

# PRÁCTICA 1

## Contexto

Con el objetivo de aplicar esta practica a un caso real, se ha decidido, aplicarlo a la realidad de la empresa Angulas Aguinaga, donde trabaja una de las integrantes del equipo. Esta empresa se dedica principalmente a la venta de diferentes derivados del pescado, a través de marcas como La Gula del Norte, Krissia, Aguinamar y Royal.

Esto así, el mercado principal de la compañía es el mercado nacional, y su canal principal, el Retail, representando la mayor parte de su facturación.

La crisis que está sufriendo el país está provocando que las cadenas sean cada vez más duras negociando con empresas marquistas como esta. Por lo tanto, ante esta situación, es muy importante tener conocimiento de lo que está ocurriendo en el mercado continuamente.

Por todo ello, surge la necesidad de hacer un pequeño análisis de mercado en una de las cadenas más relevantes para la compañía, Eroski.

## Título

Una vez explicado el contexto, y teniendo en cuenta que el objetivo del proyecto, el título establecido al proyecto es el siguiente: Análisis de mercado: Eroski Price scraping.

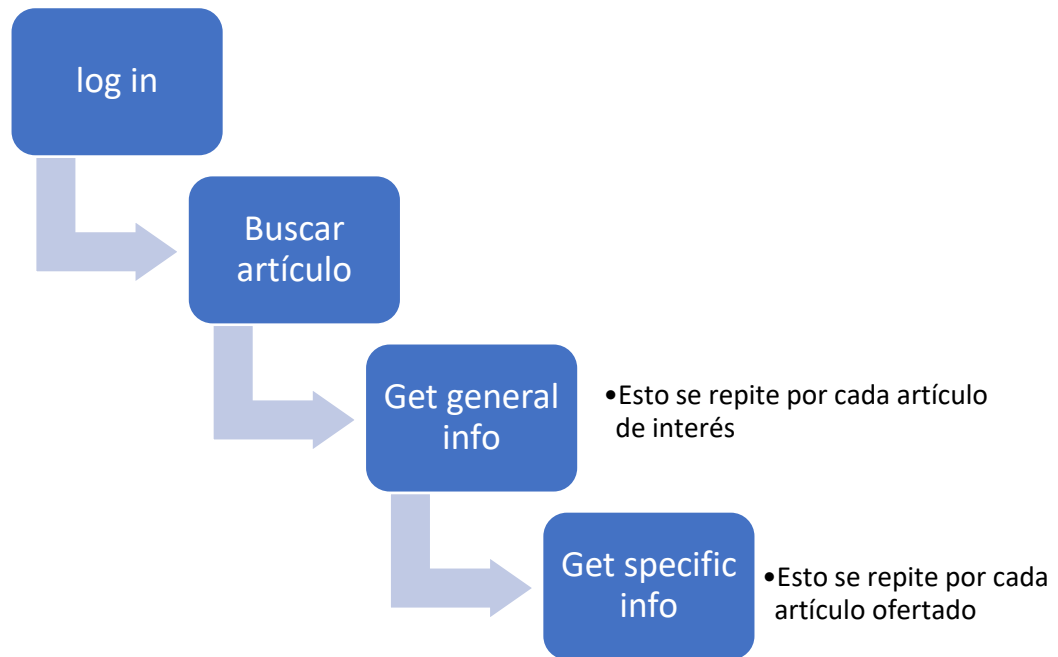
## Descripción del dataset

Se podría decir que el objetivo del proyecto es: obtener un dataset o una sábana de datos que permita analizar el mercado de los productos de Angulas Aguinaga: gula, palito de cangrejo, salmón y bacalao.

Por lo tanto, el proyecto trata de recoger los datos de la tienda online de Eroski. En concreto, se recoge la oferta de productos que Eroski ofrece al buscar los productos principales de la empresa, obteniendo el detalle de los productos y descripciones categóricas, como el precio.

El objetivo es que la parte comercial de la empresa pueda analizar y conocer lo que ofrece una de sus cadenas principales, en cuanto a sus categorías principales.

## Representación gráfica



El flujo que se ha seguido ha sido el siguiente:

- 1- Se ha iniciado sesión con un usuario creado por el equipo
- 2- Se ha buscado los artículos de interés, de forma que en la tienda online de la cadena Eroski se buscan los artículos de interés. (Los cuales están almacenados en un array)
- 3- Get General Info: Se recoge el listado de productos que aparece en cada una de las búsquedas, y la información más genérica respecto a los artículos: nombre y precio.
- 4- Get specific info: mediante un bucle se entra en cada uno de los artículos que se han buscado en el paso anterior, de forma que se recoge información específica de cada uno de ellos.

## Contenido

Teniendo en cuenta lo mencionado hasta ahora, el dataset que recogemos tiene los siguientes campos:

- Item: Descripción de los artículos que ofrece la cadena Ersoki
- Precio: El precio de venta al público de la cadena
- Precio\_kilo: Precio por kilo de producto. Este atributo va a facilitar las comparaciones entre los diferentes artículos.
- Código interno: El código interno, es el código que utiliza la cadena para hacer referencia al artículo. Este es un dato muy interesante, principalmente para los comerciales, ya que les permite comunicarse con la cadena con mayor facilidad. Al fin y al cabo, las negociaciones que hacen los comerciales con las cadenas, las hacen utilizando estos códigos internos.
- Categoría 1: Esta es la categoría general en la que ha incluido el producto la cadena
- Categoría 2: Esta sería el siguiente nivel de categoría, es decir, un nivel más específico. Estas variables se han recogido como datos informativos, y también porque posteriormente ayudarán a limpiar el dataset de artículos que no le interesan a la compañía.
- url: Link de acceso directo al artículo y su información específica.

En principio, este proyecto está planteado para que se ejecute cada semana, de forma que tanto la oferta, como los precios, se actualizarán con la realidad que ofrece Eroski cada semana.

## Propietario

Una vez estaba el proyecto orientado, se analizó si en el mercado había proyectos o análisis similares. Enseguida se pudo apreciar que son muchas las empresas que están interesadas en monitorizar tanto los precios de la competencia, como los propios. De

forma que vimos que había muchas aplicaciones y proveedores que lo que ofrecían era una monitorización de precios a través de web scraping.

Viendo las diferentes sábanas de datos que estas aplicaciones recogían, se decidió recoger los datos y los atributos mencionados anteriormente. En este caso, no hay ningún problema legal ya que únicamente se recogen datos comerciales, en ningún caso son de tipo personal. Por lo tanto, en todo momento hemos asegurado que el trabajo cumple los principios éticos y legales.

## Inspiración

En lo que respecta a la inspiración, debido a que es un proyecto aplicado a la realidad de una empresa, se han comprendido las necesidades que este proyecto iba a cubrir:

- Analizar el % de productos que Angulas Aguinaga cubre de toda la oferta de la categoría.
- Analizar el posicionamiento que tiene la empresa con sus diferentes productos en la compañía.

Cabe mencionar que a diferencia de las aplicaciones que hay en el mercado cuyo obtenido principal es monitorizar los precios de la competencia, en este caso, tiene un objetivo más analítico y comercial. Sin una perspectiva tan orientada a Insight de mercado.

## Código

A continuación, se van a explicar los principales puntos del código:

- Función Log in: Se ha iniciado sesión con un usuario creado por el grupo.

```
def log_in():  
    wait.until(EC.element_to_be_clickable((By.CLASS_NAME, "login-desktop"))).click()  
    user="maiderdorron@gmail.com"  
    psw="Flavia2023!"  
    user_input=wait.until(EC.element_to_be_clickable((By.ID, "email"))).send_keys(user)  
    psw_input=wait.until(EC.element_to_be_clickable((By.ID, "password"))).send_keys(psw)  
    wait.until(EC.element_to_be_clickable((By.XPATH, "//button[@title='Entrar']"))).click()  
    wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, "a.primary-button.primary-button-back"))).click()  
    time.sleep(1)
```

- Se han hecho varias búsquedas dinámicas, de forma que en un array se han establecido los artículos a analizar y estos se han buscado en la página web.
  - o Cabe destacar que en este caso se necesitaba hacer un scroll down hasta abajo de forma que cuanto más se bajaba, más artículos se cargaban; así, hasta que se cargaran todos los artículos.

```
def look_up_it(lista):  
    for i in lista:  
        search=driver.find_element(By.ID, value="searchTerm")  
        search.send_keys(i)  
        time.sleep(0.5)  
        driver.find_element(By.CLASS_NAME, "search-button").click()  
        last_height = driver.execute_script("return document.body.scrollHeight")  
  
        while True:  
            # Scroll down to bottom  
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight)")  
            # Wait to load page  
            time.sleep(SCROLL_PAUSE_TIME)  
            # Calculate new scroll height and compare with last scroll height  
            new_height = driver.execute_script("return document.body.scrollHeight")  
            if new_height == last_height:  
                break  
            last_height = new_height  
  
    get_items_prices()
```

- Se han recogido los artículos resultantes de las diferentes búsquedas.

```
def get_items_prices():  
    # precio  
    get_price = driver.find_elements(By.CLASS_NAME, "price-offer-now")  
    for t in get_price:  
        prices.append(t.text)  
  
    #items  
    get_products = driver.find_elements(By.CSS_SELECTOR, "h2.product-title.product-title-resp")  
    for t in get_products:  
        items.append(t.text)  
  
    #urls  
    get_urls=driver.find_elements(By.CSS_SELECTOR, "h2.product-title a")  
    for element in get_urls:  
        urls.append(element.get_attribute("href"))
```

- Se ha recogido información de cada uno de los artículos, entrando así, en cada link de cada artículo y cogiendo cierta información al respecto.
  - o Cabe destacar que no todos los artículos tenían los mismos campos, por lo que en este caso se han tenido que tratar las excepciones de forma que, aunque no esté dicho atributo el programa no dejara de funcionar.

```
def get_price_kilo(urls):  
    for element in urls:  
        driver.get(element)  
        categorias=driver.find_elements(By.CSS_SELECTOR, "div.col.col-lg-12.m_breadcrumb__path a")  
        cat_1.append(categorias[1].text)  
        cat_2.append(categorias[2].text)  
  
        link = None  
        try:  
            link = driver.find_element(By.CLASS_NAME, "price-product")  
            precio_kilo.append(link.text)  
        except NoSuchElementException:  
            precio_kilo.append('NA')
```

## Dataset

Tal y como se ha mencionado en anteriores apartados, la sábana de datos que se consigue es la oferta de productos de Eroski y su respectiva información.

- Dataset resultante: <https://doi.org/10.5281/zenodo.7859477>

## Video

El video con la respectiva presentación se encuentra en el siguiente link:

- [https://drive.google.com/file/d/1S7KUwmjPYIFrB8BNOeW6VzZQI57XpZdP/view?usp=share\\_link](https://drive.google.com/file/d/1S7KUwmjPYIFrB8BNOeW6VzZQI57XpZdP/view?usp=share_link)