

# PRÁCTICA 1

---

## EQUIPO

Los integrantes de esta práctica somos:

- Ignacio Such Ballester (isuch@uoc.edu)
- Andrés Fonts Santana (afontss@uoc.edu)

## WEB SCRAPING DE LA PÁGINA WEB MISHIGEEK.COM

### 1 CONTEXTO

El contexto que hemos escogido en esta práctica de Web Scraping la recopilación de valoraciones de juegos de mesa. Hemos escogido la página web [www.mishigeek.com](http://www.mishigeek.com) para realizar esta tarea.

[MishiGeek.com](http://MishiGeek.com) contiene decenas de reseñas. En cada una se publican diversos tipos de valoraciones como "Originalidad" y "Mecánicas". También se asigna una valoración total y una clasificación cualitativa como "Suspenso", "Recomendado" o "Juegaco".

### 2 TÍTULO

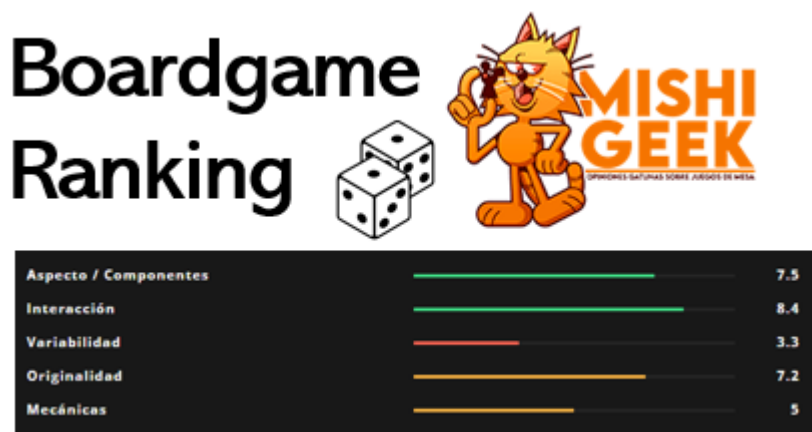
El título del dataset es **Boardgames Ranking**.

### 3 DESCRIPCIÓN DEL DATASET

El dataset Boardgames Ranking contiene 138 registros. Cada registro corresponde los datos extraídos de la reseña publicada en [MishiGeek](http://MishiGeek) de un juego de mesa.

Para cada reseña hemos identificado 21 atributos de interés. El formato del dataset obtenido es un fichero CSV.

### 4 REPRESENTACIÓN GRÁFICA



### 5 CONTENIDO

Cada registro de Boardgames Ranking cuenta con los siguientes campos:

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
nombre	string	Nombre del juego de mesa
n_jug	string	Número de jugadores
duracion	string	Duración de una partida, en minutos
fecha	string	Fecha de lanzamiento
dureza	string	Nivel de complejidad del juego
edad	string	Edad mínima recomendada o rango de edad
precio	float	Precio del juego en euros
genero	string	Género al que pertenece el juego
editorial	string	Editorial que publica el juego
diseño	string	Diseñadores del juego
val_asp	float	Valoración en cuanto Aspecto / Componentes
val_inter	float	Valoración en cuanto a nivel de Interacción
val_div	float	Valoración en cuanto a nivel de Diversión
val_var	float	Valoración en cuanto Variabilidad
val_rej	float	Valoración del nivel de Rejugabilidad
val_org	float	Valoración en cuanto a Originalidad
val_mec	float	Valoración de la Calidad de las Mecánicas
val_lec	float	Valoración de los lectores
n_votos	int	Número de votos con que cuenta val_lec
val_glob	float	Valoración global del juego
val_cual	string	Valoración Cualitativa

Los datos de nuestro dataset se han recogido de publicaciones en [MishiGeek](#) comprendidas entre las fechas 02/10/2018 y 11/04/2022.

Hemos estructurado la solución en 3 partes:

1. Script `get_reviews_list.py`. Analiza las páginas de reseñas y obtiene las direcciones url de cada reseña publicada en [MishiGeek](#). La lista se escribe en un fichero CSV.
2. Función `get_ratings`. Definida dentro de `get_ratings.py`, dada una reseña obtiene cada uno de los campos.
3. Script `gen_boardgame_dataset.py`. Itera sobre la lista de enlaces obtenida por `get_reviews_list.py` y obtiene de cada enlace los campos llamando a la función `get_rating`.

Hemos empleado la librería *Requests* para acceder a las urls así como *Beautiful Soup (bs4)* para navegar por el contenido de la url.

## 6 AGRADECIMIENTOS

Para poder realizar la tarea de *web scraping* hemos necesitado tomar tres medidas básicas:

- Solicitar permiso a [MishiGeek.com](https://mishigeek.com) para realizar la actividad.
- Consultar *robots.txt* de [MishiGeek](https://mishigeek.com) y modificar la cabecera de la petición HTTP para evitar el bloqueo de [MishiGeek](https://mishigeek.com),
- Prevenir la saturación del servidor debido a nuestras conexiones.
  - El tiempo medio entre peticiones sucesivas al servidor es de 3.5 segundos. Accedemos a cada *url* de forma secuencial.
  - El código se estructura en dos scripts de python. En primer lugar se realiza una búsqueda de enlaces con reseñas y en segundo lugar se analizan los enlaces con reseñas para obtener el *dataset*. Así prevenimos también realizar centenares de peticiones al servidor en un espacio corto de tiempo.

Los datos recogidos son propiedad de Javier Rodríguez Menéndez, autor del blog [MishiGeek](https://mishigeek.com). Agradecemos la colaboración de Javier al consentir explícitamente la labor de *web scrpaing* en su dirección web, con la condición de tratar sus datos de manera privada y no distribuirlos de forma pública.

Hemos identificado análisis similares al que presentamos en este proyecto:

- **Board Games - Kaggle.** Este *dataset* contiene datos sobre 20.000 juegos de mesa publicados en el portal [BoardGamesGeek](https://boardgamesgeek.com). El acceso al *dataset* es abierto, publicado en Kaggle: [board-games](https://kaggle.com/datasets/mishigeek/board-games). Para generar el conjunto de datos han filtrado aquellos juegos publicados que tengan como mínimo 30 valoraciones de usuarios.
- **Analysis of Boardgames (Dinesh Vatvani).** En este caso el autor del estudio ha modificado el *scraper* de BoardGamesGeek para explorar un *dataset* más completo. En su blog publica un análisis muy interesante sobre miles de datos: [BGG-Analysis-Part-1](https://dineshvattani.com/blog/bgg-analysis-part-1). Su proyecto de *web scraping* está disponible aquí: [scraper\\_and\\_data](https://dineshvattani.com/blog/scraper-and-data)
- **Board Game Data - Kaggle.** Dataset publicado en Kaggle que contiene 5000 juegos, los datos son extraídos de BoardGamesGeek. Enlace: [Board Game Data](https://kaggle.com/datasets/mishigeek/board-game-data)
- **bgg-games-data - Kaggle.** Dataset de nuevo extraído de BoardGamesGeek con datos de más de 270.000 juegos de mesa. Este data set recoge datos sobre todos los juegos publicados en BoardGamesGeek hasta el 15/7/2020.

## 7 INSPIRACIÓN

Los análisis anteriores que hemos identificado se basan en el portal BoardGamesGeek (BGG). Es sin duda el mayor portal de juegos de mesa online. Nuestro dataset se basa en reseñas publicadas por MishiGeek y aporta una visión diferente a los datasets publicados. Encontramos interesante capturar qué opiniones hay fuera BGG y aportar este valor diferencial a los análisis anteriores.

Con nuestro dataset podríamos realizar análisis como:

- Identificar los mejores juegos publicados cada año según mishigeek.

- Saber qué editoriales de juegos de mesa tienen más éxito segun mishigeek.
- Elaborar comparaciones entre reseñas publicadas entre diferentes portales como por ejemplo estudiar sesgos en las valoraciones.
- Comparar precios según diferentes atributos.
- etc.

## 8 LICENCIA


El propietario de MishiGeek no ha permitido publicar los datos de forma abierta, por lo que no podemos licenciarlos. Contemplamos por eso dos supuestos:

- En el hipotético caso de que sí accediera a publicarlos escogeríamos la licencia *CC BY-NC-ND*. Esta es la más restrictiva, permite compartir los datos bajo la condición de no modificarlos ni usarlos con fines comerciales y siempre acreditando al autor/a del *dataset*.
- Generaremos un *dataset* ficticio que compartiremos bajo la licencia *CC BY*, que permite compartir, modificar y distribuir los datos con fines comerciales y no comerciales, siempre que se acredite al autor/a.

## 9 CÓDIGO

El código utilizado puede consultarse dentro del directorio `/py` del repositorio `boardgame-reviews`.

## 10 DATASET

Hemos publicado una simulación del dataset obtenido en Zenodo  [DOI](#).

## 11 VÍDEO

El enlace para el video se ha indicado en el documento de entrega de la práctica.

## MEJORAS IDENTIFICADAS PARA FUTURAS VERSIONES

Tras realizar esta primera solución que obtiene el dataset de las reseñas publicadas en Mishigeek identificamos una serie de mejoras que podrían realizarse en futuras iteraciones sobre nuestra solución:

- Implementar una asignación menos *ad-hoc* de las categorías del dataset. En la función `info2list` definimos un diccionario para asignar las categorías obtenidas por `get_ratings` al campo adecuado del dataset. Por ejemplo, el precio del juego puede aparecer indicado como "PVP Recomendado", "PvP recomendado", o "PVP". Una mejora consiste en utilizar funciones regex que sean más robustas y permitan asignar correctamente categorías a campos sin necesidad de poblar un diccionario que podría llegar a ser muy difícil de mantener.
- Gestionar los enlaces de reseñas no analizados. Hemos detectado que ejecutando nuestra solución en ocasiones hay enlaces de reseñas que no son analizados, probablemente por un error momentáneo de conexión. Las urls son operativas, creemos que una mejora sería capturar las urls no analizadas y realizar una "segunda vuelta" de análisis exclusivamente sobre las urls no analizadas en la primera iteración. Podríamos hacer un `append` de ambos dataset para obtener un dataset final más completo.

## CONTRIBUCIONES

<b>Contribuciones</b>	<b>Firma</b>
Investigación previa	IS, AF
Redacción de las respuestas	IS, AF
Desarrollo del código	IS, AF