

Examen Final de GRAU-IA

(15 de enero de 2020)

Duración: 2 horas 30 minutos

1. (5 puntos) Debido al creciente número de plataformas de streaming de contenido audiovisual, escoger dónde gastar el dinero para acceder a contenido de nuestro agrado es cada vez más complicado. Para ayudar en este propósito, queremos crear una herramienta llamada QuéVeo que permitirá a los usuarios obtener una recomendación sobre las plataformas que deberían contratar. Para el desarrollo de nuestra herramienta queremos utilizar un sistema basado en el conocimiento que, en base al poder adquisitivo del usuario, sus preferencias personales y con quién suele visionar productos audiovisuales, nos permita recomendarle varios packs de contenidos seleccionados de diferentes plataformas.

En QuéVeo tenemos un registro con todas las plataformas de contenidos que existen. Cada plataforma tiene asociados uno o varios packs de contenido (para reflejar el hecho de que algunos contenidos pueden ser ofrecidos como compra o alquiler). Un pack tiene un coste mensual y un catálogo, que puede ser distinto para cada país. Cada catálogo incluye un conjunto de obras. Cada obra tiene un formato determinado: película, serie o documental. De cada obra queremos guardar su título, el nombre y apellidos de sus directores, sus géneros narrativos, su edad mínima recomendada y sus nacionalidades (lista de países de las entidades productoras). Los géneros que puede tener una película o una serie son acción, ciencia ficción, comedia, drama, fantasía, melodrama, musical, romántica, thriller y/o terror. Los géneros que puede tener un documental son sociopolítico, artístico, divulgativo, de naturaleza y/o falso documental. De cada película, además, queremos guardar su duración. Las series se componen de temporadas, y cada temporada se compone de episodios. De cada episodio también queremos almacenar su duración.

Para QuéVeo también son relevantes las bases de datos que existen en Internet con información de las obras audiovisuales que se producen en todo el mundo. Cada base de datos contiene registros para un conjunto de obras, y para cada una de ellas se almacena la puntuación de los usuarios, la puntuación de la crítica y si la obra cumple con ciertas categorías especiales: si la obra es de culto, si es un clásico, si es un *blockbuster*, si es de autor y/o si es de serie B. Cada obra de los catálogos de las plataformas estará relacionada con la información de las obras almacenadas en los registros de las bases de datos.

Para utilizar QuéVeo necesitamos del usuario su nombre, su primer apellido, su país de residencia, su número de teléfono, su correo electrónico, su año de nacimiento (para poder estimar su edad) y cuánto dinero está dispuesto a gastarse en plataformas cada mes. Además queremos guardar sus géneros favoritos, el formato de obra que prefiere ver y las categorías especiales en las que tiene especial interés.

Sabemos que cuando un usuario contrata una plataforma de streaming es muy probable que no sea el único consumidor del contenido. Por lo tanto del usuario se pide que nos diga si vive solo, en pareja, en familia o en un piso compartido. Si el usuario vive en familia también nos interesa guardar el número de niños menores de 10 años, y si vive en un piso compartido queremos guardar el número de convivientes y la diferencia de edades entre la edad más alta y la edad más baja. En cualquier caso queremos almacenar la media de horas totales de contenido audiovisual que se consume en su casa cada semana.

Debido a la complejidad del dominio de QuéVeo buscamos simplificar, identificando un conjunto de características que nos permitan categorizar a cada usuario del sistema:

- la **diversidad de los espectadores**, que puede ser *baja*, *media* o *alta*. La diversidad será *baja* si el número de convivientes en casa del usuario (incluyendo el propio usuario) es uno o dos, *media* si el número de convivientes es más de dos y el usuario vive en familia con como mucho un niño o en un piso compartido con una diferencia de edades menor de 10, y *alta* en cualquier otro caso.
- el **nivel de cinefilia**, que dependerá de cuántas preferencias ha configurado el usuario sumando sus géneros preferidos y las categorías especiales seleccionadas. El nivel puede ser *nulo* (sin preferencias), *poco-cinefilo* (como máximo 5 preferencias), *algo-cinefilo* (entre 5 y 10 preferencias) o *cinefilo* (más de 10 preferencias).
- el **volumen de consumo**, que dependerá del número de horas de consumo audiovisual proporcionado por el usuario y que puede ser *escaso* (menos de 10 horas), *normal* (entre 10 y 40) o *alto* (más de 40).
- el **gasto potencial**, que puede ser *bajo*, *normal* o *elevado*. El gasto será *bajo* si el dinero que el usuario está dispuesto a gastarse por mes es menor de 10 euros, *medio* si está dispuesto a gastarse menos de 50 euros y *elevado* en el resto de casos.

Estas categorías nos permiten identificar el tipo de recomendaciones que podemos hacer al usuario. Para clasificar los diferentes tipos de recomendaciones definimos las siguientes características categóricas:

- la **variedad**: con esta característica queremos identificar si es una prioridad que haya variedad de contenidos. Si la diversidad de los espectadores es *alta* o el usuario es *cinefilo* la prioridad de recomendar packs de contenido con mucha variedad será *prioridad-muy-alta*. Si la diversidad es *baja* y además el nivel de cinefilia es *nulo* o *poco-cinefilo* asignaremos *sin-prioridad*. En el resto de usuarios nos guiaremos por el volumen de consumo: si es *alto*, asignaremos *prioridad-alta* y en cualquier otro caso *prioridad-media*.

- el **número de packs de contenido** a recomendar: recomendaremos sólo *un-pack* de contenido si el volumen de consumo es *escaso*, el gasto potencial es *bajo* o la diversidad es *baja*. Alternativamente, recomendaremos *dos-a-cuatro* packs si ni el volumen de consumo es *escaso*, ni el gasto potencial es *bajo* ni la diversidad es *baja*, excepto si el gasto potencial es *elevado* y el volumen de consumo es *alto* en cuyo caso el número de packs a recomendar será *cinco-o-más*.
- si hay que **considerar categorías especiales**: con esta característica identificaremos si las categorías especiales que el usuario especificó como preferencias serán relevantes a la hora de construir la recomendación. Su valor será *sí* si el nivel de cinefilia es *algo-cinéfilo* o *cinéfilo* y, o bien la diversidad de los espectadores es *baja*, o bien el volumen de consumo es *alto*, y será *no* para cualquier otro usuario.

El objetivo es construir un sistema capaz de generar una recomendación de packs de contenidos a cada usuario de QuéVeo.

- (2,5 puntos) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción y todas las relaciones necesarias. Identifica qué conceptos forman parte de los datos de entrada del problema y qué conceptos forman parte de la solución. Justifica las decisiones de diseño que has realizado. En el caso de las subclases de una clase, justifica con un texto breve la necesidad de crear dichas subclases. Para cada concepto de la ontología lista todos sus atributos (para cada atributo se ha de indicar su nombre y su tipo). [Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente.]
 - (2,5 puntos) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, justificando si el tipo de solución que se pide requiere de refinamiento o no. Lista las variables que forman parte del Problema Abstracto (nombre y valores posibles). Lista las variables que forman parte de la Solución Abstracta (nombre y valores posibles). Da al menos 4 ejemplos de reglas (suficientemente variadas, utilizando diferentes conceptos) para cada una de las fases de esta metodología, usando los conceptos de la ontología desarrollada en el apartado anterior. Para las reglas podéis usar la notación de alto nivel que usamos en los ejercicios de problemas (*si ... entonces ...*) o una notación más próxima a la del lenguaje CLIPS (siempre que queden claro los elementos del antecedente de la regla y del consecuente).
2. (5 puntos) Las redes de distribución de contenidos (CDNs en inglés) son redes de ordenadores que se colocan entre los usuarios y los proveedores de contenido (por ejemplo páginas web) para hacer caching de los contenidos. Con esto se consigue aligerar la carga de los proveedores y reducir la latencia de las peticiones de los usuarios.

Queremos utilizar planificación automática para gestionar el funcionamiento de una red CDN, y usaremos un modelo muy simplificado del funcionamiento real de este tipo de sistemas. Como escenario, consideraremos que tenemos un conjunto de usuarios haciendo peticiones a un conjunto de proveedores de contenido y que tenemos en medio una red de distribución formada por un conjunto de ordenadores. Cada nodo (ordenador) de la red de distribución puede almacenar un número determinado (que puede variar entre nodos) de contenidos en su caché.

Cada proveedor de contenido almacena o produce un subconjunto de contenidos que puede ofrecer, y consideraremos que al inicio de la ejecución del proceso de planificación tenemos como entrada un conjunto de peticiones realizadas por los usuarios. Cada petición especifica el contenido que debe ser entregado al usuario y puede ser resuelta únicamente por un nodo de la red de distribución. El mismo contenido puede ser solicitado por más de un usuario.

En este contexto simplificado, la comunicación entre los diferentes componentes del sistema se regiría por el siguiente protocolo:

1. El cliente realiza una petición por un contenido.
2. Un nodo de la red CDN procesa esta petición y entrega el contenido al cliente.

Es decir, que no hay comunicación directa entre el cliente y los proveedores. Sin embargo, otro aspecto importante es decidir cuál de los nodos será el que intercepte la petición y se encargue de entregar el contenido al cliente. Si un nodo de la red de distribución tiene en su caché el contenido especificado por una petición no resuelta, este mismo nodo puede entregar los contenidos directamente al usuario. En caso de que no haya ningún nodo con ese contenido, cualquier nodo con alguna posición libre en su caché realizará una petición al proveedor de ese contenido. El proveedor debe responder a esta petición, tras lo cual inmediatamente el nodo entrega el contenido al cliente y lo guarda en su caché al mismo tiempo. Si no hay ningún nodo que pueda satisfacer la petición de un usuario y las cachés de todos los nodos están llenas, cualquier nodo puede liberar una posición cualquiera de su caché. En resumen:

1. En primer lugar, se escoge un nodo cualquiera (y sólo uno) que tenga el contenido en cuestión en su caché, y este nodo entrega el contenido al usuario.
2. En caso de que 1. no se pueda cumplir, se escoge un nodo cualquiera (y sólo uno) que tenga alguna posición libre en su caché.
3. En caso de que ni 1. ni 2. se cumplan, se escoge un nodo cualquiera (y sólo uno) y éste liberará una posición cualquiera de su caché para atender a la petición.

Tanto en 2. como en 3. el nodo de la red CDN debe obtener el contenido. Para ello deberá lanzar una petición a algún proveedor que ofrezca ese contenido, recibir el contenido del proveedor, guardarlo en una posición de su caché y entregar el contenido al usuario.

Queremos implementar un sistema que permita ejecutar este proceso, tomando como punto de partida las peticiones creadas, los nodos y proveedores existentes, qué contenidos tienen los proveedores y cuántas posiciones de caché tiene cada nodo. El resultado es un plan que lista el orden en el que se realizan las acciones necesarias para atender a todas las peticiones cumpliendo con los protocolos especificados en los párrafos anteriores.

Nos interesa que el sistema registre, de manera explícita, los hits de caché, las peticiones de contenido de los nodos a los proveedores, las respuestas de los proveedores a los nodos, y las entregas de contenido a los usuarios.

- a) (3,5 puntos) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a cantidades arbitrarias de clientes, nodos, proveedores, contenidos y peticiones de contenido.
- b) (1,5 puntos) Para probar el sistema nos dan el siguiente ejemplo de configuración:
 - 3 clientes: C1, C2, C3.
 - 2 nodos CDN con 3 posiciones caché disponibles y vacías: N1, N2.
 - 8 contenidos: O1, O2, O3, O4, O5, O6, O7 y O8.
 - 4 proveedores: V1 (que provee O1 y O2), V2 (O3, O4, O5), V3 (O6, O7) y V4 (O8).
 - Las siguientes peticiones iniciales:

<i>petición</i>	<i>cliente</i>	<i>contenido</i>
P1	C1	O1
P2	C2	O1
P3	C1	O2
P4	C3	O3
P5	C2	O4
P6	C2	O5
P7	C2	O6
P8	C3	O7
P9	C1	O8
P10	C1	O1

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

Las notas se publicarán el día **27 de enero**.