

# REDFLIX

Práctica 3: Inteligencia Artificial

2016-2017 FIB

David Marín Medina

Ricard Meyerhofer Parra

Alex Cuello Ortiz

<b>Introducción</b>	<b>2</b>
<b>Dominio</b>	<b>2</b>
Características del dominio	2
Justificación de los predicados	3
<b>Operadores</b>	<b>4</b>
Dominio Básico (dominioBasico.pddl)	5
Primera Extensión (dominioE1.pddl)	5
Segunda Extensión (dominioE2.pddl)	5
Tercera Extensión (dominioE3b.pddl)	6
Cuarta Extensión (dominioE4.pddl)	6
<b>Juegos de prueba</b>	<b>6</b>
Generador juegos de prueba	7
Juegos de prueba nivel básico	7
Problema Básico A (problemaBa.pddl)	7
Problema Básico B (problemaBb.pddl)	8
Problema Básico C (problemaBc.pddl)	9
Juegos de prueba de la extensión 1	10
Problema Extensión 1 A (problemaE1a.pddl)	10
Problema Extensión 1 B (problemaE1b.pddl)	11
Problema Extensión 1 C (problemaE1c.pddl)	12
Juegos de prueba de la extensión 2	13
Problema Extensión 2 A (problemaE2a.pddl)	13
Problema Extensión 2 B (problemaE2b.pddl)	14
Problema Extensión 2 C (problemaE2c.pddl)	16
Juegos de prueba de la extensión 3	17
Problema Extensión 3 A (problemaE3a.pddl)	17
Problema Extensión 3 B (problemaE3b.pddl)	19
Problema Extensión 3 C (problemaE3c.pddl)	20
Juegos de prueba de la extensión 4	23
Problema Extensión 4 A (problemaE4a.pddl)	23
Problema Extensión 4 B (problemaE4b.pddl)	24
Problema Extensión 4 C (problemaE4c.pddl)	26

## Introducción

En esta última práctica la empresa REDFLIX, distribuidora de contenido audiovisual, entre ellos series de televisión y películas. Nos han encargado la tarea de realizar una aplicación capaz de planificar el visionado de las series preferidas de cada uno de los usuarios. Con los conocimientos obtenidos en la asignatura de Inteligencia Artificial poniendo en práctica dichos conocimientos sobre planificación vamos a realizar esta tarea.

## Dominio

### Características del dominio

Como hemos comentado, es claramente un problema de planificación. Para ello utilizaremos el lenguaje de Planning Domain Definition Language (pddl). Se trata de un lenguaje declarativo, el cual requiere una definición del dominio en forma de predicados y acciones.

Cada acción tiene definidos unas precondiciones y efectos, cuando los elementos del sistema cumplen las precondiciones se ejecutan los efectos. De esta forma se mueve por el dominio hasta llegar a una solución válida.

Los elementos del dominio serán (en función de la extensión) por tanto los contenidos del catálogo (series y películas), los contenidos que son predecesores a un contenido, los contenidos que son paralelos a un contenido, los contenidos que el usuario ya ha visto y los que quiere ver.

Para ello necesitaremos los siguientes predicados:

- predecesor
- paralelo
- quiereVer
- visto
- yaPlanificado
- diaOcupado

Predecesor, quiereVer, visto y yaPlanificado nos son requeridos en todas los dominios. El paralelo es requerido a partir de la segunda extensión.

El predicado diaOcupado lo usamos solo en el caso básico y en la primera extensión.

## Justificación de los predicados

### **Predecesor (predecesor ?content - contenido ?pred - contenido)**

Es el predicado que nos indica el orden de visualización entre dos contenidos (donde el predecesor tendrá ser visualizado los días anteriores. Content es el que tiene un predecesor ?pred que es otro contenido. Por ejemplo si tenemos:

(predecesor Rambo2 Rambo1)

Para ver Rambo2, tendremos que ver Rambo1 si no la hemos visto antes. El predicado de predecesor en el primer nivel que se nos pedía solo puede haber uno pero a partir de la primera extensión, podemos tener N predecesores y así pues ir encadenando predecesores.

Este predicado es necesario para poder expresar la relación de predecesor entre dos contenidos, de otra manera no sería posible expresar esta relación.

### **Paralelo (paralelo ?content1 - contenido ?content2 - contenido)**

Es el predicado que nos indica la relación de paralelismo entre dos contenidos. Por lo tanto estos contenidos si son paralelos deben visualizarse el mismo día o en el día anterior o día posterior. Un contenido puede tener M contenidos paralelos.

Este predicado es necesario para poder expresar la relación de paralelo entre dos contenidos, de otra manera no sería posible expresar esta relación.

### **QuiereVer (quiereVer ?content - contenido)**

Este predicado nos es útil para expresar qué es lo que quiere ver el usuario y qué no. Si no queremos ver un contenido, no tenemos que planificarlo por lo tanto, nos es muy útil para poder discernir sobre qué ver y qué no ver. Así pues, sólo veremos este contenido si lo quiere ver o si es paralelo o predecesor de otro contenido ("recursivamente", podría ser el predecesor de otro predecesor de lo que queremos ver) que queremos ver.

**Visto (visto ?content - contenido)**

Es el predicado que nos indica que un contenido ya ha sido visualizado y por lo tanto no tenemos que ver otra vez y no tenemos que ver por ejemplo en el caso de que sea un predecesor o un paralelo de otro contenido.

**YaPlanificado (yaPlanificado ?content - contenido)**

Es el predicado que nos indica que un contenido ya ha sido planificado para una futura visualización.

Esto nos sirve para no volver a planificar contenidos y para marcar qué contenidos vamos a ver y cuáles no (por lo cual solo veremos los contenidos que estén bajo la etiqueta de yaPlanificado una vez terminemos nuestra ejecución).

**DiaOcupado (diaOcupado ?d - dia)**

Este predicado solo lo usamos en los primeros casos (básico y primera extensión) donde lo que hacíamos era asignar a un contenido un día así pues a tantos contenidos teníamos tantos días usábamos. Esta asignación posteriormente la cambiamos por una mejor en la asignación usando las funciones (numdia ?d - dia) y (diaAsig ? content - contenido).

Con estos predicados podemos representar adecuadamente, y de forma completa, todas las características del dominio:

- Los contenidos que el usuario ha visto y que desea visualizar.
- Los contenidos que son predecesores, paralelos o ambos respecto otros contenidos.

## Operadores

En nuestro caso nos gustaría resaltar que el básico y la extensión uno entregado son lo mismo. Eso es debido a que por una parte, hicimos directamente que hubiera N predecesores y que por otro lado, como no nos quedaba claro si incorporar los días o no, los añadimos ante la duda.

Una vez clarificado esto, procedemos a el análisis de los operadores usados en cada uno de los dominios que hemos realizado.

## Dominio Básico (dominioBasico.pddl)

Este dominio cuenta con una sola acción llamada *planning*, que cuenta con un parámetro de tipo contenido y otro de tipo día.

La precondition, comprueba que el contenido no esté incluido en el plan de visualización ni se haya visualizado previamente, y que se puede añadir al día deseado, es decir, que no esté el día ocupado. Además, en caso de cumplir con lo anterior también se comprueba que, todo contenido predecesor se haya visualizado o haya sido planeado con anterioridad. sea predecesor de un contenido que el usuario quiere visualizar o no tenga ningún predecesor. Finalmente, la acción utiliza el predicado *planning* con el contenido si cumple las condiciones. Y marca ya el día como ocupado y que dicho contenido ya ha sido planificado.

## Primera Extensión (dominioE1.pddl)

Tal y como hemos mencionado anteriormente, al realizar el Domino Básico realizamos la tarea que se nos pide en esta extensión por lo que vienen a ser lo mismo. De este modo el planificador es capaz de construir un plan para poder llegar a ver los contenidos objetivo, donde para todo contenido que pertenece al plan, todos sus contenidos predecesores pertenecen al plan y están en días anteriores.

## Segunda Extensión (dominioE2.pddl)

En esta extensión decidimos incorporar las funciones y re-implementar el código. Los casos que tenemos que considerar en este caso en nuestra lógica son los siguientes:

- No tienen predecesores ni paralelos
- Tienen predecesores pero no paralelos
- No tienen predecesores pero sí que tienen paralelos
- Tienen predecesores y paralelos

De este modo el planificador es capaz de construir un plan para poder llegar a ver los contenidos objetivo donde para todo contenido que pertenece al plan, todos sus contenidos paralelos pertenecen al plan y están en el día anterior o en el mismo día.

### Tercera Extensión (dominioE3b.pddl)

Esta extensión es una mejora de la segunda extensión donde hemos añadido una función (numAssig ?d - día) que nos permite llevar un control de los contenidos visualizados por día. Simplemente añadiendo un predicado que no permita superar el límite de asignaciones ( $(\text{numAsig } ?d) \leq 3$ ), y obviamente un incrementador de estas ( $\text{increase } (\text{numAsig } ?d) \ 1$ ) ya tendríamos la nueva extensión.

### Cuarta Extensión (dominioE4.pddl)

En esta extensión manteniendo la estructura de la anterior hemos eliminado (numAsig ?d - día) dado que ahora no estamos valorando este caso. Y en su lugar añadimos (minutosOcupados ?d - día) (minutos ?content - contenido) ya que queremos controlar que no se superen los 200 minutos al día. Para ello debemos controlar esto con esta condición: ( $\geq 200 + (\text{minutos } ?\text{content}) (\text{minutosOcupados } ?d)$ ). Y con tal de que tenga sentido debemos incrementar a los minutos ocupados los minutos del contenido.

De este modo el planificador controla que en el plan generado no se superen los 200 minutos al día.

## Juegos de prueba

Para esta práctica, hemos ideado una serie de juegos de prueba de dificultad incremental para cada Extensión incluyendo el problema básico.

En ellos trataremos el problema básico de la Extensión correspondiente, una ligera extensión y un problema de gran tamaño que incluso en el caso de la Extensión 3 y la Extensión 4 el resolutor puede demorarse más de un minuto.

## Generador juegos de prueba

El generador de juegos de prueba que hemos realizado ha sido realizado con C++ donde damos la posibilidad de generar juegos de prueba (los que queramos) del caso base y de cada una de las extensiones.

Para ejecutar el generador de pruebas solamente tenemos que compilarlo y ejecutarlo (generará los tests en la misma carpeta donde estemos). Hemos adjuntado un makefile donde con los comandos make podremos generar el código compilado de forma sencilla y con un make clean borrar los tests creados y el ejecutable.

## Juegos de prueba nivel básico

### Problema Básico A (problemaBa.pddl)

El problema básico nos plantea generar un plan de visionado donde cada contenido tendrá de 0 a 1 predecesor y ningún paralelo. El juego de pruebas A es el siguiente:

Juego de pruebas
<pre>(:objects   Rambo1 Rambo2   SpiritedAway - contenido   dia1 dia2 dia3 dia4 dia5 - dia ) (:init   (predecesor Rambo2 Rambo1)   (quiereVer Rambo2)   (quiereVer SpiritedAway) )</pre>

En este juego de pruebas tendremos un par de películas donde una será la predecesora de la otra y una película que no tiene ninguna dependencia.

El usuario querrá ver la segunda de las películas con dependencia por lo que el plan tendrá que añadir su predecesora además de la segunda película sin dependencias.

**Salida del solucionador para problemaBa.pddl**



```

step  0: PLANNING SPIRITEDAWAY DIA1
      1: PLANNING RAMBO1 DIA2
      2: PLANNING RAMBO2 DIA3

time spent:  0.00 seconds total time

```

### Problema Básico B (problemaBb.pddl)

El juego de pruebas B extiende el primero y le añade una serie más y otra película sin precedencias:

Juego de pruebas
<pre> (:objects   Rambo1 Rambo2   Rocky1 Rocky2   CastilloEnElCielo   SpiritedAway - contenido   dia1 dia2 dia3 dia4 dia5 dia6 - dia )  (:init   (predecesor Rambo2 Rambo1)   (predecesor Rocky2 Rocky1)    (quiereVer Rambo2)   (quiereVer Rocky2)   (quiereVer CastilloEnElCielo)   (quiereVer SpiritedAway) ) </pre>

Este juego de pruebas tiene que añadir al plan otra película con una precedencia y otra película sin. De esta manera el planificador, irá asignando una película por día debido a que hay días suficientes.

Salida del solucionador para problemaBb.pddl
<pre> step  0: PLANNING SPIRITEDAWAY DIA1       1: PLANNING CASTILLOENELCIELO DIA2       2: PLANNING ROCKY1 DIA3       3: PLANNING ROCKY2 DIA4       4: PLANNING RAMBO1 DIA5       5: PLANNING RAMBO2 DIA6  time spent:  0.00 seconds total time </pre>

## Problema Básico C (problemaBc.pddl)

El juego de pruebas B extiende el primero y le añade una serie más y otra película sin precedencias:

Juego de pruebas
<pre>(:objects   Rambo1 Rambo2   Rocky1 Rocky2   ElSenorDeLosAnillos1 ElSenorDeLosAnillos2   Doraemon1 Doraemon2   ShinChan1 ShinChan2   KillBill1 KillBill2   - contenido   dia1 dia2 dia3 dia4 dia5 dia6 dia7 dia8 dia9 dia10 dia11 dia12 - dia )  (:init   (predecesor Rambo2 Rambo1)   (predecesor Rocky2 Rocky1)   (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)   (predecesor Doraemon2 Doraemon1)   (predecesor ShinChan2 ShinChan1)   (predecesor KillBill2 KillBill1)    (quiereVer Rambo2)   (quiereVer Rocky2)   (quiereVer ElSenorDeLosAnillos2)   (quiereVer Doraemon2)   (quiereVer ShinChan2)   (quiereVer KillBill2) )</pre>

Este último juego de pruebas añade un pelín de dificultad; el plan está compuesto de seis series de dos películas donde dentro de cada serie una de las películas es predecesor de la otra.

El planificador asignará a cada película un día, respetando todos los predecesores ya que es condición necesaria haber visto la primera película de cada serie para ver la segunda.

Salida del solucionador para problemaBc.pddl
<pre>step 0: PLANNING KILLBILL1 DIA1       1: PLANNING KILLBILL2 DIA2       2: PLANNING SHINCHAN1 DIA3       3: PLANNING SHINCHAN2 DIA4</pre>

```

4: PLANNING DORAEMON1 DIA5
5: PLANNING DORAEMON2 DIA6
6: PLANNING ELSENORDELOSANILLOS1 DIA7
7: PLANNING ELSENORDELOSANILLOS2 DIA8
8: PLANNING ROCKY1 DIA9
9: PLANNING ROCKY2 DIA10
10: PLANNING RAMBO1 DIA11
11: PLANNING RAMBO2 DIA12

```

```
time spent: 0.01 seconds total time
```

## Juegos de prueba de la extensión 1

### Problema Extensión 1 A (problemaE1a.pddl)

La primera de las extensiones nos incrementa el número de predecesores de 0 a N pero todavía sin ningún paralelo, con los predecesores asignados a días anteriores. El juego de pruebas A es el siguiente:

#### Juego de pruebas

```

(:objects
  Rocky1 Rocky2 Rocky3
  SpiritedAway - contenido
  dia1 dia2 dia3 dia4 dia5 dia6 - dia
)

(:init
  (predecesor Rocky2 Rocky1)
  (predecesor Rocky3 Rocky2)

  (quiereVer Rocky3)
  (quiereVer SpiritedAway)
)

```

Este primer juego de pruebas de la extensión 1 prueba el problema básico. Añadimos una película más, extendiendo el primero de los problemas del nivel básico.

El planificador asignará una película a cada día respetando los predecesores de la primera serie de películas.

Salida del solucionador para problemaE1a.pddl	
step	0: PLANNING SPIRITEDAWAY DIA1 1: PLANNING ROCKY1 DIA2 2: PLANNING ROCKY2 DIA3 3: PLANNING ROCKY3 DIA4  time spent: 0.00 seconds total time

### Problema Extensión 1 B (problemaE1b.pddl)

El segundo juego de pruebas añade otra serie de películas con dos predecesores al problema anterior:

Juegos de pruebas	
(:objects	Rocky1 Rocky2 Rocky3 ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 SpiritedAway - contenido dia1 dia2 dia3 dia4 dia5 dia6 dia7 - dia )
(:init	<pre> (predecesor Rocky2 Rocky1) (predecesor Rocky3 Rocky2)  (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1) (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)  (quiereVer Rocky3) (quiereVer ElSenorDeLosAnillos3) (quiereVer SpiritedAway) </pre> )

En este juego de pruebas el planificador tendrá dos series con dos predecesores cada una, con lo que asignará un capítulo por día a cada película.

Salida del solucionador para problemaE1b.pddl	
step	0: PLANNING SPIRITEDAWAY DIA1

```

1: PLANNING ELSENORDELOSANILLOS1 DIA2
2: PLANNING ELSENORDELOSANILLOS2 DIA3
3: PLANNING ELSENORDELOSANILLOS3 DIA4
4: PLANNING ROCKY1 DIA5
5: PLANNING ROCKY2 DIA6
6: PLANNING ROCKY3 DIA7

```

```
time spent: 0.00 seconds total time
```

### Problema Extensión 1 C (problemaE1c.pddl)

Este último juego de pruebas para la primera extensión tiene más complejidad.

En el tenemos dos series de películas con dos predecesores, otra con siete predecesores y una película sin predecesores.

#### Juegos de pruebas

```

(:objects
  Rocky1 Rocky2 Rocky3
  ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3
  HarryPotter1 HarryPotter2 HarryPotter3 HarryPotter4 HarryPotter5
  HarryPotter6 HarryPotter7 HarryPotter8
  SpiritedAway - contenido
  dia1 dia2 dia3 dia4 dia5 dia6 dia7 dia8 dia9 dia10 dia11 dia12 dia13
  dia14 dia15 - dia
)

(:init
  (predecesor Rocky2 Rocky1)
  (predecesor Rocky3 Rocky2)

  (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)
  (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)

  (predecesor HarryPotter2 HarryPotter1)
  (predecesor HarryPotter3 HarryPotter2)
  (predecesor HarryPotter4 HarryPotter3)
  (predecesor HarryPotter5 HarryPotter4)
  (predecesor HarryPotter6 HarryPotter5)
  (predecesor HarryPotter7 HarryPotter6)
  (predecesor HarryPotter8 HarryPotter7)

  (quiereVer Rocky3)
  (quiereVer ElSenorDeLosAnillos3)
  (quiereVer SpiritedAway)
  (quiereVer HarryPotter8)
)

```

El planificador tiene días suficientes para planificarlo todo. En la planificación tendrán que aparecer todas las series de películas respetando cada uno de sus predecesores aunque cabe la posibilidad de que no caigan en días consecutivos, aunque no es condición necesaria.

El usuario en este caso quiere ver la última de todas las cadenas de películas y la película sin predecesores por lo que todo el contenido predecesor tendrá que formar parte del plan.

Salida del solucionador para problemaE1c.pddl	
step	0: PLANNING SPIRITEDAWAY DIA1 1: PLANNING HARRYPOTTER1 DIA2 2: PLANNING HARRYPOTTER2 DIA3 3: PLANNING HARRYPOTTER3 DIA4 4: PLANNING HARRYPOTTER4 DIA5 5: PLANNING HARRYPOTTER5 DIA6 6: PLANNING HARRYPOTTER6 DIA7 7: PLANNING HARRYPOTTER7 DIA8 8: PLANNING HARRYPOTTER8 DIA9 9: PLANNING ELSENORDELOSANILLOS1 DIA10 10: PLANNING ELSENORDELOSANILLOS2 DIA11 11: PLANNING ELSENORDELOSANILLOS3 DIA12 12: PLANNING ROCKY1 DIA13 13: PLANNING ROCKY2 DIA14 14: PLANNING ROCKY3 DIA15
time spent:	0.00 seconds total time

## Juegos de prueba de la extensión 2

### Problema Extensión 2 A (problemaE2a.pddl)

La primera de las extensiones responde al problema de los contenidos paralelos. A demás de tener contenido con 0 a N predecesores, ahora tendremos de 0 a M contenidos paralelos de algunos de los contenidos.

El problema A es sencillo, contiene una serie de tres películas las cuales una es predecesora y otra es paralela.

Al ser uno de ellos paralelo, el resolvedor podrá colocar la película paralela un día antes o el mismo día que su contenido paralelo.

Juegos de pruebas
<pre> (:objects   ElSenorDeLosAnillos1 ElSenorDeLosAnillos2   ElSenorDeLosAnillos3 - contenido   dia1 dia2 - dia )  (:init   (= (numdia dia1) 1)   (= (numdia dia2) 2)    (= (diaAsig ElSenorDeLosAnillos1) 0)   (= (diaAsig ElSenorDeLosAnillos2) 0)   (= (diaAsig ElSenorDeLosAnillos3) 0)    (paralelo ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)   (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)    (quiereVer ElSenorDeLosAnillos3) ) </pre>

El planificador, al tener solo dos días para realizar el plan, tendrá que colocar dos películas en un mismo día, cumpliendo así la condición de su contenido paralelo, y colocando el descendiente al día siguiente.

Salida del solucionador para problemaE2a.pddl
<pre> step  0: PLANNING ELSENORDELOSANILLOS1 DIA1       1: PLANNING ELSENORDELOSANILLOS2 DIA1       2: PLANNING ELSENORDELOSANILLOS3 DIA2       3: REACH-GOAL  time spent:  0.00 seconds total time </pre>

Problema Extensión 2 B (problemaE2b.pddl)

El segundo juego de pruebas de la extensión2 nos propone planificar una serie de ocho películas en dos días.

Juegos de pruebas
<pre> (:objects </pre>

```

        HarryPotter1    HarryPotter2    HarryPotter3    HarryPotter4    HarryPotter5
HarryPotter6
    HarryPotter7 HarryPotter8 - contenido
    dia1 dia2 dia3 dia4 - dia
)

(:init
    (= (numdia dia1) 1)
    (= (numdia dia2) 2)
    (= (numdia dia3) 3)
    (= (numdia dia4) 4)
    (= (diaAsig HarryPotter1) 0)
    (= (diaAsig HarryPotter2) 0)
    (= (diaAsig HarryPotter3) 0)
    (= (diaAsig HarryPotter4) 0)
    (= (diaAsig HarryPotter5) 0)
    (= (diaAsig HarryPotter6) 0)
    (= (diaAsig HarryPotter7) 0)
    (= (diaAsig HarryPotter8) 0)
    (predecesor HarryPotter2 HarryPotter1)
    (paralelo    HarryPotter3 HarryPotter2)
    (predecesor HarryPotter4 HarryPotter3)
    (predecesor HarryPotter5 HarryPotter4)
    (paralelo    HarryPotter6 HarryPotter5)
    (predecesor HarryPotter7 HarryPotter6)
    (predecesor HarryPotter8 HarryPotter7)
    (quiereVer HarryPotter8)
)

```

El planificador, teniendo en cuenta que hay dos parejas de películas paralelas, podrá realizar un planing en el que el usuario pueda ver las 8 películas en tan solo cuatro días haciendo que las películas paralelas se puedan ver en días anteriores o iguales que su contenido paralelo.

Salida del solucionador para problemaE2b.pddl	
step	0: PLANNING HARRYPOTTER1 DIA1 1: PLANNING HARRYPOTTER2 DIA2 2: PLANNING HARRYPOTTER3 DIA1 3: PLANNING HARRYPOTTER4 DIA2 4: PLANNING HARRYPOTTER5 DIA3 5: PLANNING HARRYPOTTER6 DIA2 6: PLANNING HARRYPOTTER7 DIA3 7: PLANNING HARRYPOTTER8 DIA4 8: REACH-GOAL
time spent: 0.00 seconds total time	



## Problema Extensión 2 C (problemaE2c.pddl)

El último de los juegos de prueba de la extensión 2 contiene una serie de tres películas, otra de ocho, otra más de dos y tendrá que realizar un planing en once días como mostramos a continuación.

Juegos de pruebas
<pre>(:objects   ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3   HarryPotter1 HarryPotter2 HarryPotter3 HarryPotter4 HarryPotter5   HarryPotter6   HarryPotter7 HarryPotter8   KillBill1 KillBill2 - contenido   dia1 dia2 dia3 dia4 dia5 dia6 dia7 dia8 dia9 dia10 dia11 - dia ) (:init   (= (numdia dia1) 1)   (= (numdia dia2) 2)   (= (numdia dia3) 3)   (= (numdia dia4) 4)   (= (numdia dia5) 5)   (= (numdia dia6) 6)   (= (numdia dia7) 7)   (= (numdia dia8) 8)   (= (numdia dia9) 9)   (= (numdia dia10) 10)   (= (numdia dia11) 11)   (= (diaAsig ElSenorDeLosAnillos1) 0)   (= (diaAsig ElSenorDeLosAnillos2) 0)   (= (diaAsig ElSenorDeLosAnillos3) 0)   (= (diaAsig HarryPotter1) 0)   (= (diaAsig HarryPotter2) 0)   (= (diaAsig HarryPotter3) 0)   (= (diaAsig HarryPotter4) 0)   (= (diaAsig HarryPotter5) 0)   (= (diaAsig HarryPotter6) 0)   (= (diaAsig HarryPotter7) 0)   (= (diaAsig HarryPotter8) 0)   (= (diaAsig KillBill1) 0)   (= (diaAsig KillBill2) 0)    (paralelo ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)   (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)   (predecesor HarryPotter2 HarryPotter1)   (paralelo HarryPotter3 HarryPotter2)   (predecesor HarryPotter4 HarryPotter3)   (predecesor HarryPotter5 HarryPotter4)   (paralelo HarryPotter6 HarryPotter5)   (predecesor HarryPotter7 HarryPotter6)   (predecesor HarryPotter8 HarryPotter7)   (predecesor KillBill2 KillBill1))</pre>

```
(quiereVer ElSenorDeLosAnillos3)
(quiereVer HarryPotter8)
(quiereVer KillBill2)
)
```

El solucionador generará un planing dentro de los 11 días, donde la saga de ocho películas se verá en entre los días 8 y 11 respetando los predecesores y con sus contenidos paralelos dentro del margen del día anterior o mismo día que su contenido paralelo.

Las otras dos series de películas aparecen en el planing en un orden adecuado aunque el planing distancie el visionado de alguna de las películas en días posteriores.

#### Salida del solucionador para problemaE2c.pddl

```
step 0: PLANNING KILLBILL1 DIA1
      1: PLANNING KILLBILL2 DIA11
      2: PLANNING ELSENORDELOSANILLOS1 DIA1
      3: PLANNING ELSENORDELOSANILLOS2 DIA2
      4: PLANNING ELSENORDELOSANILLOS3 DIA11
      5: PLANNING HARRYPOTTER1 DIA8
      6: PLANNING HARRYPOTTER2 DIA9
      7: PLANNING HARRYPOTTER3 DIA8
      8: PLANNING HARRYPOTTER4 DIA9
      9: PLANNING HARRYPOTTER5 DIA10
     10: PLANNING HARRYPOTTER6 DIA9
     11: PLANNING HARRYPOTTER7 DIA10
     12: PLANNING HARRYPOTTER8 DIA11
     13: REACH-GOAL
```

```
time spent: 0.25 seconds total time
```

## Juegos de prueba de la extensión 3

### Problema Extensión 3 A (problemaE3a.pddl)

El problema de la tercera extensión trata sobre limitar a tres contenidos al día, es decir, asignar a cada día como máximo tres películas o series.

En este primer juego de pruebas tenemos una serie de distintas películas y una serie de 3 películas con dos predecesores donde el usuario quiere ver todas las películas y la última de las películas con dos predecesores.

Juegos de pruebas
<pre>(:objects   django   elOrfanato   jumanji   intocable   ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 - contenido   dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia ) (:init   (= (numdia dia01) 01)    (= (numAsig dia01) 0)   (= (numdia dia02) 02)    (= (numAsig dia02) 0)   (= (numdia dia03) 03)    (= (numAsig dia03) 0)   (= (numdia dia04) 04)    (= (numAsig dia04) 0)   (= (numdia dia05) 05)    (= (numAsig dia05) 0)   (= (numdia dia06) 06)    (= (numAsig dia06) 0)   (= (numdia dia07) 07)    (= (numAsig dia07) 0)   (= (numdia dia08) 08)    (= (numAsig dia08) 0)   (= (numdia dia09) 09)    (= (numAsig dia09) 0)   (= (numdia dia10) 10)    (= (numAsig dia10) 0)   (= (numdia dia11) 11)    (= (numAsig dia11) 0)   (= (diaAsig django) 0)   (= (diaAsig elOrfanato) 0)   (= (diaAsig jumanji) 0)   (= (diaAsig intocable) 0)   (= (diaAsig ElSenorDeLosAnillos1) 0)   (= (diaAsig ElSenorDeLosAnillos2) 0)   (= (diaAsig ElSenorDeLosAnillos3) 0)   (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)   (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)   (quiereVer django)   (quiereVer elOrfanato)   (quiereVer jumanji)   (quiereVer intocable)   (quiereVer ElSenorDeLosAnillos3) )</pre>

Con esta entrada buscamos que el planificador coloque películas en los distintos días y en el momento de asignar 3 películas a un día no asigne más. Al tener una película con predecesores, esta debería planificarse a partir del segundo día al estar completo el pimer día por las películas anteriores.

Salida del solucionador para problemaE3a.pddl	
step	0: PLANNING DJANGO DIA01 1: PLANNING ELORFANATO DIA01 2: PLANNING JUMANJI DIA01 3: PLANNING INTOCABLE DIA02 4: PLANNING ELSENORDELOSANILLOS1 DIA02 5: PLANNING ELSENORDELOSANILLOS2 DIA03 6: PLANNING ELSENORDELOSANILLOS3 DIA04
time spent:	0.00 seconds total time

### Problema Extensión 3 B (problemaE3b.pddl)

Este segundo juego de pruebas amplía el catálogo con varias series y películas todas con predecesores.

Juegos de pruebas
<pre>(:objects   breakingBad01 breakingBad02 breakingBad03   narcos1 narcos2 narcos3   pokemon1 pokemon2 pokemon3   juegoDeTronos1 juegoDeTronos2 juegoDeTronos3 - contenido   dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia )  (:init   (= (numdia dia01) 01)    (= (numAsig dia01) 0)   (= (numdia dia02) 02)    (= (numAsig dia02) 0)   (= (numdia dia03) 03)    (= (numAsig dia03) 0)   (= (numdia dia04) 04)    (= (numAsig dia04) 0)   (= (numdia dia05) 05)    (= (numAsig dia05) 0)   (= (numdia dia06) 06)    (= (numAsig dia06) 0)   (= (numdia dia07) 07)    (= (numAsig dia07) 0)   (= (numdia dia08) 08)    (= (numAsig dia08) 0)   (= (numdia dia09) 09)    (= (numAsig dia09) 0)   (= (numdia dia10) 10)    (= (numAsig dia10) 0)   (= (numdia dia11) 11)    (= (numAsig dia11) 0)   (= (diaAsig breakingBad01) 0)   (= (diaAsig breakingBad02) 0)   (= (diaAsig breakingBad03) 0)   (predecesor breakingBad02 breakingBad01)   (predecesor breakingBad03 breakingBad02)   (= (diaAsig narcos1) 0)   (= (diaAsig narcos2) 0)</pre>

```

(= (diaAsig narcos3) 0)
(predesor narcos2 narcos1)
(predesor narcos3 narcos2)
(= (diaAsig pokemon1) 0)
(= (diaAsig pokemon2) 0)
(= (diaAsig pokemon3) 0)
(predesor pokemon2 pokemon1)
(predesor pokemon3 pokemon2)
(= (diaAsig juegoDeTronos1) 0)
(= (diaAsig juegoDeTronos2) 0)
(= (diaAsig juegoDeTronos3) 0)
(predesor juegoDeTronos2 juegoDeTronos1)
(predesor juegoDeTronos3 juegoDeTronos2)
(quiereVer breakingBad03)
(quiereVer narcos3)
(quiereVer pokemon3)
(quiereVer juegoDeTronos3)
)

```

Al tener todas la series predecesores, el resolutor tendrá que ir asignando los distintos contenidos a días consecutivos hasta la última serie de contenido, que se encontrará los primeros tres días llenos y se asignarán a los tres días siguientes.

#### Salida del solucionador para problemaE3b.pddl

```

step  0: PLANNING BREAKINGBAD01 DIA01
      1: PLANNING BREAKINGBAD02 DIA02
      2: PLANNING BREAKINGBAD03 DIA03
      3: PLANNING NARCOS1 DIA01
      4: PLANNING NARCOS2 DIA02
      5: PLANNING NARCOS3 DIA03
      6: PLANNING POKEMON1 DIA01
      7: PLANNING POKEMON2 DIA02
      8: PLANNING POKEMON3 DIA03
      9: PLANNING JUEGODETRONOS1 DIA04
     10: PLANNING JUEGODETRONOS2 DIA05
     11: PLANNING JUEGODETRONOS3 DIA06

```

```
time spent:  0.00 seconds total time
```

#### Problema Extensión 3 C (problemaE3c.pddl)

Este último juego de pruebas reunirá una serie de nueve capítulos, otra de seis, una película sin predecesores y una trilogía de películas, combinando todo lo visto hasta ahora.

## Juegos de pruebas

```

(:objects
  breakingBad01 breakingBad02 breakingBad03 breakingBad04 breakingBad05
  breakingBad06 breakingBad07 breakingBad08 breakingBad09
  narcos1 narcos2 narcos3 narcos4 narcos5 narcos6
  Jumanji
  ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 - contenido
  dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia
)

(:init
  (= (numdia dia01) 01)    (= (numAsig dia01) 0)
  (= (numdia dia02) 02)    (= (numAsig dia02) 0)
  (= (numdia dia03) 03)    (= (numAsig dia03) 0)
  (= (numdia dia04) 04)    (= (numAsig dia04) 0)
  (= (numdia dia05) 05)    (= (numAsig dia05) 0)
  (= (numdia dia06) 06)    (= (numAsig dia06) 0)
  (= (numdia dia07) 07)    (= (numAsig dia07) 0)
  (= (numdia dia08) 08)    (= (numAsig dia08) 0)
  (= (numdia dia09) 09)    (= (numAsig dia09) 0)
  (= (numdia dia10) 10)    (= (numAsig dia10) 0)
  (= (numdia dia11) 11)    (= (numAsig dia11) 0)
  (= (diaAsig breakingBad01) 0)
  (= (diaAsig breakingBad02) 0)
  (= (diaAsig breakingBad03) 0)
  (= (diaAsig breakingBad04) 0)
  (= (diaAsig breakingBad05) 0)
  (= (diaAsig breakingBad06) 0)
  (= (diaAsig breakingBad07) 0)
  (= (diaAsig breakingBad08) 0)
  (= (diaAsig breakingBad09) 0)
  (predecesor breakingBad02 breakingBad01)
  (predecesor breakingBad03 breakingBad02)
  (paralelo breakingBad04 breakingBad03)
  (predecesor breakingBad05 breakingBad04)
  (predecesor breakingBad06 breakingBad05)
  (predecesor breakingBad07 breakingBad06)
  (predecesor breakingBad08 breakingBad07)
  (predecesor breakingBad09 breakingBad08)
  (= (diaAsig narcos1) 0)
  (= (diaAsig narcos2) 0)
  (= (diaAsig narcos3) 0)
  (= (diaAsig narcos4) 0)
  (= (diaAsig narcos5) 0)
  (= (diaAsig narcos6) 0)
  (predecesor narcos2 narcos1)
  (predecesor narcos3 narcos2)
  (paralelo narcos4 narcos3)
  (predecesor narcos5 narcos4)
  (predecesor narcos6 narcos5)
  (= (diaAsig Jumanji) 0)
  (= (diaAsig ElSenorDeLosAnillos1) 0)
  (= (diaAsig ElSenorDeLosAnillos2) 0)
  (= (diaAsig ElSenorDeLosAnillos3) 0)
  (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)
  (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)
)

```

```

(quiereVer breakingBad09)
(quiereVer narcos6)
(quiereVer Jumanji)
(quiereVer ElSenorDeLosAnillos3)
)

```

Este juego de pruebas busca incluir todo los problemas hasta ahora para que el resolovedor genere un planning acorde a todas las restricciones. Las primeras series las colocará sin la restricción de de tres contenidos asignados por día y a partir de la segunda série se encontrará con el primer día lleno, teniendo que asignar contenido a partir del segundo. En este juego de pruebas, aunque teniendo contenidos paralelos no representan un cambio significativo en el planing ni ninguna mejora, aunque le añaden complejidad al resolovedor, que se refleja en el tiempo que tarda en generar el plan.

Salida del solucionador para problemaE3c.pddl	
step	0: PLANNING NARCOS1 DIA01 1: PLANNING NARCOS2 DIA02 2: PLANNING NARCOS3 DIA03 3: PLANNING NARCOS4 DIA04 4: PLANNING NARCOS5 DIA05 5: PLANNING NARCOS6 DIA06 6: PLANNING JUMANJI DIA01 7: PLANNING ELSENORDELOSANILLOS1 DIA01 8: PLANNING ELSENORDELOSANILLOS2 DIA02 9: PLANNING ELSENORDELOSANILLOS3 DIA11 10: PLANNING BREAKINGBAD01 DIA02 11: PLANNING BREAKINGBAD02 DIA06 12: PLANNING BREAKINGBAD03 DIA07 13: PLANNING BREAKINGBAD04 DIA06 14: PLANNING BREAKINGBAD05 DIA07 15: PLANNING BREAKINGBAD06 DIA08 16: PLANNING BREAKINGBAD07 DIA09 17: PLANNING BREAKINGBAD08 DIA10 18: PLANNING BREAKINGBAD09 DIA11 19: REACH-GOAL
time spent:	135.28 seconds total time

## Juegos de prueba de la extensión 4

### Problema Extensión 4 A (problemaE4a.pddl)

El primero de los test de la extensión 4 nos busca comprobar la restricción de 200 minutos al día con distintas películas de corta duración ficticia y una serie de tres películas de larga durada.

Juegos de prueba
<pre>(:objects   django   elOrfanato   jumanji   intocable   ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 - contenido   dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia )  (:init   (= (numdia dia01) 01)    (= (minutosOcupados dia01) 0)   (= (numdia dia02) 02)    (= (minutosOcupados dia02) 0)   (= (numdia dia03) 03)    (= (minutosOcupados dia03) 0)   (= (numdia dia04) 04)    (= (minutosOcupados dia04) 0)   (= (numdia dia05) 05)    (= (minutosOcupados dia05) 0)   (= (numdia dia06) 06)    (= (minutosOcupados dia06) 0)   (= (numdia dia07) 07)    (= (minutosOcupados dia07) 0)   (= (numdia dia08) 08)    (= (minutosOcupados dia08) 0)   (= (numdia dia09) 09)    (= (minutosOcupados dia09) 0)   (= (numdia dia10) 10)    (= (minutosOcupados dia10) 0)   (= (numdia dia11) 11)    (= (minutosOcupados dia11) 0)   (= (diaAsig django) 0)    (= (minutos django) 20)   (= (diaAsig elOrfanato) 0) (= (minutos elOrfanato) 20)   (= (diaAsig jumanji) 0)    (= (minutos jumanji) 30)   (= (diaAsig intocable) 0)  (= (minutos intocable) 60)   (= (diaAsig ElSenorDeLosAnillos1) 0) (= (minutos ElSenorDeLosAnillos1) 160)   (= (diaAsig ElSenorDeLosAnillos2) 0) (= (minutos ElSenorDeLosAnillos2) 170)   (= (diaAsig ElSenorDeLosAnillos3) 0) (= (minutos ElSenorDeLosAnillos3) 140)   (predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)   (predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)   (quiereVer django)   (quiereVer elOrfanato)   (quiereVer jumanji)   (quiereVer intocable)   (quiereVer ElSenorDeLosAnillos3) )</pre>



El resolutor tal como marca la estrictión asigna contenido al primer día sin problemas hasta que se encuentra con una película que al intentar asignarla rebasa los 200 minutos y a partir de ahí asigna nuevo contenido a distintos días ya que las películas duran demasiado para asignar varias a un mismo día.

Salida del solucionador para problemaE4a.pddl	
step	0: PLANNING DJANGO DIA01
	1: PLANNING ELORFANATO DIA01
	2: PLANNING JUMANJI DIA01
	3: PLANNING INTOCABLE DIA01
	4: PLANNING ELSENORDELOSANILLOS1 DIA02
	5: PLANNING ELSENORDELOSANILLOS2 DIA03
	6: PLANNING ELSENORDELOSANILLOS3 DIA04
time spent:	0.00 seconds total time

#### Problema Extensión 4 B (problemaE4b.pddl)

Este segundo problema para la cuarta de las extensión añade más complejidad al problema anterior. Todo el contenido está enlazado con predecesores y sucesores y tienen duración variada.

Juegos de prueba	
(:objects	breakingBad01 breakingBad02 breakingBad03
	narcos1 narcos2 narcos3
	pokemon1 pokemon2 pokemon3
	juegoDeTronos1 juegoDeTronos2 juegoDeTronos3
	ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 - contenido
	dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia
)	
(:init	
	(= (numdia dia01) 01) (= (minutosOcupados dia01) 0)
	(= (numdia dia02) 02) (= (minutosOcupados dia02) 0)
	(= (numdia dia03) 03) (= (minutosOcupados dia03) 0)
	(= (numdia dia04) 04) (= (minutosOcupados dia04) 0)
	(= (numdia dia05) 05) (= (minutosOcupados dia05) 0)
	(= (numdia dia06) 06) (= (minutosOcupados dia06) 0)
	(= (numdia dia07) 07) (= (minutosOcupados dia07) 0)

```

(= (numdia dia08) 08) (= (minutosOcupados dia08) 0)
(= (numdia dia09) 09) (= (minutosOcupados dia09) 0)
(= (numdia dia10) 10) (= (minutosOcupados dia10) 0)
(= (numdia dia11) 11) (= (minutosOcupados dia11) 0)
(= (diaAsig breakingBad01) 0) (= (minutos breakingBad01) 45)
(= (diaAsig breakingBad02) 0) (= (minutos breakingBad02) 45)
(= (diaAsig breakingBad03) 0) (= (minutos breakingBad03) 45)
(predecesor breakingBad02 breakingBad01)
(predecesor breakingBad03 breakingBad02)
(= (diaAsig narcos1) 0) (= (minutos narcos1) 32)
(= (diaAsig narcos2) 0) (= (minutos narcos2) 32)
(= (diaAsig narcos3) 0) (= (minutos narcos3) 32)
(predecesor narcos2 narcos1)
(predecesor narcos3 narcos2)
(= (diaAsig ElSenorDeLosAnillos1) 0) (= (minutos ElSenorDeLosAnillos1) 160)
(= (diaAsig ElSenorDeLosAnillos2) 0) (= (minutos ElSenorDeLosAnillos2) 170)
(= (diaAsig ElSenorDeLosAnillos3) 0) (= (minutos ElSenorDeLosAnillos3) 140)
(predecesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)
(predecesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)
(= (diaAsig pokemon1) 0) (= (minutos pokemon1) 22)
(= (diaAsig pokemon2) 0) (= (minutos pokemon2) 22)
(= (diaAsig pokemon3) 0) (= (minutos pokemon3) 22)
(predecesor pokemon2 pokemon1)
(predecesor pokemon3 pokemon2)
(= (diaAsig juegoDeTronos1) 0) (= (minutos juegoDeTronos1) 50)
(= (diaAsig juegoDeTronos2) 0) (= (minutos juegoDeTronos2) 50)
(= (diaAsig juegoDeTronos3) 0) (= (minutos juegoDeTronos3) 50)
(predecesor juegoDeTronos2 juegoDeTronos1)
(predecesor juegoDeTronos3 juegoDeTronos2)
(quiereVer breakingBad03)
(quiereVer narcos3)
(quiereVer ElSenorDeLosAnillos3)
(quiereVer pokemon3)
(quiereVer juegoDeTronos3)
)

```

El solucionador como en casos anteriores va asignado contenido a los distintos días sin rebasar el límite de los 200 minutos al día y no encuentra ningún problema.

Podemos observar como primero asigna los contenidos pequeños dejando para el final la trilogía de películas con más minutos de duración, asignándoles cada a una a los tres días siguientes para no rebasar la limitación.

Salida del solucionador para problemaE4b.pddl	
step	0: PLANNING BREAKINGBAD01 DIA01
	1: PLANNING BREAKINGBAD02 DIA02
	2: PLANNING BREAKINGBAD03 DIA03
	3: PLANNING NARCOS1 DIA01
	4: PLANNING NARCOS2 DIA02

```

5: PLANNING NARCOS3 DIA03
6: PLANNING POKEMON1 DIA01
7: PLANNING POKEMON2 DIA02
8: PLANNING POKEMON3 DIA03
9: PLANNING JUEGODETRONOS1 DIA01
10: PLANNING JUEGODETRONOS2 DIA02
11: PLANNING JUEGODETRONOS3 DIA03
12: PLANNING ELSENORDELOSANILLOS1 DIA04
13: PLANNING ELSENORDELOSANILLOS2 DIA05
14: PLANNING ELSENORDELOSANILLOS3 DIA06

```

```
time spent: 0.08 seconds total time
```

#### Problema Extensión 4 C (problemaE4c.pddl)

Este es el último juego de pruebas de la cuarta extensión. Con este juego de pruebas, buscamos que el resolutor genere un planning con todas las restricciones y problemas de las demás extensiones, a excepción de la extensión 3 que ambas son exclusivas la una de la otra.

Este juego de pruebas contiene una serie de nueve capítulos con contenido paralelo, otra de seis con también contenido paralelo, una película sin predecesores y una trilogía.

#### Juegos de prueba

```

(:objects
  breakingBad01 breakingBad02 breakingBad03 breakingBad04 breakingBad05
  breakingBad06 breakingBad07 breakingBad08 breakingBad09
  narcos1 narcos2 narcos3 narcos4 narcos5 narcos6
  Jumanji
  ElSenorDeLosAnillos1 ElSenorDeLosAnillos2 ElSenorDeLosAnillos3 - contenido
  dia01 dia02 dia03 dia04 dia05 dia06 dia07 dia08 dia09 dia10 dia11 - dia
)
(:init
  (= (numdia dia01) 01)    (= (minutosOcupados dia01) 0)
  (= (numdia dia02) 02)    (= (minutosOcupados dia02) 0)
  (= (numdia dia03) 03)    (= (minutosOcupados dia03) 0)
  (= (numdia dia04) 04)    (= (minutosOcupados dia04) 0)
  (= (numdia dia05) 05)    (= (minutosOcupados dia05) 0)
  (= (numdia dia06) 06)    (= (minutosOcupados dia06) 0)
  (= (numdia dia07) 07)    (= (minutosOcupados dia07) 0)
  (= (numdia dia08) 08)    (= (minutosOcupados dia08) 0)
  (= (numdia dia09) 09)    (= (minutosOcupados dia09) 0)
  (= (numdia dia10) 10)    (= (minutosOcupados dia10) 0)
  (= (numdia dia11) 11)    (= (minutosOcupados dia11) 0)
  (= (diaAsig breakingBad01) 0) (= (minutos breakingBad01) 45)
  (= (diaAsig breakingBad02) 0) (= (minutos breakingBad02) 45)

```

```

(= (diaAsig breakingBad03) 0) (= (minutos breakingBad03) 45)
(= (diaAsig breakingBad04) 0) (= (minutos breakingBad04) 45)
(= (diaAsig breakingBad05) 0) (= (minutos breakingBad05) 45)
(= (diaAsig breakingBad06) 0) (= (minutos breakingBad06) 45)
(= (diaAsig breakingBad07) 0) (= (minutos breakingBad07) 45)
(= (diaAsig breakingBad08) 0) (= (minutos breakingBad08) 45)
(= (diaAsig breakingBad09) 0) (= (minutos breakingBad09) 45)
(predesor breakingBad02 breakingBad01)
(predesor breakingBad03 breakingBad02)
(paralelo breakingBad04 breakingBad03)
(predesor breakingBad05 breakingBad04)
(predesor breakingBad06 breakingBad05)
(predesor breakingBad07 breakingBad06)
(predesor breakingBad08 breakingBad07)
(predesor breakingBad09 breakingBad08)
(= (diaAsig narcos1) 0) (= (minutos narcos1) 32)
(= (diaAsig narcos2) 0) (= (minutos narcos2) 32)
(= (diaAsig narcos3) 0) (= (minutos narcos3) 32)
(= (diaAsig narcos4) 0) (= (minutos narcos4) 32)
(= (diaAsig narcos5) 0) (= (minutos narcos5) 32)
(= (diaAsig narcos6) 0) (= (minutos narcos6) 32)
(predesor narcos2 narcos1)
(predesor narcos3 narcos2)
(paralelo narcos4 narcos3)
(predesor narcos5 narcos4)
(predesor narcos6 narcos5)
(= (diaAsig Jumanji) 0) (= (minutos Jumanji) 160)
(= (diaAsig ElSenorDeLosAnillos1) 0) (= (minutos ElSenorDeLosAnillos1) 160)
(= (diaAsig ElSenorDeLosAnillos2) 0) (= (minutos ElSenorDeLosAnillos2) 170)
(= (diaAsig ElSenorDeLosAnillos3) 0) (= (minutos ElSenorDeLosAnillos3) 140)
(predesor ElSenorDeLosAnillos2 ElSenorDeLosAnillos1)
(predesor ElSenorDeLosAnillos3 ElSenorDeLosAnillos2)
(quiereVer breakingBad09)
(quiereVer narcos6)
(quiereVer Jumanji)
(quiereVer ElSenorDeLosAnillos3)
)

```

El solucionador da prioridad a los elementos de menor complejidad, asignando primero la película sin conexiones y la trilogía y dejando para el final la serie de seis capítulos, y la de nueve capítulos y con contenido paralelo.

Podemos ver como en todo momento ningún día rebosa los 200 minutos de contenido y cómo va asignado progresivamente entre los días que tiene disponible.

**Salida del solucionador para problemaE4c.pddl**

step 0: PLANNING JUMANJI DIA01  
1: PLANNING ELSENORDELOSANILLOS1 DIA02  
2: PLANNING ELSENORDELOSANILLOS2 DIA03  
3: PLANNING ELSENORDELOSANILLOS3 DIA11  
4: PLANNING NARCOS1 DIA01  
5: PLANNING NARCOS2 DIA09  
6: PLANNING NARCOS3 DIA10  
7: PLANNING NARCOS4 DIA09  
8: PLANNING NARCOS5 DIA10  
9: PLANNING NARCOS6 DIA11  
10: PLANNING BREAKINGBAD01 DIA04  
11: PLANNING BREAKINGBAD02 DIA05  
12: PLANNING BREAKINGBAD03 DIA06  
13: PLANNING BREAKINGBAD04 DIA05  
14: PLANNING BREAKINGBAD05 DIA06  
15: PLANNING BREAKINGBAD06 DIA07  
16: PLANNING BREAKINGBAD07 DIA08  
17: PLANNING BREAKINGBAD08 DIA09  
18: PLANNING BREAKINGBAD09 DIA10  
19: REACH-GOAL

time spent: 132.04 seconds total time