, mayormente servilletas o papel film. Por esta razón, se ha establecido que aproximadamente 1 metro de papel pesa alrededor de 80 gramos. Por lo tanto, un metro de papel equivale a 0.08 kilogramos.
- ud: unidad, en este caso también se ha decidido usar la conversión de 1 ya que no tenemos ningún factor de conversión posible.

Posteriormente, se calculó la columna *Precio final*, donde se multiplica la columna *Precio Pack* y las conversiones, para obtener los precios de la forma más estandarizada posible.

Antes de empezar a realizar el cálculo del IPC, se revisaron los procedimientos de cálculos descritos en el apartado 3.2.1. y 3.2.2. Para empezar a realizar el cálculo del IPC se necesita un mes base o mínimamente un mes anterior. Por ello, se realizó de nuevo el scrapeo de los supermercados el día 15/05/2024, casi un mes después de la primera extracción (18/04/2024). Se obtuvo:

- Carrefour: 6895 productos
- DIA: 2812 productos
- Mercadona: 5118 productos

Prácticamente se obtuvieron los mismos productos que en la extracción anterior. Posteriormente, se llevó a cabo el mismo proceso de limpieza y tratamiento de datos que en el mes de abril. En el caso de Carrefour, se lograron categorizar 6765 productos, con una pérdida mínima, similar a la ocurrida anteriormente y sin mayor relevancia. Además, para las dos bases de datos se creó la columna *Mes* para poder distinguir entre el mes de abril y mayo.

Por último, se anexaron las dos tablas, la de abril y mayo, y se agruparon por categoría, supermercado y mes. Obteniendo como resultado, una tabla con el precio medio y la cantidad vendida por cada categoría.

En este punto, se empezó a calcular el IPC por cada supermercado y uno en general. En primer lugar, se calcularon los índices relativos, estos se calculan dividiendo el precio medio de la categoría (i) en el mes actual (m), entre el precio medio de la categoría (i) en el mes base o anterior (m-1):

$$Índice\ Relativo\ = \frac{Precio\ medio\ categoria_i^m}{Precio\ medio\ categoria_i^{m-1}} \qquad (5.1)$$

Una vez calculados los índices relativos para cada categoría en mayo en comparación con abril, se procedió a calcular los índices ponderados. Las ponderaciones de cada categoría se determinaron según las unidades vendidas de cada una. Para ello, se sumaron las unidades vendidas de cada categoría en ambos meses y se distribuyeron las ponderaciones de acuerdo con el porcentaje de ventas total, obteniendo así las siguientes ponderaciones:

Figura 5.3.: Ponderaciones por cada categoría.

Categorias	Pesos
aceites-salsas-y-especias	0,0170
agua-refrescos-y-zumos	0,0316
arroz-pastas-y-legumbres	0,0099
azucar-chocolates-y-caramelos	0,0202
bebe	0,0391
cafe-cacao-e-infusiones	0,0228
carniceria	0,0274
cervezas-vinos-y-bebidas-con-alcohol	0,1124
charcuteria-y-quesos	0,0364
congelados	0,0826
conservas-caldos-y-cremas	0,0139
frutas	0,0127
galletas-bollos-y-cereales	0,0211
leche-huevos-y-mantequilla	0,0131
limpieza-y-hogar	0,1401
mascotas	0,0310
panes-harinas-y-masas	0,0285
patatas-fritas-encurtidos-y-frutos-secos	0,0233
perfumeria-higiene-salud	0,2468
pescados-mariscos-y-ahumados	0,0095
pizzas-y-platos-preparados	0,0298
verduras	0,0198
yogures-y-postres	0,0109

Fuente: Elaboración propia.

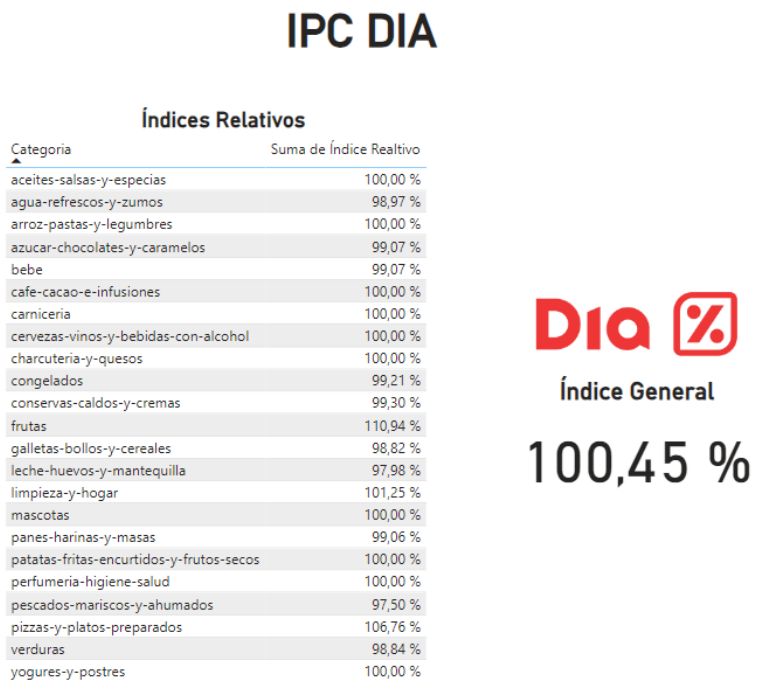
Luego, se calculó el índice general para cada categoría y supermercado. Este se calcula sumando la multiplicación de todos los índices relativos de cada categoría (i) con sus respectivos pesos:

$$Índice\ General\ = \sum Índice\ Relativo_i\ *\ Peso_i \qquad (5.2)$$

Como resultado, se obtuvieron los índices relativos de cada supermercado y categoría, así como un índice general para cada supermercado. Toda esta información se recopiló en *Dashboard* de PowerBI para una mejor visualización de los datos.

En primer lugar, observemos el *Dashboard* de Dia. Podemos ver que los precios de abril a mayo han subido un 0.45%. Esta subida ha sido impulsada por el incremento en los precios de las frutas, las pizzas y platos preparados, todas con un IPC superior al 100%. También se observa que algunas categorías han mantenido sus precios, es decir, su índice ha sido del 100%. Entre ellas se encuentran: aceites, salsas y especias, arroz, pastas y legumbres, café, cacao e infusiones… Por otro lado, algunas categorías han disminuido su precio de abril a mayo, obteniendo un IPC inferior al 100%, como agua, refrescos y zumos, azúcar, chocolates y caramelos…

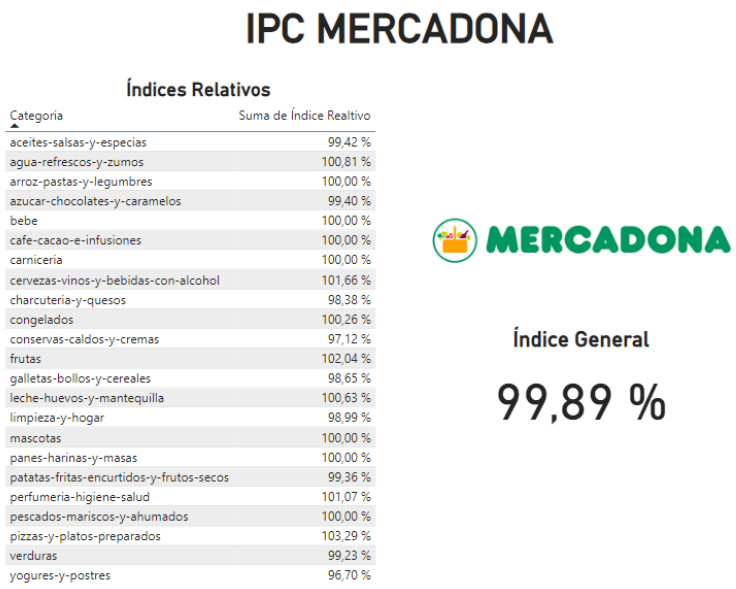
Figura 5.4.: Dashboard DIA.



Fuente: Elaboración propia. Resultado de PowerBI.

Para Mercadona, ha habido una disminución de los precios respecto el mes de abril de un 0.11%. Esto es debido a las categorías que han obtenido un índice relativo inferior al 100%. Las bajadas de precio más fuertes han estado en las categorías: yogures y postes, limpieza y hogar, galletas, bollos y cereales, conservas, caldos y cremas, charcutería y quesos. Además, igual que en el supermercado Dia las categorías de frutas, pizzas y platos preparados han tenido un aumento de precio.

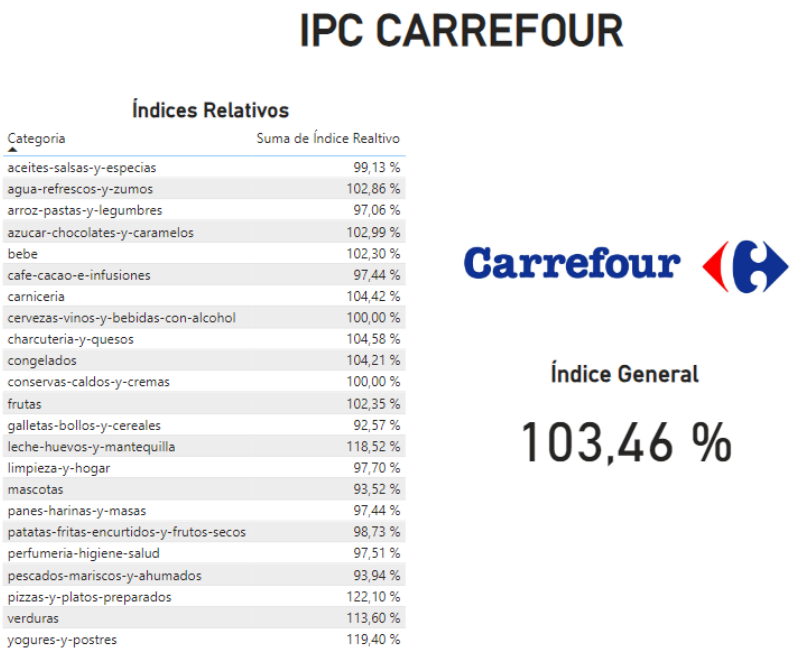
Figura 5.5.: Dashboard Mercadona.



Fuente: Elaboración propia. Resultado de PowerBI.

Carrefour ha sido el supermercado con el mayor incremento de precios, registrando una subida del 3.46% respecto al mes de abril. En este caso, muchas categorías experimentaron aumentos de precios, y en comparación con otros supermercados, este incremento ha sido bastante significativo. Entre las categorías con mayores subidas se encuentran frutas, pizzas y platos preparados. Además, también hubo importantes incrementos en charcutería y quesos, congelados, leche, huevos y mantequilla, verduras, yogures y postres. Por otro lado, ha habido fuertes bajadas de precio en las categorías: galletas, bollos y cereales, mascotas, pescados, mariscos y ahumados, de entre un 7.5% y 6% de bajada.

Figura 5.6.: Dashboard Carrefour.



Fuente: Elaboración propia. Resultado de PowerBI.

Por último, se calculó el IPC total para los tres supermercados conjuntamente. Para ello, primero se asignaros diferentes pesos a cada supermercado según su cuota de mercado. Si consultamos el apartado 3.1.1. nos especifica las cuotas de mercado de los supermercados elegidos: Mercadona (26.2%), Carrefour (9.9%) y Dia (4.1%). Según estos porcentajes se han asignado unos nuevos para que llegue al 100% y poder realizar unas ponderaciones lo más parecidas a la realidad. Para realizar este ajuste de cuotas de mercado primero se han sumado todas las ponderaciones:

$$26.2 + 9.9 + 4.1 = 40.2 \qquad (5.2.a)$$

Luego se han recalculado las nuevas cuotas de mercado de la siguiente manera:

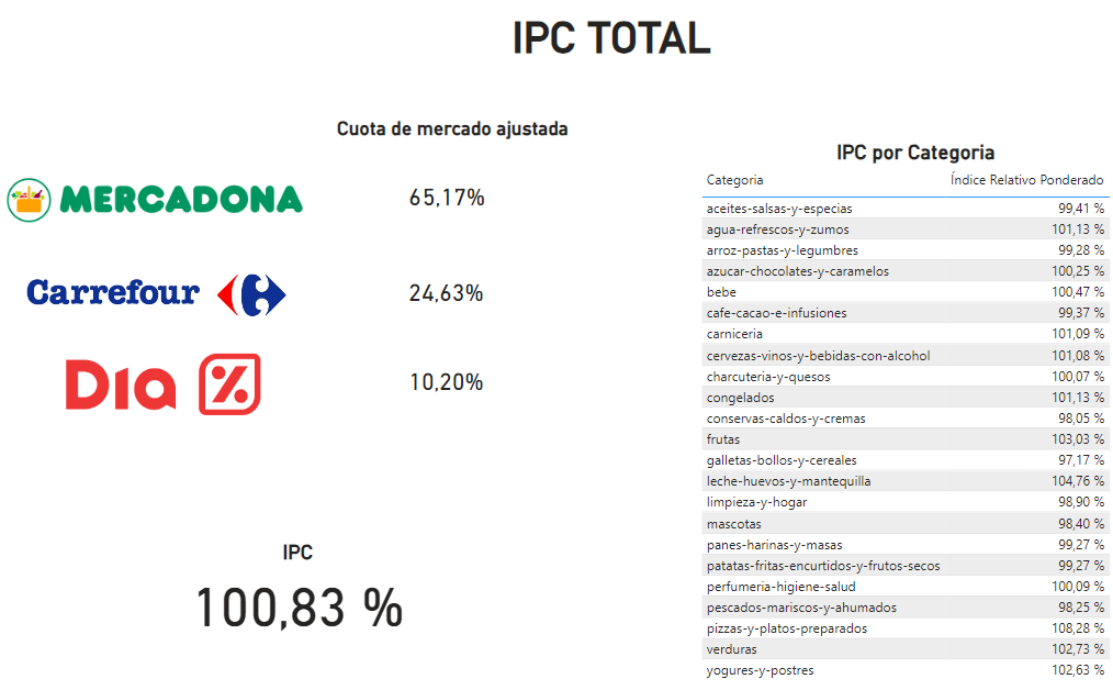
$$Mercadona = \frac{26.2}{40.2} * 100 = 65.17\% \qquad (5.2.b)$$

$$Carrefour = \frac{9.9}{40.2} * 100 = 24.63\% \qquad (5.2.c)$$

$$Dia = \frac{4.1}{40.2} * 100 = 10.20\% \qquad (5.2.d)$$

Con estas ponderaciones por cada uno de los supermercados, se calcularon los índices relativos multiplicados por la nueva cuota de mercado y un índice general, obtenido como resultado el siguiente *Dashboard*:

Figura 5.7.: Dashboard IPC Total.



Fuente: Elaboración propia. Resultado de PowerBI.

En esta diapositiva, se pueden observar las nuevas cuotas de mercado ajustadas por cada supermercado, el IPC por categoría y el IPC general. Como se observa ha habido un

incremento de precios de abril a mayo de un 0.83% (siendo abril la base 100). Esto ha sido consecuencia de aquellas categorías con un IPC superior al 100%, las más importantes han estado: frutas, leche, huevos y mantequilla, pizzas y platos preparados, verduras, yogures y postres. Todas ellas han tenido subidas de ente un 2.63% a un 8.28%. Por otro lado, ha habido categorías con un precio más bajo respecto a abril, entre ellas se encuentran: conservas, caldos y cremas, galletas, bollos y cereales, limpieza y hogar, mascotas, pescados, mariscos y ahumados.

En resumen, de abril a mayo ha habido un aumento general de precios, principalmente impulsado por un incremento significativo en los precios del supermercado Carrefour. En contraste, Mercadona, que tiene la mayor cuota de mercado, ha experimentado una leve disminución en sus precios. Como resultado, el índice general no muestra un gran aumento en los precios de los productos de los supermercados.

CONCLUSIONES

En este trabajo se han cumplido los objetivos iniciales, comenzando por el marco teórico donde se investigaron en profundidad las nuevas metodologías de recogida de datos del INE. Al recopilar y comparar información de diversas fuentes, se concluye que tanto el web scraping como el scanner data son metodologías en desarrollo que sustituirán o complementarán la forma tradicional de recogida de datos en un futuro cercano. Estas conclusiones se basan en las numerosas ventajas que ofrecen estos métodos y en el constante avance tecnológico, lo que permitirá ganar tiempo y eficiencia en el proceso de recopilación de datos estadísticos. La principal diferencia entre estas dos metodologías radica en la obtención de los datos: mientras que en el scanner data los supermercados proporcionan las bases de datos mediante convenios, en el web scraping se extrae la información directamente de la web o de alguna API pública o privada accesible.

En cuanto al cálculo del IPC, se analizó detalladamente la metodología tradicional y su proceso de cálculo. Para el scanner data y el web scraping, este proceso es diferente debido a la naturaleza distinta de la información obtenida. Se estudió en profundidad el cálculo del IPC para estas dos metodologías, observando que, aunque el método de cálculo es el mismo, el tratamiento inicial de los datos varía según la metodología y el supermercado, aunque sigue un esquema similar.

El segundo objetivo fue replicar el proceso de recogida de datos mediante web scraping utilizando Python, uno de los softwares estadísticos más potentes a nivel mundial. Para ello, se aprendió a usar Python desde cero, dedicando muchas horas a cursos de formación. Una vez alcanzado un nivel intermedio, se avanzó en el aprendizaje enfocado a realizar web scraping. El proyecto se desarrolló en Visual Studio Code, una herramienta de edición de código que facilita la comprensión de programas avanzados.

Tras adquirir las herramientas necesarias para realizar web scraping, se estudió el marco legal, concluyendo que es completamente legal siempre y cuando se respete el uso adecuado de la información y las condiciones de cada sitio web. También se investigó el funcionamiento de las webs para entender dónde buscar la información necesaria. Se exploraron diversas webs para encontrar APIs ocultas que facilitaran la extracción de datos.

Finalmente, se realizó la extracción de datos del primer supermercado, DIA, escribiendo un código desde cero. Esta tarea fue laboriosa, pero resultó exitosa. Para los demás supermercados, se encontró un código público en GitHub que permitía la extracción simultánea de datos, obteniendo todas las bases de datos necesarias para su posterior uso.

El tercer objetivo fue calcular un IPC a partir de los datos obtenidos. Primero, se trató la limpieza de los datos utilizando Power Query, logrando automatizar este proceso. Se encontraron algunas dificultades con los datos de Carrefour debido a la falta de definición en las categorías, lo que requirió mapeo manual. Finalmente, se mapearon las categorías de los tres supermercados tomando como referencia DIA para obtener un cálculo del IPC eficiente. Luego, con PowerBI, se crearon dashboards que mostraban el IPC por supermercado y

categoría, además de un IPC general ponderado según la cuota de mercado real de cada supermercado.

Concluyendo, este trabajo ha sido muy enriquecedor, ya que se han tratado a fondo las nuevas metodologías de extracción de datos para el cálculo del IPC, actualmente en desarrollo por múltiples países y promovidas por la Unión Europea. Es fundamental aprovechar la tecnología actual para mejorar la eficiencia y calidad de los datos estadísticos. Además, se ha logrado realizar la extracción de datos de tres importantes supermercados nacionales y replicar el proceso de cálculo del IPC con la información obtenida. Como sociedad, es crucial avanzar en la tecnología en todos los ámbitos para obtener información valiosa y datos de calidad mediante la estadística.

BIBLIOGRAFIA

DIA. (24 de 01 de 2023). *AVISO LEGAL Y CONDICIONES DE USO*. Obtenido de https://s1.dia.es/medias/h16/h9a/11331458760734.pdf

Documentation, P. (s.f.). *El tutorial de Python*. Obtenido de https://docs.python.org/es/3/tutorial/

Eustat. (s.f.). *Clasificación de bienes y servicios. COICOP.* Obtenido de https://www.eustat.eus/documentos/datos/codigos/egv_Clas_bienes_serv_coicop_c.pdf

Faizulinartem. (18 de 04 de 2024). *JavaScript deshabilitado en Chrome*. Obtenido de https://chromewebstore.google.com/detail/javascript-desactivat-a-c/jbjdopljoflabmgndibnfmnmojbafaog

FRIKIdelTO. (6 de Octubre de 2021). *Curso PYTHON desde CERO orientado a WEB SCRAPING*. Obtenido de https://www.youtube.com/playlist?list=PLheIVUbpfWZ17lCcHnoaa1RD59juFR06C

INE. (Abril de 2017). *Índice de Precios de Consumo. Base 2016.* Obtenido de https://www.ine.es/metodologia/t25/t2530138_16.pdf

INE. (Febrero de 2020). *Metodología de cálculo del scanner data en el IPC y IPCA.* Obtenido de https://www.ine.es/metodologia/t25/ipc_scanner_data.pdf

Luam, J. (22 de 04 de 2024). *Supermarket-Price-Scraper*. Obtenido de https://github.com/joseluam97/Supermarket-Price-Scraper/tree/main/supermarket

Moure, B. (30 de Septiembre de 2022). *Curso de PYTHON desde CERO para PRINCIPIANTES*. Obtenido de https://www.youtube.com/watch?v=Kp4Mvapo5kc

Moure, B. (18 de Noviembre de 2022). *Curso de PYTHON desde CERO para PRINCIPIANTES [Intermedio]*. Obtenido de https://www.youtube.com/watch?v=TbcEqkabAWU&t=10381s

ScrapeOps. (s.f.). *Python Requests/BS4 Beginners Series*. Obtenido de https://scrapeops.io/python-web-scraping-playbook/

Undetected-chromedriver. (02 de 2024). Obtenido de https://github.com/ultrafunkamsterdam/undetected-chromedriver

WEBGRAFIA

Anti-bloqueado scraping Tutorial: Extraer datos fácilmente | Octoparse. (2023, 13 marzo). https://www.octoparse.es/blog/scrape-websites-sin-ser-bloqueado

Axarnet. (2023, 25 agosto). Qué es el web scraping y cómo funciona. Axarnet. https://axarnet.es/blog/web-scrapping#herramientas-populares-de-web-scraping

A, T. M. (2023, 18 diciembre). Web Scraping Carrefour Data using Python and Selenium. Datahut Blog. https://www.blog.datahut.co/post/web-scraping-carrefour-data-using-python-and-selenium

Beautiful Soup Documentation — Beautiful Soup 4.12.0 documentation. (s. f.). https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Best 30 free Web Scraping tools 2024 | Octoparse. (2023, 10 septiembre). https://www.octoparse.com/blog/top-30-free-web-scraping-software

Cómo funciona la web - Aprende desarrollo web | MDN. (2023, 20 agosto). MDN Web Docs. https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Cuotas de mercado de la distribución - Kantar. (s. f.). https://www.kantarworldpanel.com/es/grocery-market-share/spain/snapshot

Desarrollo de Aplicaciones WEB. Páginas web dinámicas. Rafael Barzanallana. Universidad de Murcia. (2019, 31 marzo). https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-13-paginas-web-dinamicas.html

Dominar la rotación de proxy la clave para un web scraping ininterrumpido - FasterCapital. (2024, 24 marzo). FasterCapital. https://fastercapital.com/es/contenido/Dominar-la-rotacion-de-proxy--la-clave-para-un-web-scraping-ininterrumpido.html

Hasson, E. (2023, 7 diciembre). Is web scraping illegal? | Imperva. Blog. https://www.imperva.com/blog/is-web-scraping-illegal/

Holcombe, J. (2024, 22 abril). What is web scraping? How to legally extract web content. Kinsta. https://kinsta.com/knowledgebase/what-is-web-scraping/

Ignacio González Veiga (2016, julio). El método scanner data para la utilización de bases de datos de empresas en el IPC http://www.revistaindice.com/numero68/p29.pdf

INE. (2022, enero). Principales novedades metodológicas del Índice de Precios de Consumo Base 2021 https://ine.es/metodologia/t25/principales_caracteristicas_base_2021.pdf

INE - Instituto Nacional de Estadística. (s. f.). Sección prensa / Encuesta de Presupuestos Familiares (EPF). INE - Instituto Nacional de Estadística. https://www.ine.es/prensa/epf_prensa.htm

INE - Instituto Nacional de Estadística. (s. f.). Sección prensa / Índice de Precios de Consumo (IPC). INE - Instituto Nacional de Estadística. https://www.ine.es/prensa/ipc_prensa.htm

Javi Data Science. (2023, 2 febrero). Aprende a hacer Web Scraping. Extracción de datos web con python y Selenium. [Vídeo]. YouTube. https://www.youtube.com/watch?v=JKwfzexrQS0

Keyweo Communication SL. (2023, 2 marzo). Robots.txt: definición, utilidad y consejos SEO - Keyweo. Keyweo. https://www.keyweo.com/es/seo/glosario/robots-txt/

Mendoza, M. L. (2023, 14 abril). Páginas web estáticas vs páginas web dinámicas. OpenWebinars.net. https://openwebinars.net/blog/paginas-web-estaticas-vs-paginas-web-dinamicas/

MoureDev by Brais Moure. (2022, 30 septiembre). Curso de PYTHON desde CERO para PRINCIPIANTES [Vídeo]. YouTube. https://www.youtube.com/watch?v=Kp4Mvapo5kc

MoureDev by Brais Moure. (2022, 18 noviembre). Curso de PYTHON desde CERO para PRINCIPIANTES [Intermedio] [Vídeo]. YouTube. https://www.youtube.com/watch?v=TbcEqkabAWU

Nyamande, M., & Nyamande, M. (2023, 24 julio). Raspado web libre de bloqueos. Bright Data. https://brightdata.es/blog/datos-web/web-scraping-without-getting-blocked

Quiroga, R. (2022, 29 diciembre). Introducción al web scraping usando R. Programming Historian. https://programminghistorian.org/es/lecciones/introduccion-al-web-scraping-usando-r

Referencia de Elementos HTML - HTML: Lenguaje de etiquetas de hipertexto | MDN. (2023, 5 enero). https://perma.cc/95F8-PRLD

Saalu, S., & Saalu, S. (2024, 9 abril). Cómo usar proxies para rotar direcciones IP en Python. Bright Data. https://brightdata.es/blog/procedimientos/python-ip-rotation

Scrapy 2.11 documentation — Scrapy 2.11.1 documentation. (2024, 27 abril). https://docs.scrapy.org/en/latest/

The Selenium Browser Automation Project. (2023, 17 abril). Selenium. https://www.selenium.dev/documentation/

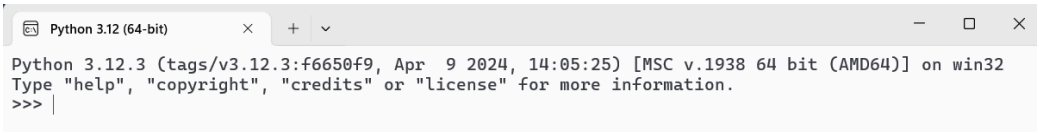
Zanini, A., & Zanini, A. (2024, 29 enero). Qué es un servidor proxy en Python. Bright Data. https://brightdata.es/blog/101-proxy/python-proxy-server

¿Es legal el web scraping?: De webscraping y legalidad. (2017, 22 marzo). Diario de un E-Letrado. https://diariodeuneletrado.wordpress.com/2017/03/22/es-legal-el-web-scrapping-de-webscrapping-y-legalidad/

ANEXO

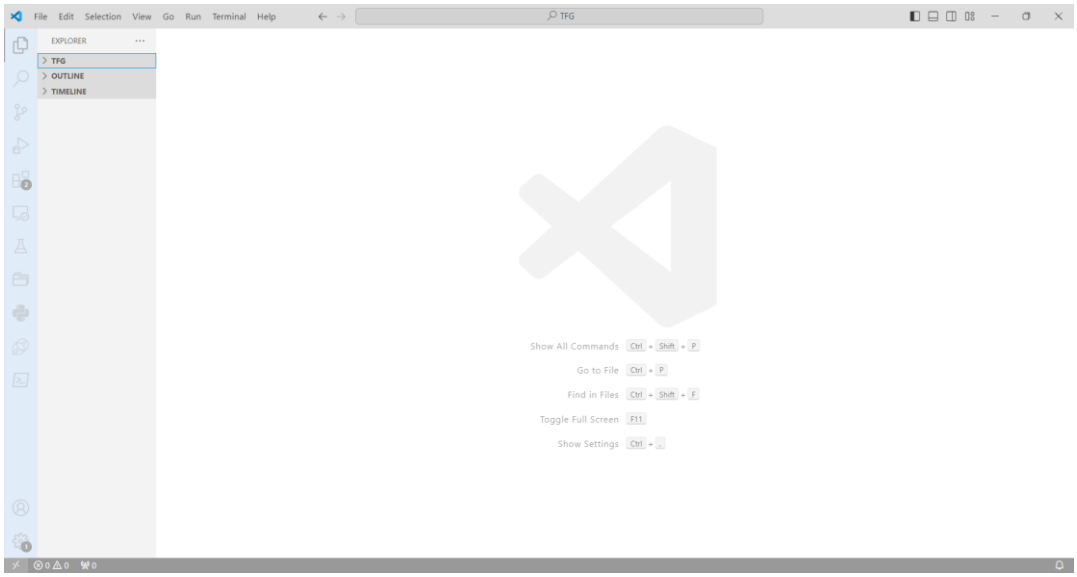
1. Descarga e instalación de Python y las librerías necesarias

Dado que Python es un software libre, su instalación es muy sencilla, al mismo tiempo que sus librerías. Para instalar Python en el ordenador es necesario acceder a la siguiente página: https://www.python.org/. En esta página se encuentra un apartado llamado *Download* donde podremos descargar la última versión de Python, a día 10/04/2024 se ha descargado la versión 3.12.3. Cuando se complete la descarga abriremos el archivo .exe y seguiremos sus indicaciones. Una vez acabada la instalación ya tendremos Python en nuestro ordenador, este tiene el siguiente aspecto:



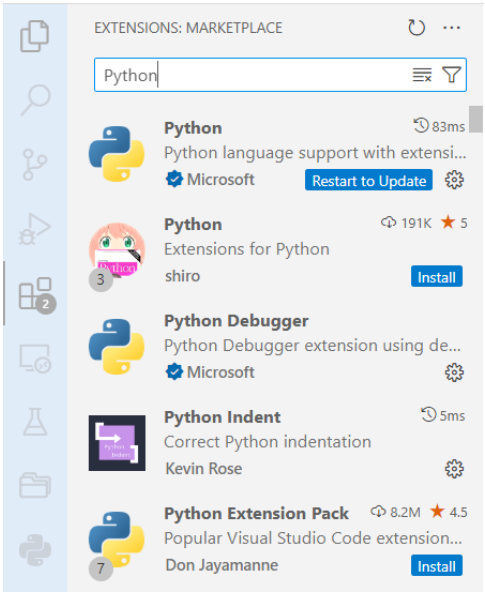
Fuente: consola de Python

Esta es la terminal de Python desde donde podremos crear nuestros programas, pero para crear programas más largos y complejos queremos un editor de código que en este caso se ha elegido el Visual Studio Code, ya que su visualización es algo parecida a RStudio, aunque hay otros programas como Anaconda. Para su descarga accederemos al siguiente sitio web: https://code.visualstudio.com/. En esta página encontramos un botón de *Download for Windows* donde lo descargaremos. Una vez descargado ejecutaremos el fichero .exe hasta completar la descarga. El aspecto de Visual Studio Code es el siguiente:



Fuente: visualización consola Visual Studio Code

Desde este programa podemos acceder a las carpetas de nuestro ordenador. Se creará una carpeta para este proyecto y se accederá a ella. Seguidamente, crearemos un archivo con la extensión .py donde es un archivo de código Python. Además, se instalarán algunas extensiones para poder usar Python en Visual Studio:



Fuente: Visual Studio Code

Instalaremos la primera extensión para poder usar Python y podremos empezar a crear nuestro programa. Primeramente, se debe conocer el lenguaje de Python que es bastante similar al de R para ello se han visualizado una serie de cursos sobre Python para tener un mayor conocimiento del programa y poder coger habilidad con él. Luego, ya se ha procedido a la instalación de las librerías necesarias, esto se debe de realizar desde la terminal con el condigo *pip install nombre_de _la_librería_que_quieras*:

```
PS C:\Users\Usuari\Documents\UNI 4\TFG> pip install pandas
Requirement already satisfied: pandas in c:\users\usuari\appdata\local\progra
1.3.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\usuari\appd
\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2017.3 in c:\users\usuari\appdata\local\
ages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\usuari\appdata\local
kages (from pandas) (1.21.6)
Requirement already satisfied: six>=1.5 in c:\users\usuari\appdata\local\prog
 (from python-dateutil>=2.7.3->pandas) (1.16.0)
```

Fuente: visualización terminal Visual Studio Code

A partir de aquí ya podrás cargar la librería en tu consola y empezar a crear tu programa.

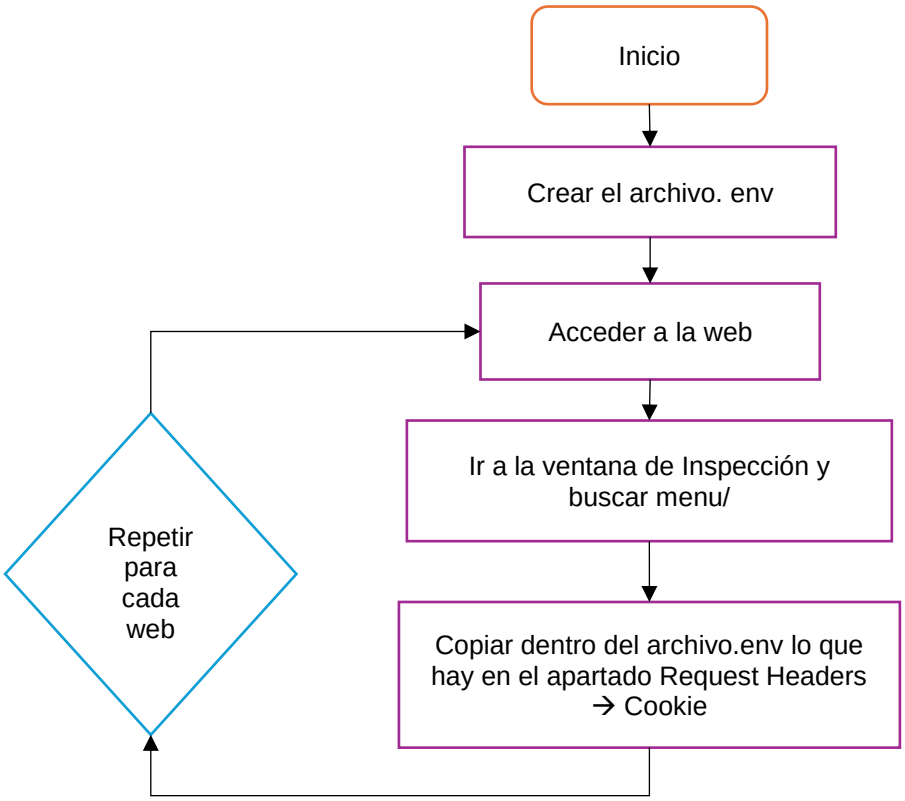
2. Función para scrapear los supermercados: Carrefour, DIA y Mercadona

En primer lugar, cabe mencionar que, para cada una de las webs, se ha seguido el código de buenas prácticas y se han revisado todos los archivos *robots.txt* y los términos y condiciones de cada página web, el análisis en detalle se encuentra en el anexo 2.2.

Dentro del repositorio de GitHub encontramos un archivo de texto llamado *requeriments.txt* con todas las librerías necesarias para scrapear los datos. Por tanto, debemos entrar en la terminal de Python y poner '*pip install requirements.txt'*, este comando nos instala todas las librerías necesarias.

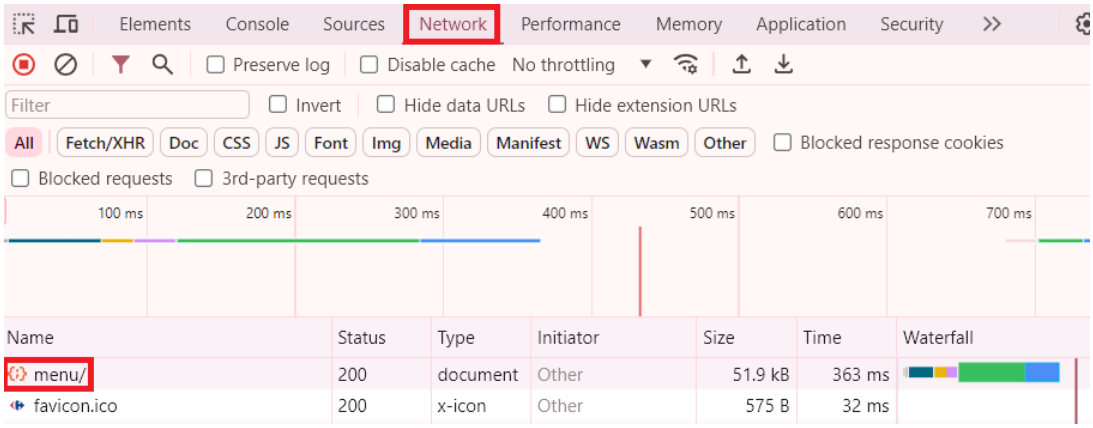
El segundo paso implica la creación de un archivo.env, para poder especificar las *Cookies* de los diferentes supermercados. En este caso, se cargarán las *Cookies* para DIA y Carrefour, ya que para Mercadona no es necesario. Para completar el archivo, hay que seguir una serie de pasos, los cuales están descritos en un diagrama de flujo para facilitar su comprensión. Luego, se detallarán estos pasos con mayor claridad.

Diagrama 4.1: Diagrama de flujo para la creación del archivo.env



1. Ir a la URL de la que se desea extraer los datos: https://www.carrefour.es/cloud-api/categories-api/v1/categories/menu/
2. Vamos a la ventana de Inspección, seguidamente al apartado Network y Ctrl+R.

Figura 4.19: Código fuente apartado Network



Fuente: Carrefour.es

De esta ventana, nos interesa el apartado de *menu/.*

3. Hacemos clic en *menu/* y vamos al apartado *Headers* y buscamos *Request Headers* y dentro de este el elemento *Cookie*:

Figura 4.19: Elemento Cookie

Fuente: Carrefour.es

Todo el código que nos aparece en *Cookie* lo copiamos y lo pegamos en nuestro archivo *.env*. Repetimos los siguientes pasos para el supermercado DIA.

4. El archivo *.env* debe de tener la siguiente forma:

Figura 4.20: Archivo.env

```
COOKIE_CARREFOUR = 'visid_incap_258278=rCD/+aIWRQ
COOKIE_DIA = 'OptanonAlertBoxClosed=2024-04-17T17
```

Fuente: Repositorio de GitHub

Una vez ya tenemos configuradas las Cookies ya podemos ejecutar el código de '*python main_supermarket.py*':

Diagrama 4.2: Diagrama de flujo para el código 'python main_supermarket.py'

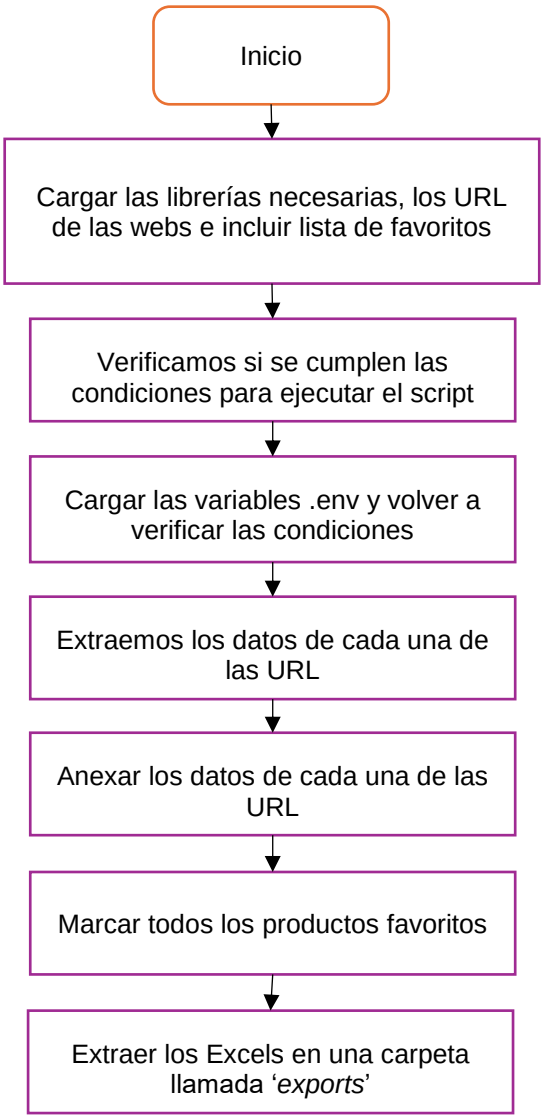


Figura 4.21: Primera parte del código main_supermarket.py

```
from excel_data import *
from supermarket.mercadona import *
from supermarket.carrefour import *
from supermarket.dia import *
from datetime import *
from dotenv import load_dotenv
import pandas as pd

URL_CATEGORY_CARREFOUR = "https://www.carrefour.es/cloud-api/categories-api/
URL_PRODUCTS_BY_CATEGORY_CARREFOUR = "https://www.carrefour.es/cloud-api/plp
URL_CATEGORY_DIA = "https://www.dia.es/api/v1/plp-insight/initial_analytics/
URL_PRODUCTS_BY_CATEGORY_DIA = "https://www.dia.es/api/v1/plp-back/reduced"

LIST_IDS_FAVORITES_PRODUCTS_MERCADONA = ["4749", "4193", "4954", "4957", "19
                                        "11801", "67658", "11178", "13594", "2794", "4523", "279
                                        "35884", "2870", "15611", "86074", "53141", "59124", "59
                                        "10381", "20727", "10199", "40732", "40180", "71380", "4

LIST_IDS_FAVORITES_PRODUCTS_CARREFOUR = []
LIST_IDS_FAVORITES_PRODUCTS_DIA = []
```

Fuente: Repositorio de GitHub

Esta primera parte del código nos carga las librerías necesarias. Primero importamos '*excel_data*', '*supermarket.mercadona*', '*supermarket.carrefour*' y '*supermarket.dia*'. Estas son archivos .py que se han creado antes de realizar este código para que se pueda ejecutar, para obtener más detalles sobre ellas consultar el repositorio o en el apartado 4.3.3.1.1. donde se analiza el código para el supermercado DIA. Básicamente, '*excel_data*' sirve para tratar los datos en Excel, es decir, exportarlos, darles formato y poder almacenarlos, las demás son los códigos para scrapear cada una de las webs.

Seguidamente, se obtienen todas las URL de las webs que se van a usar para obtener información de los productos, como se puede observar son APIs de las webs. Además, se definen listas de IDs favoritos para cada uno de los supermercados, en este caso, se seleccionan algunos productos favoritos para Mercadona. Luego, con la función '*check_favortites_products*' se marcarán aquellos productos que hemos marcado su ID como favorito, creado una columna con el valor TRUE/FALSE. Esta función no tiene mucha utilidad, simplemente sirve para marcar aquellos productos favoritos especificando su ID.

Figura 4.22: Segunda parte del código main_supermarket.py

```
def checkContions(ruta_env, ruta):
    # Si la carpeta export no existe
    if not os.path.exists(ruta):
        print("En la raiz del proyecto debe haber una carpeta llamada 'export'")
        return False

    # Si el archivo .env no existe
    if not os.path.exists(ruta_env):
        print("En la raiz del proyecto debe encontrarse el archivo .env del proyecto")
        return False

    # Si la carpeta export no existe
    if not os.path.exists(ruta):
        print("En la raiz del proyecto debe haber una carpeta llamada 'export'")
        return False

    # Si la variable de sesion COOKIE_CARREFOUR no existe
    if os.getenv('COOKIE_CARREFOUR') is None:
        print("En el archivo .env debe existir una variable llamada 'COOKIE_CARREFOUR'
        return False

    if os.getenv('COOKIE_CARREFOUR') == "TU_COOKIE_CARREFOUR":
        print("Debe cumplimentar en el archivo .env la variable llamada 'COOKIE_CARREFO
        return False

    # Si la variable de sesion COOKIE_DIA no existe
    if os.getenv('COOKIE_DIA') is None:
        print("En el archivo .env debe existir una variable llamada 'COOKIE_DIA' con la
        return False

    if os.getenv('COOKIE_DIA') == "TU_COOKIE_DIA":
        print("Debe cumplimentar en el archivo .env la variable llamada 'COOKIE_DIA' co
        return False
```

Fuente: Repositorio de GitHub

En esta segunda parte se crea la función '*checkContions*', que verifica si se cumplen ciertas condiciones necesarias para ejecutar el script, como la presencia de carpetas y archivos, además de la presencia de ciertas variables de entorno con valores específicos. Si alguna de las condiciones falla nos envía un mensaje de error con el problema en concreto, de esta manera, si algo falla será fácil detectar el porqué.

Figura 4.22: Tercera parte del código main_supermarket.py

```
if checkContions(ruta_env, ruta) == True:
    print("")
    print("-----------------------------------MERCADONA-----------------------------------")
    print("")
    df_mercadona = gestion_mercadona(ruta)
    print("")
    print("-----------------------------------CARREFOUR-----------------------------------")
    print("")
    df_carrefour = gestion_carrefour(ruta)
    print("")
    print("-------------------------------------DIA-------------------------------------")
    print("")
    df_dia = gestion_dia(ruta)

    # Unir los DataFrames verticalmente
    if(len(df_carrefour.columns) != 0):
        df_supermercados = pd.concat([df_mercadona, df_carrefour], ignore_index=True)
    else:
        df_supermercados = pd.concat([df_mercadona], ignore_index=True)

    if(len(df_dia.columns) != 0):
        df_supermercados = pd.concat([df_supermercados, df_dia], ignore_index=True)

    #Marcar los productos favoritos
    df_supermercados = check_favortites_products(df_supermercados)

    #Export Excel
    export_excel(df_supermercados, ruta, "products", "Productos")
```

Fuente: Repositorio de GitHub

En esta tercera parte, se cargan las variables de entorno desde el archivo .env y verifica las condiciones necesarias para ejecutar el script, si lo cumple realiza lo siguiente:

- Extrae los datos para cada uno de los supermercados.
- Une los DataFrame de los productos de los diferentes supermercados.
- Marca los productos favoritos en el DataFrame resultante.
- Exporta toda la información de los productos a un archivo Excel en la carpeta especificada (‘*export*’).

2.1. Código para scrapear los datos de DIA

En este apartado, se pretende analizar una de las tres funciones de los supermercados para saber cómo rastrear los datos de las webs. Las tres funciones son prácticamente idénticas, la del DIA y Carrefour no tienen casi ninguna diferencia. Por otro lado, la del Mercadona lo único que no incluye es la verificación de las *Cookies* y la definición de los *Headers* porque no son necesarios.

Figura 4.23: Primera parte del código dia.py

```
from excel_data import *
import os
from datetime import *
import requests
from dotenv import load_dotenv
import pandas as pd
import time

URL_CATEGORY_DIA = "https://www.dia.es/api/v1/plp-insight/initial_analytics/charcuter
URL_PRODUCTS_BY_CATEGORY_DIA = "https://www.dia.es/api/v1/plp-back/reduced"

HEADERS_REQUEST_DIA = {
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
    'Accept-Encoding': 'gzip, deflate, br',
    'Accept-Language': "es-GB,es;q=0.9",
    'Cookie': '',
    'Sec-Ch-Ua': '"Not_A Brand";v="8", "Chromium";v="120", "Google Chrome";v="120"',
    'Sec-Ch-Ua-Mobile': '?0',
    'Sec-Ch-Ua-Platform': '"Windows"',
    'Sec-Fetch-Mode': 'navigate',
    'Sec-Fetch-Site': 'none',
    'Sec-Fetch-User': '?1',
    'Upgrade-Insecure-Requests': '1',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
}
```

Fuente: Repositorio de GitHub

En esta primera parte se importan las librerías y funciones necesarias. La '*excel_data'* sirve para tratar los datos en Excel, *'os'* proporciona funciones para interactuar con el sistema operativo, *'datetime'* se utiliza para trabajar con fechas y horas, *'requests'* se utiliza para realizar solicitudes HTTP, *'dotenv'* se usa para cargar las variables de entorno desde un archivo .env, *'pandas'* se usa para trabajar con estructuras de datos tabulares y *'time'* proporciona funciones relacionadas con el tiempo. Luego, define dos URL, la primera para obtener información sobre la categoría productos del DIA y la segunda para obtener la subcategoría de cada uno de los productos. Por último, se proporciona un pequeño diccionario '*HEADERS_REQUEST_DIA*' que contiene encabezados de solicitud HTTP que se utilizarán al realizar solicitudes a la API de DIA.

Figura 4.24: Segunda parte del código dia.py

```
def gestion_dia(ruta):

    #Add cookie into header json
    HEADERS_REQUEST_DIA["Cookie"] = os.getenv('COOKIE_DIA')

    # Registra el tiempo de inicio
    tiempo_inicio_dia = time.time()

    # Ejecutar las acciones para obtener los productos
    list_categories = get_ids_categorys_dia()
    df_dia = get_products_by_category_dia(list_categories, ruta)

    # Registra el tiempo de finalización
    tiempo_fin_dia = time.time()

    # Calcula la duración total
    duracion_total_dia = tiempo_fin_dia - tiempo_inicio_dia

    # Obtener minutos y segundos
    minutos_dia = int(duracion_total_dia // 60)
    segundos_dia = int(duracion_total_dia % 60)

    # Mostrar el log con el tiempo de ejecución en minutos y segundos
    print(f"La ejecución de DIA ha tomado un tiempo de {minutos_dia} minutos y {segundos_dia} segundos")

    return df_dia
```

Fuente: Repositorio de GitHub

En esta segunda parte se crea la función 'gestión_dia', primero configura las *Cookies* y luego, realiza un registro del tiempo al inicio de la ejecución. Seguidamente, obtiene los ID de cada

producto y registra el tiempo de finalización de la ejecución. Calcula la duración total de la ejecución, le da formato y lo imprime en la consola. Por último, nos retorna un *DataFrame* con los datos de los productos de DIA.

Figura 4.25: Tercera parte del código dia.py

```
def get_products_by_category_dia(list_categories, ruta):

    df_products = pd.DataFrame()

    for index, stringCategoria in enumerate(list_categories):
        print(str(index+1)+"/" + str(len(list_categories)) + " - Procesando los productos de la categ
        url = URL_PRODUCTS_BY_CATEGORY_DIA + str(stringCategoria)

        #Obtener los productos de la categoria
        list_products_data = requests.get(url, headers=HEADERS_REQUEST_DIA)
        list_products = list_products_data.json()

        try:
            df_productos = pd.json_normalize(list_products["plp_items"], sep="_")

            # Concatenar "https://www.dia.es" a la columna "url"
            df_productos['url'] = 'https://www.dia.es' + df_productos['url']
            df_productos['image'] = 'https://www.dia.es' + df_productos['image']
            df_productos['categoria'] = stringCategoria
            df_productos['supermercado'] = "Dia"

            # Seleccionar y renombrar columnas
            selected_columns = ['object_id', 'display_name', 'prices_price', 'prices_price_per_unit',
            renamed_columns = {
                'object_id': 'Id',
                'display_name': 'Nombre',
                'prices_price': 'Precio',
                'prices_price_per_unit': 'Precio Pack',
                'prices_measure_unit': 'Formato',
                'categoria': 'Categoria',
                'supermercado': 'Supermercado',
                'url': 'Url',
                'image': 'Url_imagen'
                }
```

Fuente: Repositorio de GitHub

En esta tercera parte, se crea la fusión '*get_products_by_category_dia'*, para verla completa ir al repositorio. Primero crea un *DataFrame* vacío para almacenar los datos. Luego, se crea un bucle para iterar sobre cada categoría en la lista de categorías proporcionada. Esta lista de categorías es la que se ha definido en la primera parte del código. A partir de aquí, obtiene los datos para cada una de las categorías mediante solicitudes HTTP GET a la URL usando los encabezados definidos anteriormente y la respuesta la convierte en formato JSON. Además, realiza un procesamiento de los datos, es decir, normaliza los datos JSON y se realizan operaciones de limpieza como: concatenar las URL's, seleccionar y renombrar columnas y agregar información adicional. Por último, se realiza una concatenación de los *DataFrames* y exporta los datos a Excel.

Figura 4.26: Cuarta parte del código dia.py

```
def get_ids_categorys_dia():

    try:
        list_category_dia_data = requests.get(URL_CATEGORY_DIA, headers=HEADERS_REQUEST_DIA)
        list_category_dia = list_category_dia_data.json()
    except:
        print("ERROR - La cookie proporcionada para el supermercado DIA ha caducado")
        return []

    info = list_category_dia['menu_analytics']

    # Crear el DataFrame resultante
    data = procesar_nodo(info)
    df_resultado = pd.DataFrame(data, columns=['id', 'parameter', 'path'])

    # Utilizar pandas para operaciones de datos
    # Filtrar solo las filas con valores en la columna 'parameter'
    df_resultado = df_resultado[df_resultado['parameter'].notna()]

    category_ids = df_resultado["path"].explode().dropna().astype(str).tolist()


    return category_ids
```

Fuente: Repositorio de GitHub

En esta cuarta parte del código, se obtienen los ID para cada una de las categorías mediante la función '*get_ids_category_dia*'.

Es importante tener en cuenta que, en la implementación completa del código, hay una función distinta para cada supermercado, como ya se ha comentado, pero aquí se muestra solo el ejemplo del supermercado DIA para tener una idea de cómo rastrea los datos. Además, también hay una función donde concatena los datos de los tres supermercados y los exporta en un Excel conjunto.

2.2. Análisis de los archivos robots.txt

Primeramente, se analizará la web de Carrefour https://www.carrefour.es/ para realizar web scraping.

Inicialmente, se debe revisar este archivo, ya que es una sugerencia de como interactuar con el sitio web (explicado en el apartado 4.1.2.1.). En este caso, se revisará el archivo robots.txt del Carrefour (https://www.carrefour.es/robots.txt).

Robotx.txt Carrefour I

```
User-agent:*
Disallow: /ayuda/mas-info
Disallow: /nuevatienda/
Disallow: /buscador/
Disallow: /city/
Disallow: /supermercado/ajax/*
Disallow: /supermercado/CambioCentro
Disallow: /supermercado/ConfirmarPedido
Disallow: /supermercado/ConfirmarPendientePedido
Disallow: /supermercado/forms/*
Disallow: /supermercado/MiCarrito
Disallow: /supermercado/mylists
Disallow: /supermercado/myorders
Disallow: /supermercado/orderDetail
Disallow: /supermercado/Paypal
Disallow: /supermercado/PaypalProcesandoPago
Disallow: /supermercado/ProcesandoPago
Disallow: /supermercado/ProcesandoSeguridad
Disallow: /supermercado/ProcesoPago
Disallow: /supermercado/ProcesandoPagoPendiente
Disallow: /supermercado/Test_list_products
Disallow: /MiCarrito/ProcessIframe
Disallow: /MiCarrito/Process3ds
Disallow: /pedido-exitoso
Disallow: /pedido-erroneo
Disallow: /ProcesoPago
Disallow: /cloud-api/one-cart-api/v1/carts/current/process_payment
Disallow: /cloud-api/one-cart-api/v1/carts/current/payment_redirect
Disallow: /myaccount/*
Disallow: /order_locator
Disallow: /logout
Disallow: /refresh_token
Disallow: /supermercado/all-reviews/*
Disallow: /supermercado/review-form/*
Disallow: /browse/*
Disallow: /HuchaPromocion*
Disallow: /e-commerce/www/*
User-agent: TwengaBot
Crawl-delay: 60
```

Fuente: Carrefour robotx.txt

En esta primera parte del archivo para todos los agentes (*user-agent:* *): se prohíben todas las rutas y recursos del sitio web con la directiva *Disallow*. Además, se excluye el acceso a ciertas URL relacionadas con la administración de cuentas y usuarios (*/myaccount/*, /order_locator, /logout*). Por último, para el agente *TwengaBot* se establece un tiempo entre peticiones de 60 segundos (*Crawl-delay: 60*).

Robotx.txt Carrefour II

```
#FILTERS
User-agent:*
Disallow: */F-*Z*Z*
Disallow: */supermercado/*/F-*Z*Z*
Disallow: */N-*Z*Z*
Disallow: /*-euros-a-*
Disallow: /*-vendido-por-carrefour*
Disallow: */F-*Z*/g


#PARAMETERS
User-agent:*
Disallow: /c?q=*
Disallow: /?q=*
Disallow: /c?Ntt=*
Disallow: /*login?*
Disallow: /c?sort=*
Disallow: /?gclsrc=aw.ds*
Disallow: /p?awaid=*
Disallow: /?userLoginOrigin=*
Disallow: /shopping_cart?_requestid=*
Disallow: /?source=*
Disallow: /*c?+*
Disallow: /*c?La=*
Disallow: /*c?No=*
Disallow: /*c?Nr=*
Disallow: /*c?Ns=*
Disallow: /c?No=*
Disallow: /c?categoryId=
Disallow: /*?promoId=*
Disallow: /a?id=*
Disallow: /c?selected_permanent_facets=*
Disallow: /c?filter=*
Disallow: /*?tcmUri=*
Disallow: /*c?permanent_facets_lost=*
Disallow: /*?sort=*
```

Fuente: Carrefour robotx.txt

En este caso, también se dirige a todos los agentes (*user-agent: **): excluye todas las URL que contienen los patrones especificados en la imagen.

Figura 4.3: Robotx.txt Carrefour III

```
#New bots
User-agent: Pinterestbot
Allow: .jpg
Allow: .png
```

Fuente: Carrefour robotx.txt

Aquí nos especifica que para el agente Pinterestbot, se permite acceso (*Allow:*) a los archivos con extensiones .jpg y .png.

Robotx.txt Carrefour IV

```
# Begin block Bad-Robots from robots.txt
User-agent: asterias
Disallow:/
User-agent: BackDoorBot/1.0
Disallow:/
User-agent: Black Hole
Disallow:/
User-agent: BlowFish/1.0
Disallow:/
User-agent: BotALot
Disallow:/
User-agent: BuiltBotTough
Disallow:/
User-agent: Bullseye/1.0
Disallow:/
User-agent: BunnySlippers
Disallow:/
User-agent: Cegbfeieh
Disallow:/
User-agent: CheeseBot
Disallow:/
User-agent: CherryPicker
Disallow:/
User-agent: CherryPickerElite/1.0
Disallow:/
User-agent: CherryPickerSE/1.0
Disallow:/
User-agent: CopyRightCheck
Disallow:/
User-agent: cosmos
Disallow:/
User-agent: Crescent
Disallow:/
User-agent: Crescent Internet ToolPak HTTP OLE Control v.1.0
Disallow:/
User-agent: DittoSpyder
Disallow:/
```

Fuente: Carrefour robotx.txt

En este apartado, excluye bots que podrían ser maliciosos o no deseados al sitio web. La directiva *Disallow* indica que los bots listados (*asterias, BackDoorBot/1.0, Black Hole, BlowFish/1.0, BotALot* …) no tienen permitido acceder a ninguna parte del sitio web, lo indica con la /. Previene actividades maliciosas o no deseadas.

Robotx.txt Carrefour V

```
#Sitemaps
Sitemap: https://www.carrefour.es/_includes/sitemap.xml
Sitemap: https://www.carrefour.es/tecno/dyn/MEDIA/siteindex.xml
Sitemap: https://www.carrefour.es/crs/cdn-static/sitemap-food/index.xml
Sitemap: https://www.carrefour.es/crs/cdn-static/sitemap-bodega/index.xml
```

Fuente: Carrefour robotx.txt

Este último trozo indica que Carrefour tiene cuatro *sitemaps* disponibles. Un *sitemaps*, mapa del sitio web, enumera las páginas de un sitio web, es muy útil para los sitios web grandes, ya que proporcionan a los motores de búsqueda una lista de todas las páginas disponibles para buscar información. Ayudando a la visibilidad del sitio en los resultados de búsqueda. Cada una de estas cuatro URL proporciona acceso a diferentes partes del sitio web, proporciona a los motores de búsqueda indexar de manera eficiente y completa al contenido del sitio web.

Por último, se revisará el archivo robots.txt para la web de Mercadona https://www.mercadona.es/robots.txt.

Archivo robots.txt Mercadona

```
# www.robotstxt.org/

User-agent: *
Disallow:

Sitemap: https://www.mercadona.es/sitemap.xml
```

Fuente: mercadona.es

En este caso, vemos que este archivo tiene una dimensión mucho más pequeña que Carrefour y DIA. En este caso, '*User-agent:* * 'se refiere a todos los robots de búsqueda, '*Disallow*:', como está en blanco, significa que no hay ninguna restricción específica para los robots de qué partes se pueden rastrear y no. Por último, nos indica un '***Sitemap'*** este indica donde se puede encontrar el archivo XML, dentro de este se encuentran todas las URL del sitio web que el propietario deja scrapear y nos aporta información adicional sobre cada una de ellas como la URL, la frecuencia de actualización y la importancia en la web.

3. Descarga e instalación de PowerBI

PowerBI es una herramienta de Microsoft gratuita, por ello, su instalación es muy sencilla. Vamos a la siguiente web: https://powerbi.microsoft.com/es-es/downloads/. En esta página encontraremos un apartado llamado Microsoft Power BI Desktop, PowerBI para el escritorio, le damos a descargar, se nos abrirá la tienda de Microsoft le damos las autorizaciones para descargar y ya tendremos listo el PowerBI en nuestro escritorio.

Donde descargar PowerBI Desktop

Fuente: powerbi.microsoft.com