

Pregunta 1:

/*

Plataforma: TDA que representa el catálogo de películas, consiste en una lista, donde cada elemento es una lista con las películas de cierto genero

E: Elemento a agregar en una lista

ColaG: Cola de la lista que contiene las películas de cierto genero

LF: Lista que contiene la lista a retornar

CabezaP: Cabeza del TDA plataforma

ColaP: Cola del TDA plataforma

Min: Entero que representa la duración mínima de una película

Max: Entero que representa la duración máxima de una película

CabezaF: Cabeza de la lista final (lista a retornar)

ColaF: Cola de la lista final (lista a retornar)

ListaGenero: Lista que representa la sublista de un género de películas

L: Lista cualquiera

Predicados

agregarAlFinal(L, E, LF)

crearPelicula(Titulo, Genero, Duracion, Restriccion, Pelicula)

buscarGenero(Plataforma, Genero, LF)

tituloUnico([ListaGenero, Titulo])

existeGenero(Plataforma, Genero)

agregarGenero(Plataforma, Genero, NuevaPlataforma)

peliculaAlGenero(Plataforma, Pelicula, Genero, Plataforma2)

eliminarPeliculaGenero(ListaGenero, Titulo, Genero, NuevaListaGenero)

filtrarTiempo(ListaGenero, Min, Max, NuevaListaGenero)

agregarPelicula(Plataforma, Titulo, Genero, Duracion, Restriccion, Plataforma2)

Nombre: Christofer Rodríguez

Rut:20.239.786-7

Profesor: Víctor Flores

eliminarPelicula(Plataforma, Genero, Titulo, Plataforma2)

filtrarPorDuracion(Plataformaa, Min, Max, LF)

Metas

Principales: agregarPelicula, eliminarPelicula y filtrarPorDuracion.

*/

%Clausulas

%Reglas

agregarAlFinal([], E, [E]).

agregarAlFinal([Cabeza|Cola], E, [Cabeza|NuevaCola]) :- agregarAlFinal(Cola, E, NuevaCola).

crearPelicula(Titulo, Genero, Duracion, Restriccion, Pelicula):-

Pelicula = [Titulo, Genero, Duracion, Restriccion].

buscarGenero([[Genero|ColaG]|_], Genero, [Genero|ColaG]):- !.

buscarGenero(_|ColaP, Genero, LF):- buscarGenero(ColaP, Genero, LF), !.

tituloUnico([[Titulo, _,_]|_], Titulo):- !.

tituloUnico(_|ColaG, Titulo):- tituloUnico(ColaG, Titulo), !.

existeGenero([[Genero|_|_], Genero):- !.

existeGenero(_|Cola, Genero):- existeGenero(Cola,Genero), !.

agregarGenero(Plataforma, Genero, NuevaPlataforma):- NuevoGenero = [Genero],
agregarAlFinal(Plataforma, NuevoGenero, NuevaPlataforma).

peliculaAlGenero([[Genero|ColaG]|ColaP], Pelicula, Genero, Plataforma2):-
agregarAlFinal(ColaG, Pelicula, NuevaColaG), Plataforma2 =
[[Genero|NuevaColaG]|ColaP], !.

peliculaAlGenero([CabezaP|ColaP], Pelicula, Genero, Plataforma2):- Plataforma2 =
[CabezaP|NuevaColaP], peliculaAlGenero(ColaP, Pelicula, Genero, NuevaColaP), !.

eliminarPeliculaGenero([[Titulo,_,_]|ColaG], Titulo, _, ColaG).

eliminarPeliculaGenero([OtroTitulo|ColaG], Titulo, Genero, [OtroTitulo|NuevaColaG]):-
eliminarPeliculaGenero(ColaG, Titulo, Genero, NuevaColaG), !.

filtrarTiempo([], _, _, []).

filtrarTiempo([[Titulo, Duracion, Restriccion]|Cola], Min, Max, [CabezaF|ColaF]):-

Duracion >= Min, Duracion <= Max, CabezaF = [Titulo, Duracion, Restriccion],
filtrarTiempo(Cola, Min, Max, ColaF), !.

filtrarTiempo([_|Cola], Min, Max, LF):- filtrarTiempo(Cola, Min, Max, LF), !.

%A)

agregarPelicula(Plataforma, Titulo, Genero, Duracion, Restriccion, Plataforma2):-

 existeGenero(Plataforma, Genero),

 buscarGenero(Plataforma, Genero, ListaGenero),

 not(tituloUnico(ListaGenero, Titulo)),

 crearPelicula(Titulo, Genero, Duracion, Restriccion, Pelicula),

 peliculaAlGenero(Plataforma, Pelicula, Genero, Plataforma2),!.

agregarPelicula(Plataforma, Titulo, Genero, Duracion, Restriccion, Plataforma2):-

 agregarGenero(Plataforma, Genero, PlataformaNueva),

 crearPelicula(Titulo, Genero, Duracion, Restriccion, Pelicula),

 peliculaAlGenero(PlataformaNueva, Pelicula, Genero, Plataforma2).

%B)

eliminarPelicula([[[Genero, [Titulo,_,_]]|ColaP], Genero, Titulo, ColaP).

Nombre: Christofer Rodríguez

Rut:20.239.786-7

Profesor: Víctor Flores

eliminarPelicula([[Genero|ColaG]|ColaP], Genero, Titulo, [NuevoGenero|ColaP]):-
eliminarPeliculaGenero([Genero|ColaG], Titulo, Genero, NuevoGenero), !.

eliminarPelicula([CabezaP|ColaP], Genero, Titulo, [CabezaP|NuevaColaP]):-
eliminarPelicula(ColaP, Genero, Titulo, NuevaColaP), !.

%C)

filtrarPorDuracion([], _, _, []).

filtrarPorDuracion([_|ColaG]|ColaP], Min, Max, [CabezaF|ColaF]):-

 filtrarTiempo(ColaG, Min, Max, CabezaF), not(CabezaF = []),
filtrarPorDuracion(ColaP, Min, Max, ColaF), !.

filtrarPorDuracion([_|ColaP], Min, Max, ColaF):- filtrarPorDuracion(ColaP, Min, Max,
ColaF), !.

D) agregarPelicula([], "Pelicula1", "Comedia", 120, "Todo Espectador", Plataforma2),
agregarPelicula(Plataforma2, "Pelicula2", "Terror", 180, "Adultos", Plataforma3),
agregarPelicula(Plataforma3, "Pelicula3", "Accion", 160, "Adolescentes", Plataforma4),
agregarPelicula(Plataforma4, "Pelicula4", "Comedia", 200, "Todo Espectador",
Plataforma5), eliminarPelicula(Plataforma5, "Accion", "Pelicula3", Plataforma6),
filtrarPorDuracion(Plataforma6, 150, 190, PeliculasFiltradas).

Nombre: Christofer Rodríguez

Rut:20.239.786-7

Profesor: Víctor Flores

Pregunta 2:

1)

estudiante(_, Estudiante, _,CodigoInformatica).

2)

profesorDe(NombreCarrera, Profesor):-

```
carrera(CodigoCarrera, NombreCarrera),  
asignatura(CodigoAsignatura, _, _, _, CodigoCarrera),  
seccion(_, RutProfesor, CodigoAsignatura, _),  
profesor(RutProfesor, Profesor, _, _).
```

3) Ya que no se especifica a que se refiere con prerrequisito, se asume que el Código de prerrequisito hace referencia que el estudiante tenga una nota aprobatoria (mayor o igual a 4) en la asignatura con código de asignatura] igual al código de prerrequisito.

puedeInscribir(RutEstudiante, NombreAsignatura):-

asignatura(_, NombreAsignatura, _,CodigoPrerrequisito,_),

seccion(CodigoSeccion, _, CodigoPrerrequisito, _),

calificacion(CodigoSeccion, RutEstudiante, Nota),

Nota >= 4.

4)

`mismaAsignatura(X, Y, NombreAsignatura):-`

`asignatura(CodigoAsignatura, NombreAsignatura, _, _, _),`

`seccion(CodigoSeccion, _, CodigoAsignatura, _),`

`inscripcion(RutX, CodigoSeccion),`

`inscripcion(RutY, CodigoSeccion),`

`not(RutX = RutY),`

`estudiante(RutX, X, _, _),`

`estudiante(RutY, Y, _, _).`

5)

numeroMayor([], Mayor, Mayor).

numeroMayor([E|Cola], Mayor, X):- E > Mayor, numeroMayor(Cola, E, X), !.

numeroMayor(_|Cola, Mayor, X):- numeroMayor(Cola, Mayor, X), !.

mayorNota(CodigoSeccion, Rut):-

findall(X, calificacion(_,_,X), ListaNotas),

numeroMayor(ListaNotas, 0, MejorNota),

calificacion(CodigoSeccion, Rut, MejorNota).