

Práctica de Planificación

Albert Pita, Carlos Gascón, Alejandro Alarcón

2º Cuatrimestre - Curso 2019/2020



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**

Índice general

Introducción	2
El problema	2
Descripción del problema	2
Conocimiento del sistema	2
Resultado	2
Extensiones	3
Dominio	4
Tipos de datos	4
Predicados	5
Funciones	6
Operadores	6
Desarrollo de los modelos	8
Juegos de prueba	10
Juego de Prueba 1	10
Juego de Prueba 2	13
Juego de Prueba 3	16
Juego de Prueba 4	19
Juego de Prueba 5	22
Conclusiones	26

Introducción

El problema

Descripción del problema

La cadena de gimnasios *Couching Potato*, nos ha solicitado que le creamos un programa para organizar en plan de entrenamiento quincenal para sus usuarios. Para llevar esta tarea a cabo se deben respetar un conjunto de restricciones.

Estas restricciones son las siguientes:

Precursores - Un ejercicio es precursor de otro ejercicio cuando el primero se tiene que hacer inmediatamente antes que el segundo, sin poder existir ejercicios intercalados entre ambos.

Preparadores - Un ejercicio es preparador de otro cuando antes de realizar el segundo se debe haber realizado el primero en el mismo día, pudiendo hacer ejercicios intercalados entre ambos.

N+1 - Todos los ejercicios tienen un nivel asignado. Este nivel se puede ir incrementando, hasta llegar a un nivel máximo 10. Sin embargo, solo se podrá ejercitar un ejercicio a nivel N si antes se ha ejercitado a nivel N - 1 en algún día anterior.

Conocimiento del sistema

Ejercicios que se pueden hacer - El sistema debe tener conocimiento de los posibles ejercicios que se pueden realizar en el gimnasio.

Ejercicios precursores - Será necesario conocer las relaciones de precursión entre ejercicios.

Ejercicios preparadores - También será necesario conocer que ejercicios sirven de preparación para otros ejercicios.

Nivel inicial de los ejercicios - El sistema deberá conocer el nivel inicial que tiene el usuario para cada ejercicio que existe en el gimnasio.

Nivel objetivo de los ejercicios - También será necesario conocer el nivel que el usuario espera alcanzar para cada ejercicio durante el plan quincenal.

Resultado

La salida del programa consistirá en un plan de entrenamientos con una duración de 15 días, en el cual, para cada día, se indicará que ejercicios se deben realizar, además del nivel.

Siguiendo este plan, se podrá garantizar que el usuario, partiendo de sus propios niveles para cada ejercicio, es capaz de llegar a sus niveles objetivo durante la duración del plan.

Los ejercicios que se presenten serán los mínimos necesarios para cumplir los objetivos, sin introducirse mejoras ni recomendaciones. Quedará a cargo del usuario la gestión del tiempo restante en caso de existir.

Para todo ejercicio, se realizarán sus ejercicios precursores inmediatamente antes que este en el mismo día.

Para todo ejercicio, se realizarán sus ejercicios preparadores en el mismo día, aunque no necesariamente inmediatamente antes.

Para todo ejercicio con nivel N , se cumplirá que en algún día anterior se ha realizado en mismo ejercicio con nivel $N - 1$.

Extensiones

Problema Básico

En las relaciones de preparación que se plantean, los ejercicios solo pueden tener 0 o 1 preparadores y ningún precursor. Se deberán cumplir las restricciones del nivel de los ejercicios, de modo que el resultado de la ejecución deberá representar un plan de entrenamiento de 15 días durante el cual el usuario pueda subir el nivel de los ejercicios llegando a sus objetivos.

Extensión 1

En esta extensión, los ejercicios podrán tener un número aleatorio de preparadores, pero aún no se introducirán relaciones de precursión. De este modo, al objetivo del problema básico le debemos añadir la dificultad de asignar los ejercicios preparadores antes de los ejercicios a los que estos preparan.

Extensión 2

En esta extensión, a las dificultades que aparecen en la extensión 1 deberemos añadir la aparición de ejercicios precursores, aunque cabe notar que cada ejercicio solo podrá tener como máximo un ejercicio precursor. De este modo, se deberán respetar tanto las restricciones que aparecen con los ejercicios precursores, así como las de los ejercicios preparadores y los niveles asociados a los ejercicios.

Extensión 3

En esta extensión, se deberá limitar el número de ejercicios por día a 6 ejercicios. Debemos tener en cuenta que dentro de esta limitación de los ejercicios por día, se deberán cumplir las restricciones de precursión y preparación.

Extensión 4

En esta extensión, se modificará el conocimiento del sistema, y se asociará un tiempo a cada ejercicio. Además de esto, se modificará la limitación del número máximo de ejercicios por día que plantea la extensión 3 a una limitación de suma de tiempos, que limitará el tiempo de los entrenamientos a 90 minutos, manteniendo, otra vez, todas las restricciones relacionadas con las tipologías de ejercicios y sus relaciones que se plantean anteriormente.

Dominio

Tipos de datos

Ejercicio

Modela cada uno de los ejercicios que se pueden ejercitar en el gimnasio y que el usuario podrá introducir un nivel inicial y un nivel objetivo.

La implementación varía según las extensiones. En el programa básico el ejercicio solo consta de niveles que irán aumentando a medida que se ejercitan los ejercicios. En la extensión 1 se introducen los conceptos de ejercicios preparadores, y en la extensión 2 se añaden las relaciones de precursión. Finalmente, en la extensión 4, se añade un valor de tiempo asociado a cada ejercicio, de modo que la suma de los tiempos de todos los ejercicios pertenecientes a un mismo entrenamiento no deberá superar los 90 minutos.

Nivel

El nivel es un tipo asociado a los ejercicios que funciona como contador. Va asociado a cada uno de los ejercicios, de manera que cada ejercicio puede tener un nivel diferente.

En la entrada del programa, el usuario deberá introducir los niveles de los que parte por cada ejercicio así como los niveles a los que pretende llegar.

Día

Modela los 15 días de los que consta el plan de entrenamiento que se recomendará al usuario. Estos días siguen una estructura de sucesión, es decir, tienen un orden explícito.

Cada entrenamiento ocurre en un día diferente, y el orden de los días es importante para la verificación de los niveles de entrenamiento, para asegurar que nunca se realiza un ejercicio de nivel N si antes no se ha realizado ese mismo ejercicio con un nivel $N - 1$ en un día anterior.

Además, también es importante tener en cuenta de que los ejercicios precursores y preparadores se deben realizar el mismo día (y con anterioridad) que el ejercicio del cuál son precursores o preparadores.

Predicados

(preparador ?p - ejercicio ?e - ejercicio)

El ejercicio p sirve para preparar el ejercicio e . Este predicado se introdujo para representar la relación de *preparador* que puede existir entre ejercicios.

(precursor ?p - ejercicio ?e - ejercicio)

El ejercicio p es precursor del ejercicio e , Este predicado se introdujo para representar la relación de *precursor* que puede existir entre ejercicios.

(diaAnterior ?d1 - dia ?d2 - dia)

El día $d1$ precede al día $d2$.

(diaActual ?d - dia)

El día d es el día en el que en ese momento el planeador está asignando ejercicios.

(nivelPosterior ?n - nivel ?n1 - nivel)

El nivel $n1$ es un nivel anterior al nivel n .

(nivelActual ?e - ejercicio ?n - nivel)

El ejercicio e tiene el nivel actual n .

(subidoNivel ?e - ejercicio)

El ejercicio e ya ha subido de nivel en el día que el planeador está organizando en ese momento.

(ejercicioHoy ?e - ejercicio)

El ejercicio e se ha asignado en el día que el planeador está organizando en ese momento.

(ultimoEjercicio ?e - ejercicio)

El ejercicio e es el último ejercicio que ha sido asignado en el día que el planeador está organizando en ese momento.

Funciones

(nivelEjercicio ?e - ejercicio)

Indica el nivel que tiene el ejercicio e en un momento concreto.

(tiempoEjercicio ?t - ejercicio)

Indica el tiempo que necesita un ejercicio t para su realización.

(tiempoEjercitado)

Indica el tiempo acumulado por los ejercicios asignados en un mismo día.

(contadorEjercicios)

Indica el número de ejercicios que el día que se está organizando en ese momento lleva acumulados.

Operadores

Subir nivel

- `subirNivel(?e - ejercicio ?n - nivel ?n1 - nivel)`
- `subirNivel(?e - ejercicio)`

Este operador permite subir el nivel de un ejercicio. Este operador es necesario para que se cumpla el objetivo, ya que sin él ningún ejercicio podría subir su nivel y por lo tanto el nivel objetivo que marca el usuario para cada ejercicio no se alcanzaría.

Cabe destacar que este operador se especifica con un número diferente de parámetros a lo largo de las diferentes extensiones de la práctica. Esto se debe a que la introducción de flujos y el consecuente cambio en la forma de gestionar los niveles de los ejercicios hicieron que el segundo y tercer parámetros resultaran inútiles.

En la primera de sus dos versiones (correspondiente a las versiones del problema anteriores a la extensión 3), el ejercicio e sube de su nivel actual n al siguiente nivel $n1$. En cambio, en la siguiente versión (correspondiente a las extensiones 3 y 4 del problema) el operador solo necesita el ejercicio e , el cual subirá a su siguiente nivel.

Realizar ejercicio

- `realizarEjercicio(?e - ejercicio)`

Este operador es el encargado de realizar la asignación de un ejercicio con un nivel concreto a un día. Para ello comprueba que el ejercicio no se esté realizando ya ese mismo día, que no este ya a nivel máximo, y que se cumplan todos los requerimientos necesarios.

Para poder asignar un ejercicio a un día, se debe asegurar que todos sus ejercicios preparadores se estén realizando ese mismo día y que en caso de tener un ejercicio precursor, este se debe estar realizando inmediatamente antes.

Además, en el caso de la extensión 3, se debe asegurar que no se superen los 6 ejercicios diarios. Similarmente, en la extensión 4, se debe comprobar que la suma del tiempo asociado al posible ejercicio a asignar más el de todas las asignaciones previas de ese día, no supere los 90 minutos.

Pasar día

- `pasarDia(?d1 ?d2 - dia)`

Este operador está a cargo del secuenciamiento de los días, de modo que esos sigan un orden. Se aplica cada vez que se terminan las acciones posibles para el día que se está tratando.

Dependiendo de la expansión en la que estemos, su ejecución puede ser debida a que no existen más posibles ejercicios que realizar (problema básico y extensiones 1 y 2), que ya se hayan asignado 6 ejercicios (extensión 3) o que se hayan asignado ejercicios que completan el entrenamiento de 90 minutos (extensión 4).

Desarrollo de los modelos

Nivel Básico

En el punto inicial del problema, solo nos tenemos que preocupar de que se cumplan las restricciones de las relaciones de preparación, que además estaban limitadas a un preparador por ejercicio, y de la restricción de los niveles. En este momento fue cuando se crearon las tres acciones que se utilizarían a lo largo de los diferentes estados del problema y lo mismo ocurre con los predicados, de los cuales se crearon la mayoría.

Así pues, los predicados que tenemos en este momento son: *preparador*, *diaAnterior*, *diaActual*, *nivelPosterior*, *nivelActual*, *subidoNivel* y *ejercicioHoy*.

Extensión 1

La diferencia con el nivel básico es que ahora el número de ejercicios preparadores que puede tener un ejercicio asciende a N , lo cual no introduce cambios en nuestro, ya que aunque ahora la restricción de preparación se tiene que comprobar para todos los posibles ejercicios preparatorios que pueda tener un ejercicio esto ya lo hacíamos en la anterior versión.

Extensión 2

Con esta extensión, a todo lo contemplado en las extensiones anteriores se le tiene que añadir las relaciones de precursión, ya que ahora un ejercicio podrá tener de 0 a 1 precursor. Esto implica cambios en el conjunto de los predicados y en la acción de *realizarEjercicio*.

En lo referente a los predicados, se añade uno nuevo, *ultimoEjercicio*. En lo referente a la acción *realizarEjercicio*, se debe modificar la precondition para que también compruebe que la restricción de precursión se cumpla.

Extensión 3

Con esta extensión se produce el cambio más drástico. Ahora se añade que por cada día, el número máximo de ejercicios que se puede realizar es de 6. Además es en esta extensión cuando se introducen los fluentes. Por lo tanto, en esta versión se producen cambios en todos los aspectos del código excepto en el conjunto de predicados, que se queda igual que con la extensión anterior.

Respecto a los fluentes, se han añadido las funciones *nivelEjercicio* y *contadorEjercicios*, con los propósitos de gestionar los niveles y la restricción de los 6 ejercicios por día respectivamente. Para las acciones, modifican las condiciones y efectos de todas las acciones para cumplir la nueva restricción. Además en la acción *subirNivel* el número de parámetros cambia por la nueva gestión con fluentes que se hace de los niveles de los ejercicios.

Extensión 4

Esta expansión es muy similar a la anterior, y de hecho repite la misma idea, solo que esta vez no hay que controlar que se hagan como máximo 6 ejercicios en un día, sino que lo que hay que hacer es limitar que el tiempo acumulado en un día por los diferentes ejercicios realizados no supere los 90 minutos.

Para poder controlar esta restricción se han realizado diversos cambios. Se ha eliminado la función *contadorEjercicios* y se han añadido un par nuevas: *tiempoEjercicio* y *tiempoEjercitado*. En las acciones, se han realizado cambios en la precondition y efecto de *realizarEjercicio* y *pasarDia* para que se respeta la restricción del tiempo.

Juegos de prueba

Juego de Prueba 1

Introducción

Este juego de pruebas pretende mostrar la capacidad del programa para crear las secuencias de ejercicios según sus niveles además de la posibilidad de añadir ejercicios preparadores, aunque cada ejercicio podrá tener como mucho un único ejercicio preparador.

Entrada

```
(define (problem test-00) (:domain planificador)
  (:objects
    ej1 ej2 ej3 ej4 ej5 - ejercicio
    d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 - dia
    n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 - nivel
  )
  (:init
    (preparador ej2 ej1)
    (preparador ej3 ej2)
    (preparador ej5 ej4)
    (diaAnterior d1 d2)
    (diaAnterior d2 d3)
    (diaAnterior d3 d4)
    (diaAnterior d4 d5)
    (diaAnterior d5 d6)
    (diaAnterior d6 d7)
    (diaAnterior d7 d8)
    (diaAnterior d8 d9)
    (diaAnterior d9 d10)
    (diaAnterior d10 d11)
    (diaAnterior d11 d12)
    (diaAnterior d12 d13)
    (diaAnterior d13 d14)
    (diaAnterior d14 d15)
    (diaActual d1)
    (nivelPosterior n1 n2)
    (nivelPosterior n2 n3)
```

```

(nivelPosterior n3 n4)
(nivelPosterior n4 n5)
(nivelPosterior n5 n6)
(nivelPosterior n6 n7)
(nivelPosterior n7 n8)
(nivelPosterior n8 n9)
(nivelPosterior n9 n10)
(nivelActual ej1 n1)
(nivelActual ej2 n1)
(nivelActual ej3 n1)
(nivelActual ej4 n1)
(nivelActual ej5 n1)
)
(:goal (and
  (nivelActual ej1 n5)
  (nivelActual ej2 n5)
  (nivelActual ej3 n5)
  (nivelActual ej4 n5)
  (nivelActual ej5 n5)
)
)
)

```

Salida

```

0: REALIZAREJERCICIO EJ3
1: REALIZAREJERCICIO EJ2
2: REALIZAREJERCICIO EJ1
3: REALIZAREJERCICIO EJ5
4: REALIZAREJERCICIO EJ4
5: SUBIRNIVEL EJ1 N1 N2
6: SUBIRNIVEL EJ2 N1 N2
7: SUBIRNIVEL EJ3 N1 N2
8: SUBIRNIVEL EJ4 N1 N2
9: SUBIRNIVEL EJ5 N1 N2
10: PASARDIA D1 D2
11: REALIZAREJERCICIO EJ3

```

12: REALIZAREJERCICIO EJ2
13: REALIZAREJERCICIO EJ1
14: REALIZAREJERCICIO EJ5
15: REALIZAREJERCICIO EJ4
16: SUBIRNIVEL EJ1 N2 N3
17: SUBIRNIVEL EJ2 N2 N3
18: SUBIRNIVEL EJ3 N2 N3
19: SUBIRNIVEL EJ4 N2 N3
20: SUBIRNIVEL EJ5 N2 N3
21: PASARDIA D2 D3
22: REALIZAREJERCICIO EJ3
23: REALIZAREJERCICIO EJ2
24: REALIZAREJERCICIO EJ1
25: REALIZAREJERCICIO EJ5
26: REALIZAREJERCICIO EJ4
27: SUBIRNIVEL EJ1 N3 N4
28: SUBIRNIVEL EJ2 N3 N4
29: SUBIRNIVEL EJ3 N3 N4
30: SUBIRNIVEL EJ4 N3 N4
31: SUBIRNIVEL EJ5 N3 N4
32: PASARDIA D3 D4
33: REALIZAREJERCICIO EJ3
34: REALIZAREJERCICIO EJ2
35: REALIZAREJERCICIO EJ1
36: REALIZAREJERCICIO EJ5
37: REALIZAREJERCICIO EJ4
38: SUBIRNIVEL EJ1 N4 N5
39: SUBIRNIVEL EJ2 N4 N5
40: SUBIRNIVEL EJ3 N4 N5
41: SUBIRNIVEL EJ4 N4 N5
42: SUBIRNIVEL EJ5 N4 N5

Justificación

Como podemos ver, en la salida se respetan las condiciones impuestas por la regla de preparador, de modo que los ejercicios preparadores se realizarán antes que los ejercicios que estos preparan.

Además, se puede observar que en todo momento los ejercicios de un nivel se realizan poste-

riormente a esos mismos ejercicios con niveles inferiores.

Juego de Prueba 2

Introducción

En este juego de prueba se pretende mostrar los cambios realizados que permiten el correcto funcionamiento del programa para la extensión 1, que permite que un ejercicio tenga un número aleatorio de preparadores.

Entrada

```
(define (problem test-01) (:domain planificador)
  (:objects
    ej1 ej2 ej3 ej4 ej5 - ejercicio
    d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 - dia
    n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 - nivel
  )
  (:init
    (preparador ej1 ej2)
    (preparador ej3 ej2)
    (preparador ej4 ej2)
    (preparador ej3 ej4)
    (diaAnterior d1 d2)
    (diaAnterior d2 d3)
    (diaAnterior d3 d4)
    (diaAnterior d4 d5)
    (diaAnterior d5 d6)
    (diaAnterior d6 d7)
    (diaAnterior d7 d8)
    (diaAnterior d8 d9)
    (diaAnterior d9 d10)
    (diaAnterior d10 d11)
    (diaAnterior d11 d12)
    (diaAnterior d12 d13)
    (diaAnterior d13 d14)
    (diaAnterior d14 d15)
    (diaActual d1)
    (nivelPosterior n1 n2)
```

```

(nivelPosterior n2 n3)
(nivelPosterior n3 n4)
(nivelPosterior n4 n5)
(nivelPosterior n5 n6)
(nivelPosterior n6 n7)
(nivelPosterior n7 n8)
(nivelPosterior n8 n9)
(nivelPosterior n9 n10)
(nivelActual ej1 n1)
(nivelActual ej2 n1)
(nivelActual ej3 n1)
(nivelActual ej4 n1)
(nivelActual ej5 n1)
)
(:goal (and
  (nivelActual ej1 n5)
  (nivelActual ej2 n5)
  (nivelActual ej3 n5)
  (nivelActual ej4 n5)
  (nivelActual ej5 n5)
)
)
)

```

Salida

```

0: REALIZAREJERCICIO EJ1
1: REALIZAREJERCICIO EJ3
2: REALIZAREJERCICIO EJ4
3: REALIZAREJERCICIO EJ2
4: REALIZAREJERCICIO EJ5
5: SUBIRNIVEL EJ1 N1 N2
6: SUBIRNIVEL EJ2 N1 N2
7: SUBIRNIVEL EJ3 N1 N2
8: SUBIRNIVEL EJ4 N1 N2
9: SUBIRNIVEL EJ5 N1 N2
10: PASARDIA D1 D2

```

- 11: REALIZAREJERCICIO EJ1
- 12: REALIZAREJERCICIO EJ3
- 13: REALIZAREJERCICIO EJ4
- 14: REALIZAREJERCICIO EJ2
- 15: REALIZAREJERCICIO EJ5
- 16: SUBIRNIVEL EJ1 N2 N3
- 17: SUBIRNIVEL EJ2 N2 N3
- 18: SUBIRNIVEL EJ3 N2 N3
- 19: SUBIRNIVEL EJ4 N2 N3
- 20: SUBIRNIVEL EJ5 N2 N3
- 21: PASARDIA D2 D3
- 22: REALIZAREJERCICIO EJ1
- 23: REALIZAREJERCICIO EJ3
- 24: REALIZAREJERCICIO EJ4
- 25: REALIZAREJERCICIO EJ2
- 26: REALIZAREJERCICIO EJ5
- 27: SUBIRNIVEL EJ1 N3 N4
- 28: SUBIRNIVEL EJ2 N3 N4
- 29: SUBIRNIVEL EJ3 N3 N4
- 30: SUBIRNIVEL EJ4 N3 N4
- 31: SUBIRNIVEL EJ5 N3 N4
- 32: PASARDIA D3 D4
- 33: REALIZAREJERCICIO EJ5
- 34: REALIZAREJERCICIO EJ1
- 35: REALIZAREJERCICIO EJ3
- 36: REALIZAREJERCICIO EJ4
- 37: REALIZAREJERCICIO EJ2
- 38: SUBIRNIVEL EJ1 N4 N5
- 39: SUBIRNIVEL EJ2 N4 N5
- 40: SUBIRNIVEL EJ3 N4 N5
- 41: SUBIRNIVEL EJ4 N4 N5
- 42: SUBIRNIVEL EJ5 N4 N5

Justificación

En la salida de este juego de prueba se puede observar como se respetan las limitaciones de los ejercicios preparadores, realizándose estos en el mismo día pero con anterioridad al ejercicio

que preparan. Podemos ver que un ejercicio puede tener más de un ejercicio preparador.

Juego de Prueba 3

Introducción

En este juego de pruebas se pueden apreciar el correcto funcionamiento del programa para la extensión 2, donde aparecen los ejercicios precursores, que se deberán realizar inmediatamente antes que sus consecuentes.

Entrada

```
(define (problem test-00) (:domain planificador)
  (:objects
    ej1 ej2 ej3 ej4 ej5 - ejercicio
    d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 - dia
    n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 - nivel
  )
  (:init
    (precursor ej3 ej2)
    (precursor ej2 ej1)
    (precursor ej5 ej4)
    (preparador ej4 ej1)
    (diaAnterior d1 d2)
    (diaAnterior d2 d3)
    (diaAnterior d3 d4)
    (diaAnterior d4 d5)
    (diaAnterior d5 d6)
    (diaAnterior d6 d7)
    (diaAnterior d7 d8)
    (diaAnterior d8 d9)
    (diaAnterior d9 d10)
    (diaAnterior d10 d11)
    (diaAnterior d11 d12)
    (diaAnterior d12 d13)
    (diaAnterior d13 d14)
    (diaAnterior d14 d15)
    (diaActual d1)
    (nivelPosterior n1 n2)
```

```

(nivelPosterior n2 n3)
(nivelPosterior n3 n4)
(nivelPosterior n4 n5)
(nivelPosterior n5 n6)
(nivelPosterior n6 n7)
(nivelPosterior n7 n8)
(nivelPosterior n8 n9)
(nivelPosterior n9 n10)
(nivelActual ej1 n1)
(nivelActual ej2 n1)
(nivelActual ej3 n1)
(nivelActual ej4 n1)
(nivelActual ej5 n1)
)
(:goal (and
  (nivelActual ej1 n5)
  (nivelActual ej2 n5)
  (nivelActual ej3 n5)
  (nivelActual ej4 n5)
  (nivelActual ej5 n5)
)
)
)

```

Salida

```

0: REALIZAREJERCICIO EJ3
1: REALIZAREJERCICIO EJ2
2: SUBIRNIVEL EJ2 N1 N2
3: SUBIRNIVEL EJ3 N1 N2
4: REALIZAREJERCICIO EJ5
5: SUBIRNIVEL EJ5 N1 N2
6: REALIZAREJERCICIO EJ4
7: SUBIRNIVEL EJ4 N1 N2
8: PASARDIA D1 D2
9: REALIZAREJERCICIO EJ3
10: REALIZAREJERCICIO EJ2

```

11: SUBIRNIVEL EJ2 N2 N3
12: SUBIRNIVEL EJ3 N2 N3
13: REALIZAREJERCICIO EJ5
14: SUBIRNIVEL EJ5 N2 N3
15: REALIZAREJERCICIO EJ4
16: SUBIRNIVEL EJ4 N2 N3
17: PASARDIA D2 D3
18: REALIZAREJERCICIO EJ3
19: REALIZAREJERCICIO EJ2
20: SUBIRNIVEL EJ2 N3 N4
21: SUBIRNIVEL EJ3 N3 N4
22: REALIZAREJERCICIO EJ5
23: SUBIRNIVEL EJ5 N3 N4
24: REALIZAREJERCICIO EJ4
25: SUBIRNIVEL EJ4 N3 N4
26: PASARDIA D3 D4
27: REALIZAREJERCICIO EJ3
28: REALIZAREJERCICIO EJ2
29: SUBIRNIVEL EJ2 N4 N5
30: SUBIRNIVEL EJ3 N4 N5
31: REALIZAREJERCICIO EJ5
32: REALIZAREJERCICIO EJ4
33: SUBIRNIVEL EJ4 N4 N5
34: SUBIRNIVEL EJ5 N4 N5
35: PASARDIA D4 D5
36: REALIZAREJERCICIO EJ5
37: REALIZAREJERCICIO EJ4
38: REALIZAREJERCICIO EJ3
39: REALIZAREJERCICIO EJ2
40: REALIZAREJERCICIO EJ1
41: SUBIRNIVEL EJ1 N1 N2
42: PASARDIA D5 D6
43: REALIZAREJERCICIO EJ5
44: REALIZAREJERCICIO EJ4
45: REALIZAREJERCICIO EJ3
46: REALIZAREJERCICIO EJ2

47: REALIZAREJERCICIO EJ1
 48: SUBIRNIVEL EJ1 N2 N3
 49: PASARDIA D6 D7
 50: REALIZAREJERCICIO EJ5
 51: REALIZAREJERCICIO EJ4
 52: REALIZAREJERCICIO EJ3
 53: REALIZAREJERCICIO EJ2
 54: REALIZAREJERCICIO EJ1
 55: SUBIRNIVEL EJ1 N3 N4
 56: PASARDIA D7 D8
 57: REALIZAREJERCICIO EJ5
 58: REALIZAREJERCICIO EJ4
 59: REALIZAREJERCICIO EJ3
 60: REALIZAREJERCICIO EJ2
 61: REALIZAREJERCICIO EJ1
 62: SUBIRNIVEL EJ1 N4 N5

Justificación

En la salida de este juego de prueba se puede observar como los ejercicios se realizan inmediatamente después que sus precursores, en el caso de tenerlos. Además se siguen manteniendo todas las limitaciones establecidas en las extensiones previas.

Juego de Prueba 4

Introducción

Este juego de prueba pretende mostrar las modificaciones que se plantean para la extensión 3, que plantea la limitación de los ejercicios por día.

Entrada

```

(define (problem test-03) (:domain planificador)
  (:objects
    ej1 ej2 ej3 ej4 ej5 ej6 ej7 - ejercicio
    d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 - dia
  )
  (:init
    (precursor ej3 ej2)
  )

```

```

(preparador ej1 ej2)
(diaAnterior d1 d2)
(diaAnterior d2 d3)
(diaAnterior d3 d4)
(diaAnterior d4 d5)
(diaAnterior d5 d6)
(diaAnterior d6 d7)
(diaAnterior d7 d8)
(diaAnterior d8 d9)
(diaAnterior d9 d10)
(diaAnterior d10 d11)
(diaAnterior d11 d12)
(diaAnterior d12 d13)
(diaAnterior d13 d14)
(diaAnterior d14 d15)
(diaActual d1)
(= (nivelEjercicio ej1) 1)
(= (nivelEjercicio ej2) 1)
(= (nivelEjercicio ej3) 1)
(= (nivelEjercicio ej4) 1)
(= (nivelEjercicio ej5) 1)
(= (nivelEjercicio ej6) 1)
(= (nivelEjercicio ej7) 1)
(= (contadorEjercicios) 0)
)
(:goal (and
  (= (nivelEjercicio ej1) 3)
  (= (nivelEjercicio ej2) 4)
  (= (nivelEjercicio ej3) 3)
  (= (nivelEjercicio ej4) 3)
  (= (nivelEjercicio ej5) 3)
  (= (nivelEjercicio ej6) 3)
  (= (nivelEjercicio ej7) 3)
))
)
)

```

Salida

0: REALIZAREJERCICIO EJ1
1: REALIZAREJERCICIO EJ3
2: REALIZAREJERCICIO EJ2
3: REALIZAREJERCICIO EJ4
4: REALIZAREJERCICIO EJ6
5: REALIZAREJERCICIO EJ5
6: SUBIRNIVEL EJ5
7: SUBIRNIVEL EJ6
8: SUBIRNIVEL EJ4
9: SUBIRNIVEL EJ2
10: SUBIRNIVEL EJ3
11: SUBIRNIVEL EJ1
12: PASARDIA D1 D2
13: REALIZAREJERCICIO EJ1
14: REALIZAREJERCICIO EJ3
15: REALIZAREJERCICIO EJ2
16: REALIZAREJERCICIO EJ4
17: REALIZAREJERCICIO EJ5
18: SUBIRNIVEL EJ5
19: SUBIRNIVEL EJ4
20: SUBIRNIVEL EJ3
21: SUBIRNIVEL EJ1
22: REALIZAREJERCICIO EJ6
23: SUBIRNIVEL EJ6
24: SUBIRNIVEL EJ2
25: PASARDIA D2 D3
26: REALIZAREJERCICIO EJ1
27: REALIZAREJERCICIO EJ3
28: REALIZAREJERCICIO EJ2
29: REALIZAREJERCICIO EJ7
30: SUBIRNIVEL EJ2
31: SUBIRNIVEL EJ7
32: PASARDIA D3 D4
33: REALIZAREJERCICIO EJ7

Justificación

En la salida de este juego de prueba se puede observar como para cada día existe un máximo de 6 entrenamientos.

Juego de Prueba 5

Introducción

En este juego de prueba queremos comprobar que se respeta la restricción introducida por la extensión 4. Esta restricción indica que la cantidad de tiempo acumulado por los ejercicios que se realizan un mismo día no debe exceder los 90 minutos.

Entrada

```
(define (problem test-04) (:domain planificador)
  (:objects
    ej1 ej2 ej3 ej4 ej5 ej6 ej7 - ejercicio
    d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 - dia
  )
  (:init
    (precursor ej3 ej2)
    (preparador ej1 ej2)
    (diaAnterior d1 d2)
    (diaAnterior d2 d3)
    (diaAnterior d3 d4)
    (diaAnterior d4 d5)
    (diaAnterior d5 d6)
    (diaAnterior d6 d7)
    (diaAnterior d7 d8)
    (diaAnterior d8 d9)
    (diaAnterior d9 d10)
    (diaAnterior d10 d11)
    (diaAnterior d11 d12)
    (diaAnterior d12 d13)
    (diaAnterior d13 d14)
    (diaAnterior d14 d15)
```

```

(diaActual d1)
(= (nivelEjercicio ej1) 1)
(= (nivelEjercicio ej2) 1)
(= (nivelEjercicio ej3) 1)
(= (nivelEjercicio ej4) 1)
(= (nivelEjercicio ej5) 1)
(= (nivelEjercicio ej6) 1)
(= (nivelEjercicio ej7) 1)
(= (tiempoEjercicio ej1) 20)
(= (tiempoEjercicio ej2) 20)
(= (tiempoEjercicio ej3) 10)
(= (tiempoEjercicio ej4) 10)
(= (tiempoEjercicio ej5) 10)
(= (tiempoEjercicio ej6) 10)
(= (tiempoEjercicio ej7) 20)
(= (tiempoEjercitado) 0)
)
(:goal (and
  (= (nivelEjercicio ej1) 5)
  (= (nivelEjercicio ej2) 5)
  (= (nivelEjercicio ej3) 5)
  (= (nivelEjercicio ej4) 5)
  (= (nivelEjercicio ej5) 5)
  (= (nivelEjercicio ej6) 5)
  (= (nivelEjercicio ej7) 5)
))
)
)

```

Salida

```

0: REALIZAREJERCICIO EJ1
1: REALIZAREJERCICIO EJ3
2: REALIZAREJERCICIO EJ2
3: REALIZAREJERCICIO EJ4
4: REALIZAREJERCICIO EJ6
5: REALIZAREJERCICIO EJ5

```


6: SUBIRNIVEL EJ5
7: SUBIRNIVEL EJ6
8: SUBIRNIVEL EJ4
9: SUBIRNIVEL EJ2
10: SUBIRNIVEL EJ3
11: SUBIRNIVEL EJ1
12: PASARDIA D1 D2
13: REALIZAREJERCICIO EJ1
14: REALIZAREJERCICIO EJ3
15: REALIZAREJERCICIO EJ2
16: REALIZAREJERCICIO EJ4
17: REALIZAREJERCICIO EJ6
18: REALIZAREJERCICIO EJ5
19: SUBIRNIVEL EJ5
20: SUBIRNIVEL EJ6
21: SUBIRNIVEL EJ4
22: SUBIRNIVEL EJ2
23: SUBIRNIVEL EJ3
24: SUBIRNIVEL EJ1
25: PASARDIA D2 D3
26: REALIZAREJERCICIO EJ1
27: REALIZAREJERCICIO EJ3
28: REALIZAREJERCICIO EJ2
29: REALIZAREJERCICIO EJ4
30: REALIZAREJERCICIO EJ6
31: REALIZAREJERCICIO EJ5
32: SUBIRNIVEL EJ5
33: SUBIRNIVEL EJ6
34: SUBIRNIVEL EJ4
35: SUBIRNIVEL EJ2
36: SUBIRNIVEL EJ3
37: SUBIRNIVEL EJ1
38: PASARDIA D3 D4
39: REALIZAREJERCICIO EJ1
40: REALIZAREJERCICIO EJ3
41: REALIZAREJERCICIO EJ2

42: REALIZAREJERCICIO EJ4
43: REALIZAREJERCICIO EJ5
44: SUBIRNIVEL EJ5
45: SUBIRNIVEL EJ4
46: SUBIRNIVEL EJ2
47: SUBIRNIVEL EJ3
48: SUBIRNIVEL EJ1
49: REALIZAREJERCICIO EJ6
50: SUBIRNIVEL EJ6
51: PASARDIA D4 D5
52: REALIZAREJERCICIO EJ7
53: SUBIRNIVEL EJ7
54: PASARDIA D5 D6
55: REALIZAREJERCICIO EJ7
56: SUBIRNIVEL EJ7
57: PASARDIA D6 D7
58: REALIZAREJERCICIO EJ7
59: SUBIRNIVEL EJ7
60: PASARDIA D7 D8
61: REALIZAREJERCICIO EJ7
62: SUBIRNIVEL EJ7

Justificación

Como se puede observar en el output, esta restricción se mantiene para todos los días del plan de entrenamiento quincenal.

Conclusiones

En esta práctica hemos resuelto un problema de planificación, modelizándolo para usar un sistema de planificación automática en su resolución.

Aunque se ha tratado con una práctica simple con un dominio limitado, nos ha permitido ver el tipo de problemas sobre los que aplicar técnicas de planificación, la capacidad expresiva de lenguajes formales de definición de dominios y el potencial de Fast Forward y del Metric Fast Forward.

Para ello, hemos analizado el problema del plan de entrenamiento físico quincenal, en el que intervienen varias restricciones, tipos de objetos, predicados y acciones a realizar. Hemos desarrollado la práctica siguiendo el desarrollo incremental guiado por diferentes extensiones del problema.

Con todo esto, hemos podido comprobar que este tipo de sistemas de planificación automática son capaces de ofrecer al usuario soluciones correctas en un tiempo razonable.