

Práctica 1

Tipología y ciclo de vida de los datos

Índice

Uoc

- 1.Contexto
- 2.Título
- 3.Descripción del dataset
- 4.Representación gráfica
- 5.Contenido
- 6.Propietario
- 7.Inspiración
- 8.Licencia
- 9.Código
- 10.Dataset
- 11.Vídeo



1.Contexto

Para esta práctica se ha decidido recolectar los datos de todas las bicicletas en venta en la página web [deporvillage.com](https://www.deporvillage.com). El objetivo de este dataset es de poder observar las variaciones de precio de las bicicletas a lo largo del tiempo como hacen algunas páginas que rastrean el precio de productos online, como por ejemplo [camelcamelcamel](https://www.camelcamelcamel.com). En el mercado de las bicicletas se han observado grandes fluctuaciones desencadenadas por la pandemia que han causado en primera instancia una subida generalizada de los precios por la gran demanda de artículos que está tardando en estabilizarse. Gracias a los datos que puede recolectar este scraper será posible monitorizar periódicamente los precios y la disponibilidad de bicicletas y hacer un estudio más profundizado sobre este tema. Asimismo se recogen también todas las características de las bicicletas a la venta, por ejemplo para que otra página pueda hacer un estudio de mercado comparando los productos.

Deporvillage.com es un sitio que vende productos deportivos de todo tipo y es uno de los más grandes en España en este sector. La dirección de la página es www.deporvillage.com. Para este ejemplo se utilizará la categoría bicicletas: <https://www.deporvillage.com/bicicletas>

2.Título

El título del dataset será `deporvillage_bicicletas_[fecha]`

El campo fecha recogerá la fecha de descarga de los datos para poder comparar los precios y los productos en distintas fechas.

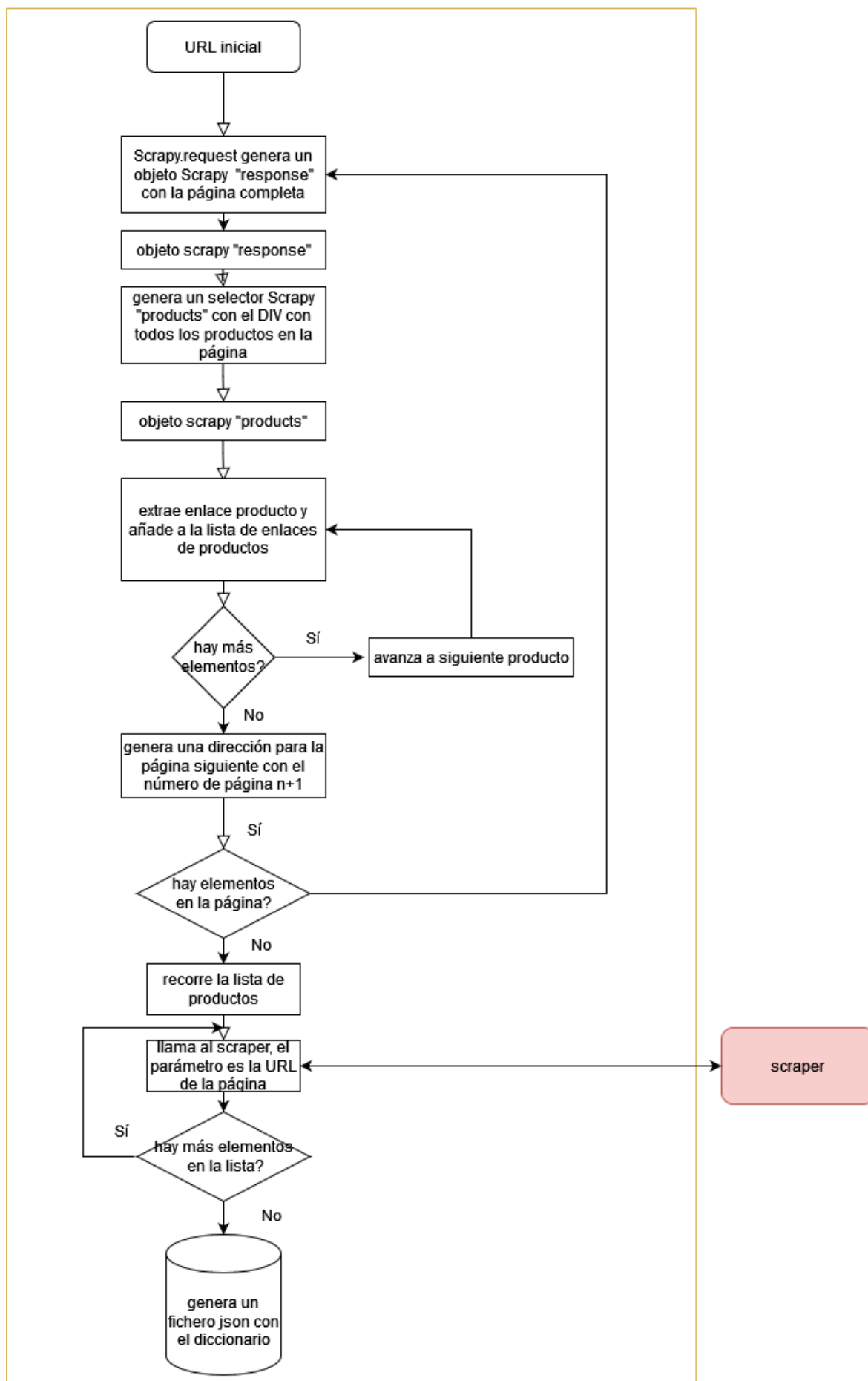
3.Descripción del dataset

El dataset generado por el scraper contiene todos los datos de las bicicletas a la venta en la página <https://www.deporvillage.com/bicicletas> en la fecha de descarga de los datos.

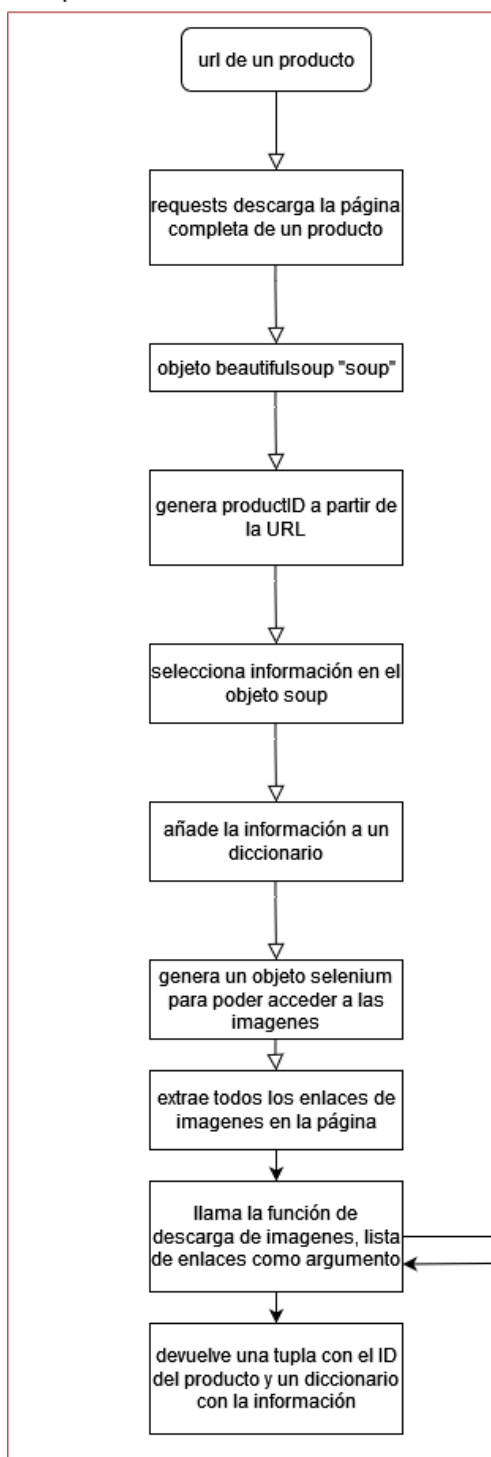
El dataset contiene las informaciones esenciales para identificar los productos (modelo de la bicicleta), el precio de venta, y todas las características disponibles de cada bicicleta para poder luego, asimismo descarga también las imágenes de cada modelo y las guarda en una carpeta dedicada.

4.Representación gráfica

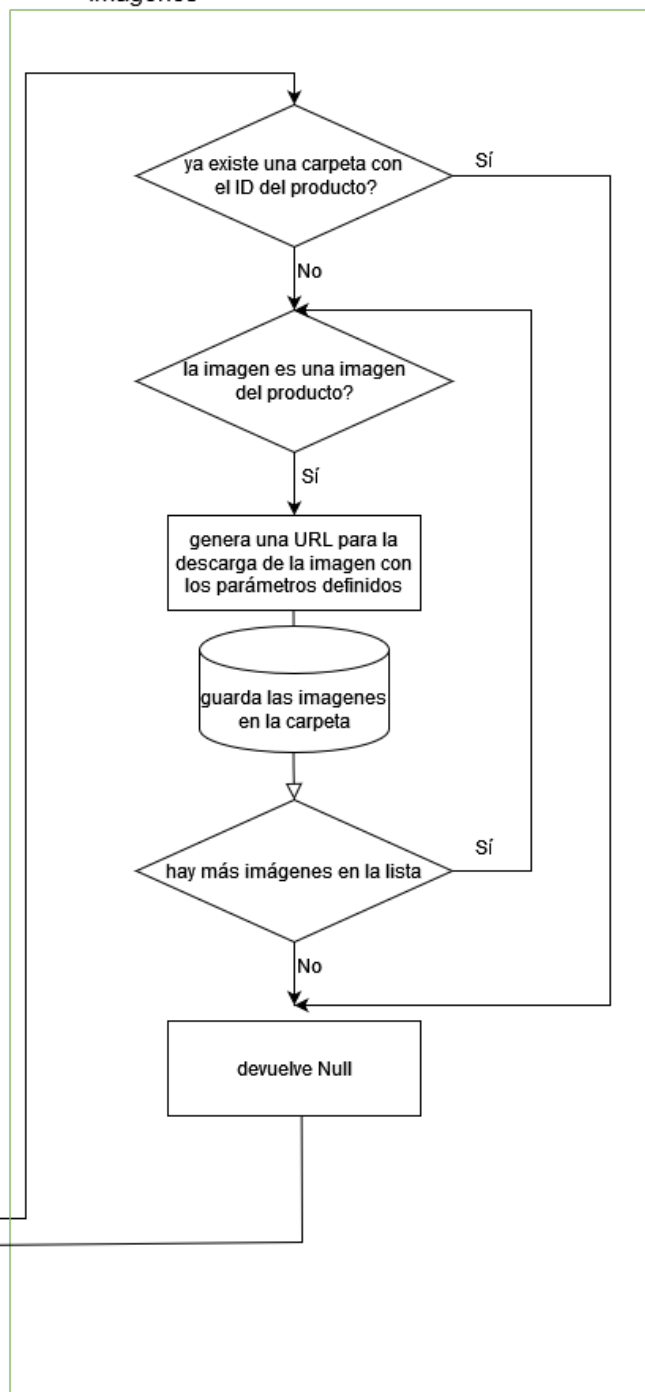
Crawler



Scraper



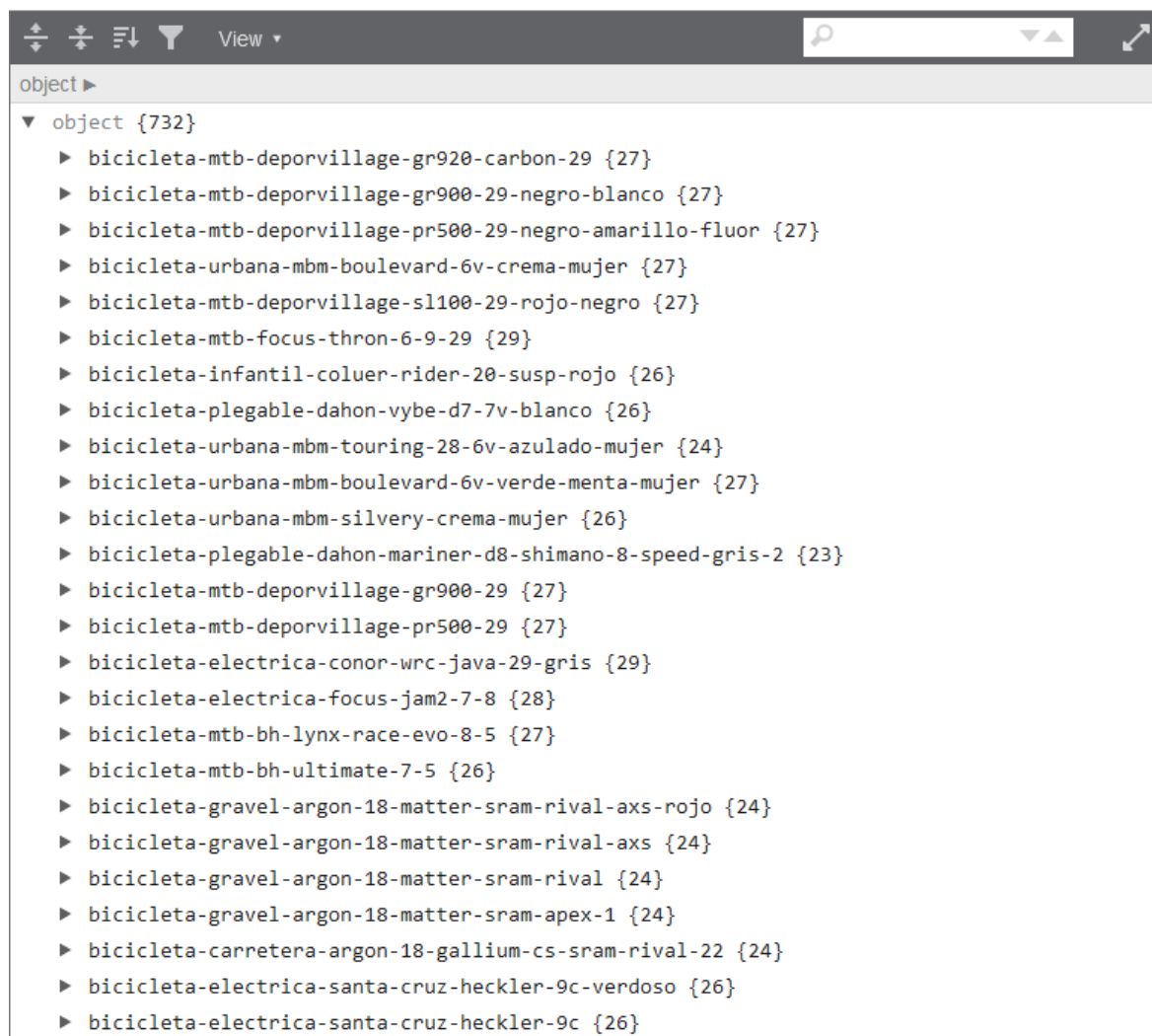
Descarga de imagenes



5.Contenido

El contenido del dataset es:

1. Un fichero JSON que contiene todos los datos scrapeados de las bicicletas, el fichero tendrá el siguiente formato:



```

object
  ▼ object {732}
    ▶ bicicleta-mtb-deporvillage-gr920-carbon-29 {27}
    ▶ bicicleta-mtb-deporvillage-gr900-29-negro-blanco {27}
    ▶ bicicleta-mtb-deporvillage-pr500-29-negro-amarillo-fluor {27}
    ▶ bicicleta-urbana-mbm-boulevard-6v-crema-mujer {27}
    ▶ bicicleta-mtb-deporvillage-sl100-29-rojo-negro {27}
    ▶ bicicleta-mtb-focus-thron-6-9-29 {29}
    ▶ bicicleta-infantil-coluer-rider-20-susp-rojo {26}
    ▶ bicicleta-plegable-dahon-vybe-d7-7v-blanco {26}
    ▶ bicicleta-urbana-mbm-touring-28-6v-azulado-mujer {24}
    ▶ bicicleta-urbana-mbm-boulevard-6v-verde-menta-mujer {27}
    ▶ bicicleta-urbana-mbm-silvery-crema-mujer {26}
    ▶ bicicleta-plegable-dahon-mariner-d8-shimano-8-speed-gris-2 {23}
    ▶ bicicleta-mtb-deporvillage-gr900-29 {27}
    ▶ bicicleta-mtb-deporvillage-pr500-29 {27}
    ▶ bicicleta-electrica-conor-wrc-java-29-gris {29}
    ▶ bicicleta-electrica-focus-jam2-7-8 {28}
    ▶ bicicleta-mtb-bh-lynx-race-evo-8-5 {27}
    ▶ bicicleta-mtb-bh-ultimate-7-5 {26}
    ▶ bicicleta-gravel-argon-18-matter-sram-rival-axs-rojo {24}
    ▶ bicicleta-gravel-argon-18-matter-sram-rival-axs {24}
    ▶ bicicleta-gravel-argon-18-matter-sram-rival {24}
    ▶ bicicleta-gravel-argon-18-matter-sram-apex-1 {24}
    ▶ bicicleta-carretera-argon-18-gallium-cs-sram-rival-22 {24}
    ▶ bicicleta-electrica-santa-cruz-heckler-9c-verdoso {26}
    ▶ bicicleta-electrica-santa-cruz-heckler-9c {26}
  
```

Cada entrada se identifica con el ID del producto que es parte de su URL y contendrá la siguiente información:

Nombre producto: El nombre comercial del producto

Precio Original: El precio sin descontar (si no hay descuentos este precio no existe)

Precio Venta: El precio al que se vende la bicicleta

Marca: Marca

Breadcrumb: El recorrido de categorías que permite ubicar el producto dentro de la estructura de la página web

Tags: Los tags que caracterizan el producto

Estrellas: Número de estrellas de valoración de un producto

Descripción: La descripción extendida del producto

Talla: Las tallas disponibles

Componentes: Un listado de componentes que varía en función del producto, ciertas bicicletas tendrán componentes que otras no tienen como por ejemplo un cambio delantero, o la batería en el caso de bicicletas eléctricas, etc. Por tanto, esta para del fichero JSON es variable.

Datetime: hora y fecha en que se ha descargado la información.

2. Una estructura de carpetas que contiene las imágenes de cada producto. El nombre de la carpeta es el ID del producto, este ID coincide con la URL de la página y con el identificador del JSON.

Name	Date modified	Type
bicicleta-bmx-atala-crime-20	21/04/2023 14:34	File folder
bicicleta-bmx-umit-20	21/04/2023 14:20	File folder
bicicleta-carretera-argon-18-gallium-cs-...	21/04/2023 13:59	File folder
bicicleta-carretera-argon-18-gallium-cs-...	21/04/2023 15:41	File folder
bicicleta-carretera-argon-18-gallium-cs-...	21/04/2023 15:42	File folder
bicicleta-carretera-argon-18-gallium-cs-...	21/04/2023 15:41	File folder
bicicleta-carretera-argon-18-sum-pro-sr...	21/04/2023 15:41	File folder
bicicleta-carretera-argon-18-sum-sram-f...	21/04/2023 15:42	File folder

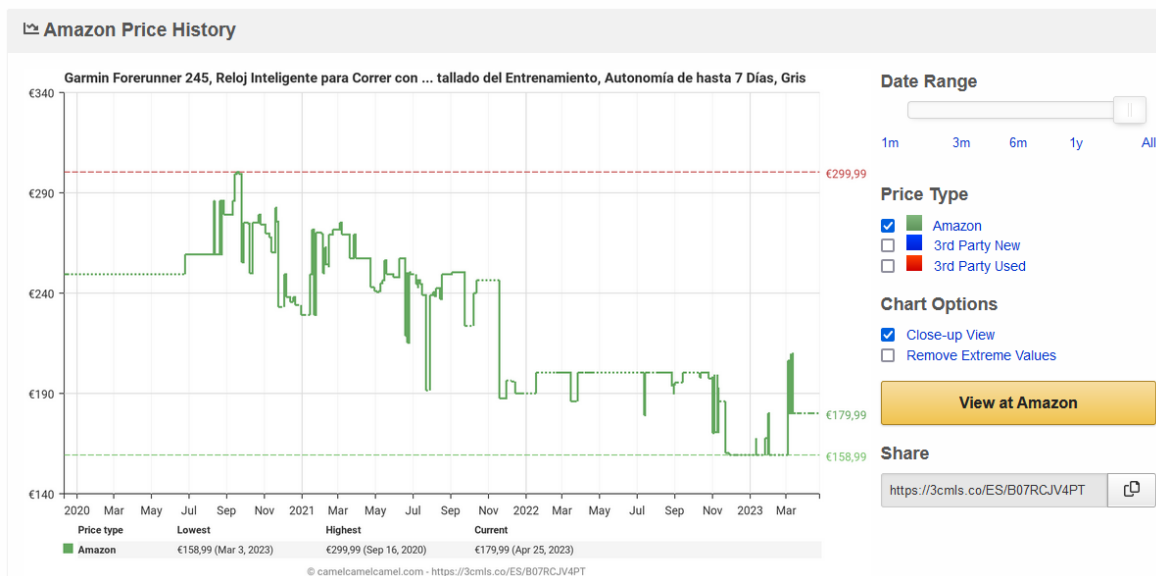
Los datos se han descargado el día 21 de abril y otra vez el día 23 del mismo mes. Como se ha comentado antes la idea es ir descargando estos datos periódicamente para poder hacer comparaciones de precios.

6. Propietario

El propietario de los datos es deporvillage.com. Tratándose de una página web de comercio electrónico los datos están disponibles para el público y el scraper simplemente los recoge todos. El scraper no sobrecarga ni daña el sitio web así que se entiende que no hay ningún incumplimiento de las condiciones legales que se pueden encontrar en la página: <https://www.deporvillage.com/condiciones-legales> y tampoco con las condiciones que se encuentran en: <https://www.deporvillage.com/condiciones-generales-compra-web-app>

Las marcas registradas y los derechos de autor siguen perteneciendo a deporvillage.com.

Experiencias parecidas de productos que permiten ver el historial de precios a lo largo del tiempo son por ejemplo <https://es.camelcamelcamel.com/> para los productos de amazon (para todas las páginas nacionales de amazon) o por ejemplo <https://steampricehistory.com/> para los precios de los videojuegos en Steam. Asimismo las empresas con un negocio parecido al de deporvillage (venta de artículos deportivos) tendrán mucho interés en hacer periódicamente revisiones de los precios de la competencia para ajustar los suyos.



Ejemplo de visualización de las variaciones de precio de un producto en amazon.es visualizada en camelcamelcamel.com

Para realizar el scraping se ha comprobado que no generase problemas con el fichero <https://www.deporvillage.com/robots.txt> y para no saturar el servidor se han espaciado las requests de 0,5 segundos. Se ha llevado a cabo el scraping en varias ocasiones y nunca se han tenido problemas de saturación del servidor o con la dirección IP.

7.Inspiración

Este conjunto de datos es un conjunto de ejemplo de los datos que se pueden recabar de una página como deporvillage.com. El ejemplo de las bicicletas es interesante porque sus precios varían de manera significativa a lo largo del tiempo. Hay webs comerciales que permiten trazar el precio de ciertos productos a lo largo del tiempo como las que se han mencionado en el apartado anterior. Para hacer este tipo de análisis, como la de camelcamelcamel.com, sería suficiente escrapear el precio del producto, que sería un proceso mucho más rápido y sencillo que lo que lleva a cabo este scraper. Sin embargo descargar toda la información técnica de las bicicletas es una forma para generar también una base de datos de componentes y elementos que permitiría a otra página de artículos deportivos analizar la competencia y comparar los precios y las características de los productos. Una vez obtenido el database completo con todas las bicicletas se pueden analizar otros parámetros tales como:

- Rango de precios de bicicletas de carretera (o eléctricas, o de montaña)
- Rango de precios de bicicletas con componentes de alta gama (grupos Shimano Ultegra por ejemplo, o componentes de carbono) y compararlos con bicicletas con componentes de gama media o baja.

- Analiza las políticas de descuentos y detectar patrones de subida y bajada de precios a lo largo del año.
- Acceder a las características de productos propios de deporvillage (que tiene su propia marca de bicicletas)
- Detectar cuales son los segmentos de mercado con más o menos productos, en que rango de precios se encuentran, etc.

8.Licencia

Los datos ya son de dominio público por tanto se utiliza la licencia CC0 1.0 Universal tanto para el dataset como para el código. Los derechos de las imágenes pertenecen a los respectivos propietarios, así como la responsabilidad sobre la exactitud de la información. El proyecto no recoge ni publica ninguna información que no fuera ya de dominio público.

9.Código

El código se puede dividir en los siguientes bloques principales:

1. Crawler

Para el Crawler se utiliza Scrapy. El objetivo de esta parte del código es de navegar todas las páginas de la categoría bicicletas de la web deporvillage.com y generar un listado con todos los enlaces válidos a modelos de bicicletas en venta.

Este bloque se puede dividir en dos partes:

En la primera se extraen de la página todos los enlaces contenidos en el “div”

```
products = response.css('div.ProductList_list-item__qgx2K')
```

Este div solo contiene enlaces a bicicletas, por tanto podemos acceder a ellos mediante un bucle que añade los enlaces a una lista:

```
for product in products:
```

```
    product_link = product.css('a::attr(href)').get()
```

```
    products_to_scrape.append(product_link)
```

La segunda parte del bloque prevé la navegación dentro de la web hacia la página siguiente. La forma más sencilla de hacerlo es aprovechando del patrón de generación de las páginas que utiliza el formato “dirección_página_web?p2”.

Cada vez que se scrapea una página se abre la página siguiente generando una nueva dirección con ese patrón, si la página resultante no tiene enlaces útiles significa que la anterior era la última página. Si la página tiene contenido útil entonces se sigue con el proceso añadiendo más enlaces a la lista y continuando recursivamente con la navegación autónoma.

2. Scraper

En el scraper se utilizan BeautifulSoup y Selenium.

Gran parte del trabajo de scraping se hace con BS ya que es relativamente fácil individuar mucha de la información más relevante:

product_id: es un código único que identifica el producto y se obtiene desde la URL del mismo producto

Nombre Producto: se obtiene del elemento "h1" con "itemprop = name" de la página

Precio Original y Precio Venta: estos dos elementos están contenidos en div con un identificador de clase único así que es fácil acceder a ellos. En el caso de que el precio original (sin descontar no existe) se utiliza solamente el precio de venta (retail)

Marca: se puede encontrar en un div

Breadcrumb: se puede extraer como lista de una lista, es útil para saber en que parte de la web se encuentra la página

Tags: se encuentran como elementos de texto dentro de un mismo div que se puede seleccionar con BS

Estrellas: se encuentran en un DIV que se puede encontrar con BS, dentro de este DIV se cuentan los elementos que se identifican como estrellas activas

Descripción: en este apartado entra mucha información que se encuentra toda dentro del mismo DIV identificado por el "itemprop: description". Como la información no está completamente estructurada y varia de producto en producto (por ejemplo entre bicis eléctricas y no...) es necesario recorrer todo el contenido en un bucle y detectar las etiquetas que describen las características y luego el contenido de la descripción. Esto es posible gracias a un patrón que se repite que es que la etiqueta acaba siempre con ":".

La galería de imágenes no es accesible directamente en el objeto BeautifulSoup ya que estas se cargan a parte con JS. Para acceder a ellas se utiliza "selenium" que descarga todos los enlaces de imágenes de la página una vez abierta, esto incluye entre otras cosas los enlaces de las imágenes de la galería.

3. Descarga de imágenes

Como comentado antes Selenium descarga todos los enlaces de imágenes en la página, esta lista de enlaces contiene las imágenes del producto pero también otras imágenes de la página tal como los logos, será necesario por tanto poder seleccionar solamente ciertas imágenes basándose en una palabra clave. Asimismo las imágenes se generan dinámicamente en base a unos parámetros de resolución, calidad, etc. Será por tanto necesario descargar estos ficheros con

la calidad que más nos interesa. Para hacer todo esto se crea una función que se llama `get_pictures(url_list, search_key, height, weidth, quality, producto_id, folder_path, headers)`.

Esta función hace los pasos siguientes:

1. Comprueba que no existe ya una carpeta con las imágenes para este producto, si existe se considera que ya se han descargado y no hace falta volver a descargarlas.
 2. Recorre la lista de direcciones y selecciona solamente las imágenes que contienen la “search_key” es decir la palabra clave que identifica las imágenes de un producto de las que son logos o marcas, en este caso la palabra es “producto”. Si la dirección no contiene esa palabra se descarta la dirección, si no sigue al paso siguiente
 3. Genera una nueva dirección con los parámetros que ha recibido en entrada para descargar una imagen a la resolución deseada y con la calidad deseada, el patrón de este URL se repite y es fácil de generar añadiendo simplemente el nombre de la imagen al final.
 4. Finalmente se hace una requests y se descarga la imagen seleccionada. Si se ha podido descargar con éxito se guarda la imagen en la carpeta utilizando el mismo nombre que tiene en la página web.
4. Creación del fichero JSON

Una vez terminado el proceso de scraping tendremos un diccionario con toda la información sobre los productos, los precios etc. Al finalizar la ejecución el programa genera un fichero JSON con el diccionario. Es importante tener cuidado con el encoding UTF-8 ya que las descripciones contienen muchos caracteres que si no se codificarían mal.

10.Dataset

El dataset se encuentra publicado a la dirección siguiente

<https://zenodo.org/record/7855025>

Y en la propia carpeta de github /dataset

Y también en la carpeta de drive:

https://drive.google.com/drive/folders/1OVwyDvOHRp2bqqXJH0ncM1Z66IyAQBxw?usp=share_link

11.Vídeo

El vídeo se encuentra en la página de VideoPAC y se puede encontrar también al enlace siguiente:

https://drive.google.com/file/d/157f3w69sMRmOP1tTTq9qP8hQ5268B9eW/view?usp=share_link

Contribuciones	Firma
Investigación previa	Marco Rizzetto
Redacción de las respuestas	Marco Rizzetto
Desarrollo del código	Marco Rizzetto
Participación en el vídeo	Marco Rizzetto