

# Documentación de la práctica de Planificación

Laboratorio de Inteligencia Artificial

1<sup>er</sup> Cuatrimestre - curso 2020/2021

Grau en Informàtica

Departament de Ciències de la Computació



**FIB**

Facultat d'Informàtica  
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Índice general

1. Organización, evaluación y entrega	2
2. Objetivos de aprendizaje	3
3. El problema	4
4. Guión de la práctica	7
5. Rúbrica de evaluación	9

## Organización, evaluación y entrega

Esta es la documentación de la práctica de planificación para los alumnos de Inteligencia Artificial del Grado en Informática. En este documento tenéis:

- Los objetivos de aprendizaje de la práctica correspondientes al temario de la asignatura
- La descripción del problema que debéis resolver
- Lo que tenéis que incluir en el informe que deberéis entregar como resultado de la práctica
- La planificación semanal de la práctica incluyendo los objetivos que debéis ir cubriendo cada semana.

La práctica se debe hacer **preferentemente en grupos de 3 personas**. Intentad no hacerla solos ya que os llevará mucho más trabajo.

La práctica se debe desarrollar con **Fast Forward v2.3**, el planificador que se presentará en las clases de laboratorio.

Planificad bien el desarrollo de la práctica y no lo dejéis todo para el último día, ya que no seréis capaces de acabarla y hacer un buen trabajo. En este documento tenéis indicaciones sobre el desarrollo de la práctica que os ayudarán a planificar vuestro trabajo.

La valoración de la práctica dependerá de la calidad del modelado del problema, de la cobertura del problema que hagáis, de las extensiones que abordéis y de la calidad de los juegos de prueba.

La entrega de la documentación será el día **11 de enero** en formato electrónico según las instrucciones que aparecerán en el racó.

## Objetivos de aprendizaje

El objetivo de esta práctica es enfrentarse a un problema sencillo de síntesis que se puede resolver mediante un planificador en el espacio de estados para que construya la solución.

Los objetivos específicos de esta práctica son:

- Implementar mediante un lenguaje de descripción (PDDL) el dominio (predicados y acciones) y varios ejemplos de problemas (objetos, estados inicial y final)
- Aplicar una metodología de desarrollo basada en prototipado rápido y diseño incremental.
- Saber escoger juegos de prueba suficientemente representativos para demostrar el funcionamiento del sistema y explicar los resultados.
- Tomar contacto con lenguajes de representación de acciones que se puedan usar con planificadores modernos. Se ha de demostrar cierta comprensión y madurez a la hora de utilizar el lenguaje PDDL.
- Conectar lo que se ha hecho en la práctica de Sistemas Basados en el Conocimiento con lo que puede hacer el planificador.

## El problema

La empresa REDFLIX distribuye diferentes contenidos audiovisuales, entre ellos series de televisión y películas. Para satisfacer las necesidades de públicos particulares quiere una aplicación capaz de planificar el visionado de sus series preferidas.

Se nos pide una herramienta que ha de **tener en cuenta**, por un lado, **las películas y los capítulos de series televisivas que el usuario ha visto ya**, y por el otro un **listado de películas y capítulos que quiere ver en los próximos días**, haciendo un plan de visionado diario en el que se le recomiende en qué orden ha de ver los capítulos o películas intentando **balancear un poco el número de contenidos o el de minutos que le tocaría ver cada día**. El plan de visionado ha de tener en cuenta que a menudo hay dependencias entre contenidos:

- **Contenidos predecesores**: es típico de series de televisión o de historias divididas en varias películas. Llamaremos **contenidos predecesores** a aquellos contenidos que **se han de visionar antes de un cierto contenido para poder disfrutarlo mejor**. Por ejemplo, si el usuario quiere ver la película *Star Trek III: en busca de Spock* y solo ha visto los capítulos de la 1a y 2a temporada de la serie clásica de Star Trek, el planificador ha de incluir en el plan de visionado todos los capítulos de la 3a temporada (en el orden correcto) y las dos primeras películas de Star Trek.
- **Contenidos paralelos**: esto ocurre en aquellos casos en que los contenidos **pertenecen a universos ficticios donde las historias no solo pasan una después de la otra, sino también en paralelo, por lo que es bueno ver las historias más o menos en el orden en el que ocurren**. Este fenómeno es muy típico de algunas series que han tenido varias subseries emitiéndose a la vez (Stargate, Star Trek, CSI, NCIS, Doctor Who/Torchwood...) y que a veces incluso cruzan historias y subtramas entre ellas.

Por lo tanto al sistema se le ha de **dar conocimiento sobre**:

- los contenidos del catálogo
- los contenidos predecesores a un contenido
- los contenidos paralelos a un contenido
- los contenidos que el usuario ya ha visto
- los contenidos que el usuario quiere ver

El **resultado** es un plan de visionado que

- refleja los **contenidos mínimos que el usuario ha de ver para llegar a ver los que quiere en las mejores condiciones**

- para **cada contenido**, indica en **que día ha de verlo**
- para **todos los contenidos** del plan se cumple en todo momento que **sus predecesores se ven en días anteriores**
- para **todos los contenidos** del plan se cumple en todo momento que **sus paralelos se ven en el mismo día o en el día anterior** (o en el día siguiente, para la relación simétrica).

El plan de visionado **no tiene porque llenar todos los minutos máximos de visionado cada día (200 minutos)**, solo ha de intentar hacer un plan factible. Si el usuario tiene pocos contenidos objetivo el plan puede llegar a ser un solo contenido al día.

## Problema básico y extensiones

- **Nivel básico:** En el plan de visionado todos los contenidos tienen **0 o 1 predecesores** y **ningún paralelo**. El planificador es capaz de encontrar un plan para poder llegar a ver los contenidos objetivo encadenando contenidos, donde cada contenido tiene solo uno o ningún predecesor.
- **Extensión 1:** Los contenidos pueden tener de **0 a N predecesores** pero **ningún paralelo**. 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 estan **en días anteriores**.
- **Extensión 2:** Extensión 1 + los contenidos pueden tener de **0 a M contenidos paralelos**. 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**.
- **Extensión 3:** El planificador controla que **no** se coloquen **más de 3 contenidos al día**.
- **Extensión 4:** Los **contenidos** tienen **asignados** el **número de minutos de duración**. El planificador controla que en el plan generado **no** se **superen los 200 minutos al día**. (Aviso: según como modeléis el problema es posible que necesitéis utilizar Metric-FF).

Según las extensiones que decidáis abordar la nota de la práctica será diferente:

- Nivel básico: la nota máxima es un 6
- Nivel básico + extensión 1: la nota máxima es un 7
- Nivel básico + extensión 2: la nota máxima es un 8
- extensión 3: +1 punto en la nota
- extensión 4: +2 puntos en la nota

En principio las extensiones 3 y 4 son mutuamente excluyentes, pero si un grupo presenta una versión del prototipo que cubre la extensión 3 y otra versión que cubre la extensión 4, puede obtener +3 puntos (con lo que la nota de la práctica puede llegar a 11).

## Nota extra

Los juegos de prueba los podéis hacer a mano, pero se asignará un punto extra a los grupos que hagan un programa (no importa el lenguaje) que pueda generar ficheros con juegos de prueba generados aleatoriamente y **los usen**.

# Documentación a entregar

La documentación debe incluir:

- Un documento en el que se describa, de forma razonada
  - La forma en la que se ha modelado el dominio (variables, predicados y acciones)
  - La forma en la que se modelan los problemas a resolver (objetos, estado inicial y final)
  - Una breve explicación de como habéis desarrollado los modelos (de una sola vez, por iteraciones)
  - El conjunto de problemas de prueba (mínimo 2), explicando para cada uno que es lo que intentan probar y su resultado. Podéis partir de los juegos de prueba para el nivel básico e ir añadiendo los elementos que cada extensión requiera.
- Código en PDDL del dominio que habéis modelado para cada extensión y los problemas de prueba.
- Un fichero que recolecte la traza de la resolución de los problemas de prueba.

## Capítulo 4

### Guión de la práctica

#### Primera semana: Fast Forward/Enunciado/creación del primer prototipo (14 a 20 de diciembre)

Esta primera semana la deberéis dedicar a leer el enunciado, a hacer un **modelo inicial de dominio y problema** y a **crear un modelo en PDDL que llegue al nivel básico**.

Esta semana se os explicará el funcionamiento del planificador Fast Forward. Es importante que leáis la documentación sobre PDDL y Fast Forward que se os dará, miréis los ejemplos que tenéis e intentéis ejecutarlos.

Tened en cuenta que modelar dominios en PDDL necesita una forma de pensar algo diferente a la que estáis acostumbrados con los lenguajes imperativos y lógicos, por lo que es importante que empecéis cuanto antes a ver cómo funciona.

**Si tenéis planeada alguna de las extensiones deberíais de ponerlas ya con ellas a media semana**, ya que la **última semana** deberéis dedicar algo de tiempo a la **documentación** y a las **pruebas**.

En esta práctica es importante planificar vuestro trabajo, no lo dejéis todo para el último momento.

#### Segunda semana: Prototipo definitivo / Juegos de prueba y documentación (21 a 27 de diciembre)

En esta semana deberíais tener ya un planificador que, como mínimo, es **capaz de crear planes en el nivel básico**. A **principios de la semana** ya deberíais haber **fijado todas las extensiones que queréis intentar hacer y tenerlas algo avanzadas** a media semana.

Mirad los ejemplos de problemas modelados en PDDL que tenéis en la web de la asignatura y en otras páginas en Internet para inspiraros.

Deberéis **plantearos los casos que queréis probar** y **mirar que los resultados que esperáis** sean los **correctos**. Haced una **lista de casos pensando los diferentes escenarios** que es capaz de resolver vuestro sistema.

Pensad que los **casos** han de ser **suficientemente variados tanto en lo que respecta a elementos que intervienen como su complejidad**. Tened en cuenta que estos casos os servirán de juegos de prueba, por lo que estáis matando dos pájaros de un tiro. Aprovechad para **guardar los resultados y documentarlos**.

También deberíais **ser capaces de explicar los resultados** que obtenéis **en función del conocimiento que habéis programado**.

Las **pruebas deberíais documentarlas adecuadamente** explicando cual es el **escenario de la prueba** y cuales son los **resultados que da el sistema**.



El resto de la documentación debería explicar todo el proceso de desarrollo y los diferentes prototipos que habéis creado por el camino.

No hace falta que esperéis hasta el último día para entregar. Si acabáis la práctica y la documentación antes podéis entregarla ya durante la semana.

## Rúbrica de evaluación

Esta es la rúbrica de evaluación de la práctica. La corrección se hará según estos criterios y siguiendo las pautas que se detallan para cada nivel de evaluación.

Deberéis seguir estos criterios a la hora de escribir vuestra documentación y explicar qué habéis hecho en el desarrollo de la práctica y como lo habéis hecho.

Criterio \ Valoración	Mal	Regular	Bien
Dominio	<ul style="list-style-type: none"> <li>El dominio se representa de manera incompleta o inadecuada (predicados innecesarios)</li> </ul>	<ul style="list-style-type: none"> <li>Se representa completa y adecuadamente las características del dominio</li> <li>La explicación de la representación del dominio es superficial</li> </ul>	<ul style="list-style-type: none"> <li>Se representa completa y adecuadamente las características del dominio</li> <li>Se explica detalladamente el significado de cada predicado y se justifica su necesidad</li> </ul>
Operadores	<ul style="list-style-type: none"> <li>El conjunto de operadores es inadecuado o incompleto</li> </ul>	<ul style="list-style-type: none"> <li>El conjunto de operadores es adecuado y completo</li> <li>La explicación/justificación de los operadores es superficial</li> </ul>	<ul style="list-style-type: none"> <li>El conjunto de operadores es adecuado y completo</li> <li>Se explica cada operador y se justifica detalladamente su necesidad para la resolución del problema</li> </ul>
Juegos de prueba	<ul style="list-style-type: none"> <li>Juegos de prueba inadecuados para el problema planteado</li> </ul>	<ul style="list-style-type: none"> <li>Juegos de prueba adecuados</li> <li>No se justifica la elección de los juegos de prueba</li> </ul>	<ul style="list-style-type: none"> <li>Juegos de prueba adecuados</li> <li>Se justifica la elección de los juegos de prueba</li> <li>Se explica la solución obtenida</li> </ul>
Completado de los niveles	<ul style="list-style-type: none"> <li>La solución propuesta para los diferentes niveles es inadecuada o incompleta</li> </ul>		<ul style="list-style-type: none"> <li>La solución propuesta para los diferentes niveles es adecuada y completa</li> </ul>