Juegotea

Víctor del Pino Castilla

Leila Ruiz Casanova

Javier Sesé

# Índice

[Índice 2](#_Toc535425967)

[Introducción 3](#_Toc535425968)

[Obtención de datos 4](#_Toc535425969)

[Herramientas de extracción 5](#_Toc535425970)

[Bibliografía 6](#_Toc535425971)

# Introducción

En el presente documento se describen las acciones realizadas para la realización de un sistema de recomendación de juegos de mesa.

Los sistemas de recomendación Online permiten que los usuarios obtengan propuestas de productos en base a sus gustos y acciones pasadas, los mejores ejemplos los encontramos en las plataformas de comercio electrónico (Amazon y eBay) y más concretamente en el ámbito del ocio los encontramos que nos recomiendan música (Spotify), series y películas (Netflix, Amazon Video).

Para el nicho de los juegos de mesa actualmente no hay un sistema de recomendación especializado, consultando páginas web genéricas de comercio electrónico y especializadas encontramos diferentes grados de recomendaciones de productos, pero no un sistema concreto de recomendación de juegos.

Nuestro objetivo principal es desarrollar un sistema que dado uno o varios juegos como entrada ofrezca una salida con diversos juegos relacionados. Para establecer esta relación entre los juegos y ofrecer las mejores recomendaciones disponemos de la información en la plataforma BGG [*Board* *Game Geek*][[1]](#footnote-1), esta es una red social centrada en los juegos de mesa que entre otra funcionalidad dispone de un sistema de calificación que usaremos como base para establecer las relaciones entre los juegos.

En las siguientes páginas se desarrolla la creación del sistema de recomendación

Bienvenidos al juego desconectado.

# Obtención de datos

Para obtener los datos hemos utilizado la red social BGG en la que los usuarios tienen el interés común sobre los juegos de mesa, dispone de una ficha sobre la inmensa mayoría de los juegos de mesa que existen en la que se pueden encontrar además de las características de cada uno, imágenes, videos de partidas, estadísticas, expansiones, un foro para cada uno de los juegos, expansiones, se pueden subir ficheros con modificaciones, reglas propias. Como secciones importantes además dispone de un foro, una sección de noticias, un apartado de compra/venta para juegos, blogs de los usuarios, capacidades para que los usuarios generen sus colecciones de juegos y la parte más interesante para este estudio permite que los usuarios valoren cada uno de los elementos disponibles.

Con toda esta información disponible hay que seleccionar los datos relevantes que nos permitan efectuar recomendaciones de calidad.

Para ello en una primera aproximación planteamos generar un grafo bipartito en el que existan nodos representando a los usuarios y a los juegos en el que las aristas estén restringidas a unir un nodo usuario a un nodo juego siempre que ese usuario haya valorado a ese juego con una nota superior a un umbral, de forma que el grafo abarque la totalidad de los juegos y los usuarios disponibles, y se obtenga la recomendación para un juego, en base a los juegos relacionados con los usuarios que están relacionados con el juego dado como entrada.

Debido a la dificultad para obtener la totalidad de los datos y tratarlos por recomendaciones de los profesores se abandona esta idea y se plantea la posibilidad de generar dos grafos no bipartitos uno de juegos y otro de usuarios.

Debido a que nuestro objetivo es generar recomendaciones para juegos de mesa nos centramos en el grafo de juegos abandonando el grafo de usuarios.

Para obtener los datos que necesitamos para este análisis disponemos de dos API V1[[2]](#footnote-2) y V2[[3]](#footnote-3) [*Application Program Interface*] publicadas por la propia BGG y la información que podemos obtener a realizar operaciones usando la técnica Web Scraping[[4]](#footnote-4).

La estrategia que plantemos para obtener los datos consiste en, obtener los usuarios que han valorado un juego conocido, de esta forma podemos confiar que un gran abanico de tipos de jugadores, y obtener todas las valoraciones realizadas por estos usuarios y en base a estos datos generar la información de los nodos y aristas.

Para obtener los datos hemos utilizado el juego Azul[[5]](#footnote-5) y establecido la nota umbral para relacionar los juegos en 9

## Herramientas de extracción

Se han generado una serie de scripts en lenguaje Python en su versión 3.6[[6]](#footnote-6) con las librerías Request[[7]](#footnote-7) y Network X[[8]](#footnote-8) que nos permiten obtener la información necesitada, están pensados para ser ejecutados de forma secuencial obteniendo la información y refinándola según se realizan los siguientes pasos:



1. Obtenemos los nombres de todos los usuarios que han valorado un juego.
2. Obtenemos todos los juegos valorados por cada uno de los usuarios y las notas dadas a cada juego.
3. Limpiamos la información de cada juego obtenida en el paso 2 y generamos una lista de los juegos únicos que tenemos disponibles.
4. Limpiamos la información de cada relación obtenida en el paso 2 y generamos una lista con las aristas que relacionan los juegos y sus pesos asociados

### Obtención de usuarios

En un primer intento probamos a realizar un scraping de la web para obtener los datos, pero comprobamos que debido a la forma de construir la web y las respuestas que nos devolvía incrementaban demasiado el nivel de dificultad, por lo que abandonamos esta línea de trabajo para obtener los datos mediante la API.

Para obtener los usuarios de los que obtendremos los datos hemos generado el script *01\_usuarios.py* para esto obtenemos de la API V2 mediante peticiones web con la librería Request los usuarios que han valorado el juego Azul, esta API nos devuelve un XML[[9]](#footnote-9) con la información del juego y lotes con los usuarios y la nota que estos le han asignado de 100 en 100 por lo que tenemos que recorrer todas las páginas para obtener estos usuarios que guardamos en un fichero para procesarlo después

Para ejecutarlo con la instrucción: python3 01\_usuarios.py

### Obtención de Juegos y Notas

Este paso se asocia con el script *02\_datosJuegos.py*, en el que recorremos todos los usuarios recogidos en el fichero generado en el paso 1, para cada uno de estos datos realizamos una petición con la librería Request, a la API V1 de la que obtendremos una respuesta con el código 202, que nos indica que el servidor ha aceptado la petición y está generando los datos, cada segundo se vuelve a realizar la petición, para obtener los datos generados en el servidor en formato XML[[10]](#footnote-10), recorremos los resultados para obtener los juegos que se han valorado y obtener la nota que se ha establecido.

Según recorremos los resultados, vamos almacenando en un fichero todos los juegos a los que se hace referencia con sus datos asociados, y almacenamos en una lista los juegos valorados por cada uno de los usuarios, si superan la nota umbral de 9. Una vez hemos recogido todos los datos del usuario generamos la relación de todos los pares de juegos que le gustan y se almacena en un fichero.

Para ejecutarlo con la instrucción: python3 02\_datosJuegos.py

### Refinado de los juegos

En este paso recogido en el script *03\_constructorNodos.py*, recorre el fichero de nodos generado en el paso 2 y elimina los duplicados y genera un fichero preparado para importar en Gephi[[11]](#footnote-11) con todos los juegos disponibles

Para ejecutarlo con la instrucción: python3 03\_constructorNodos.py

### Refinado de las aristas

En este paso el script *04\_constructorAristas.py*, lee todos los pares de juegos generados en el paso 2 y calcula el peso que tiene la arista en función de las repeticiones que se encuentren de ese par de juegos y genera un fichero preparado para importar en Gephi con todas las aristas para relacionar los juegos disponibles estableciendo su peso.

Para ejecutarlo con la instrucción: python3 04\_constructorAristas.py

# Red generada

Para generar la red disponemos de un fichero con los nodos y otro con las aristas, disponemos de 37.275 nodos con 1,425,054 aristas, estas aristas relacionan 8,414 juegos que son sobre los que vamos a realizar la recomendación.

En el grafo que obtenemos los nodos representan cada uno de los juegos y las aristas representan que dos usuarios han valorado el juego con una nota superior a el umbral establecido, que consideraremos indicativo como que ambos juegos le gustan a ese usuario, y estableciendo el peso de la arista, como la suma de las veces que este par de juegos le ha gustado a un usuario, de esta forma tenemos un indicador de cuanto gusta el juego de un extremo, a los que les gusta el juego del otro extremo.

# Algoritmo de recomendación

Para realizar la recomendación

# Ejecución del sistema de recomendación

Magia

# Bibliografía

CARO MARTÍNEZ, M. (2017). *SISTEMAS DE RECOMENDACIÓN BASADOS EN TÉCNICAS DE PREDICCIÓN DE ENLACES PARA JUECES EN LÍNEA.*

1. <https://www.boardgamegeek.com/> [↑](#footnote-ref-1)
2. <https://boardgamegeek.com/wiki/page/BGG_XML_API> [↑](#footnote-ref-2)
3. <https://boardgamegeek.com/wiki/page/BGG_XML_API2> [↑](#footnote-ref-3)
4. <https://es.wikipedia.org/wiki/Web_scraping> [↑](#footnote-ref-4)
5. <https://boardgamegeek.com/boardgame/230802/azul/ratings?rated=1> [↑](#footnote-ref-5)
6. <https://www.python.org/downloads/release/python-360/> [↑](#footnote-ref-6)
7. <http://docs.python-requests.org/en/master/> [↑](#footnote-ref-7)
8. <https://networkx.github.io/> [↑](#footnote-ref-8)
9. <https://api.geekdo.com/xmlapi2/thing?id=230802&comments=1&page=1&pagesize=100> [↑](#footnote-ref-9)
10. <https://www.boardgamegeek.com/xmlapi/collection/victordpc?rated=1> [↑](#footnote-ref-10)
11. <https://gephi.org/> [↑](#footnote-ref-11)