IBEX 35: *Valores de sus componentes.*

Realizado por: *Leonor Pallares Ortega y Alberto Suárez Rodríguez.*



**CONTEXTO**.

Los datos has sido extraídos de la web [www.investing.com](https://es.investing.com/). Hemos elegido esta web porque es una plataforma de mercados financieros que proporciona datos en tiempo real, cotizaciones, gráficos, herramientas financieras, noticias de última hora y análisis de 250 mercados del mundo a través de sus 44 ediciones internacionales. Con más de 21 millones de usuarios mensuales y más de 180 millones de sesiones, [.investing.com](https://www.investing.com) es una de las tres mejores webs financieras del mundo según SimilarWeb (se especializa en análisis web, tráfico web y rendimiento) y Alexa. Los datos recogidos se centran en el mercado de índices, principalmente en el IBEX35 (índice de referencia en España) y sus componentes ( desde la fecha [[[[[(((((--- ----poner fecha-------------)))))]]] hasta la fecha actual cada vez que se ejecute el web scraping.

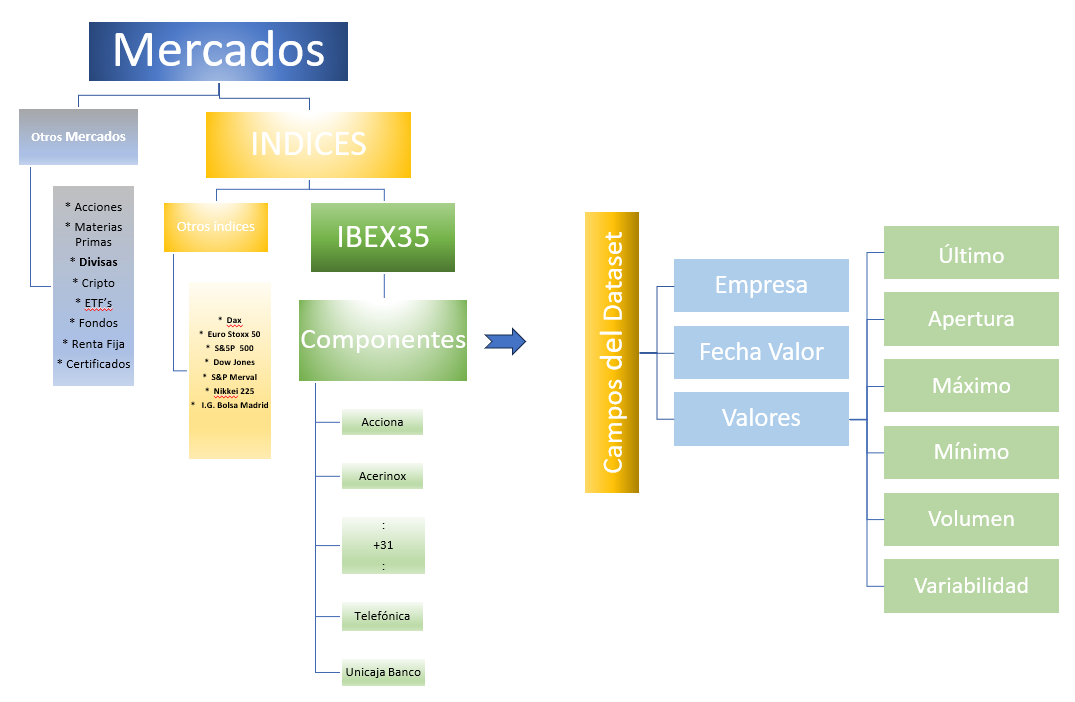
**DATASET.**

Definición del título: El titulo escogido para el dataset extraído mediante web scraping en esta primera práctica de la asignatura es: “Ibex35: Valores de sus componentes”.

Descripción: El conjunto de datos que se extraerá en esta práctica se compone de una serie de cotizaciones diarias del conjunto de empresas con más liquidez en las cuatro bolsas españolas (Madrid, Barcelona, Bilbao y Valencia).

Representación Gráfica:

Para representar gráficamente el dataset se ha construido un esquema en el que se identifica cada uno de los campos. A continuación, se adjunta el esquema comentado:



Contenido:

El dataset extraído contiene 8 variables. A continuación, se detallará el contenido de cada uno de ellos.

**Empresa**: Corresponde a los nombres de empresa que componen el IBEX 35

**Fecha**: Fecha en la que se recogen los datos de la empresa.

**Último**: El precio de cierre de las acciones de la empresa en la fecha especificada.

**Apertura**: El precio de apertura de las acciones de la empresa en la fecha especificada.

**Máximo**: El precio más alto al que llegaron las acciones de la empresa en la fecha especificada.

**Mínimo**: El precio más bajo al que llegaron las acciones de la empresa en la fecha especificada.

**Volumen**: La cantidad total de acciones de la empresa que se negociaron en la fecha especificada.

**Variabilidad**: La variación porcentual entre el precio de cierre y el precio de apertura de las acciones.

Por otro lado, comentar que los datos están comprendidos dentro del período que abarca entre la fecha dd/mm/yyyy hasta el último día de publicación de los valores existente en la web [.investing.com](https://www.investing.com) .

**PROPIETARIO**:

El propietario de la web es *FUSION MEDIA LIMITED*.

El propietario de la web es *FUSION MEDIA LIMITED,* investing.com es una plataforma de mercados financieros que proporciona datos en tiempo real, cotizaciones, gráficos, herramientas financieras, noticias de última hora y análisis de 250 mercados del mundo a través de sus 44 ediciones internacionales. Hemos analizado [*https://www.investing.com/robots.txt*](https://www.investing.com/robots.txt)y *https://es.investing.com/about-us/terms-and-conditions* y no presenta ninguna restricción al scraping de datos que hemos realizado, solo restringe el scraping de datos de usuario dados de alta en la web.

A continuación, se presenta un análisis de datos sobre IBEX35 encontrado en Internet para mostrar que tipo de análisis se pueden desarrollar con esta temática de datos.

* Se ha encontrado un ejemplo, en el expansión <https://www.expansion.com> Donde a partir de los valores obtenidos, igual que en nuestra practica, se realizan diferentes estudios que generan un análisis técnico, un análisis de riesgos etc..

Por ejemplo:

En la pestaña valores refleja los datos marcados en la sesión del día.

.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

En la pestaña Análisis tecnico:

En función de los valores obtenidos y con las comparaciones realizadas en el análisis, recomiendan que hacer con las acciones a corto plazo, que tendencia siguen las acciones de cada compañía perteneciente al IBEX 35

Tabla

Descripción generada automáticamente

Dicha información se puede consultar en la web siguiente:

<https://www.expansion.com/mercados/cotizaciones/indices/ibex35_I.IB.html>

.

El ejemplo propuesto se asemeja debido a que estudia las mismas variables a partir de también de los valores publicados en los mercados y puede servir como referencia.

**INSPIRACIÓN:**

El dataset obtenido puede usarse para distintos análisis. En primer lugar, para hacer regresiones e intentar predecir futuros valores lo cual es útil para inversores. También para estudiar los comportamientos del mercado y optimizar carteras de inversión. Otra opción es la evaluación de riesgos estudiando la volatilidad de cada empresa.

Esas serían algunas de las posibles opciones. Además, se podría enriquecer el dataset con otras variables como, por ejemplo, el sector económico de las empresas, o tamaño de las empresas, para realizar otros estudios de interés como analizar la evolución de un sector económico en concreto.

Se considera que se puede comparar con el ejemplo mencionado en el ejercicio anterior (propietario) por las razones comentadas, la similitud de las variables y la idea de analizar los valores con el fin de ayudar a posibles inversores, etc.

**LICENCIA:**

La licencia escogida es ***Released Under CC BY-NC-SA 4.0 License (Atribución-NoComercial-CompartirIgual 4.0 Internacional).***

El motivo por el cual usaremos esta licencia es para permitir que otros usuarios puedas usar el dataset, modificarlo, copiarlo y compartirlo, siempre que se otorgue atribución a los autores, y además tener control sobre su uso comercial. Finalmente, destacar que en ningún momento hemos tenido la intención de publicar el dataset de forma restringida ya que no contiene ningún tipo de información confidencial y, por lo tanto, puede servir de ayuda a cualquier futuro análisis que se pueda realizar

RECURSOS

1. RICHARD LAWSON. Web scraping with Python. Copyright © 2015 Packt Publishing
2. RYAN MITCHELL. Web scraping with Python, collecting data from the modern web. Copyright © 2015 Ryan Mitchell. All rights reserved.

**CÓDIGO:**

En este ejercicio se adjuntará el código de Python con el que se ha generado el dataset (se

adjunta mediante imágenes separadas para poder explicar más fácilmente el código, ya que este en una sola captura de pantalla no cabe:

En el código que se adjunta podemos distinguir cuatro partes claramente seccionadas: la importación de las librerías necesarias, la definición de la función que extrae la información de la página web escogida y finalmente la ejecución de la función y creación del archivo csv que contiene el dataset creado.

A continuación se detallarán los procedimientos seguidos en cada una de estas partes:

* Importación de las librerías necesarias

Captura de pantalla de computadora

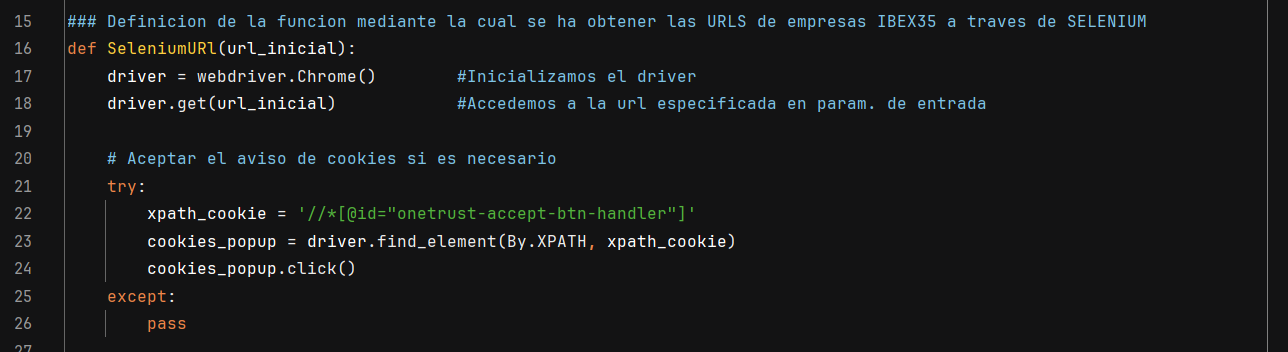
Descripción generada automáticamente

En esta parte del código se han importado las librerías “datetime”, “timedelta” y “relativedelta”

para trabajar con fechas, las librerías “requests” , “BeautifulSoup” y selenium para realizar el web scraping, la librería “pandas” especializada para trabajar con estructuras de datos, la librería “csv” para crear el archivo csv que contiene el dataset obtenido, y la librería “os” para trabajar con las rutas de carpetas del pc.

* Definición de la función “SeleniumURL”

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En esta segunda parte del código se ha definido la función “SeleniumURL” que tiene como parámetro de entrada “url\_inicial”, esta función nos ayudará a automatizar el navegador, Chrome en nuestro caso, como si fuera una “marioneta”, para ello es necesario disponer de driver que nos permite interactuar con el navegador y los sitios web.

Para ello primero inicializamos el driver, en nuestro caso con webdriver.chrome(), y procedemos a navegar por la web accediendo a la url\_inicial utilizando el método .get().

Se implementa la aceptación de avisos de cookies para poder seguir navegando configurando el xpath\_cookie y haciendo clic en el boton correspondiente utilizando la función .click().

* Procedemos a localizar los elementos de tag “a” dentro del DOM, utilizando el acceso por nombre de etiqueta a través de la función find\_elements\_by\_tag\_name(), localizador By.TAG\_NAME.
* Creamos una lista vacía llamada “enlaces\_filtrados”, la cual se irá informado con los enlaces que se van encontrando cuando cumplen las condiciones de que en sus atributos de href contenga “equities” y en el atributo title contenga “CFD”, consiguiendo tener las URL de cada una de las empresas que componen el IBEX35.
* Eliminamos posibles duplicados de la lista enlaces\_filtrados con la función set() y esta nueva lista se la asignaremos a la variable lista\_de\_urls .
* Creamos una lista vacía llamada “lista\_de\_urls2” que contendrá las urls que nos llevan a la web de valores históricos de cada una de las empresas de IBEX35, que servirá para obtener nuestros valores con la función “HistIbex35”, definida mas adelante.
* Una vez creada la nueva lista, procedemos a obtener los links requeridos a través de una iteración sobre enlaces\_filtrados
* Dentro de la iteración (bloque for)
  + abrimos URL apartir de la función get()
  + obtenemos los elementos que contengan el valor “//a[contains(@href, "historical") utilizando el acceso por XPath a través de la función find\_elements\_by\_xpath(), localizador By.XPATH.
  + Una vez localizados los elementos anteriores, procederemos a realizar una iteración sobre “elementos\_a” para obtener los links que contienen la información de los valores históricos de cada una de las empresas utilizando la función get\_attribute(“href”) , esta nueva información se almacenará en la variable de tipo lista “lista\_de\_urls2, creada anteriormente.
  + Una obtenidos los links procedemos eliminar duplicados con la función set() y ha ordenar los elementos con la función (sort) consiguiendo así el listado de las URLs requeridas para la obtención de los datos necesarios para el dataset.
  + Por último, procedemos a salir del navegador con la función .quit() para liberar los recursos que pueda estar utilizando.
* Definición de la función “HistIbex35”

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

En esta tercera parte del código se ha definido la función “HistIbex35” que nos ayudará a obtener la información mediante web scraping de la página web de “Datos históricos” de investing.com. Esta función tiene el parámetro de entrada “url\_complet” que corresponde a la URL de la página web comentada por cada una de las empresas que componen el IBEX35, actualmente el periodo de tiempo que se está obteniendo es fijo de 1 mes teniendo en cuenta el día de ejecución del web scraping. A continuación, se detallan los cálculos y pasos que se realizan dentro de la función:

* + En primer lugar se crea una lista vacía llamada “dd”, la cual se irá llenando con la información extraída mediante el web scraping. De esta forma se obtendrán los nombres de los campos del dataset como primer valor de la lista dd, y posteriormente se irá insertando la información de estos campos.
  + Después de obtener la URL completa que se pasa por parametro, se realiza el web scraping. Para ello se ejecuta la sentencia “requests.get(url\_complet)” para obtener la web que se está solicitando. Una vez hecho esto, se ejecuta la función “BeautifulSoup” para obtener el código fuente de dicha web, una vez aquí se ejecuta la sentencia “find(‘hi’)” con el objetivo de obtener el nombre de la empresa IBEX35 que estamos tratando en cada llamada y finalmente se ejecuta la sentencia “find\_all(‘table’)” para obtener las partes de este código fuente que estén dentro de ‘table’, es decir, la información que queremos extraer (recordar que la información en la página web se encontraba dentro de tablas de datos y por eso buscamos por la palabra ‘table’).

Una vez se ha obtenido la información resultante de la sentencia “find\_all(‘table’)” se borran los tres últimos elementos de la lista ya que son tres tablas con estadísticas globales que para esta práctica no aportan valor alguno.

* + A continuación, con el comando .read\_html haremos que la maquina interprete el html y así recuperaremos la información en un dataFrame (datos1) con un formato mas amigable.
  + Una vez construido el DataFrame “datos1” (en la cual se encuentra la información de los valores que necesitamos) se realiza un bucle “for” para recorrer cada uno de los elementos de “datos1” (es decir, cada uno de los días del mes en cuestión) y en cada una de estas iteraciones extraemos la información que deseamos mediante las asignaciones correspondientes tal y como se muestran en la captura de pantalla.

Comentar que para los campos numéricos se ha utilizado el método format, par obtener los decimales.

* + Finalmente, creamos una lista llamada “dd\_diaX” con la información recopilada de cada día referente a la iteración que estamos realizando y se juntan a la lista “dd” que hemos al inicio de la función.
* Proceso Principal

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En el proceso principal, Después de definir las diferentes funciones se realizan sus ejecuciones

Con la siguiente secuencia

* Se define la url\_inicial donde vamos a encontrar los componentes de IBEX35 dentro de investing.com.
* Ejecutamos la función SeleniumURL (definida al inicio) utilizando como parámetro la variable “url\_inicial”, esta función nos retornará una lista con todas las URL, donde podemos encontrar historia de los valores por fecha, de cada empresa que componen IBEX35 y se la asignaremos a la variable **link\_ibex35**
  + se crea una lista vacía llamada “dataset1”, la cual se irá llenando con la información extraída mediante el web scraping. También se crea una lista llamada “campos” con los nombres de los campos que tendrá el dataset y se adjunta como primer valor de la lista “dataset1” creada anteriormente. De esta forma se obtendrán los nombres de los campos del dataset como primer valor de la lista dataset1, y posteriormente se irán anexando la información de estos campos.
  + Mas tarde se realiza un bucle “while” de forma que se irá iterando mientras existan elementos en la lista link\_ibex35 (que recordamos contiene los links por empresa hacia la web de datos históricos)
  + Ya dentro del bucle, se realiza la llamada para ejecutar la función HistIbex35 utilizando como parámetro la variable “url\_complet” y asignaremos el dataFrame que retorna a dataset, este se irá añadiendo a la variable **dataset1** con el objetivo de que esta última contenga toda la información a trasladar a un fichero .csv
  + Antes de terminar, se crea la ruta donde se creará el archivo csv con el dataset obtenido mediante la concatenación de la ruta del directorio actual y el nombre que le queramos dar al archivo csv.
  + Finalmente, se crea el archivo csv en la ruta especificada mediante la función “csv.writer”.

**PUBLICACION del DATASET:**

**VIDEO**