

[width=0.5]grafica.png

Figure 1: Enter Caption



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Proyecto de IA: Planificación

Práctica de Planificación

Santiago Cervera Pérez
Nicolás Gonzalo Longueria Millicay
Víctor Hernández Barragán

Contents

1	Introducción	2
2	Dominio	2
2.1	Nivel básico	2
2.2	Extensión 1	3
2.3	Extensión 2	3
2.4	Extensión 3	4
3	Modelado del problema	5
3.1	Nivel básico	5
3.2	Extensión 1	5
3.3	Extensión 2	6
3.4	Extensión 3	6
4	Desarrollo de los modelos	6
5	Juegos de prueba	6
5.1	Juego 1 - Nivel Básico	7
5.2	Juego 2 - Nivel Básico	8
5.3	Juego 3 - Extensión 1	10
5.4	Juego 4 - Extensión 1	11
5.5	Juego 5 - Extensión 2	12
5.6	Juego 6 - Extension 2	14
5.7	Juego 7 - Extensión 3	16
5.8	Juego 8 - Extensión 3	18
6	Tiempo de ejecución	20
7	Conclusiones	21

1 Introducción

Esta práctica tiene el fin de desarrollar un Planning Domain Definition Language (PDDDL). Concretamente se nos pide crear un plan de lectura mensual basándose en los libros que el usuario ya ha leído y los que planea leer en el próximo año. El objetivo es proporcionar recomendaciones de lectura mensual que equilibren el número de páginas a leer. El plan debe tener en cuenta las dependencias entre libros, como los libros predecesores (necesarios para disfrutar mejor de otro libro) y los libros paralelos (historias que ocurren simultáneamente). Para lograr esto, el sistema debe tener conocimiento de:

- El catálogo de libros.
- Libros predecesores a un libro.
- Libros paralelos a un libro.
- Libros que el usuario ya ha leído.
- Libros que el usuario desea leer.

El resultado esperado es un plan de lectura que refleje los libros mínimos que el usuario debe leer, indicando en qué mes debe leer cada uno. El plan garantiza que, para cada libro, sus predecesores se lean en meses anteriores. Además, asegura que, para todos los libros del plan, sus paralelos se lean en el mismo mes o en el mes anterior (o en el mes siguiente para relaciones simétricas).

El propósito de esta práctica es abordar un problema que requiere un proceso de planificación para su resolución.

Después de obtener la solución del problema junto con sus diversas extensiones, procederemos a generar problemas adicionales y exploraremos la escalabilidad en términos de tiempo que presentan las diferentes configuraciones de parámetros.

2 Dominio

2.1 Nivel básico

- **Variables**
 - Mes: Representa un mes del año.
 - Libro: Representa un libro.
- **Predicados**
 - (predecesor ?l1 - libro ?l2 - libro): Representa la relación de presidencia entre libros, donde l1 precede a l2.
 - (leído ?l1 - libro): Representa si se ha leído el libro l1.
 - (quiere ?l1 - libro): Representa si el usuario desea leer el libro l1.
 - (asignado ?l1 - libro ?m1 - mes): Representa en que mes se debería leer dicho libro.

- (mes-anterior ?m1 - mes ?m2 - mes): Representa una relación de presidencia entre meses, donde, m1 precede a m2.

- **Acciones**

- leer: Operador que, dado un libro y un mes, se encarga de crear las asignaciones de libros - mes (*asignado*). Además de tener en cuenta la restricción de libros *predecesores*, es decir, que un libro que tenga un predecesor estará asignado a un mes posterior al mismo.

2.2 Extensión 1

No hay ningún cambio de dominio entre el nivel básico y la extensión 1, más adelante se desarrollará el porqué.

2.3 Extensión 2

- **Variables**

- Mes: Representa un mes del año.
- Libro: Representa un libro.

- **Predicados**

- (predecesor ?l1 - libro ?l2 - libro): Representa la relación de presidencia entre libros, donde l1 precede a l2.
- (paralelo ?l1 - libro ?l2 - libro): Representa la relación de paralelo entre libros, como es una relación no jerárquica no importa la permutación entre l1 y l2.
- (leído ?l1 - libro): Representa si se ha leído el libro l1.
- (quiere ?l1 - libro): Representa si el usuario desea leer el libro l1.
- (asignado ?l1 - libro ?m1 - mes): Representa en que mes se debería leer dicho libro.
- (mes-anterior ?m1 - mes ?m2 - mes): Representa una relación de presidencia entre meses, donde, m1 precede a m2.
- (mes-anterior-p ?m1 - mes ?m2 - mes): Representa una relación entre meses donde m1 es el mes anterior inmediato de m2, por ejemplo, m1 enero m2 febrero.

- **Acciones**

- leer: Operador que, dado un libro y un mes, se encarga de crear las asignaciones de libros - mes (*asignado*). Además de tener en cuenta la restricción de libros *predecesores*, es decir, que un libro que tenga un predecesor estará asignado a un mes posterior al mismo. También se tienen en cuenta la restricción de libros paralelos.
- leer-pa: Operador que, dado un libro y un mes, se encarga de crear las asignaciones de libros - mes (*asignado*). Con la restricción de encargarse únicamente de la asignación de los libros paralelos que posean predecesores que ya han sido asignados, pero no sus contrapartes paralelas.

2.4 Extensión 3

- **Variables**

- Mes: Representa un mes del año.
- Libro: Representa un libro.

- **Predicados**

- (predecesor ?l1 - libro ?l2 - libro): Representa la relación de presidencia entre libros, donde l1 precede a l2.
- (paralelo ?l1 - libro ?l2 - libro): Representa la relación de paralelo entre libros, como es una relación no jerárquica no importa la permutación entre l1 y l2.
- (leído ?l1 - libro): Representa si se ha leído el libro l1.
- (quiere ?l1 - libro): Representa si el usuario desea leer el libro l1.
- (asignado ?l1 - libro ?m1 - mes): Representa en que mes se debería leer dicho libro.
- (mes-anterior ?m1 - mes ?m2 - mes): Representa una relación de presidencia entre meses, donde, m1 precede a m2.
- (mes-anterior-p ?m1 - mes ?m2 - mes): Representa una relación entre meses donde m1 es el mes anterior inmediato de m2, por ejemplo, m1 enero m2 febrero.

- **Funciones**

- (pag-por-mes ?m - mes): El fluente que controla la cantidad de páginas que han sido asignados en un mes m.
- (pag-libro ?l - libro): Fluente estático que indicia la cantidad de páginas del libro l.

- **Acciones**

- leer: Operador que, dado un libro y un mes, se encarga de crear las asignaciones de libros - mes (*asignado*). Además de tener en cuenta la restricción de libros *predecesores*, es decir, que un libro que tenga un predecesor estará asignado a un mes posterior al mismo. También se tienen en cuenta la restricción de libros paralelos y el límite de páginas mensuales.
- leer-pa: Operador que, dado un libro y un mes, se encarga de crear las asignaciones de libros - mes (*asignado*). Con la restricción de encargarse únicamente de la asignación de los libros paralelos que posean predecesores que ya han sido asignados, pero no sus contrapartes paralelas. También controla el límite de páginas mensuales.

Con todo esto podemos argumentar y justificar que nuestro planificador engloba todo el dominio descrito por el problema, con sus respectivas restricciones y precondiciones.

3 Modelado del problema

Para abordar las diversas problemáticas, presentamos las siguientes representaciones modeladas. Dado que se trata de una resolución incremental del problema, en cada extensión solo detallaremos las modificaciones con respecto a la anterior y qué elementos del problema empleamos para resolver dicha extensión.

3.1 Nivel básico

- **Objetos**

- Mes: Representa un mes del año, en todos los problemas el objeto se limitarán a ser: *enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre y diciembre*.
- Libro: Representa un libro.

- **Estado inicial**

- Se instancias todos los meses del año y se crean las relaciones de meses precedentes.
- Se establecen los libros que el usuario desea leer y las relaciones de precedencias entre libros.

- **Estado final**

```
(:goal
  (or
    (forall (?l - libro) (or
      (leido ?l)
      (imply (not (leido ?l)) (not (quiere ?l)))
    ))
  )
)
```

- Lo que se propone en el código anterior es que programa finalizará (de ser posible) luego de haberle signado un mes de lectura a todos los libros que el usuario quiera leer.

3.2 Extensión 1

Como se menciona en el dominio, respecto al dominio entre el nivel básico y extensión 1 no hubo ningún cambio, pero en cuanto al modelaje simplemente varia una pequeña cosa de la solución inicial. Ya que en la extensión 1 en vez de que lo libros tengan un único predecesor, ahora estos pueden tener N predecesores. Por lo que en la solución inicial en vez de referirnos a relaciones únicas entre libros predecesores ahora pueden ser N-arias.

3.3 Extensión 2

En esta extensión se agregan las relaciones de libros paralelos por lo que nuestro nuevo estado inicial también las inicializa.

Estado inicial

- Se instancias todos los meses del año y se crean las relaciones de meses precedentes.
- Se establecen los libros que el usuario desea leer, las relaciones de precedencias entre libros y las relaciones de paralelo.

3.4 Extensión 3

Por último, en esta nueva extensión se nos pide agregar el concepto de páginas a los libros para establecer un límite de páginas mensuales, todo ello también se tendrá que agregar al estado inicial de nuestro problema.

Estado inicial

- Se instancias todos los meses del año y se crean las relaciones de meses precedentes.
- Se establecen los libros que el usuario desea leer, las relaciones de precedencias entre libros y las relaciones de paralelo.
- Se inicializan las páginas asignadas a cada mes con un 0.
- Se establece la cantidad de páginas de cada libro.

4 Desarrollo de los modelos

En el proceso de desarrollo, hemos adherido a la metodología detallada en el enunciado, realizando extensiones de manera incremental, partiendo de un nivel inicial muy básico. En cada extensión, hemos incorporado distintas funcionalidades que nos han permitido cumplir con los requisitos establecidos en el enunciado.

Para evaluar la eficacia de los diversos modelos, hemos creado problemas específicos que han demostrado el correcto funcionamiento de estos. Posteriormente, hemos implementado un generador de problemas aleatorios para cada modelo, facilitando así el estudio de sus diferentes comportamientos y tiempos de ejecución.

5 Juegos de prueba

Se omitirá la parte de inicializar la relación de meses anteriores, ya que ocupan mucho espacio. Todos los juegos de prueba siguientes fueron creados con el **generador.cc**, básicamente nuestro generador crea varios grafos dirigidos acíclicos aleatorios (controlados por unos parámetros de entrada), y cada nodo representa un libro y cada arista una relación, que puede ser de predecesor o de paralelo, dependiendo el modelo.

5.1 Juego 1 - Nivel Básico

- Input

```
(define (problem problema)

  (:domain libros)

  (:objects
    diciembre noviembre octubre septiembre agosto julio junio mayo
    abril marzo febrero enero - mes
    libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
    libro_8 - libro
  )
  (:init
    **Inizializacion de relacion meses anteriores
    (quiere libro_0)
    (quiere libro_1)
    (quiere libro_2)
    (quiere libro_3)
    (quiere libro_4)
    (quiere libro_5)
    (quiere libro_6)
    (quiere libro_7)
    (quiere libro_8)
    (predecesor libro_0 libro_3)
    (predecesor libro_1 libro_2)
  )

  (:goal
    (forall (?l - libro) (or (leido ?l) (imply (not (leido ?l)) (not
      (quiere ?l))))))
  ))
```

- Output

step	0: LEER LIBRO_0 ENERO
	1: LEER LIBRO_1 ENERO
	2: LEER LIBRO_2 FEBRERO
	3: LEER LIBRO_3 FEBRERO
	4: LEER LIBRO_4 ENERO
	5: LEER LIBRO_5 ENERO
	6: LEER LIBRO_6 ENERO
	7: LEER LIBRO_7 ENERO
	8: LEER LIBRO_8 ENERO
	9: REACH-GOAL

- Visualización

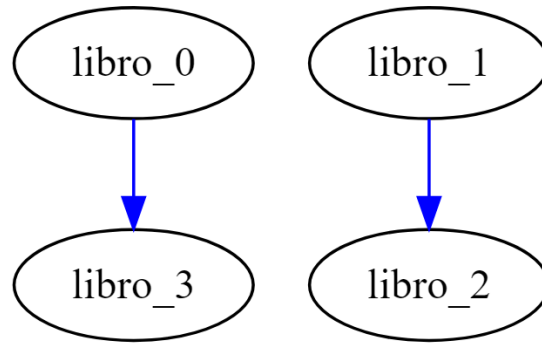


Figure 2: Visualización de la jerarquía de libros del juego de pruebas 1. (Azul = predecesor)

- **Justificación**

Este juego de pruebas muestra como un problema con libros con varios predecesores se crea un plan de lectura respetando las restricciones establecidas hasta el momento.

5.2 Juego 2 - Nivel Básico

- **Input**

```

(define (problem problema)

  (:domain libros)

  (:objects
    diciembre noviembre octubre septiembre agosto julio junio mayo
    abril marzo febrero enero - mes
    libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
    libro_8 - libro
  )

  (:init
    **Inizializacion de relacion meses anteriores
    (quiere libro_0)
    (quiere libro_1)
    (quiere libro_2)
    (quiere libro_3)
    (quiere libro_4)
    (quiere libro_5)
    (quiere libro_6)
    (quiere libro_7)
    (quiere libro_8)
    (predecesor libro_0 libro_1)
    (predecesor libro_1 libro_2)
    (predecesor libro_2 libro_3)
    (predecesor libro_3 libro_4)
    (predecesor libro_4 libro_5)
    (predecesor libro_5 libro_6)
    (predecesor libro_6 libro_7)
  )

```

```

    (predecesor libro_7 libro_8)
  )

  (:goal
    (forall (?l - libro) (or (leido ?l) (imply (not (leido ?l)) (not
      (quiere ?l))))))
  ))

```

• Output

step	0: LEER LIBRO_0 ENERO
	1: LEER LIBRO_1 FEBRERO
	2: LEER LIBRO_2 MARZO
	3: LEER LIBRO_3 ABRIL
	4: LEER LIBRO_4 MAYO
	5: LEER LIBRO_5 JUNIO
	6: LEER LIBRO_6 JULIO
	7: LEER LIBRO_7 AGOSTO
	8: LEER LIBRO_8 SEPTIEMBRE
	9: REACH-GOAL

• Visualización

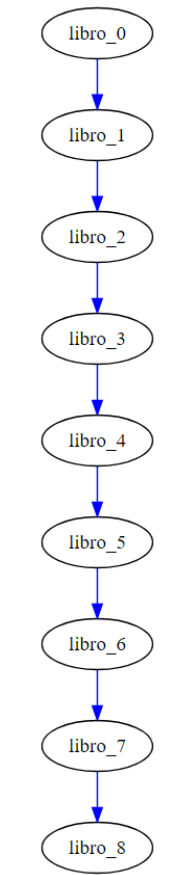


Figure 3: Visualización de la jerarquía de libros del juego de pruebas 2. (Azul = predecesor)

- **Justificación**

Este juego de pruebas muestra como un problema con una cadena de libros precedentes (saga) se crea un plan de lectura respetado las restricciones establecidas hasta el momento.

5.3 Juego 3 - Extensión 1

- **Input**

```
(define (problem problema)

  (:domain libros)

  (:objects
    diciembre noviembre octubre septiembre agosto julio junio mayo
    abril marzo febrero enero - mes
    libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
    libro_8 libro_9 libro_10 libro_11 libro_12 libro_13 - libro
  )
  (:init
    **Inizializacion de relacion meses anteriores
    (quiere libro_0)
    (quiere libro_1)
    (quiere libro_2)
    (quiere libro_3)
    (quiere libro_4)
    (quiere libro_5)
    (quiere libro_6)
    (quiere libro_7)
    (quiere libro_8)
    (quiere libro_9)
    (quiere libro_10)
    (quiere libro_11)
    (quiere libro_12)
    (quiere libro_13)
    (predecesor libro_0 libro_2)
    (predecesor libro_0 libro_3)
    (predecesor libro_1 libro_2)
    (predecesor libro_1 libro_6)
    (predecesor libro_2 libro_5)
    (predecesor libro_3 libro_5)
    (predecesor libro_4 libro_6)
    (predecesor libro_8 libro_10)
  )
  (:goal
    (forall (?l - libro) (or (leido ?l) (imply (not (leido ?l)) (not
      (quiere ?l))))))
  ))
```

- **Output**

step	0: LEER LIBRO_0 ENERO
	1: LEER LIBRO_1 ENERO
	2: LEER LIBRO_2 FEBRERO
	3: LEER LIBRO_3 FEBRERO
	4: LEER LIBRO_4 ENERO
	5: LEER LIBRO_5 MARZO
	6: LEER LIBRO_6 FEBRERO
	7: LEER LIBRO_7 ENERO
	8: LEER LIBRO_8 ENERO
	9: LEER LIBRO_9 ENERO
	10: LEER LIBRO_10 FEBRERO
	11: LEER LIBRO_11 ENERO
	12: LEER LIBRO_12 ENERO
	13: LEER LIBRO_13 ENERO
	14: REACH-GOAL

- **Visualización**

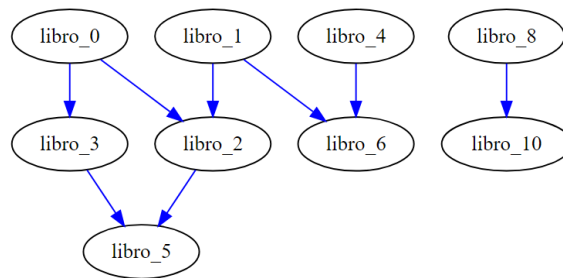


Figure 4: Visualización de la jerarquía de libros del juego de pruebas 3. (Azul = predecesor)

- **Justificación**

Este juego de pruebas muestra como un problema con libros con varios predecesores se crea un plan de lectura respetado las restricciones establecidas hasta el momento.

5.4 Juego 4 - Extensión 1

- **Input**

```

(:objects
  diciembre noviembre octubre septiembre agosto julio junio mayo
  abril marzo febrero enero - mes
  libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
  libro_8 - libro
)
(:init
  (quiere libro_0)
  (quiere libro_1)
  (quiere libro_2)
  (quiere libro_3)

```

```

(quiere libro_4)
(quiere libro_5)
(quiere libro_6)
(quiere libro_7)
(quiere libro_8)

(predesor libro_0 libro_2)
(predesor libro_0 libro_3)
(predesor libro_0 libro_4)
(predesor libro_0 libro_5)
(predesor libro_1 libro_3)
(predesor libro_1 libro_4)
(predesor libro_1 libro_5)
)

```

• Output

```

step    0: LEER LIBRO_0 ENERO
      1: LEER LIBRO_1 ENERO
      2: LEER LIBRO_2 FEBRERO
      3: LEER LIBRO_3 FEBRERO
      4: LEER LIBRO_4 FEBRERO
      5: LEER LIBRO_5 FEBRERO
      6: LEER LIBRO_6 ENERO
      7: LEER LIBRO_7 ENERO
      8: LEER LIBRO_8 ENERO
      9: REACH-GOAL

```

• Visualización

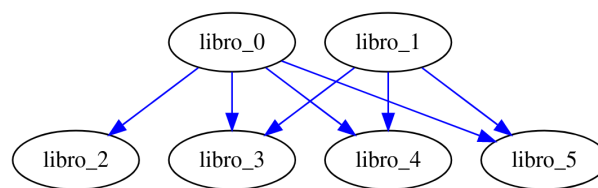


Figure 5: Visualización de la jerarquía de libros del juego de pruebas 4. (Azul = predesor, Rojo = paralelo)

• Justificación

Este juego de pruebas muestra como un problema con libros con varios predecesores donde no tiene porque ser independientes se crea un plan.

5.5 Juego 5 - Extensión 2

• Input

```

(define (problem problema)

  (:domain libros)

  (:objects
    diciembre noviembre octubre septiembre agosto julio junio mayo
    abril marzo febrero enero - mes
    libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
    libro_8 libro_9 libro_10 libro_11 - libro
  )
  (:init
    (mes-anterior-p enero febrero)
    (mes-anterior-p febrero marzo)
    (mes-anterior-p marzo abril)
    (mes-anterior-p abril mayo)
    (mes-anterior-p mayo junio)
    (mes-anterior-p junio julio)
    (mes-anterior-p julio agosto)
    (mes-anterior-p agosto septiembre)
    (mes-anterior-p septiembre octubre)
    (mes-anterior-p octubre noviembre)
    (mes-anterior-p noviembre diciembre)
    **Inizializacion de relacion meses anteriores
    (quiere libro_0)
    (quiere libro_1)
    (quiere libro_2)
    (quiere libro_3)
    (quiere libro_4)
    (quiere libro_5)
    (quiere libro_6)
    (quiere libro_7)
    (quiere libro_8)
    (quiere libro_9)
    (quiere libro_10)
    (quiere libro_11)
    (predecesor libro_1 libro_3)
    (predecesor libro_2 libro_3)
    (paralelo libro_6 libro_9)
  )
  (:goal
    (forall (?l - libro) (or (leido ?l) (imply (not (leido ?l)) (not
      (quiere ?l))))))
  ))

```

• Output

```

step  0: LEER-PA LIBRO_0 ENERO
      1: LEER-PA LIBRO_1 ENERO
      2: LEER-PA LIBRO_2 ENERO
      3: LEER-PA LIBRO_3 ENERO

```

```

4: LEER-PA LIBRO_4 ENERO
5: LEER-PA LIBRO_5 ENERO
6: LEER-PA LIBRO_6 ENERO
7: LEER-PA LIBRO_7 ENERO
8: LEER-PA LIBRO_8 ENERO
9: LEER LIBRO_9 ENERO
10: LEER-PA LIBRO_10 ENERO
11: LEER-PA LIBRO_11 ENERO
12: REACH-GOAL

```

- **Visualización**

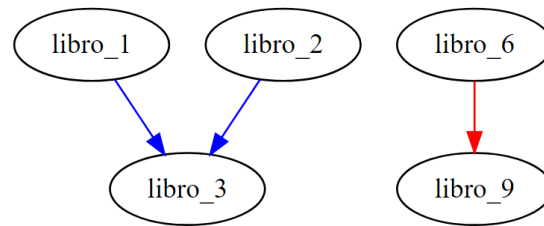


Figure 6: Visualización de la jerarquía de libros del juego de pruebas 5. (Azul = predecesor, Rojo = paralelo)

- **Justificación**

Este juego de pruebas muestra como un problema con libros con varios predecesores y un libro paralelo se crea un plan de lectura respetado las restricciones establecidas hasta el momento.

5.6 Juego 6 - Extension 2

- **Input**

```

(:objects
  diciembre noviembre octubre septiembre agosto julio junio mayo
  abril marzo febrero enero - mes
  libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
  libro_8 libro_9 libro_10 libro_11 libro_12 - libro
)
(:init
  (quiere libro_0)
  (quiere libro_1)
  (quiere libro_2)
  (quiere libro_3)
  (quiere libro_4)
  (quiere libro_5)
  (quiere libro_6)
  (quiere libro_7)
  (quiere libro_8)

```

```

(quiere libro_9)
(quiere libro_10)
(quiere libro_11)
(quiere libro_12)

(predecesor libro_0 libro_3)
(paralelo libro_0 libro_4)
(predecesor libro_0 libro_5)
(predecesor libro_0 libro_6)
(predecesor libro_1 libro_3)
(paralelo libro_2 libro_3)
(predecesor libro_2 libro_4)

(predecesor libro_7 libro_9)
(predecesor libro_7 libro_10)
(paralelo libro_7 libro_11)
(predecesor libro_7 libro_12)
(paralelo libro_8 libro_9)
(predecesor libro_8 libro_10)
(predecesor libro_8 libro_11)
(predecesor libro_8 libro_12)

```

)

• Output

```

step  0: LEER-PA LIBRO_0 ENERO
      1: LEER-PA LIBRO_1 ENERO
      2: LEER-PA LIBRO_2 ENERO
      3: LEER LIBRO_3 FEBRERO
      4: LEER LIBRO_4 FEBRERO
      5: LEER-PA LIBRO_5 ENERO
      6: LEER-PA LIBRO_6 ENERO
      7: LEER-PA LIBRO_7 ENERO
      8: LEER-PA LIBRO_8 ENERO
      9: LEER LIBRO_9 FEBRERO
     10: LEER-PA LIBRO_10 ENERO
     11: LEER LIBRO_11 FEBRERO
     12: LEER-PA LIBRO_12 ENERO
     13: REACH-GOAL

```

• Visualización

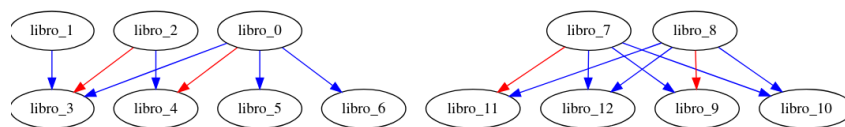


Figure 7: Visualización de la jerarquía de libros del juego de pruebas 6. (Azul = predecesor, Rojo = paralelo)

- **Justificación** Este juego de pruebas muestra como un problema con libros con varios predecesores y varios libro paralelos, con un numero elevado de estos, se crea un plan de lectura respetado las restricciones establecidas hasta el momento, incluyendo la de limite de paginas mensuales.

5.7 Juego 7 - Extensión 3

- **Input**

```
(define (problem problema)

  (:domain libros)

  (:objects
    diciembre noviembre octubre septiembre agosto julio junio mayo
    abril marzo febrero enero - mes
    libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
    libro_8 libro_9 libro_10 libro_11 libro_12 - libro
  )

  (:init
    (mes-anterior-p enero febrero)
    (mes-anterior-p febrero marzo)
    (mes-anterior-p marzo abril)
    (mes-anterior-p abril mayo)
    (mes-anterior-p mayo junio)
    (mes-anterior-p junio julio)
    (mes-anterior-p julio agosto)
    (mes-anterior-p agosto septiembre)
    (mes-anterior-p septiembre octubre)
    (mes-anterior-p octubre noviembre)
    (mes-anterior-p noviembre diciembre)
    **Inizializacion de relacion meses anteriores
    (= (pag-por-mes enero) 0)
    (= (pag-por-mes febrero) 0)
    (= (pag-por-mes marzo) 0)
    (= (pag-por-mes abril) 0)
    (= (pag-por-mes mayo) 0)
    (= (pag-por-mes junio) 0)
    (= (pag-por-mes julio) 0)
```

```

(= (pag-por-mes agosto) 0)
(= (pag-por-mes septiembre) 0)
(= (pag-por-mes octubre) 0)
(= (pag-por-mes noviembre) 0)
(= (pag-por-mes diciembre) 0)
(quiere libro_0)
(quiere libro_1)
(quiere libro_2)
(quiere libro_3)
(quiere libro_4)
(quiere libro_5)
(quiere libro_6)
(quiere libro_7)
(quiere libro_8)
(quiere libro_9)
(quiere libro_10)
(quiere libro_11)
(quiere libro_12)
(= (pag-libro libro_0) 138)
(= (pag-libro libro_1) 262)
(= (pag-libro libro_2) 190)
(= (pag-libro libro_3) 98)
(= (pag-libro libro_4) 227)
(= (pag-libro libro_5) 137)
(= (pag-libro libro_6) 153)
(= (pag-libro libro_7) 113)
(= (pag-libro libro_8) 98)
(= (pag-libro libro_9) 286)
(= (pag-libro libro_10) 120)
(= (pag-libro libro_11) 241)
(= (pag-libro libro_12) 249)
(predesor libro_0 libro_3)
(predesor libro_0 libro_4)
(predesor libro_0 libro_5)
(predesor libro_1 libro_3)
(predesor libro_1 libro_4)
(predesor libro_1 libro_5)
(predesor libro_2 libro_3)
(paralelo libro_2 libro_4)
(paralelo libro_6 libro_8)
(predesor libro_9 libro_11)
(predesor libro_9 libro_12)
(predesor libro_10 libro_11)
)
(:goal
  (forall (?l - libro) (or (leido ?l) (imply (not (leido ?l)) (not
    (quiere ?l))))))
))

```

- Output

```

step  0: LEER-PA LIBRO_0 ENERO
      1: LEER-PA LIBRO_1 ENERO
      2: LEER-PA LIBRO_2 ENERO
      3: LEER-PA LIBRO_3 FEBRERO
      4: LEER LIBRO_4 FEBRERO
      5: LEER-PA LIBRO_5 FEBRERO
      6: LEER-PA LIBRO_6 ENERO
      7: LEER-PA LIBRO_7 ENERO
      8: LEER LIBRO_8 FEBRERO
      9: LEER-PA LIBRO_9 FEBRERO
     10: LEER-PA LIBRO_10 MARZO
     11: LEER-PA LIBRO_11 ABRIL
     12: LEER-PA LIBRO_12 MARZO
     13: REACH-GOAL

```

- **Visualización**

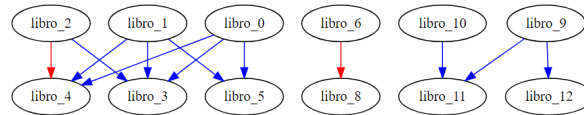


Figure 8: Visualización de la jerarquía de libros del juego de pruebas 7. (Azul = predecesor, Rojo = paralelo)

- **Justificación**

Este juego de pruebas muestra como un problema con libros con varios predecesores y varios libro paralelos se crea un plan de lectura respetado las restricciones establecidas hasta el momento, incluyendo la de limite de paginas mensuales.

5.8 Juego 8 - Extensión 3

- **Input**

```

(:objects
  diciembre noviembre octubre septiembre agosto julio junio mayo
  abril marzo febrero enero - mes
  libro_0 libro_1 libro_2 libro_3 libro_4 libro_5 libro_6 libro_7
  libro_8 libro_9 libro_10 libro_11 - libro
)
(:init
  (quiere libro_0)
  (quiere libro_1)
  (quiere libro_2)
  (quiere libro_3)
  (quiere libro_4)
  (quiere libro_5)
  (quiere libro_6)
  (quiere libro_7)

```

```

(quiere libro_8)
(quiere libro_9)
(quiere libro_10)
(quiere libro_11)

(= (pag-libro libro_0) 237)
(= (pag-libro libro_1) 81)
(= (pag-libro libro_2) 327)
(= (pag-libro libro_3) 400)
(= (pag-libro libro_4) 125)
(= (pag-libro libro_5) 69)
(= (pag-libro libro_6) 134)
(= (pag-libro libro_7) 113)
(= (pag-libro libro_8) 294)
(= (pag-libro libro_9) 261)
(= (pag-libro libro_10) 125)
(= (pag-libro libro_11) 357)

(predesor libro_0 libro_3)
(predesor libro_0 libro_4)
(predesor libro_1 libro_3)
(paralelo libro_1 libro_4)
(predesor libro_2 libro_3)
(paralelo libro_2 libro_4)

(paralelo libro_5 libro_9)
(predesor libro_5 libro_10)
(predesor libro_5 libro_11)
(paralelo libro_6 libro_9)
(predesor libro_6 libro_10)
(paralelo libro_6 libro_11)
(predesor libro_7 libro_9)
(predesor libro_7 libro_10)
(predesor libro_7 libro_11)
(predesor libro_8 libro_9)
(predesor libro_8 libro_11)

```

• Output

```

step  0: LEER-PA LIBRO_0 ENERO
      1: LEER-PA LIBRO_1 ENERO
      2: LEER-PA LIBRO_2 ENERO
      3: LEER-PA LIBRO_3 FEBRERO
      4: LEER LIBRO_4 FEBRERO
      5: LEER-PA LIBRO_5 ENERO
      6: LEER-PA LIBRO_6 ENERO
      7: LEER-PA LIBRO_7 FEBRERO
      8: LEER-PA LIBRO_8 FEBRERO
      9: LEER LIBRO_9 MARZO

```

10: LEER-PA LIBRO_10 MARZO
 11: LEER LIBRO_11 MARZO
 12: REACH-GOAL

- **Visualización**

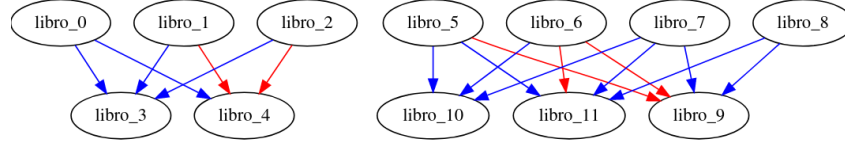


Figure 9: Visualización de la jerarquía de libros del juego de pruebas 8. (Azul = predecesor, Rojo = paralelo)

- **Justificación** Este juego de pruebas muestra como un problema con libros con varios predecesores y varios libro paralelos, con un numero superior al anterior, se crea un plan de lectura respetado las restricciones establecidas hasta el momento, incluyendo la de limite de paginas mensuales.

6 Tiempo de ejecución

En este apartado nos centraremos en analizar la evolución del tiempo de ejecución. Para poder realizar este análisis hemos hecho pruebas con los parámetros de entrada del generador: el mínimo y máximo de la altura de dependencias, el mínimo y máximo de la anchura de dependencias, la probabilidad de asociar con predecesor o paralelo 2 libros y el numero de familia de libros a crear. Podemos ver que los primeros y último parámetro hacen que aumente el numero de libros. Para simplificar, pondremos que el numero de familias a crear sea 1 y el porcentaje de crear asociaciones sea 50 y iremos subiendo tanto la altura y la anchura de estos, aunque siempre dejaremos fijo el mínimo de los dos paramentros a 1. En nuestro caso, tal y como dice en la documentación del enunciado, haremos las pruebas con la **Extensión 2**.

Número de libros	Tiempo de ejecución (seg)
2	0.00
6	0.00
9	0.01
10	0.02
12	0.81
13	3.92
14	28.82
15	293.44

Table 1: Ejecución del Planificador

Hemos creado un gráfico con estos resultados para poder visualizarlos mejor:

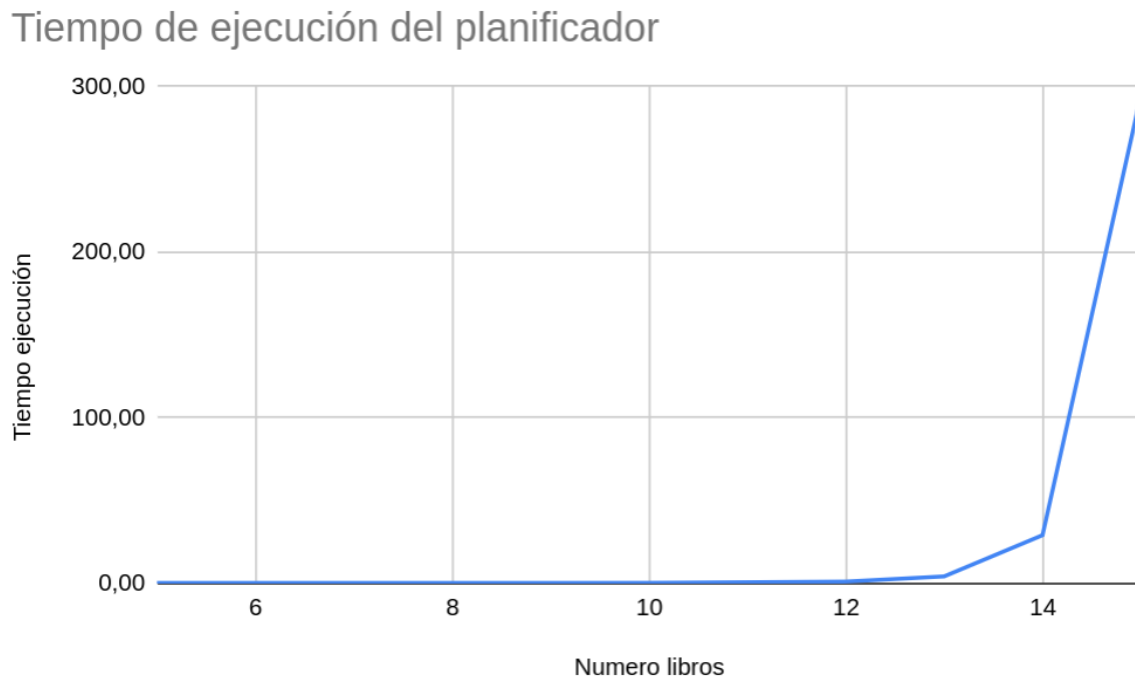


Figure 10: Gráfica del tiempo de ejecución

Tal y como podemos observar tanto en la tabla como en la gráfica, para valores pequeños (inferiores a 10) el tiempo es casi nulo, mientras que si vamos aumentando el numero de libros con los parámetros ya mencionados previamente, el tiempo aumenta de manera exponencial. También se nota que la mayor parte del tiempo se dedica a la búsqueda de la solución.

7 Conclusiones

En esta experiencia, hemos notado la estrecha relación entre los sistemas basados en conocimiento y los planificadores. Ambos se apoyan en los fundamentos de la lógica, aunque se diferencian en la autonomía, ya que los planificadores utilizan heurísticas para llegar a soluciones efectivas. Es crucial destacar que el modelado preciso de un problema y su dominio tiene un impacto significativo en la resolución, afectando desde la calidad de la solución hasta el tiempo de ejecución. Además, hemos observado un impacto significativo de la complejidad del problema en el tiempo de ejecución, especialmente al analizar cómo este crece en relación con el número de libros queridos. Esto puede ser debido a la cantidad de `forall`'s y `exists`'s implementados en el código.

Esta última práctica nos ha brindado la oportunidad de familiarizarnos y trabajar con un nuevo lenguaje de programación, así como con un enfoque novedoso para resolver problemas. Cada extensión realizada nos ha permitido explorar nuevas complejidades y extraer nuestras propias conclusiones a partir de ellas.