Máster Executive en Big Data y Business Analytics 5ª Edición (2016-2017)

ASIGNATURA:

PERSISTENCIA POLÍGLOTA Y BASES DE DATOS NO RELACIONALES

EJERCICIO FINAL:

PROPUESTA DE SISTEMA DE ALMACENIMIENTO POLÍGLOTA

Autores:
Ainhoa Calvo Ejerique
Juan Antonio García Cuevas

Fecha: 30 de septiembre de 2016

Juan García Cuevas, Ainhoa Calvo Ejerique

En aras de acabar con la piratería, un conjunto de productores independientes ha creado un portal web para visualizar películas y series online. El portal, llamado Veeme, funciona mediante suscripción mensual.

En los últimos meses, el número de usuarios de Veeme ha aumentado exponencialmente gracias a su novedad, su publicidad y a su eficaz sistema de recomendación. Debido a este gran crecimiento, se debe renovar con la máxima celeridad su sistema de almacenamiento.

Nos piden, como conocedores de las tecnologías NoSQL, que hagamos una propuesta de sistema de almacenamiento para el nuevo sistema. El nuevo sistema de almacenamiento deberá satisfacer las siguientes necesidades:

Juan García Cuevas, Ainhoa Calvo Ejerique

0.- INTRODUCCIÓN

El auge de Internet en las últimas dos décadas, con el incremento de servicios on-line, de usuarios conectados y del volumen y variedad de información, ha puesto de manifiesto que el tratamiento centralizado de los datos no es eficiente en estos nuevos escenarios.

La necesidad de garantizar en cualquier momento el servicio a un gran número de usuarios potencialmente desconocido, y de ofrecer tiempos de respuesta aceptables independientemente del número de usuarios conectados y de su ubicación, impulsó en la primera década del siglo XXI el desarrollo y auge de una nueva generación de bases de datos no relacionales, altamente distribuidas y escalables, reconocidas con el nombre genérico NoSql.

Las propiedades que proporcionan los sistema de almacenamiento distribuido son la Consistencia, Disponibilidad y Tolerancia a particiones. Pero en el año 2000, Eric Brewer determinó que un sistema distribuido sólo podía garantizar simultáneamente 2 de estas propiedades (Teorema CAP). Así pues, los diferentes diseños de los distintos sistemas gestores bases de datos los capacita para poder realizar unas tareas de forma más eficiente que otras.

Por otro lado, para proporcionar la gran variedad de servicios que existen hoy en día, las aplicaciones deben gestionar una gran diversidad y cantidad de datos. Por ello, es posible que un único tipo de sistema de almacenamiento no sea capaz de cubrir de forma eficiente todas las necesidades de una aplicación.

Surge entonces la idea de Persistencia Políglota, entendida como la utilización en una misma aplicación o empresa de varios tipos de sistemas de bases de datos, aplicando el más adecuado a cada tipo de tarea.

A lo largo de los últimos años han ido apareciendo gran cantidad y variedad de sistemas distribuidos de almacenamiento, algunos de los cuales han desaparecido o caído ya en desuso. Elegir un sistema de almacenamiento distribuido puede ser muy difícil, ya que ninguno ha conseguido aún la fama que sí han logrado las bases de datos relacionales.

Dado que nos encontramos ante un caso práctico propuesto en la asignatura de “Bases de Datos No Convencionales”, y después de haber analizado y determinado que pueden existir distintas opciones eficientes para cada uno de los sistemas propuestos en el ejercicio, hemos optado por elegir, para cada uno de los apartados del mismo, un sistema de almacenamiento distinto de entre los estudiados en el curso, basándonos, por supuesto, en el análisis del problema y en casos de éxito.

Juan García Cuevas, Ainhoa Calvo Ejerique

1. Gestión de suscripciones

Debido al alto número de usuarios concurrentes, en los pasados meses el portal ha tenido muchos problemas de disponibilidad que han provocado que, en determinadas ocasiones, los usuarios no hayan podido visualizar ninguna película ni serie aun teniendo sus suscripciones activas. Para solucionar este tema, se ha decidido utilizar un sistema de almacenamiento de alta disponibilidad que gestione la información de suscripciones.

La información a gestionar es, dado un usuario, saber si tiene la suscripción activa. La información de suscripciones no se actualiza muy a menudo, por lo tanto, no es necesario que el sistema satisfaga consistencia fuerte, pero si deberá garantizar la máxima disponibilidad.

Juan García Cuevas, Ainhoa Calvo Ejerique

1.- GESTIÓN DE SUSCRIPCIONES

Para el sistema de gestión de suscripciones se necesita una base de datos que cumpla con los siguientes requisitos:

I) Alta disponibilidad, ya que la falta de la misma ha impedido en algunas ocasiones que suscriptores activos pudieran visualizar películas o series.

II) No se necesita consistencia fuerte, ya que la información sobre suscripciones no se actualiza con demasiada frecuencia.

III) No se necesita escalado horizontal, puesto que la información sobre suscripciones no se actualiza con demasiada frecuencia.

A continuación, se muestra una tabla en la que se relacionan las bases de datos estudiadas y el cumplimiento de las mismas con los requisitos del Sistema de Suscripciones:

	Cassandra (AP)	MongoDB (CP)*	Riak (AP)	Neo4j (CA)	B.D.Rel. (CA)
Alta disponibilidad	Cumple	No cumple en configuración por defecto	Cumple	Cumple	Cumple (replicación)
No precisa consistencia fuerte	Cumple	Cumple en configuración por defecto (excede)	Cumple	Cumple	Cumple (excede)
No precisa escalado horizontal	Cumple (excede)	Cumple (excede)	Cumple (excede)	Cumple (excede)	Cumple

** MongoDB proporciona consistencia fuerte en detrimento de alta disponibilidad. No obstante, podría configurarse para proporcionar alta disponibilidad a costa de la consciencia fuerte.*

Desde el punto de vista del teorema CAP, sería recomendable utilizar bases de datos de los tipos AP (Cassandra o Riak) o CA (Neo4j, B.D. relacionales), pero no CP, por lo que descartamos MongoDB.

Descartamos también las bases de datos relacionales, ya que no suelen estar tan altamente optimizadas para las operaciones recuperación y agregado como las bases de datos NoSql.

Por otro lado, los modelos de agregación, como MongoDB, Cassandra o Riak, son una buena solución cuando no existen interrelaciones complejas entre los datos y éstos están sujetos a pocos cambios, como el caso que nos ocupa. Por ello, aunque Neo4j cumple con los requisitos del Teorema CAP, lo descartamos como la opción más conveniente, por no seguir un modelo de agregación sino de grafo.

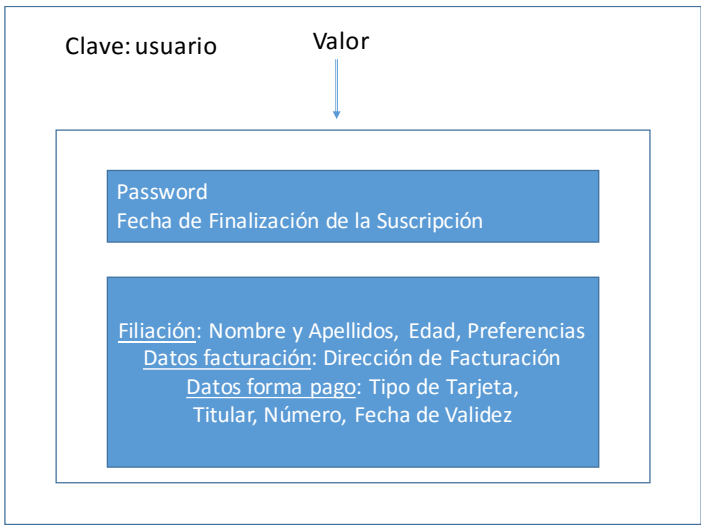
Cassandra y Riak son sistemas que poseen ciertas características comunes y que satisfacen los requisitos AP del Teorema CAP. Podría utilizarse cualquiera de los dos sistemas, aunque nos decidimos por Riak, al haberse utilizado en varios casos de éxito* como el de Wildworks (http://www.wildworks.com/), una plataforma de juegos on-line

que lanzó en 2010, junto a National Geographic Society, Animal Jam (http://www.animaljam.com/) pasando a convertirse en una de las plataformas de juegos on-line de mayor crecimiento en Estados Unidos. Wildworks utiliza Riak para la gestión de las cuentas de usuarios en distintos dispositivos (PC, móvil...)

* *http://arstechnica.com/information-technology/2016/04/power-tools-sorting-through-the-crowded-specialized-database-toolbox/*

Por tanto, la base de datos propuesta para este caso es Riak.

Estructura del agregado



Juan García Cuevas, Ainhoa Calvo Ejerique

2. Catálogo de películas y series

El catálogo de películas y series disponibles debe ser accesible por todos los usuarios por igual, aunque sólo los usuarios con una suscripción activa podrán visualizarlas completamente. El visionado de las novedades en el catálogo supone una fuente de ingresos muy importante, por lo que mantener la integridad de datos en el catálogo es de vital importancia.

El catálogo está formado por el listado de las películas y series disponibles. En algunos casos, puede ser que estas sólo estén disponibles en determinados países, debido a sus derechos de autor y restricciones comerciales.

El catálogo también incluye los actores, y la relación de las películas/series donde han actuado. Para cada película/serie se deben indicar las películas/series similares, las categorías a la que pertenece (acción, suspense, comedia, etc.), el año de su estreno y su sinopsis. Aun teniendo disponibles la gran mayoría de películas y series de los últimos años, se tiene la certeza de que el sistema no tendrá la necesidad de escalar horizontalmente porque el número de nuevas películas, series, actores y directores esperado es bastante reducido.

Juan García Cuevas, Ainhoa Calvo Ejerique

2.- CATÁLOGO DE PELÍCULAS Y SERIES

Para el sistema de gestión del catálogo de películas y series se necesita una base de datos que cumpla con los siguientes requisitos:

I) Alta disponibilidad, puesto que el visionado de las novedades es una fuente de ingresos muy importante. Además, el catálogo debe estar disponible para todos los usuarios, no sólo para los que tengan una suscripción activa.

II) Integridad de la información, consistencia fuerte, puesto que el visionado de las novedades es una fuente de ingresos muy importante.

Otras circunstancias importantes a tener en cuenta son las siguientes:

I) No se precisa escalado horizontal, porque no se espera que el número de nuevas películas, series, actores y directores no crezca mucho.

II) Diferenciación geográfica, para gestionar la disponibilidad por países según derechos de autor y restricciones comerciales.

III) Almacenamiento y recuperación de la información en forma de metadatos y relaciones.

A continuación, se muestra una tabla en la que se relacionan las bases de datos estudiadas y el cumplimiento de las mismas con los requisitos del catálogo de películas y series:

	Cassandra (AP)	MongoDB (CP)	Riak (AP)	Neo4j (CA)	B.D.Rel. (CA)
Alta disponibilidad	Cumple	No cumple en configuración por defecto	Cumple	Cumple	Cumple (replicación)
Integridad, consistencia fuerte	No cumple en configuración por defecto	Cumple en configuración por defecto	No cumple	No cumple en configuración por defecto, pero se puede configurar*	Cumple
No precisa escalado horizontal	Cumple (excede)	Cumple (excede)	Cumple (excede)	Cumple (excede)	Cumple
Diferenciación geográfica	Cumple	Cumple	Puede cumplir	Cumple perfectamente	Cumple
Metadatos y relaciones	Puede cumplir	Puede cumplir	Puede cumplir	Cumple perfectamente (nodos, aristas y propiedades)	Cumple

* *En la documentación contenida en la página web de Neo4j (http://neo4j.com/docs/operations-manual/current/introduction/#ha-architecture) se indica que, por defecto, no cumple con la propiedad de consistencia fuerte, pero se puede configurar: "Updates to slaves are eventually consistent by nature but can be configured to be pushed optimistically from master during commit.". "Availability can often be addressed with various strategies for recovery or mirroring. However, Neo4j's clustering architecture is an automated solution for ensuring Neo4j is consistently available to your application and end-users".*

Juan García Cuevas, Ainhoa Calvo Ejerique

Desde el punto de vista del teorema CAP, sería recomendable utilizar bases de datos de los tipos AC (Neo4j, B.D. relacionales) en lugar de CP (MongoDB) o AP (Cassandra, Riak), por lo que descartamos Cassandra, Riak y MongoDB.

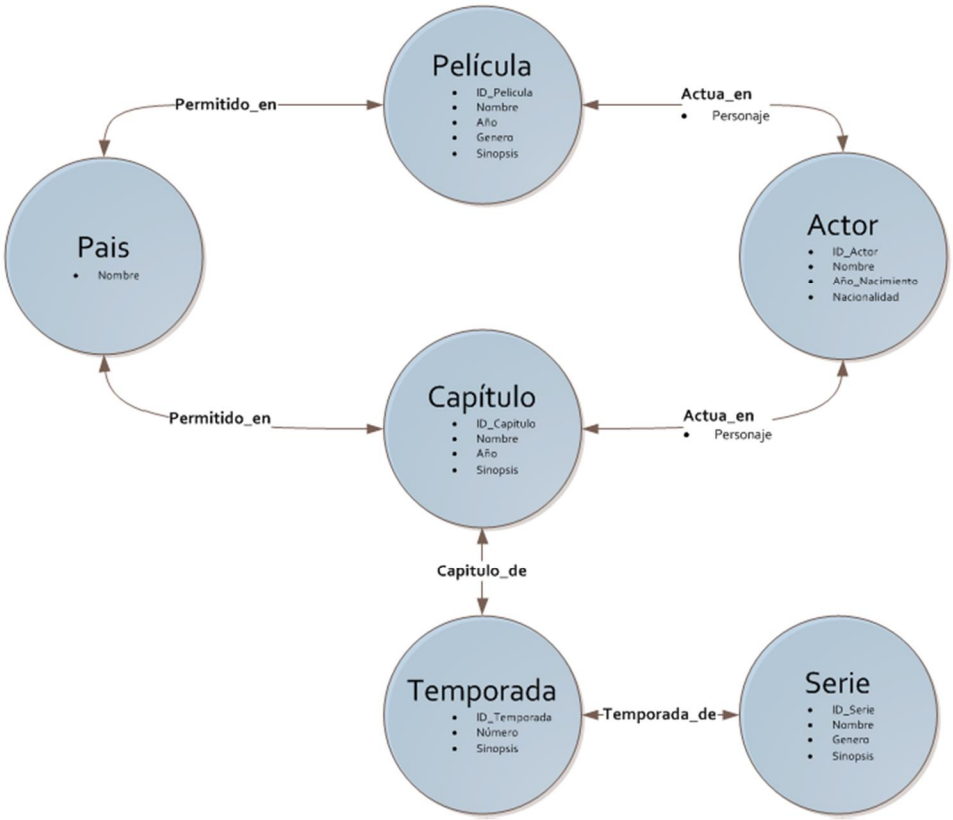
También descartamos las bases de datos relacionales, ya que el tiempo de respuesta en operaciones de recuperación con relaciones (joins) no está tan optimizado como en las bases de datos en grafo.

Seleccionamos, por tanto, un sistema de gestión de bases de datos en grafo como Neo4j, pues cumple con todos los requisitos (disponibilidad, consistencia, relaciones, …).

Como casos de éxito puede señalarse la utilización de Neo4j por Infojobs y Linkedin's. Son bolsas de empleo on-line que facilitan el encuentro entre oferta y demanda de empleo mediante el almacenamiento de los currículum de los candidatos y de las ofertas de las empresas en un formato estandarizado. Esta información se encuentra disponible en: https://neo4j.com/case-studies/infojobs/, https://neo4j.com/case-studies/linkedin-china/

Por tanto, la base de datos propuesta para este caso es Neo4j.

Estructura del grafo



Juan García Cuevas, Ainhoa Calvo Ejerique

3. Análisis de usuarios

Es de vital importancia analizar el comportamiento implícito de los usuarios. Para ello el sistema debe almacenar datos sobre la navegación de los usuarios con el fin de extraer información sobre sus preferencias. A tal fin se almacenará información sobre cada conexión que el usuario realiza al portal, indicando la hora de la conexión, la página vista y el tiempo que ha permanecido en ella.

Adicionalmente, a partir de la clave del usuario, se podrá acceder a información sobre todas sus conexiones. Veeme está teniendo mucho éxito y, a pesar de contar ya con millones de usuarios activos, duplica el número de usuarios activos cada dos meses. Por tanto, para almacenar las conexiones de los usuarios es de vital importancia escoger un sistema de almacenamiento que sea capaz de escalar horizontalmente acorde al crecimiento esperado. Por otro lado, al haber decenas/centenares de conexiones por segundo, el sistema elegido deberá garantizar una alta disponibilidad.

Juan García Cuevas, Ainhoa Calvo Ejerique

3.- ANÁLISIS DE USUARIOS

Para el sistema de análisis de usuarios se necesitaría una base de datos que cumpla con los siguientes requisitos:

I) Alta disponibilidad, ya que puede haber decenas o centenares de conexiones por segundo.

II) Escalabilidad horizontal, puesto que se espera duplicar el número de usuarios activos cada dos meses.

III) Alto rendimiento en inserción de datos, puesto que se debe almacenar la información de los centenares de conexiones por segundo que realizan los usuarios.

	Cassandra (AP)	MongoDB (CP)	Riak (AP)	Neo4j (CA)	B.D.Rel. (CA)
Alta disponibilidad	Cumple	No cumple en configuración por defecto	Cumple	Cumple	Cumple (replicación)
Escalabilidad horizontal	Cumple	Cumple	Cumple	Cumple	No Cumple
Alto rendimiento en inserción de datos	Cumple	Cumple	Cumple	Cumple	Por debajo de las B.D NoSql

Desde el punto de vista del teorema CAP, sería recomendable utilizar bases de datos del tipo AP (Cassandra, Riak).

Descartamos las bases de datos relacionales porque no cumple con la necesidad de escalabilidad horizontal, su rendimiento en la inserción de datos en grandes volúmenes de información está muy por debajo de las bases de datos NoSql, y para poder mejorar la disponibilidad requieren equipos de coste muy elevado.

Neo4j cumple con los requisitos, pero está diseñada para gestionar grafos (relaciones), por lo que no es la mejor candidata para este caso.

Descartamos también MongoDB, ya que no proporciona alta disponibilidad en su configuración por defecto.

Recomendamos en este caso el uso de Cassandra en vez de Riak porque, además de alta disponibilidad, Cassandra está especialmente diseñada para escenarios en los que hay más escrituras que lecturas, escalando (escalado horizontal) muy bien en esos casos (ideal para análisis de logs).

Un caso de uso exitoso de Cassandra es Hulu, un conocido servicio americano de streaming de películas y series. Una especie de videoclub virtual mediante el cual el usuario tiene derecho a ver cierto contenido con una tarifa plana que se consigue pagando una suscripción (http://www.hulu.com).

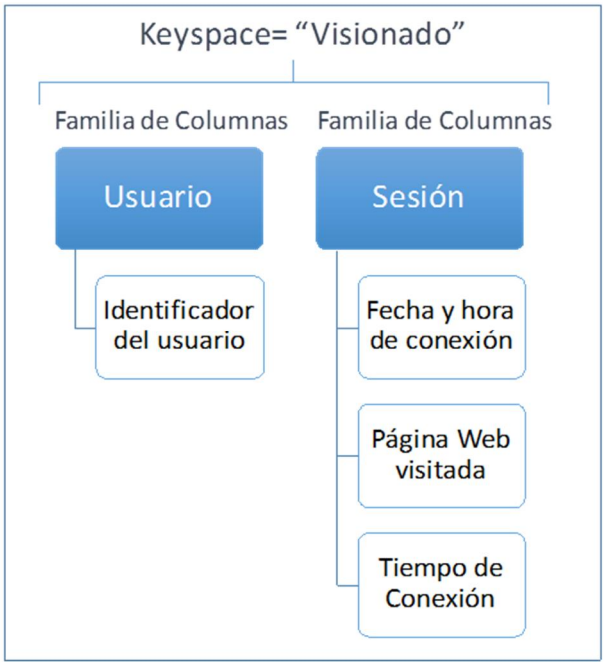
Un segundo caso de uso es el Ooyala que utiliza Cassandra para proporcionar a empresas de medios como Bloomberg, ESPN, Telegraph Media Group y Yahoo! Japan,

"*actionable analytics*", lo que les permite analizar con gran detalle la forma en que su contenido de vídeo está siendo consumid0, para optimizar su entrega y maximizar los ingresos (http://www.ooyala.com)*

**http://www.computing.co.uk/ctg/analysis/2291474/a-big-data-case-study-ooyala-s-realtime-video-analytics*

Por tanto, la base de datos propuesta para este caso es Cassandra.

Estructura del Agregado



Juan García Cuevas, Ainhoa Calvo Ejerique

4. Sistema de recomendación

Se debe actualizar el sistema para realizar las recomendaciones a los usuarios. Los usuarios pueden valorar del 1 al 10 cada una de las películas o capítulos de las series que han visualizado y, de forma independiente, pueden puntuar también a sus actores.

Por otro lado, también interesa ofrecer recomendaciones a todos los usuarios de acuerdo con sus preferencias de forma rápida. El sistema que calcula las recomendaciones para cada usuario queda fuera del ámbito de este trabajo.

Supongamos que ya está implementado, que obtiene los datos de las fuentes necesarias y recomienda películas/series para cada usuario de forma periódica. Aquí, sólo nos interesa definir la base de datos que almacene para cada usuario la información de sus votaciones, de las últimas películas y series que ha visto y de la lista de recomendaciones proporcionadas por el sistema (que será actualizada por el recomendador de forma periódica).

Juan García Cuevas, Ainhoa Calvo Ejerique

4.- SISTEMA DE RECOMENDACIÓN

Según el enunciado, la base de datos que debe diseñarse no se utilizará para realizar el "cálculo de recomendaciones" de películas y series a cada usuario, sino para almacenar y recuperar el resultado de dicho "cálculo", que será actualizado periódicamente por un proceso externo al sistema.

Se deberá poder obtener la información de cada usuario a través de una clave que lo identifique. Mediante dicha clave se accederá toda la información previamente almacenada de dicho usuario (en el proceso de cálculo de recomendaciones). La información será recuperada en un único bloque, registro o documento, y estará estructurada en sub-bloques o sub-documentos correspondientes a las votaciones del usuario, a las últimas películas y series visionadas y a las recomendaciones.

En las bases de datos relacionales el diseño lógico sería utilizar tablas distintas para los diferentes tipos de datos, y joins para establecer las relaciones entre ellos. Este modelo implica necesariamente realizar varias consultas a base de datos por cada petición de información de un usuario, con la consiguiente penalización en rendimiento. Por ello descartaremos el uso de bases de datos relacionales.

Las bases de datos en grafo están especialmente diseñadas para relacionar datos y recorrer los nodos según sus relaciones aplicando la teoría de grafos, pero en este caso la información no necesita ser almacenada en una estructura de grafo o de red. Por ello descartaremos también las bases de datos en grafo como Neo4j (*cuyo uso sí está indicado en sistemas de "calculo de recomendaciones"*).

Por otro lado, las bases de datos documentales y columnares están mejor diseñadas para facilitar la actualización de los datos que las de tipo clave-valor, por lo que también descartaremos Riak.

Cassandra es un buen candidato para resolver el problema, pero MongoDB se ajusta de manera más natural a la estructura de la información descrita más arriba, así que descartamos también Cassandra.

Por tanto, la base de datos propuesta para este caso es MongoDB.

Juan García Cuevas, Ainhoa Calvo Ejerique

Estructura del Agregado

La estructura de agregado propuesta en este caso es la siguiente:

- id_usuario
- votaciones
  - serie (sólo en series)
  - temporada (sólo en series)
  - capitulo (sólo en series)
  - titulo
  - genero
  - imagen
  - _id
  - votacion
- ultimos_visionados
  - serie (sólo en series)
  - temporada (sólo en series)
  - capitulo (sólo en series)
  - titulo
  - genero
  - imagen
  - _id
  - fecha_visionado
- recomendaciones
  - serie (sólo en series)
  - temporada (sólo en series)
  - capitulo (sólo en series)
  - titulo
  - genero
  - imagen
  - _id
  - sinopsis

Como puede observarse, incluye un identificador de usuario y tres listas: lista de votaciones, lista de últimas películas y series visionadas y lista de recomendaciones.

1. Los elementos de todas las listas incluyen los siguientes datos: título, género, imagen e identificador de la película o serie (para poder enlazar con ella). En caso de tratarse de una serie se incluirán además estos otros datos: título de la serie, temporada y capítulo.
2. Los elementos de la lista de votaciones incluyen también el dato de votación.
3. Los elementos de la lista de últimos visionados incluyen también el dato de fecha de visionado.
4. Los elementos de la lista de recomendaciones incluyen también el dato de sinopsis.

A continuación se muestra un ejemplo en formato JSON:


```
{
  "id_usuario" : "12345",

  "votaciones" : [
    {
      "titulo": "Florence Foster Jenkins",
          "genero": "Biografía, Drama, Comedia",
          "imagen": "/images/232226.png",
      "_id": "232226",
      "votacion": "5"
    },
    {
      "serie": "The Walking Dead ",
          "temporada": "6",
      "capitulo": " 16",
      "titulo": "Last Day on Earth",
          "genero": "Drama, Terror",
          "imagen": "/images/847234-6-16.png",
      "_id": "847234-6-16",
      "votacion": "7"
    }
  ],

  "ultimos_visionados" : [
    {
          "titulo": "Cuerpo de élite",
      "genero": "Comedia",
          "imagen": "/images/230659.png",
          "fecha_visionado": "23/09/2016",
      "_id": "230659"
    },
    {
      "serie": "Vikingos",
          "temporada": "3",
      "capitulo": "10",
      "titulo": "The Dead",
      "genero": "Aventura, Drama, Histórico",
          "imagen": "/images/749365-3-10.png",
          "fecha_visionado": "15/08/2016",
      "_id": "749365-3-10"
    }
  ],

  "recomendaciones" : [
    {
      "titulo": " Los siete magníficos",
      "genero": " Western, Acción, Aventura",
          "imagen": "/images/207874.png",
      "sinopsis": "En la América posterior a la Guerra Civil, en pleno Salvaje Oeste, un pequeño pueblo mexicano
llamado Rose Creek es asediado constantemente por el... ",
      "_id": "207874"
    },
    {
      "serie": "Juego de Tronos",
          "temporada": "6",
      "capitulo": "15",
      "titulo": "The Winds of Winter",
      "genero": " Drama , Fantasía ",
          "imagen": "/images/340530-6-15.png",
      "sinopsis": "Inmerso en este mundo cuyas estaciones duran décadas y en el que retazos de una magia inmemo-
rial y olvidada surgen en los rincones más sombríos y maravillosos, Ned Stark tendrá que lidiar con la traición, la
lealtad, la compasión y la sed de venganza.",
      "_id": "340530-6-15"
    }
  ]
```