

Recuperatorios2019

Prolog

Se desea programar un sistema de recomendación de películas para Netflix. Cuando un usuario inicia la aplicación, el sistema debe recomendarle una lista de películas, según lo que éste haya visto previamente. Para eso, consulta en toda la base de conocimientos la listas de películas de los usuarios de Netflix y organiza la recomendación en función de los que hayan visto las mismas películas que el usuario en cuestión, basado en un nivel de similaridad.

Por ejemplo, si tenemos 3 usuarios en Netflix, y en la base de conocimientos tenemos que las películas que vio el usuario José fueron 'Misterio a Bordo', 'The Perfection', 'Otro día para matar', 'Barreras', 'Yo soy Sam', 'Inspection'; las que vio el usuario María fueron 'Quizás para siempre', 'A pesar de todo', 'La casa del Lago', 'Barreras'; y las que vio el usuario Pedro son 'Barreras', 'Otro día para matar', 'Yo soy Sam', 'A pesar de todo'; si Pedro inicia su sesión, el sistema debe recomendar películas en función de su lista de películas vistas y de la lista de otro usuario con cuya similitud de películas vistas sea mayor a 2. En esta caso, se debería obtener la/s película/s que Pedro no vio: 'A pesar de todo'.

Se solicita, en base a este ejemplo, crear la recomendación de la/s película/s para el usuario Pedro.

Solución con nota 96

% si Pedro inicia su sesión, el sistema
% debe recomendar películas en función de
% su lista de películas vistas y de la lista
% de otro usuario con cuya similitud de
% películas vistas sea mayor a 2. En esta caso,
% se debería obtener la/s película/s que Pedro no vio:
% 'A pesar de todo'.

%Se solicita, en base a este ejemplo, crear la
%recomendación de la/s película/s para el usuario Pedro.

%base de conocimientos
vio('Jose', ['Misterio a Bordo', 'The Perfection', 'Otro día para matar', 'Barreras', 'Yo soy Sam', 'Inspection']).

vio('Maria', ['Quizás para siempre', 'A pesar de todo', 'La casa del Lago', 'Barreras']).

vio('Pedro', ['Barreras', 'Otro día para matar', 'Yo soy Sam']).

vio('Juan', ['Barreras', 'Otro día para matar', 'duro de matar']).

%miembro: determina si el elemento X está en la lista
miembro(X, [X|_]):-!.

```
miembro(X,[_|Resto]):-  
    miembro(X,Resto).
```

%funcion similitud: recibe las peliculas que vio el usuario y lista de cada usuario
%devuelve la cantidad de similitudes, donde similitud se incrementa en 1 por cada peli vista
%por el usuario que coincide con alguna peli que vio otro

```
similitud([],_,0):-!.  
similitud([X|PelisVistas],PelisAComp,Sim):-  
    not(miembro(X,PelisAComp)),  
    similitud(PelisVistas,PelisAComp,Sim),  
    !.  
similitud([X|PelisVistas],PelisAComp,Sim):-  
    miembro(X,PelisAComp),  
    similitud(PelisVistas,PelisAComp,Sim2),  
    Sim is Sim2+1,  
    !.
```

%eliminar: elimina elemento X de una lista
eliminar(_,[],[]):-!.
eliminar(X,[X|L],Res):-
 eliminar(X,L,Res),
 !.
eliminar(X,[Y|L],[Y|Res]):-
 eliminar(X,L,Res).

%elimina_repetidos: devuelve la diferencia L1-L2: peliculas que vio otro-peliculas que vio el usuario
eliminaRepetidos(L1,[],L1):-!.
eliminaRepetidos([],_,[]):-!.
eliminaRepetidos(L1,[Y|L2],Res):-
 eliminar(Y,L1,Res2),
 eliminaRepetidos(Res2,L2,Res).

%funcion recomendar recibe el usuario. Manda a buscar similitud con cada usuario que no sea el mismo.
%si similitud es mayor que 2, elimina los repetidos entre ambas listas
%la lista sin repetidos es la solucion
%devuelve las recomendaciones en comparacion con cada uno de los usuarios

```
recomendar(Usuario,Recom):-  
    vio(Usuario,PelisVistas),  
    vio(X,PelisAComp),  
    X\=Usuario,
```

```
similitud(PelisVistas,PelisAComp,Sim),
Sim >= 2,
eliminaRepetidos(PelisAComp,PelisVistas,Recom).
```

Devolucion

Similitud, eliminar repetidos y eliminar podrían haber sido parte del mismo predicado.

Funcional

Se desea crear un programa que permita consultar el costo del boletos para realizar un viaje entre dos localidades. El programa recibe tres (3) parámetros que son: origen, destino y hora de salida preferida (indicada en hh y mm) y debe devolver el costo del viaje y los posibles horarios en que se puede realizar.

Se dispone de la lista de localidades, el costo entre la localidad y la inmediata siguiente y el listado de horarios en que sale un viaje desde cada una.

Localidad	Costo del boleto	Horarios
Santa Fe	\$5	10:00, 12:00, 16:00
Recreo	\$2	10:20, 12:20, 16:20
Candiotti	\$1	10:30, 12:30, 16:30
Iriondo	\$2	10:35, 12:35, 16:35
Manucho	\$2	10:45, 12:45, 16:45
Cululu	\$12	11:00, 13:00, 17:00

Ej:

(viaje '("Recreo" "Manucho" (12 30)) --> Recreo - Manucho: \$5 - 16:20

(viaje '("Santa Fe" "Cululu" (9 0)) --> Santa Fe - Cululu: \$12 - 10:00, 12:00, 16:00

(viaje '("Santa Fe" "Iriondo" (11 0)) --> Santa Fe - Iriondo: \$8 - 12:00, 16:00

AYUDA:

```
(define tarifa '(
  ("Santa Fe" 5 ((10 . 0) (12 . 0) (16 . 0)))
  ("Recreo" 2 ((10 . 20) (12 . 20) (16 . 20)))
  ("Candiotti" 1 ((10 . 30) (12 . 30) (16 . 30)))
  ("Iriondo" 2 ((10 . 35) (12 . 35) (16 . 35)))
  ("Manucho" 2 ((10 . 45) (12 . 45) (16 . 45)))
  ("Cululu" 0 ((11 . 0) (13 . 0) (17 . 0)))
)
```

Objetos

Obtener el modelo conceptual de un sistema de información muy simplificado de la gestión de Residencias Estudiantiles de la Universidad.

La universidad brinda residencias a sus alumnos. Dichas residencias son casas que alquila la Universidad.

Tienen una dirección y una capacidad (cantidad de personas que pueden vivir en la misma).

El procedimiento es el siguiente, la Universidad firma Contratos de Alquiler con un Locador, se establece la vigencia de mismo (desde-hasta) y el monto. El locador pueden ser tanto una Inmobiliaria como un particular.

Por ejemplo: Suipacha 3023(Santa Fe) – capacidad 10 pers. – Desde: 01/01/2013 Hasta 31/12/2014.

LOCADOR: Inmobiliaria ACUARIO PROPIEDADES

Los Residentes son Alumnos que se alojan en una residencia, un determinado tiempo (desde..hasta), mientras están activos en una residencia la fecha hasta es nula. Hay residentes activos y no activos en base a esa fecha.

Requerimiento: Codificar en lenguaje JAVA especificando todas las clases necesarias, con el siguiente detalle: accesibilidad de atributos y métodos, constructores, tipo de atributo, valor de devolución de los métodos y parámetros. Desarrollar los algoritmos de los métodos que crea necesario para codificar la siguiente funcionalidad.

Universidad.mostrarContratosVigentes() : void

El método debe mostrar como salida por consola la siguiente información:

Datos del Contrato: 01/01/2013 al 31/12/2014 – Monto: \$ 72.000.-

Locador Inmobiliaria Acuario Propiedades. / Particular Juan Perez

Datos de la Residencia: Bv. Pellegrini 550.

Capacidad 8 personas - activos 6 personas

