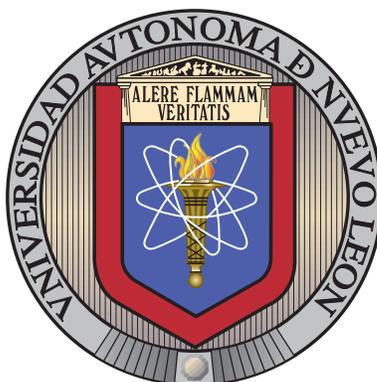


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



MODELACIÓN DE PLANES DE ESTUDIO USANDO
TÉCNICAS AVANZADAS DE PLANIFICACIÓN

POR

CRISTINA MAYA PADRÓN

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

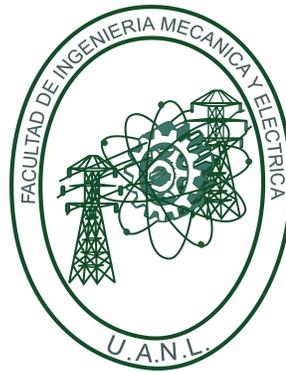
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

JUNIO 2013

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



MODELACIÓN DE PLANES DE ESTUDIO USANDO
TÉCNICAS AVANZADAS DE PLANIFICACIÓN

POR

CRISTINA MAYA PADRÓN

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

JUNIO 2013

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
División de Estudios de Posgrado

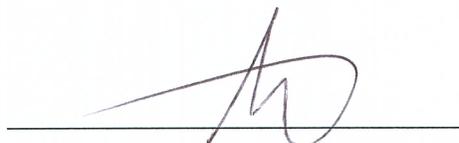
Los miembros del Comité de Tesis recomendamos que la Tesis «Modelación de Planes de Estudio usando Técnicas Avanzadas de Planificación», realizada por el alumno Cristina Maya Padrón, con número de matrícula 0937602, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dr. Romeo Sánchez Nigenda

Asesor



Dr. Fernando López Irarragorri

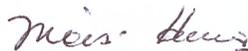
Revisor



Dra. Carolina Franco Espinosa

Revisor

Vo. Bo.



Dr. Moisés Hinojosa Rivera

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, junio 2013

*A mis pequeñas y hermosas hijas:
Shannon Cristina y Karen Naomi.*

ÍNDICE GENERAL

Agradecimientos	XIV
Resumen	XVI
1. Introducción	1
1.1. Contexto	1
1.2. Justificación	7
1.3. Antecedentes	9
1.4. Descripción del problema	12
1.4.1. Descripción del proyecto de tesis	12
1.4.2. Objetivo general	13
1.4.3. Objetivos específicos	13
1.5. Metodología propuesta	14
1.6. Descripción de capítulos	15
2. Marco Teórico	17
2.1. e-learning	17
2.1.1. Sistemas Gestores de aprendizaje	18

2.1.2.	Estándares de aprendizaje	19
2.1.3.	Objetos de aprendizaje	21
2.2.	Educación	22
2.2.1.	Objetivos de aprendizaje	22
2.2.2.	Diseño instruccional	22
2.2.3.	Diseños de aprendizaje	23
2.3.	Planificación de Inteligencia Artificial	24
2.3.1.	Algoritmos de planificación	26
2.3.2.	Problemas de planificación	28
2.3.3.	Representación de planes	32
2.3.4.	Solución	33
2.3.5.	PDDL (Planning Domain Definition Language)	34
2.3.6.	Ejemplo del lenguaje PDDL	36
2.3.7.	PDDL 2.1	43
2.3.8.	Competencias de Planificadores	50
2.3.9.	Planificador SGPLAN	51
2.3.10.	Planificador LPG	53
3.	Formulación del problema y Metodología	55
3.1.	Planteamiento del problema	55
3.2.	Planificación en Inteligencia Artificial	57
3.3.	Características del Modelo	57

3.3.1.	Representación de la estructura de la materia	60
3.3.2.	Restricciones de tiempo	64
3.3.3.	Uso de recursos educativos	64
3.3.4.	Funciones de acumulación de calidad	65
3.4.	Transformación del problema educativo a un modelo PDDL	74
3.4.1.	Requerimientos en PDDL de nuestro modelo	75
3.4.2.	Dominio	75
3.4.3.	Problema	96
4.	Pruebas y resultados	102
4.1.	Generador de instancias	102
4.1.1.	Creación de tipos de modelos y clases.	103
4.1.2.	Archivo de entrada para el generador de instancias	104
4.2.	Diseño de experimentos	105
4.3.	Pruebas y Resultados	106
4.3.1.	Modelos con requerimientos	106
4.3.2.	Modelos sin requerimientos	109
5.	Conclusiones y Recomendaciones	117
5.1.	Conclusiones	117
5.2.	Recomendaciones	119
5.3.	Trabajo futuro	120

A. Glosario	123
A.1. Términos educativos	123
A.2. Glosario de acrónimos	126
B. PDDL	127
B.1. Historia del PDDL	127
B.2. STRIPS	128
C. Modelo en PDDL	129
C.1. Modelo de 1 materia de clase pequeña con requerimientos	129
C.1.1. Dominio	129
C.1.2. Problema	139

ÍNDICE DE FIGURAS

2.1. Un plan de orden parcial para ponerse zapatos y calcetines (incluidas las condiciones previas a los pasos) y las seis posibles linealizaciones del plan.	34
2.2. Ejemplo del mundo de los bloques- Estado inicial	37
2.3. Ejemplo del mundo de los bloques- Estado final	37
2.4. Ejemplo del mundo de los bloques- Nombre de Dominio	38
2.5. Ejemplo del mundo de los bloques- Requerimientos del Dominio . . .	38
2.6. Ejemplo del mundo de los bloques- Predicados del Dominio	38
2.7. Ejemplo del mundo de los bloques- Acción pick up y su componentes. .	39
2.8. Ejemplo del mundo de los bloques- Acción Put-down y su componentes	40
2.9. Ejemplo del mundo de los bloques- Acción stack y su componentes .	41
2.10. Ejemplo del mundo de los bloques- Acción unstack y su componentes	42
2.11. Ejemplo del mundo de los bloques- Nombre de Problema	42
2.12. Ejemplo del mundo de los bloques- A que dominio pertenece el Problema	42
2.13. Ejemplo del mundo de los bloques- Objetos del problema	43
2.14. Ejemplo del mundo de los bloques- Estado Inicial definido en pddl . .	43

2.15. Ejemplo del mundo de los bloques- Estado Final definido en pddl . . .	44
2.16. Ejemplo de acciones durativas en PDDL	48
2.17. Ilustración de intervalos de tiempo de las acciones durativas. Se tiene que la acción dura 45 minutos y el recurso esta asociado a una actividad de aprendizaje.	49
3.1. Diversas actividades que realiza un estudiante.	55
3.2. Diversidad de recursos para estudio.	56
3.3. Proceso de planificación.	58
3.4. Plan de estudios de la Carrera de Mecatrónica-FIME/U.A.N.L. . . .	59
3.5. Ejemplo de materias previas como requisito para inscribir una materia en el Plan de estudios de la Carrera de Mecatrónica-FIME/U.A.N.L.	59
3.6. Figura de cómo es que se ve una materia del plan de estudios.	61
3.7. Estructura de una materia del plan de estudios	62
3.8. Dependencia de otra materia. RequiredSubject	63
3.9. Flechas para diferenciar los distintos tipos de dependencias.	63
3.10. Ejemplificación de un LO	64
3.11. Ejemplo de la materia de Estructura de datos con los temas, subtemas y actividades de aprendizaje.	69
3.12. Selección de las actividades de aprendizaje por cada subtema.	71
3.13. Tema Conceptos Básicos.	72
3.14. Tabla de acumulación actual de score.Inicial	72
3.15. Acumulación de subtemas: Arreglos y Conjuntos, del tema Conceptos Básicos	73

3.16. Acumulación de subtemas: Inserción-directa y Método-Quicksort, del tema Algoritmos-de-ordenamiento	73
3.17. Formación de la calificación. Cada tema suma el conjunto de subtemas que tiene y los multiplica por la ponderación que tiene ese tema. Al final la suma de las evaluaciones por tema, conforman dicha “calificación”.	74
3.18. Ejemplo de la función de score	80
3.19. Gráfica que ilustra cuando para realizar cualquier LA de un subtema se necesita ya sea que haya realizado al menos un LA (opción a) o que haya completado todas las LA necesarias en el subtema requerido.	83
3.20. Acción de inscribir la materia de Física II en PDDL.	84
3.21. Representación de las acciones modeladas.	95
4.1. Comparación de resultados con los planificadores SGPLAN y LPG en la clase pequeña con requerimientos.	107
4.2. Comparación de resultados con los planificadores SGPLAN y LPG en la clase mediana con requerimientos.	108
4.3. Comparación de resultados con los planificadores SGPLAN y LPG en la clase grande con requerimientos.	110
4.4. Comparación de resultados con los planificadores SGPLAN y LPG en la clase pequeña sin requerimientos.	111
4.5. Comparación de resultados con los planificadores SGPLAN y LPG en la clase mediana sin requerimientos.	112
4.6. Comparación de resultados con los planificadores SGPLAN y LPG en la clase grande sin requerimientos.	113

4.7. Concentrado de problemas resueltos por ambos planificadores. Modelando con o sin requerimientos	115
--	-----

ÍNDICE DE TABLAS

2.1. Plan del modelo del “mundo de los bloques”	44
3.1. Relación de orden y jerarquía	60
3.2. Temas de la materia Estructura de Datos del tercer semestre del plan de estudios de Ing. en mecatrónica. Seleccionamos solo dos temas: Conceptos básicos y Algoritmos de ordenamiento para nuestro ejemplo	67
3.3. Descripción de temas y subtemas de la materia de Estructura de Datos . Se seleccionaron dos subtemas por tema. Arreglos y Conjuntos del tema Conceptos básicos. Inserción directa y método Quicksort del tema Algoritmos de ordenamiento	68
3.4. Descripción de actividades de aprendizaje para cada subtema. En donde el subtema Arreglos y Conjuntos pertenece al tema Conceptos básicos. Inserción directa y método Quicksort al tema Algoritmos de ordenamiento	70
3.5. Características de PDDL que utilizamos en nuestro modelo y las cuáles son soportadas por los planificadores seleccionados	75
4.1. Descripción de los tipos de modelos y sus clases	104
4.2. Subgrupos de dominios y problemas generados	105

AGRADECIMIENTOS

Le agradezco a Dios por todas las bendiciones de mi vida y por brindarme la oportunidad de poder haber realizado esta maestría. Y cada vez que pudiera presentarse la frustración o el desánimo recordaba este versículo de Josué 1:9 que dice: “Mira que te mando que te esfuerces y seas valiente; no temas ni desmayes, porque Jehová tu Dios estará contigo en dondequiera que vayas”.

Le agradezco a mi esposo Jorge Hudtler por todo su apoyo, ánimo y comprensión, sin la cual no habría realizado esta maestría. A mis hijas que son la inspiración de todo lo que hago y que todo el tiempo preguntaban ¿Qué es una maestría?, ¿Qué es una tesis?, ¿Cuándo terminas esa tarea? (refiriéndose a la tesis). A mis padres Máxima Padrón y Epifanio Maya los cuales me han impulsado a realizar todo lo que he hecho y me han apoyado en el cuidado de sus nietas. A mis hermanas y hermano que siempre han estado ahí conmigo.

Un agradecimiento especial al Dr. Oscar Chacón por creer que podría realizar esto, por las facilidades que me brindó durante la maestría, por su preocupación al buscarme “asesorías” en las diferentes materias del primer semestre. Asimismo agradezco a mis asesores, que eran estudiantes de doctorado: Lucero Ozuna, Omar Ibarra y Jonás Velasco.

Agradezco también al Comité de selección a la maestría que confiaron en que podría realizar y culminar estos estudios: Dr. Hugo Jair Escalante, Dra. Yasmin Ríos y Dr. Arturo Berrones.

Agradezco a mi asesor de tesis el Dr. Romeo Sánchez por su apoyo, dirección y

paciencia. Por brindar las ideas que posteriormente fueron plasmadas en esta tesis. Por todas las correcciones que me hizo en este trabajo, asegurándose todo el tiempo de que estuviera bien realizado. Asimismo, agradezco a mis revisores el Dr. Fernando López y la Dra. Carolina Franco Espinosa por tomarse el tiempo de leer esta tesis, de dar su opinión y aportar ideas de cómo es que puede ser mejorado este trabajo.

Agradezco a mis compañeros de generación porque han estado conmigo, me han apoyado, asesorado y animado, además de esto, se han convertido en mis amigos: Nancy Arratia, Victoria Rebillas, Alexis Figueroa, Fernando Elizalde, Luis Infante y Andrés Castrillón.

Gracias a mis compañeros de Yalma, por recibirme cuando Cubos-b desapareció, por apoyarme en algunas dudas que he tenido, escucharme hablar de mi trabajo, darme su opinión y sobretodo por su compañerismo.

Agradezco a la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León, especialmente al Posgrado en Ingeniería de Sistemas.

Gracias al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para la realización de este trabajo de tesis.

RESUMEN

Cristina Maya Padrón.

Candidato para el grado de Maestro en Ingeniería
con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: Modelación de Planes de Estudio usando Técnicas Avanzadas de Planificación

MODELACIÓN DE PLANES DE ESTUDIO USANDO TÉCNICAS AVANZADAS DE PLANIFICACIÓN

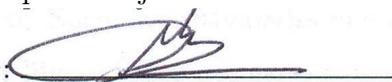
Número de páginas: 152.

OBJETIVOS Y MÉTODO DE ESTUDIO: La aplicación de técnicas avanzadas de Planificación en Inteligencia Artificial (IA) por medio de la modelación en PDDL del programa académico de un estudiante, incluyendo en el modelo la situación actual de dicho estudiante, en relación a las materias aprobadas y/o objetivos de aprendizaje satisfechos. Considerando las restricciones de tiempo y recursos del sistema educativo, con el fin de generar planes educativos que lo guíen durante su trayectoria escolar. Posteriormente se utilizan los planificadores SGPLAN y LPG, que son los algoritmos de planificación seleccionados para dar solución a los modelos generados.

CONTRIBUCIONES Y CONCLUSIONES: La principal contribución de este trabajo es la realización del modelo de planificación en PDDL que contempla la utilización de duración en sus “acciones”, la asignación de recursos a cada actividad de aprendizaje, esquemas de acumulación de calidad o “scores” en las acciones propuestas para que el estudiante realice, restricciones jerárquicas y de orden, es decir precedencia de subtemas, temas y actividades de aprendizaje.

Como conclusión decimos que nuestro modelo está desarrollado apropiadamente, considerando las necesidades que se modelaron. Sin embargo, los planificadores presentan ciertas limitantes para dar solución de una manera eficiente en las clases grandes con requerimientos, es decir de 5 a 7 temas, subtemas y actividades de aprendizaje. Los requerimientos son de precedencia a subtemas anteriores, temas y actividades de aprendizaje.

Firma del asesor:



Dr. Romeo Sánchez Nigenda

CAPÍTULO 1

INTRODUCCIÓN

1.1 CONTEXTO

Los estudiantes generalmente tienen múltiples actividades a realizar durante su trayectoria escolar. Entre las actividades que realizan se encuentran: asistir a clases, hacer tareas, leer libros, hacer exámenes, etc.

Además de lo antes mencionado, en la actualidad existe una gran diversidad de recursos disponibles para llevar a cabo dichas actividades, por ejemplo: diapositivas, ejercicios, artículos de investigación, libros, videos, computadoras, software, equipo de laboratorio, etc. Esto sin considerar que el estudiante (al menos a nivel superior) debe realizar el servicio social, prácticas profesionales, y actividades extracurriculares como actividades deportivas.

El principal problema es entonces que el estudiante tiene que decidir ¿qué estudiar?, ¿cuándo? y ¿qué utilizará? para posiblemente mejorar su rendimiento académico a través del tiempo. Es decir, necesitamos generar planes de aprendizaje personalizados (también conocidos como diseños de aprendizaje) que permitan a los estudiantes manejar sus objetivos de aprendizaje, y las actividades educativas necesarias para lograrlos optimizando su tiempo disponible. Este trabajo presentará modelos de IA basados en planificación que permitirá el diseño y generación de dichos planes tomando en consideración el plan de estudios del estudiante, su situación actual con respecto a materias no aprobadas, etc.

Lo anterior es importante ya que si el estudiante toma control de su aprendizaje tiene un menor riesgo de abandono y mayor probabilidad de avanzar satisfactoriamente en sus estudios. Según [16], durante los años de escuela secundaria, el interés de los estudiantes en la escuela y las habilidades académicas puede comenzar a rezagarse, por lo que en la transición a la preparatoria, los estudiantes que están en riesgo de abandonar la escuela necesitan apoyo individual intensivo u otros soportes que integren al estudiante en el propósito de la educación. Debido a esto, el Instituto de Ciencias de la Educación, de los Estados Unidos recomienda a los educadores y los encargados de formular políticas, considerar cómo implementar estrategias intermedias destinadas a aumentar la participación de los estudiantes. Una de las recomendaciones de este Instituto es personalizar el entorno de aprendizaje y el proceso de instrucción de los estudiantes [16].

Aunado a esto, algunas investigaciones favorecen el hecho de que los estudiantes administren su propio estudio, asegurando que “la capacidad de los estudiantes para gestionar su propio estudio es una de las habilidades más importantes que necesitan para aprender, con consecuencias que se sentirán en toda su vida” [33]. De igual manera, el alumno tiende a aprender mejor cuando las aplicaciones tecnológicas están directamente relacionadas al curriculum que tienen que cubrir y cuando la aplicación está integrada en un día escolar típico [39].

Considerando todo lo anterior, creemos que es de suma importancia generar planes educativos personalizados, donde el estudiante pueda administrar sus actividades de estudio y utilice la tecnología a su favor.

Además de ser un problema importante, podemos decir que es interesante abarcarlo ya que el modelo está compuesto de propiedades complejas que interaccionan entre sí, como: Objetivos, recursos educativos, el factor tiempo en las actividades, métricas de optimización, dependencias de orden y entre actividades de aprendizaje, etc. Además de esto, está la necesidad de identificar el estado del arte de los algoritmos que lo pueden resolver.

Es interesante porque los modelos podrían abrir la puerta a muchas otras aplicaciones o áreas de investigación como la identificación de estudiantes en riesgo, clasificación de recursos, pronósticos en el uso de los recursos (se podrían tener datos que nos permitieran identificar a tiempo la necesidad de adquirir recursos extras para satisfacer las necesidades de los estudiantes), etc.

Se estudia además la complejidad de solución de los planes generados por los modelos de planificación, ya que éste contempla propiedades como duración en las actividades, asignación de recursos, objetivos, métricas de optimización, como el tiempo, dependencias de orden entre los subtemas/temas, dependencias entre actividades, etc. Sin considerar además, que no estamos abarcando las ventanas de tiempo para diseñar planes educativos (por semestre, días, etc). Es decir, el plan comprende una programación de actividades a realizar, seleccionando aquellas que vayan de acuerdo a la métrica a optimizar y considerando su objetivo, pero no el día, hora, etc.

Es importante notar que viendo la importancia de personalización de planes educativos o cursos a estudiantes, se han realizado modelos que tratan de representarlos. En la literatura se han realizado modelos de cursos que se pretende sean en línea, algunos de estos modelos, al igual que los modelos que nosotros proponemos, consideran duración en sus actividades [29], [10], [9]. Algunas representaciones propuestas mencionan manejar duración en sus actividades, sin embargo no son claras en el manejo de las mismas [20] o simplemente no modelan las acciones durativas [15]. Nuestros modelos consideran el tiempo como una métrica discreta, lo que permite personalizar aún más las actividades de aprendizaje.

Aunado a lo anterior, nuestro modelo maneja restricciones de recursos educativos que son asociados a una actividad de aprendizaje, dicha característica es modelada en trabajos como [20] [18] [19]. Aunque, la cantidad de estos recursos es tratada como una función numérica, “recalculando” el plan cuando estas restricciones están presentes.

Nuestro modelo tiene la característica de “acumular calidad” entre las acciones, es decir, por cada actividad de aprendizaje realizada se toma el valor o score dado y se acumula por subtema y tema de cada materia, esto está descrito a detalle más adelante. Esta manera de acumulación no es tratada en ningún trabajo consultado. Solo en trabajos como [18] realizan solo sumatorias de los valores asignados a los objetos de aprendizaje.

Otra diferencia significativa es el hecho que los modelos actuales consideran personalizar un curso para el estudiante [20] [15], [18], [10], [30], [31] mientras el trabajo que presentaremos trata de abarcar múltiples cursos. Parte del análisis responderá la escalabilidad de los modelos propuestos.

Otra característica en que difiere nuestro trabajo de los demás, es que no estamos limitando nuestra modelación a las propiedades que los Sistemas Gestores de Aprendizaje (LMS) consideran. Nosotros no estamos interesados en adaptar nuestros modelos a los LMS, sino en tratar de representar los planes de estudios en planes individualizados a los estudiantes, de una manera más robusta, considerando la duración de las actividades de aprendizaje, los recursos asociados a éstas, además de la cantidad de recursos disponibles, métricas de acumulación de calidad, es decir, el score de cada actividad de aprendizaje realizada, para de alguna manera tratar de medir el nivel de aprendizaje de dicha actividad y así cumplir con los objetivos.

Así tenemos que en la literatura existen trabajos relacionados a la modelación de cursos en línea para estudiantes, conocido este proceso como e-learning. El e-learning es el proceso de formación a distancia basado en el uso de las tecnologías de la información [8]; estos trabajos se han desarrollado adaptando diseños de aprendizaje usando los estándares que manejan los Sistemas Gestores de Aprendizaje (Sistemas informáticos en servidores de internet/intranet que se encargan de administrar, entre otras cosas, cursos virtuales [3]). Así, toman directamente las plataformas de e-learning para generar los diseños de trayectorias de los estudiantes, [18], [30], [9]. Observamos que esto limita el tipo de modelos que generan, ya que su nivel de representación está sujeto a la capacidad de expresividad de los lenguajes estándares de

E-learning y los metadatos contenidos en los Objetos de Aprendizaje. Entendiéndose como Objeto de Aprendizaje cualquier recurso educativo que se pueda utilizar en el aprendizaje [32].

Nuestro trabajo busca generar modelos que representen el currículum académico de un estudiante, en donde dichos modelos no estén limitados a los LMS. Aunque los LMS son útiles para los trabajos que están vinculados con el e-learning, no es el propósito de este trabajo. De hecho, el no considerarlos nos abre la puerta para generar modelos más expresivos que nos permitan generar una secuencia de acciones que ayuden al estudiante a planificar su tiempo y actividades.

Se pretende que los modelos generados ayuden a optimizar el tiempo de aprendizaje de los estudiantes y los recursos educativos del sistema al generar secuencias de acciones que satisfagan los objetivos de aprendizaje de cada una de las materias comprendidas en el plan de estudios, y esto sirva para tratar de mejorar su rendimiento académico.

Para el diseño de nuestros modelos se analizaron estándares de e-learning compatibles con LOM (Learning Object Metadatos) [20] para representar secuencias de actividades de aprendizaje, también conocidas como Diseños de Aprendizaje, con el fin de reconocer las limitaciones de los modelos actuales, y proponer extensiones que nos permitan representar el complejo mundo real del ambiente educativo.

Una vez identificadas todas las propiedades de los modelos educativos a considerar, transformamos dichos modelos a dominios de planificación inteligente [38]. El objetivo es que, al utilizar técnicas de planificación inteligente, podamos generar autónomamente planes educativos o trayectorias escolares para los estudiantes. El propósito es que dichos planes agrupen los objetivos de aprendizaje que los estudiantes tienen que cumplir a lo largo de sus licenciaturas, junto con las acciones educativas necesarias para lograrlos.

Los modelos generados serán evaluados empíricamente utilizando algunos de los algoritmos de planificación más avanzados en la actualidad con el fin de identi-

ficar y analizar los factores críticos que pueden llegar a aumentar la complejidad de solución de la generación de planes educativos. Para esto, diseñamos un programa que genera automáticamente los modelos en el lenguaje de definición de dominios de planificación, PDDL-Planning Definition Domain Lenguaje [17], los cuáles sirven de entrada al *planificador*, el cual es un algoritmo de propósito especial, que utiliza un lenguaje de planificación formal con una sintaxis, semántica y teoría de la demostración bien definidas [36]. La teoría de la demostración especifica qué es lo que se puede inferir de los resultados de las secuencias de acción y, por lo tanto, cuáles son los planes legales[36]. El algoritmo nos permite encontrar tales planes. El proceso anterior nos da como resultado los planes de estudio de cada estudiante.

Una de las limitantes que presenta al analista humano el modelar en PDDL es representar situaciones reales totalmente adaptadas al modelo de planificación. Una de las dificultades para nosotros fue la modelación de las distintas actividades de aprendizaje, ya que cada una de ellas es particular y con características y requerimientos diferentes, además de pertenecer a diferentes temas de diferentes materias. Inicialmente se modeló cada una de ellas como una acción con sus precondiciones y efectos de manera individual, pero esto incrementaba la cantidad de predicados y de acciones en el dominio. Después se logró compactar este modelo haciendo una clasificación de las diferentes actividades de aprendizaje, incluyendo en ellas los requerimientos de pertenencia, asignación de recursos, duración, score, etc.

De igual manera ocurre con aquellas características no contempladas en el modelado, por ejemplo, nosotros decidimos sacrificar la inclusión de ventanas de tiempo en el modelo, lo cual nos permitió que el modelo fuera más compacto. Aún así creemos que esto es un área perfectible para darle una mayor representabilidad al modelo.

Las principales bondades de este trabajo es que se generan modelos válidos, considerando todas las restricciones. Se seleccionó los mejores planificadores actuales que soportaran los requerimientos del modelo, tales como acciones durativas, tipos, métricas, igualdades, constantes, de manera que pudieran dar solución a estos mod-

elos.

Dentro de limitaciones de este trabajo es que para poder obtener soluciones de los modelos, llamados planes, estamos limitados al alcance que tengan los planificadores, ya que algunas propiedades que consideramos, como la métrica de “grades”, que utilizamos para obtener una “calificación” por materia del estudiante tratando de garantizar un grado de aprendizaje en dicha materia, no puede ser soportada por los planificadores seleccionados, ya que al ir aumentando la cantidad de actividades de aprendizaje, le resulta imposible soportar esta métrica, solo fue posible considerarla en las instancias pequeñas.

1.2 JUSTIFICACIÓN

Nuestra motivación para investigar el uso de la tecnología de Planificación en el ámbito educativo radica en investigaciones recientes que sugieren que los estudiantes tienden a aprender mejor cuando las aplicaciones tecnológicas están directamente relacionadas al currículum que tienen que cubrir, cuando éstas se ajustan a las habilidades del estudiante y a su experiencia anterior, cuando les proveen de retroalimentación en su desempeño, cuando les permiten examinar sus datos y fijar sus propios objetivos de aprendizaje personalizando su entorno académico, cuando éstas le ayudan a los estudiantes a identificar que material han aprendido y que temas requieren más estudio [16] [33] [39].

Asimismo la tecnología mejora el rendimiento cuando se utiliza en entornos en los que los maestros, la comunidad, la escuela y sus administradores, apoyan su uso. Además de las mejoras de rendimiento vinculado con el apoyo administrativo a la tecnología, los resultados muestran que la integración de la tecnología con la instrucción, el desarrollo profesional de los maestros, el uso de computadora en casa y en la escuela con productos diferenciados y los puntos de entrada de los estudiantes, se combinan para mejorar el rendimiento [39].

Considerando lo anterior, creemos que la creación de trayectorias (o planes educativos personalizados) a través de técnicas de Planificación capturan las propiedades antes mencionadas, ya que permitiría a los estudiantes un control personalizado de sus actividades académicas, lo cual sería de gran beneficio no solo para el estudiante el cual podría administrar sus actividades de una manera adecuada, optimizando su tiempo, sino también a profesores que podrían detectar oportunamente aquellos alumnos que están fallando y/o en peligro de deserción; beneficiaría también a la institución, porque basándonos en técnicas de aprendizaje máquina y planes educativos históricos podríamos detectar trayectorias escolares con riesgo a abandonar la escuela, o a fallar académicamente para así intervenir adecuadamente. Además, los administradores, basándose en el uso de los recursos educativos que los planes arrojen, podrían pronosticar la demanda de los mismos, con el fin de cubrir mejor las necesidades de generaciones futuras.

Viendo la importancia de la creación de planes educativos personalizados, tanto para estudiantes, profesores, institución, etc., nuestra investigación intenta abarcar, por ahora, la creación de los modelos en PDDL, el cual es un estándar para la comunidad de IA para representar modelos de dominios de planificación [17].

Aunque las técnicas de planificación inteligente han sido útiles para sintetizar problemas complejos dado un estado inicial, una representación de objetivos y un conjunto de transiciones de estados posibles [15], es hasta muy recientemente, que se ha visto la generación de cursos de aprendizaje para estudiantes como un problema de Planificación.

Nosotros estamos proponiendo considerar, por ejemplo, el uso de recursos, tiempo, costos, la acumulación de scores de las actividades de aprendizaje, relaciones de precondiciones entre LOs (Objetos de aprendizaje), y éstos con sus subtemas, temas y materias, etc., que nos ayuden a representar de una manera más expresiva la trayectoria escolar de un estudiante con el fin de asegurar que los objetivos del plan de estudios sean cubiertos de una manera personalizada.

Nuestros modelos nos permitirán analizar las capacidades de las técnicas de planificación actuales al tratar de resolverlos, se pretende también identificar aquellos factores que aumentan la complejidad de solución de los mismos. De la misma manera existen características que no pudimos modelar y que son necesarias para aumentar la expresividad de los modelos generados y que serán presentadas y discutidas en el resto de este trabajo.

1.3 ANTECEDENTES

Actualmente, la modelación de diseños de aprendizaje ha generado gran interés bajo el paradigma de e-learning, éste es “una metodología de enseñanza-aprendizaje basada en el uso de contenidos educativos distribuidos a través de un soporte electrónico, fundamentalmente Internet” [14]. Asimismo, los cursos de formación en red son definidos como cursos donde la mayoría, sino toda, de la instrucción y de las pruebas se logran vía recursos accesibles en la Web [8].

El e-learning utiliza herramientas específicas de creación de cursos como las plataformas LMS (Learning Management System) Sistemas Gestores de aprendizaje, los cuales son sistemas informáticos en servidores de internet/intranet que se encargan de administrar cursos virtuales [3].

En estos LMS se pueden generar cursos que cumplan con más de un estándar, pero prácticamente todos son capaces de generar cursos que cumplan el estándar SCORM (del inglés Shareable Content Object Reference Model- modelo de referencia para contenidos compatibles) [6], con el cual se hace posible crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM. Este estándar ha sido desarrollado por ADL (Advanced Distributed Learning) con el objeto de contar con un modelo estándar internacional capaz de aglutinar los distintos modelos existentes hasta la fecha en el mercado del e-learning: AICC (Aviation Industry CBT [Computer-Based Training] Committee), IMS (Instructional Management System Project), ARIADNE (Alliance

of Remote Instruction Authoring and Distribution Networks for Europe), IEEE ((Institute of Electric and Electronic Engineers) Learning Technology Standards Committee), etc.

Otro estándar conocido es el LOM (Learning Object Metadata) que según la IEEE se define como “los atributos necesarios para describir adecuadamente un Objeto de aprendizaje”. Asimismo, “Los Objetos de aprendizaje (LO-Learning Object) son descritos como cualquier entidad, digital o no digital que puede ser utilizado, reutilizado o referenciado durante el aprendizaje apoyado en la tecnología” [12].

Cabe mencionar, que aunque en la actualidad existe polémica con respecto a lo que es un objeto de aprendizaje [35], [40], nosotros nos basamos en la definición dada anteriormente, para utilizar los LO dentro de nuestra modelación de planes de estudio.

Los trabajos en la literatura consideran las plataformas de los Sistemas Gestores de Aprendizaje al igual de los estándares antes mencionados para la modelación de cursos en línea, tales como LOM [20], [18], [10], [19] y el IMS [10], [19], [29]; sin embargo, debemos mencionar que nosotros vamos en otra dirección, es decir, en lugar de generar cursos en línea, tratamos de modelar el curriculum de un estudiante, el cual puede estar en un formato de educación presencial y a la vez, utilizar la tecnología como una herramienta de apoyo que los ayude de una manera individual a mejorar su desempeño de acuerdo a sus objetivos. Por tal motivo, no utilizamos los Sistemas Gestores de aprendizaje, ya que no estamos limitados a la educación a distancia, además de que observamos limitaciones en cuanto a expresividad en los modelos generados por estos trabajos.

Todos los estándares mencionados anteriormente son utilizados por los LMS para representar sus cursos o diseños de aprendizaje. Por lo mismo, los cursos generados bajo estos estándares estarán limitados en su expresividad a lo que los estándares puedan expresar. Estos estándares se generan con el propósito de administrar contenidos, en consecuencia no tienen el soporte para considerar propiedades que pudier-

an ser de interés para modelos educativos más amplios.

Por otro lado, debido al alto nivel de detalle para generar un curso de aprendizaje de una forma manual se ha utilizado recientemente la Planificación de Inteligencia Artificial para modelar cursos de aprendizaje [15].

La planificación de IA utiliza el lenguaje de descripción de dominios de planificación PDDL, el cual es un estándar para la representación de modelos de dominios de planificación dado a conocer por Drew McDermott en 1998 [17]. Éste es una estandarización de la sintaxis para expresar acciones utilizando precondiciones y post-condiciones para describir la aplicabilidad y efectos de las acciones. Además está comprendido en dos archivos llamados dominio y problema. Aunque existe cierta incorfomidad con la comunidad con algunas de las características del lenguaje PDDL, no se puede negar que la introducción de PDDL ha facilitado el desarrollo científico de la planificación.

Queriendo facilitar el diseño de dominios y problemas de planificación se ha trabajado en la realización de herramientas de modelado sin la necesidad de conocer el lenguaje PDDL [20] y [18] [19] [9]. Y aunque los resultados de las investigaciones en el área lucen promisorios, se observan limitaciones, para nuestros propósitos, con respecto al tipo de modelo educativo que generan.

Algunos de estos trabajos no consideran, o al menos no lo mencionan, el tiempo que se dedica a cada Objeto de aprendizaje [15]; otros, como ya se había mencionado, están limitados a los Sistemas Gestores de Aprendizaje (LMS) y su nivel de representación se basa en los datos contenidos en los metadatos de los objetos de aprendizaje (LOM) [20], [30], [10] esto porque están orientados a cursos en línea. Trabajos más recientes como [9] [19] manejan acciones durativas para cada Objeto de aprendizaje y flujos numéricos pero no representan de manera explícita cómo acumular calidad en el desempeño del estudiante, es decir, que calificación o score va obteniendo el estudiante al realizar tal actividad. Las consideraciones de la duración de las actividades, las restricciones de recursos asociados a los LO, la acumulación

de score en las actividades de aprendizaje, las restricciones entre materias, temas, subtemas, que consideramos importantes para poder representar un plan de estudios personalizado a cada estudiante, no están cubiertas en su conjunto en los trabajos encontrados.

Si queremos representar el complejo mundo real del ámbito educativo se necesita modelar características que extiendan dichos modelos y se adecuen a las necesidades de los estudiantes. Proporcionando acciones durativas para la realización de las actividades de aprendizaje, considerando la duración en unidades de tiempo, flujos numéricos para representar no solo el tipo de recurso asociado a un objeto de aprendizaje, sino la cantidad existente de los mismos. Esta restricción se considera ya que en la actualidad los recursos no son ilimitados.

A modo de conclusión, nosotros pretendemos no solo modelar un curso como en e-learning, sino representar modelos que además, no estén limitados a los Sistemas Gestores de Aprendizaje y proporcionen una secuencia de acciones que ayude al alumno a planificar su tiempo y actividades escolares con respecto a todo su currículum académico (plan de estudios), optimizando a su vez los recursos del sistema educativo.

Aunado a esto, la investigación aquí planteada abre las puertas a otras líneas de investigación que permitan la detección oportuna de problemas de aprendizaje en los estudiantes utilizando sus planes educativos. Esto permitirá en un futuro anticipar riesgos de deserción escolar o mal desempeño académico actuando proactivamente con modificaciones a los planes de estudio personalizados.

1.4 DESCRIPCIÓN DEL PROBLEMA

1.4.1 DESCRIPCIÓN DEL PROYECTO DE TESIS

La finalidad de este proyecto es la aplicación de técnicas avanzadas de Planificación en Inteligencia Artificial al ámbito educativo. La idea principal es modelar

el programa académico de un alumno de educación a nivel superior (aunque no está limitado a este nivel), incluyendo en dicho modelo la situación actual del estudiante, considerando las restricciones de tiempo y recursos del sistema educativo, con el fin de generar planes educativos que lo guíen durante su trayectoria escolar para intentar mejorar su rendimiento académico.

1.4.2 OBJETIVO GENERAL

El objetivo es crear un Modelo de Planificación en Inteligencia Artificial que genere planes de estudio personalizados a cada estudiante, que incluya de una manera satisfactoria diferentes esquemas de acumulación de calidad o scores de calificaciones, que ayuden a determinar si el estudiante se está desviando del plan educativo y de los objetivos deseados, tomando en cuenta además, las restricciones de tiempo y recursos del sistema educativo, con el fin de ayudar a su formación académica.

1.4.3 OBJETIVOS ESPECÍFICOS

- Identificar los factores asociados que se necesitan para modelar la trayectoria escolar de un estudiante. Dicha identificación y análisis se realiza tomando en cuenta los estándares de e-learning y la capacidad de modelado de PDDL, esto con la finalidad de determinar los factores que sean considerados importantes para incluirlos en el modelo.
- Basándonos en el análisis realizado del objetivo anterior, el siguiente objetivo consiste en generar un modelo basado en técnicas de inteligencia artificial que represente las propiedades de una trayectoria escolar, incluyendo las actividades de aprendizaje necesarias para completar un curriculum académico.
- Una vez que se tiene el modelo en PDDL, se construyen las librerías de planificadores más actuales que pueda utilizar para resolver el modelo de planificación considerando las características y requerimientos del mismo.

- Teniendo el modelo de planificación y los planificadores que puedan resolver dicho modelo, el objetivo siguiente es desarrollar un generador para implementar diferentes tipos o clases de problemas que pudieran representar situaciones del mundo real a la hora de representar los planes educativos. Es decir, se pretende capturar las diferentes clases de problemas para identificar si las tecnologías actuales son suficientes para resolverlos.
- Resolver computacionalmente los modelos generados con algoritmos de planificación, identificando existencia de soluciones y calidad de las mismas.
- Identificar los factores críticos de los modelos de planificación que pudieran aumentar la complejidad de solución por los algoritmos actuales de planificación y en consecuencia limitarían el uso de estos modelos en tecnologías actuales. A este respecto se buscaría proponer posibles soluciones a los mismos.

1.5 METODOLOGÍA PROPUESTA

La metodología que se utilizó para el desarrollo de esta investigación es la siguiente:

Se realizará un análisis exhaustivo sobre las plataformas y estándares educativos utilizados por los Sistemas Gestores de Aprendizaje, los estándares utilizados, los objetos de aprendizaje, etc., para ver como están representados sus modelos de planificación.

Se estudiará el lenguaje estándar de planificación o PDDL por sus siglas en inglés de Planning Domain Definition Language, el cuál es una herramienta de modelación de la Planificación de Inteligencia Artificial, incluyendo la sintaxis, requerimientos, acciones durativas, tipos, etc. Una vez hecho el análisis, se determinarán aquellas propiedades importantes y necesarias para construir el modelo de planificación tomando en cuenta los trabajos realizados en e-learning y la utilización de

los estándares. Teniendo en cuenta los factores que son considerados en el objetivo anterior, se realiza el modelo en PDDL para la generación de planes de estudio.

Se construyen las librerías de planificadores que se puedan utilizar en el modelo, así como la selección de planificadores actuales, descartando aquellos cuyas características no se adapten a los requerimientos del modelo generado en PDDL.

Se desarrolla un programa generador (en el lenguaje C), el cual genera de manera automática el modelo de planificación, tanto el archivo de dominio como el archivo de problema, modelando diferentes clases de problemas.

Se procede a realizar pruebas de algoritmos de planificación seleccionados con el modelo desarrollado anteriormente. Se realizan pruebas ejecutando el generador con los modelos de planificación con el fin de identificar los factores críticos de los modelos de planificación que pudieran dar problema a los algoritmos actuales de planificación.

1.6 DESCRIPCIÓN DE CAPÍTULOS

Para describir el desarrollo de la presente investigación la tesis se ha estructurado de la siguiente manera:

1. En este capítulo se presenta una introducción al proyecto de investigación, justificación, descripción del problema y cómo se está lidiando actualmente con él, objetivos general y específicos, además de la metodología propuesta.
2. El capítulo 2 es el marco teórico, donde se definen y describen los términos utilizados en esta investigación, tales como Sistemas Gestores de Aprendizaje, Estándares de aprendizaje, Diseños de Aprendizaje, Objetos de aprendizaje, Planificación en Inteligencia Artificial, PDDL (Planning Domain Definition Language, así como Planificadores, que son algoritmos de planificación seleccionados y utilizados para la solución del modelo propuesto.

3. El capítulo 3 describe el problema, la metodología utilizada en esta investigación. Se describe además la modelación a través de técnicas de planificación, las propiedades de dichos modelos y características del problema educativo en un modelo PDDL.
4. El capítulo 4 describe la manera en que se llevó a cabo la experimentación, las características y tamaños de las instancias, las pruebas y resultados de los experimentos, con diferentes tamaños de instancias (clases) variando la cantidad de materias, el tipo de modelo generado y el planificador a resolver.
5. El capítulo 5 muestra las conclusiones, recomendaciones y el trabajo futuro, y cómo este trabajo será la base para diversas líneas de investigación en el área educativa y la Planificación.

CAPÍTULO 2

MARCO TEÓRICO

2.1 E-LEARNING

Nuestro trabajo de investigación queda enmarcado dentro del ámbito de e-learning. Al hablar de e-learning tenemos que señalar que los términos o expresiones que se han utilizado para hacer referencia a ello han sido diferentes: aprendizaje en red, teleformación, aprendizaje virtual, etc. Con todas ellas nos referimos por lo general a la formación que utiliza la red como tecnología de distribución de la información, sea esta red abierta (Internet) o cerrada (intranet) [8].

Son muchas las definiciones y traducciones más o menos literales del término e-Learning. Desde aprendizaje electrónico a virtual pasando por aprendizaje flexible. Una aproximación conceptual, podría ser la siguiente: “una metodología de enseñanza-aprendizaje basada en el uso de contenidos educativos distribuidos a través de un soporte electrónico” [14].

Se trata, en definitiva de relacionar, tres ideas clave: una metodología de enseñanza y de aprendizaje; contenidos educativos; y el potencial del acceso a la información, principalmente vía Internet, o sea, tecnologías de información y comunicación. Es decir, el uso de contenidos electrónicos o digitales a través de tecnologías de información y comunicación en el ámbito educativo.

Se observa que después de utilizar el Internet en educación, esta modalidad no es válida únicamente para la educación a distancia, es válida para la formación a lo

largo de la vida, la formación continua y un excelente y necesario complemento a la formación presencial de los jóvenes universitarios o profesionales. Es una modalidad para aquellos que necesitan organizar su bien máspreciado, el tiempo. Aunque hemos de mencionar que estudiar usando tecnologías no quiere decir que el aprendizaje es rápido o fácil, ya que cualquier proceso de aprendizaje requiere esfuerzo y tiempo [14].

Con lo anterior vemos que hoy en día necesitamos una metodología específica que tenga en cuenta el contexto en el que se produce el proceso educativo, un contexto asíncrono, propiciado por Internet y sus sistemas de comunicación, o bien un contexto híbrido, resultado de la mezcla inteligente de la presencialidad y la no presencialidad [14]. Y es este enfoque el que estamos tomando para dirigir nuestro trabajo.

2.1.1 SISTEMAS GESTORES DE APRENDIZAJE

Una de las herramientas en este marco de referencia (e-learning) que ayudan a la administración de procesos educativos digitales son los Sistemas Gestores de Aprendizaje, LMS de sus siglas en inglés Learning Management System. Estos Sistemas Gestores de Aprendizaje (LMS) “son sistemas informáticos en servidores de internet/intranet que se encargan de administrar cursos virtuales” [3].

Los LMS son sistemas especializados en tecnologías de aprendizaje basados principalmente en internet que se utilizan para dar información y entrenamiento, todo ello en un paradigma abierto de aprendizaje a distancia [31]. Entre las actividades básicas de los Sistemas Gestores de Aprendizaje se observan:

- La gestión de usuarios, ya sea el maestro, el estudiante y el administrador del sistema.
- La gestión administrativa de los cursos virtuales, dando seguimiento a las actividades que el alumno desarrolla con la herramienta.
- La gestión de herramientas de comunicación, desde el punto de vista básico

pueden ser foros, e-mail, chat, videoconferencia, etc.

Una vez mencionado lo anterior, he de mencionar que no es nuestro propósito dar una explicación a detalle de cómo es que trabajan los LMS, ya que nosotros en este trabajo de tesis no lo utilizamos dentro de nuestra metodología de trabajo. Además de que, nuestro trabajo, en un futuro, podría ser una herramienta más de un LMS. Lo que queremos concluir aquí es que nuestra metodología pudiera utilizarse dentro de un LMS.

2.1.2 ESTÁNDARES DE APRENDIZAJE

Un estándar no es más que un conjunto de reglas o normas que especifican cómo debe realizarse un determinado servicio, cómo debe producirse un determinado producto o cómo debe realizarse un determinado proceso de modo que se garantice una cierta calidad y compatibilidad con otros productos o servicios [42].

Estos estándares son generados o bien por organizaciones internacionales ya sean públicas o privadas, e incluso por organizaciones gubernamentales.

Dentro del manejo de los LMS están inmersos estándares tecnológicos de aprendizaje, es decir un conjunto de reglas en común para las compañías dedicadas a la tecnología de e-learning, donde establecen cómo los fabricantes pueden construir cursos on-line y las plataformas sobre las cuales son impartidos estos cursos de tal manera que puedan interactuar unas con otras.

En la actualidad, dos normas son las que dominan el mercado de e-learning [6]. Estos son el estándar SCORM, de la iniciativa ADL y el estándar AICC (Aviation Industry CBT [Computer-Based Training] Committee) de la organización AICC.

A continuación mencionamos los estándares más utilizados:

- El estándar SCORM (del inglés Shareable Content Object Reference Model - Modelo de referencia para contenidos compatibles) [6] hace posible crear con-

tenidos que puedan importarse dentro de LMS diferentes, siempre que éstos soporten la norma SCORM. Este estándar ha sido desarrollado por ADL (Advanced Distributed Learning) con el objeto de contar con un modelo estándar internacional capaz de aglutinar los distintos modelos existentes hasta la fecha en el mercado del e-Learning.

- IMS (Instructional Management System Project) de Global Learning Consortium, Inc.(<http://www.imsproject.org/>). Cuya misión es apoyar el uso de las tecnologías en la enseñanza y cuya principal actividad es desarrollar un conjunto de especificaciones y software prototipo para facilitar el crecimiento y la viabilidad de aprendizaje distribuido en el Internet [34].
- ARIADNE (Alliance of Remote Instruction Authoring and Distribution Networks for Europe),(<http://www.ariadne-eu.org/>). Se enfoca en el desarrollo de herramientas y metodologías para producir, manejar y reusar elementos pedagógicos basados en computadora y planes de estudios soportados por Internet [5].
- IEEE ((Institute of Electric and Electronic Engineers) Learning Technology Standards Committee). Esta es una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Dentro de esta asociación se ha integrado un comité encargado de los estándares técnicos, buenas prácticas y guías recomendadas para el aprendizaje de tecnologías.

Además de los estándares antes mencionados, el IEEE desarrolló el estándar LOM (Learning Object Metadata) que según la IEEE se define como “los atributos necesarios para describir adecuadamente un Objeto de aprendizaje” [12]. Este se crea para complementar el trabajo directo en los estándares que se centra en permitir múltiples objetos de aprendizaje (descritos en la siguiente sección) para trabajar juntos dentro de un entorno de aprendizaje abierto distribuido. En donde su finalidad es permitir a los estudiantes y/o profesores buscar, evaluar, adquirir, Objetos de aprendizaje.

Al realizar el análisis de los trabajos relacionados a nuestra investigación, se observa que éstos hacen uso de LMS con el manejo de algunos estándares [20], [18], [10], [19], [29], [30], [31].

Ésto nos da la pauta para analizar cómo podemos utilizar esta información y crear no un curso en línea, sino modelar el currículum de un estudiante, de licenciatura, por ejemplo, donde pueda utilizar la tecnología como una herramienta de apoyo que ayude de una manera individual a mejorar el desempeño del estudiante de acuerdo a objetivos, no limitando la representación de los modelos a estos estándares, ya que algunas limitaciones observadas, es que existen algunas propiedades como el tiempo, desempeño del estudiante, acumulación de “scores”, entre otras, que se desarrollan en el ambiente educativo del mundo real que no pueden ser representadas adecuadamente.

2.1.3 OBJETOS DE APRENDIZAJE

Aunque en la actualidad existe polémica con respecto a lo que es un Objeto de Aprendizaje [35], [40], podemos decir que “Los Objetos de aprendizaje (LO-Learning Object) son descritos como cualquier entidad, digital o no digital que puede ser utilizado, reutilizado o referenciado durante el aprendizaje apoyado en la tecnología” [12].

Asimismo “un objeto de aprendizaje (LO) puede basarse en un texto electrónico, una simulación, un sitio web, una imagen gráfica, una película QuickTime, un applet de Java o cualquier otro recurso educativo que se puede utilizar en el aprendizaje” de un objetivo particular del currículum académico a cubrir [32].

Tomando en cuenta su definición y los recursos que pueden ser utilizados como objetos de aprendizaje, en este trabajo, comprenden un amplia gama de *actividades de aprendizaje* tales como lecturas, ejercicios, experimentos, diapositivas, videos, imágenes, participación de foros, chat, etc.

De acuerdo a lo anterior, utilizamos este concepto para modelar las actividades

que un estudiante pudiera realizar, para por ejemplo, aprobar un subtema de una materia de su currículum, en donde además se vincula ese LO con un recurso, ya sea una computadora, un libro, etc. Aunque debemos puntualizar que no es nuestro objetivo, su creación, etiquetado ni resguardo.

2.2 EDUCACIÓN

Una vez descritos algunos conceptos de e-Learning, es necesario describir los conceptos de educación que se utilizarán en el transcurso de este trabajo, para introducir al lector.

2.2.1 OBJETIVOS DE APRENDIZAJE

Los objetivos de aprendizaje establecen unas metas o conductas concretas y específicas del alumno y cuya adquisición acerca progresivamente el logro del fin determinado. Se trata de pasos intermedios que permiten conseguir las metas o fines que se pretenden, y se refiere a aquello que el alumno será capaz de hacer al finalizar la acción formativa, en unas condiciones dadas y con unos medios determinados [1].

Así, los *Objetivos de Aprendizaje* describen lo que el estudiante va a ser capaz de aprender, entender o hacer como resultado del desarrollo de una lección o tema que involucra un conjunto de *Actividades de Aprendizaje* por medio de las cuales se obtienen nuevos conocimientos, habilidades o aptitudes.

2.2.2 DISEÑO INSTRUCCIONAL

Desde un punto de vista didáctico, la *instrucción* consiste en la aplicación práctica de las teorías de aprendizaje que a su vez requieren su concreción a través de métodos que se acomoden a las distintas situaciones del aprendizaje. Estos métodos constituyen *teorías de diseño instruccional* la cual se define como “una teoría que ofrece una guía explícita sobre como enseñar a aprender y mejorar”. Las cuales

se caracterizan por estar orientadas a la practica e identifica métodos educativos (modos de favorecer y facilitar el aprendizaje) [2].

Del mismo modo, se define *Diseño Instruccional* como aquel que integra un conjunto de *actividades de aprendizaje* con el ambiente de aprendizaje y permite estructurar dichas actividades en base a teorías pedagógicas que les permitan mejorar su eficiencia, eficacia y atractivo dependiendo de las características de cada grupo de estudiantes [31]. En otras palabras, el diseño instruccional es una secuencia personalizada de objetos de aprendizaje.

2.2.3 DISEÑOS DE APRENDIZAJE

Se les llama *Diseños de aprendizaje LD* a la especificación de los diseños instruccionales estándar, de manera que podríamos decir que son sinónimos, solo que los diseños de aprendizaje se utilizan para referirse a una especificación estándar y el diseño instruccional para hablar del término educativo general.

Así, un Diseño de Aprendizaje (Learning Design) es definido en la especificación IMS como “una descripción de un método que permite a los alumnos alcanzar ciertos objetivos de aprendizaje por medio del desarrollo de ciertas actividades de aprendizaje en un cierto orden en el contexto de un cierto ambiente de aprendizaje”. Otra definición es, la aplicación de un modelo pedagógico para un objetivo de aprendizaje, un grupo objetivo y un contexto determinado [22].

Entonces vemos que los Diseños de aprendizaje (LD-Learning Design) son la aplicación del conocimiento del diseño de aprendizaje cuando se desarrolla una unidad completa de aprendizaje, por ejemplo, un curso, una lección, un currículum, o un evento de aprendizaje [26].

Los Diseños de aprendizaje son básicamente un plan de actividades educativas que se llevan a cabo con cierto orden con la finalidad de cumplir cierto objetivos de aprendizaje, y el objetivo de esta Tesis es diseñar los modelos que nos permitan generar dichos planes educativos de forma autónoma utilizando técnicas de IA.

2.3 PLANIFICACIÓN DE INTELIGENCIA ARTIFICIAL

Como lo mencionamos en la sección anterior, partiendo de la necesidad de generar planes de estudio personalizados que nos permitan cubrir un conjunto de objetivos de aprendizaje, emplearemos técnicas de Planificación de Inteligencia Artificial.

Russell y Norvig en [37] definen la *Inteligencia Artificial* como el “estudio de la acción racional”, lo que significa que la Planificación - la elaboración de un plan de acción para alcanzar un objetivo - es una parte crítica y racional para la consecución de un conjunto de objetivos.

La *Planificación* en Inteligencia Artificial (IA) es la disciplina que centra su proceso de solución de problemas en “la búsqueda y articulación de una secuencia de acciones que permiten alcanzar un objetivo” [38]; es decir, permite al agente utilizar creencias sobre acciones y sus consecuencias para buscar una solución dentro del más abstracto espacio de los planes, no en el de situaciones.[36]. Así, los algoritmos de planificación se pueden considerar como demostradores de teoremas de propósito especial capaces de razonar de manera eficiente con los axiomas que describen las acciones.

Por lo tanto, definimos *Planificación* como el proceso de búsqueda y articulación de una secuencia de acciones que permitan alcanzar un objetivo [38].

TENEMOS PUES, QUE LA PLANIFICACIÓN DE IA PUEDE SATISFACER DE UNA MANERA MAS ROBUSTA LAS NECESIDADES DE GENERACIÓN DE *diseños instruccionales*, YA QUE:

- Recaba una gran cantidad de información acerca del mundo circundante para definir un estado inicial del entorno del problema con el cual habrá de enfrentarse.

- Define un conjunto de reglas de actuación en base a determinadas condiciones, precondiciones que una vez que se cumplen, permiten llevar a cabo una o varias acciones que generan efectos sobre el estado del mundo, es decir, definir el dominio por medio del cuál será posible cambiar el estado inicial de un problema para llegar hasta un estado que lo solucione.
- Establece una meta que suele representarse como una acción o conjunto de acciones que se quieran llevar a cabo, además de una serie de condiciones que debe cumplir el mundo circundante una vez que se hayan ejecutado dichas acciones meta.

IDEAS CLAVE DE LA PLANIFICACIÓN [36]:

- La primera de las ideas clave para la planificación es la de la “apertura” de la representación de *estados, metas y acciones*. Los *estados y metas* se representan mediante conjuntos de oraciones; las *acciones*, mediante descripciones lógicas de condiciones previas y efectos. Esto permite al *planificador* establecer conexiones directas entre estados y acciones.
- La segunda clave fundamental en la planificación consiste en que el planificador es libre de añadir acciones al plan siempre que ello sea necesario, en vez de hacerlo de forma gradual empezando por el estado inicial. Al tomar primero decisiones “obvias” o “importantes” el planificador puede disminuir el factor de ramificación correspondiente a elecciones futuras y disminuir la necesidad de retroceder a decisiones arbitrarias. La representación de estados como conjuntos de oraciones lógicas desempeña un papel determinante para hacer posible esta libertad.
- La tercera clave de la planificación consiste en que la mayoría de las partes del mundo son independientes de otras partes. Lo anterior permite elegir una meta conjuntiva y resolverla mediante una estrategia de *divide y vencerás*. Para resolver todo el problema se agrupan todos los planes. Claro, esto es

dependiendo del tipo de problema que se está modelando, porque no todo problema es independiente entre si.

Los algoritmos tipo *divide y vencerás* son eficientes porque casi siempre es más fácil resolver pequeñas cantidades de subproblemas en vez de uno solo grande. Esta estrategia falla cuando el costo que implica la combinación de las soluciones de cada uno de los subproblemas es demasiado alto. Muchos acertijos son de este tipo, por fortuna, el mundo real es un sitio mayormente benigno en el que las submetas tienden a ser casi independientes entre sí. De no serlo, en la totalidad del mundo real sería imposible resolver problemas satisfactoriamente.

2.3.1 ALGORITMOS DE PLANIFICACIÓN

La manera de resolver nuestro modelo (descrito más adelante) es utilizando algoritmos de planificación automatizada, dando como resultado planes de acciones educativas para satisfacer un conjunto de objetivos de aprendizaje para el estudiante.

Para esto se crean los archivos de *dominio y problema* en el lenguaje *PDDL* (PDDL-lenguaje de definición de dominios de planificación, descrito a detalle más adelante), y así los algoritmos de búsqueda que conforman el motor de planificación se encargan de buscar la secuencia de acciones que lleve a alcanzar la meta definida en el archivo de problema. Siendo, por lo tanto, esta secuencia de acciones a la cual se le denomina comúnmente *plan*, la salida proporcionada por un proceso de planificación.

Para una planificación práctica es necesario:

- Restringir el lenguaje utilizado para definir los problemas. Cuando se usa un lenguaje restrictivo, el espacio de búsqueda disminuye.
- Para encontrar una solución emplee un algoritmo de propósito especial, denominado *planificador*, en vez de un demostrador de teoremas de propósito

general. Aunque éstos van de la mano, ya que cada vez que definimos un nuevo lenguaje para la descripción de problemas necesitamos también un nuevo algoritmo de planificación para procesar tal lenguaje.

Tomando en cuenta lo anterior, tenemos que los *planificadores* son algoritmos de propósito especial, los cuales utilizan un lenguaje de planificación formal con una sintaxis, semántica y teoría de la demostración bien definidas [36]. La teoría de la demostración especifica qué es lo que se puede inferir de los resultados de las secuencias de acción y, por lo tanto, cuáles son los planes legales. El algoritmo nos permite encontrar tales planes.

El método clásico empleado actualmente por la mayoría de los planificadores se caracteriza por describir estados y operadores mediante un lenguaje restringido conocido como lenguaje STRIPS, o derivaciones de éste. El lenguaje STRIPS es adecuado para los algoritmos de planificación eficientes, al tiempo que conserva mucho de la expresividad de las representaciones hechas mediante el cálculo de situaciones.

ESPACIO DE ESTADOS

El espacio de estados provee un modelo de acción básica para problemas relacionados con acciones deterministas e información completa [7]. Un *espacio de estados* consiste de un conjunto finito de estados S , un conjunto finito de acciones A , una función de transición de estado f que describe cómo las acciones se asignan de un estado a otro, y una función de costo $c(a,s) > 0$ que mide el costo de hacer una acción a en el estado s . Un espacio de estados extendido con un estado inicial dado s_0 y un conjunto S_G de estados objetivo son llamados *modelos de estado*.

Formalmente un modelo de estado es un tuple $S = \langle S, s_0, S_G, A, f, c \rangle$ donde:

- S es un conjunto finito y no vacío de estados s .
- $s_0 \in S$ es el estado inicial.

- $S_G \subseteq S$ es un conjunto no vacío de estados objetivo.
- $A(s) \subseteq A$ denotan las *acciones aplicables* en cada estado $s \in S$.
- $f(a,s)$ denota la función de transición de estados para todo $s \in S$ y $a \in A(s)$.
- $c(a,s)$ representa el costo de hacer una acción a en el estado s .

2.3.2 PROBLEMAS DE PLANIFICACIÓN

Un problema de planificación en STRIPS es representado por un Tuple $P = \langle A, O, I, G \rangle$ donde A es un conjunto de átomos que comprenden el universo del problema (con estos se forman los estados), O es un conjunto de operadores, $I \subseteq A$ y $G \subseteq A$ codifican el estado inicial y estado objetivo. Los operadores $op \in O$ se asume que todos son “*grounded*” (es decir, con las variables sustituidas por constantes). Cada operador tiene una precondition, listas add y delete (que corresponden a los efectos del operador) denotadas como $Prec(op)$, $Add(op)$ y $Del(op)$ respectivamente. Todos ellos son dados por conjuntos de átomos de A .

Un problema strips $P = \langle A, O, I, G \rangle$ define un espacio de estados $S_P = \langle S, s_0, S_G, A(\cdot), f, c \rangle$ donde:

- $S1$. El estado $s \in S$ son conjuntos de átomos de A .
- $S2$. El estado inicial s_0 es I .
- $S3$. El estado objetivo $s \in S_G$ son tales que $G \subseteq s$.
- $S4$. Las acciones $a \in A(s)$ son los operadores $op \in O$ tal que $Prec(op) \subseteq s$. Es decir, las acciones aplicables al estado s .
- $S5$. La función de transición f asigna un estado s en un estado $s' = s - Del(a) + Add(a)$ para $a \in A(s)$.
- $S6$. Todos los costos de la acción $c(a,s)$ son 1.

PARTES PRINCIPALES DE UN PROBLEMA

Los problemas de planificación se representan mediante oraciones lógicas que describen las tres partes principales de un problema:

- **Estado inicial:** Una oración lógica arbitraria acerca de una situación S_0 . Teniendo en cuenta el siguiente ejemplo sencillo: “Me encuentro en casa, quiero prepararme un cereal pero no tengo leche ni plátanos”, tal oración sería:

$$En(Casa, S_0) \wedge \neg Hay(Leche, S_0) \wedge \neg Hay(Plátanos, S_0)$$

- **Estado meta:** Una consulta lógica para encontrar situaciones adecuadas. En el ejemplo anterior, la consulta sería:

$$\exists s En(Casa, s) \wedge Hay(Leche, s) \wedge Hay(Plátanos, s)$$

- **Operadores:** Conjunto de la especificación de acciones, utilizando la representación de acciones. Por ejemplo, el siguiente es un axioma de estado sucesor relacionado con la acción $Comprar(Leche)$:

$$\forall a, s, Tiene(Leche, Resultado) (a, s) \Leftrightarrow [(a = Comprar(Leche) \wedge En(Supermercado, s) \vee (Hayleche, s) \wedge a \neq Entregar(Leche))]$$

REPRESENTACIONES DE ESTADOS Y OBJETIVOS

En el lenguaje STRIPS, los estados se representan mediante conjunciones de literales básicas en las que no hay funciones, es decir, solamente existen predicados aplicados a signos constantes, posiblemente negados. Por ejemplo, tomando en cuenta el problema sencillo de “Me encuentro en casa y quiero prepararme un cereal, pero no tengo leche, ni plátanos”, el estado inicial del problema se especificaría de la siguiente manera:

$$En(Casa) \wedge \neg Hay(Leche) \wedge \neg Hay(Plátanos) \wedge \dots$$

Aquí podemos ver que las especificaciones de un estado no necesariamente tiene que estar completa. Por otro parte, en muchos sistemas de planificación se

adopta la convención -análoga a la convención “negativa como falla”, (closed-world assumption) usada en la programación lógica- de que si en la descripción del estado no se menciona una determinada literal positiva entonces se toma como falsa dicha literal.

Asimismo, las metas también se caracterizan mediante una conjunción de literales. Tomando en cuenta el ejemplo anterior, la meta se representaría de la manera siguiente:

$$En(Casa) \wedge Hay(Leche) \wedge Hay(Plátanos)$$

En las metas pueden figurar también variables. Por ejemplo, la meta de encontrarse en una tienda en donde vendan leche se representaría así:

$$En(x) \wedge Vende(x, Leche)$$

REPRESENTACIÓN DE ACCIONES

Nuestros operadores STRIPS constan de tres elementos:

- La **especificación de la acción**, es lo que de hecho devuelve el agente al ambiente para así proceder a hacer algo. Cuando está dentro del planificador sirve como nombre de una posible acción.
- La **condición previa** es una conjunción de átomos (literales positivas) que dice qué debe existir (ser verdad) antes de poder aplicar el operador.
- El **efecto** de un operador es una conjunción de literales (positivas o negativas) que dice de qué manera cambia la situación al aplicar el operador.

Para ejemplificar lo anterior se muestra en ejemplo de la sintaxis que se empleará para formar un operador STRIPS que vaya de un sitio a otro:

$$Op(ACCION): Ir(allí), PRECOND: En(aquí) \wedge Ruta(aquí, allí),$$

EFECTO: $En(allí) \wedge \neg En(aquí)$

En este ejemplo se ilustra que todo lo que hay en la condición previa implícitamente se refiere a la situación inmediatamente previa a la acción así como todo lo que hay en el efecto implícitamente se refiere a la situación resultante de la acción.

Al operador que tiene variables se le conoce como *esquema de operador*, puesto que no corresponde a una sola acción ejecutable, sino a una familia de acciones, una por cada concretización de las variables. Por lo general sólo es posible ejecutar operadores totalmente concretizados; nuestros algoritmos de planificación permitirán asegurar que a cada variable corresponda un valor en el momento de aplicar.

Se dice que un operador o es **aplicable** a un estado s siempre y cuando exista una manera de concretizar las variables que tiene o , de manera que cada una de las condiciones previas de o también sea válida en s , es decir, si $Condprev(o) \subset s$. En el estado así obtenido, todas las literales positivas de $Efecto(o)$ son válidas, como también sucede con todas las literales que eran válidas en s , excepto aquellas que eran literales negativas en $Efecto(o)$. Por ejemplo, si en la situación inicial están presentes las literales

$En(Casa), Camino(Casa, Supermercado), \dots$

se puede aplicar la acción $Ir(Supermercado)$, y en la situación resultante están las literales

$\neg En(Casa), En(Supermercado), Camino(Casa, Supermercado), \dots$

Cómo podrán darse cuenta, la condición $En(casa)$ que era verdadera antes de aplicar la acción, es negada en el efecto de dicha acción, porque ahora ya no está en casa, sino en el supermercado.

2.3.3 REPRESENTACIÓN DE PLANES

Antes de poder buscar a través de un espacio de planes, es necesario poder representarlos. Estableceremos lo que es una buena representación de planes mediante la consideración de los planes parciales de un problema sencillo: “ponerse un par de zapatos”. La meta es la conjunción de $ZapatoDerechoPuesto \wedge ZapatoIzquierdoPuesto$, el estado inicial no tiene ninguna literal y los operadores son

$$Op(\text{ACCION: } ZapatoDerecho, \text{ CONDPREV: } CalcetínDerechoPuesto, \\ \text{EFECTO: } ZapatoDerechoPuesto)$$

$$Op(\text{ACCION: } CalcetínDerecho, \text{ EFECTO: } CalcetínDerechoPuesto)$$

$$Op(\text{ACCION: } ZapatoIzquierdo, \text{ CONDPREV: } CalcetínIzquierdoPuesto, \\ \text{EFECTO: } ZapatoIzquierdoPuesto)$$

$$Op(\text{ACCION: } CalcetínIzquierdo, \text{ EFECTO: } CalcetínIzquierdoPuesto)$$

Un plan parcial para este problema constaría de los dos pasos $ZapatoDerecho$ y $ZapatoIzquierdo$. Pero ¿Cuál es el primero? Muchos planificadores aplican el principio de *compromiso mínimo*, que afirma que uno sólo debe preocuparse de tomar decisiones relacionadas con aquello que actualmente interese a uno, dejando las otras decisiones para después.

Si al plan se añade un tercer paso, $CalcetínDerecho$, es necesario estar seguros de que ponerse el calcetín derecho se efectúe antes de ponerse el zapato derecho, es decir (CONDPREV: $NOT(ZapatoDerechoPuesto)$) en la acción $CalcetínDerecho$. Aunque no nos importe cómo se producen en relación con el zapato izquierdo.

Al planificador capaz de representar planes en los que algunos pasos se orde-

nan (antes o después) en relación con los demás y otros pasos no están ordenados se conoce *planificador de orden parcial*[36]. La alternativa es un planificador de *orden total*, en el cual los planes están formados por una sola lista de pasos. Al plan totalmente ordenado que se obtiene de un plan P añadiendo restricciones de ordenamiento se conoce como *linealización* de P .

A un *plan* [36] se le define formalmente como una estructura de datos constituida por: Un conjunto de los pasos del plan. Cada uno de éstos es uno de los operadores del problema. Un conjunto de restricciones para el ordenamiento de los pasos. Las restricciones de ordenamiento tienen la forma $O_i \prec O_j$, que se lee “ O_i antes de O_j ” y significa que el paso O_i debe producirse en algún momento antes del paso O_j (aunque no es necesario inmediatamente antes).

En el plan inicial, antes de efectuar cualquier refinamiento, sencillamente se describe el problema no resuelto. Consta de dos pasos, denominados *Iniciar* y *Terminar*, regidos por la restricción de ordenamiento $Iniciar \prec Terminar$. Tanto *Iniciar* como *Terminar* tienen acciones nulas asociadas, por lo que al momento de ejecutar el plan se les ignora. El paso *Iniciar* no tiene condiciones previas y su efecto consiste en añadir todas las proposiciones que son verdaderas en el estado inicial. En la meta del paso *Terminar* el estado meta es su condición previa y no tiene efecto alguno.

2.3.4 SOLUCIÓN

Al resultado del proceso de Planificación se llama “solución”. El término *solución* según [36] es un *plan* que un agente puede ejecutar y garantiza el logro de la meta. Es decir, una secuencia de acciones que, ejecutada en el estado inicial, da como resultado un estado final que satisface el objetivo.

Decimos que una **solución** es un plan *completo* y *consistente*. Un *plan completo* es aquel en el que cada condición previa de todas las acciones se logra mediante otra acción. Una acción logra una condición si ésta es uno de los efectos producidos por otra acción, y si ninguna otra acción tiene posibilidad de eliminar la condición.

Un *plan consistente* es aquel en el que no hay contradicciones en las restricciones de ordenamiento o enlazamiento. La contradicción se produce cuando tanto $O_i \prec O_j$ como $O_j \prec O_i$ se cumplen, o cuando se cumple que tanto $v=C$ como $v=D$ (para dos constantes diferentes C y D). Tanto \prec como $=$ son transitivos y, por ejemplo, un plan en el que $O_1 \prec O_2$, $O_2 \prec O_3$, y $O_3 \prec O_1$ es inconsistente.

El plan parcial de la figura 2.1 es una solución puesto que se satisfacen todas las condiciones previas. Con base en las definiciones anteriores es fácil darse cuenta de que toda linealización de una solución es también una solución. Por lo anterior, el agente puede ejecutar los pasos en cualquier orden que esté de acuerdo con las restricciones, y aun así se puede garantizar el logro de la meta.

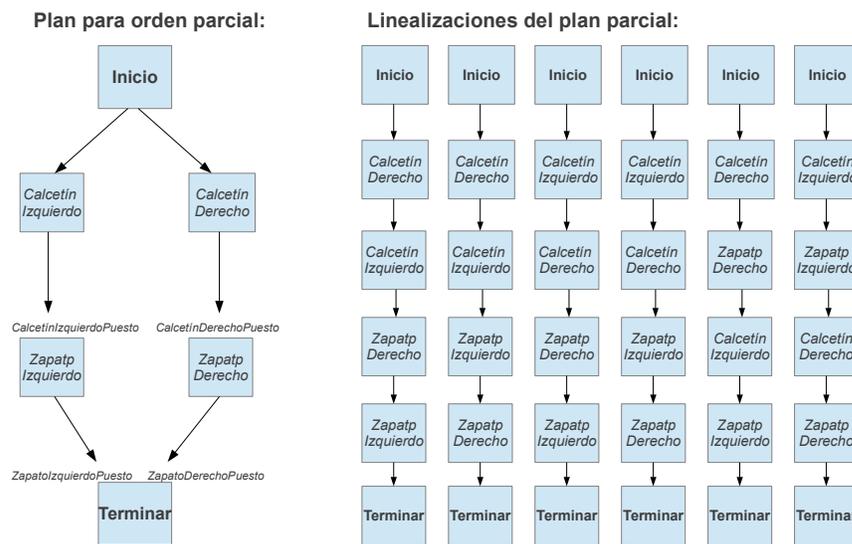


Figura 2.1: Un plan de orden parcial para ponerse zapatos y calcetines (incluidas las condiciones previas a los pasos) y las seis posibles linealizaciones del plan.

2.3.5 PDDL (PLANNING DOMAIN DEFINITION LANGUAGE)

La *Planificación* en Inteligencia Artificial utiliza un lenguaje de modelación el cuál es llamado *PDDL* (acrónimo de Planning Domain Definition Language), que traducido es Lenguaje de Definición de Dominio de Planificación.

PDDL es un lenguaje centrado en las acciones inspirado en las formulaciones “Strips” de problemas de planificación. Éste es una estandarización de la sintaxis para expresar acciones utilizando precondiciones y post-condiciones para describir la aplicabilidad y efectos de las acciones [17]. La sintaxis está inspirada en Lisp [41] (acrónimo de LIST Processing), así que gran parte de la estructura de la descripción del dominio es una lista como Lisp de las expresiones entre paréntesis.

PDDL está organizado en un par de archivos:

- Archivo de *dominio*.- que describe las reglas de actuación,
- Archivo de *problema*.- que describe el estado del mundo circundante y las metas.

ELEMENTOS DE PDDL

A continuación se describen los elementos de PDDL y las acciones.

PDDL define lo que nosotros necesitamos para definir un problema de búsqueda:

1. Estado inicial: Es el estado del mundo en el que iniciará.
2. Operadores: La descripción de los operadores. Es decir, las acciones que están disponibles en un estado y el efecto de aplicar dicha acción
3. Estado final: Son los objetivos que queremos que sean verdad.

ELEMENTOS DE LAS ACCIONES

Los componentes de las acciones son:

- Especificación de la acción: nombre de la acción.

- La condición previa: es una conjunción de átomos (literales positivas) que dice qué debe ser verdad antes de poder aplicar el operador.
- El efecto de un operador: es una conjunción de literales (positivas o negativas) que dice de qué manera cambia la situación al aplicar el operador.

2.3.6 EJEMPLO DEL LENGUAJE PDDL

A continuación se muestra el ejemplo típico de planificación, el **mundo de los bloques** [38]. Este dominio consiste en un conjunto de bloques con forma de cubo situados en una mesa. Los bloques pueden ser amontonados, pero únicamente podemos situar uno sobre otro. Un brazo mecánico puede cambiar bloques de sitio, tanto sobre la mesa como sobre otros bloques. El brazo puede tomar sólo un bloque por cada instante de tiempo, de modo que no puede tomar uno si aún no ha soltado otro anterior. El objetivo será construir una o más configuraciones de bloques, que quedarán especificados en términos de cuántos bloques están sobre cuántos otros bloques.

Para esto, necesitamos describir los componentes de una tarea de planificación PDDL [27], las cuales se separan en dos archivos:

En el Archivo Dominio están:

- Los predicados: Estos representan relaciones entre los objetos. Nos ayudan a describir un problema del mundo real tratando de representar los conceptos de nuestro problema a través de relaciones entre los objetos que lo conforman.
- Acciones / Operadores: Maneras de cambiar el estado del mundo.

En el Archivo Problema están:

- Objetos: Las cosas en el mundo que nos interesan.
- Estado inicial: El estado del mundo en el que se empieza. Es decir, el punto de partida de la búsqueda.

- Especificación Objetivo: Comprueba si el estado actual corresponde a una solución del problema.

En el ejemplo del mundo de los bloques, tenemos en la figura 2.2 el *estado inicial* en donde tenemos cuatro objetos que son los bloques A, B, C, D y todos los bloques están sobre la mesa.

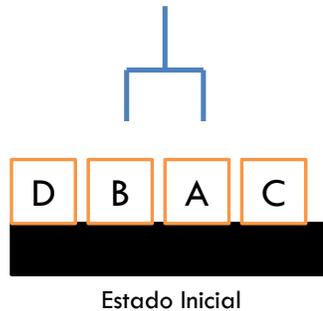


Figura 2.2: Ejemplo del mundo de los bloques- Estado inicial

El objetivo es que su estado final sea que los bloques estén apilados de la siguiente manera: el bloque A es el único que está sobre la mesa, luego el bloque B está apilado sobre el bloque A, el C sobre el B, y el D sobre C respectivamente. Véase figura 2.3.

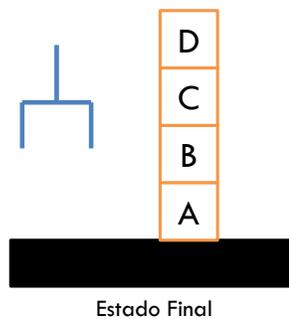


Figura 2.3: Ejemplo del mundo de los bloques- Estado final

DOMINIO

Empezamos definiendo el *dominio* el cual consta de un nombre, en la figura 2.4 se muestra la sintaxis. *Los requerimientos* en nuestro ejemplo sencillo es STRIPS,

figura 2.5. Y los predicados, figura 2.6, representan los conceptos de nuestro problema a través de relaciones entre los objetos que lo conforman, por ejemplo, en el caso del predicado $on(a,b)$ estoy estableciendo una relación entre a y b , en este caso es que el bloque a se encuentra encima del bloque b .

En nuestro ejemplo tenemos 5 predicados: $(on ?x ?y)$ hace referencia a que el objeto x se encuentra sobre el objeto y , $(ontable ?x)$ indica que el objeto x está sobre la mesa, $(clear ?x)$ indica que el objeto x esta “limpio”, es decir que no tiene ningún objeto apilado, $(handempty)$ indica que el brazo mecánico está disponible, $(holding ?x)$ contrario al anterior predicado, indica que el brazo mecánico tiene sostenido un objeto.

Nombre del Dominio:

```
(define (domain BLOCKS)
```

Figura 2.4: Ejemplo del mundo de los bloques- Nombre de Dominio

Requerimientos:

```
(:requirements :strips)
```

Figura 2.5: Ejemplo del mundo de los bloques- Requerimientos del Dominio

Predicados:

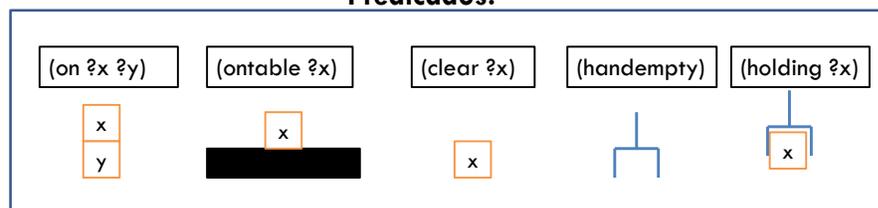


Figura 2.6: Ejemplo del mundo de los bloques- Predicados del Dominio

ACCIONES

Después de tener definido lo anterior, se procede a modelar las *acciones* posibles a ejecutar. En nuestro ejemplo tenemos cuatro acciones que son: “pick up”, “put down”, “stack” y “unstack”.

acción: “pick-up”. Esta acción hace referencia a “levantar un bloque de la mesa”. En la figura 2.7 se observa cómo se representa esta acción, tanto en sintaxis como en imagen para ilustrar. Nótese que la acción debe tener un nombre. Los componentes de esta acción son:

- *parámetros*: aquí solo necesita un bloque.
- *precondición*: es para indicar los requisitos que se deben cubrir para poder aplicar esta acción, los cuales se definen con los *predicados*: “clear”, que dice que el objeto no tiene nada encima, “ontable” que el objeto está sobre la mesa y “handempty” que indica que el brazo mecánico esta libre.
- *efectos*: son las consecuencias de aplicar esta acción, y al igual que en las precondiciones se utilizan los predicados que definimos anteriormente, así como sus negaciones. En este ejemplo sencillo los predicados utilizados son “holding” que indica que el objeto esta siendo tomado, y las negaciones de los predicados “clear”, “ontable”, “handempty”.

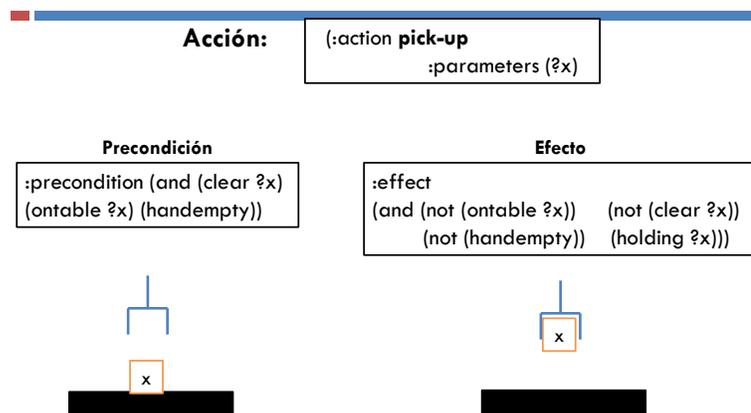


Figura 2.7: Ejemplo del mundo de los bloques- Acción pick up y su componentes.

acción: “put-down”. Esta acción hace referencia a “dejar un bloque sobre la mesa”, en la figura 2.8 se observa cómo se representa esta acción, tanto en sintaxis como en imagen para ilustrar. Los componentes de esta acción son:

- *parámetros*: aquí solo necesita un bloque.

- *precondición*: es estar sosteniendo el bloque x , los predicados para indicar esto son: “holding”.
- *efectos*: como la precondición era estar sosteniendo el bloque x , en efectos se hace la negación de que el bloque está sostenido “holding”, y ahora decimos que el brazo mecánico esta disponible con el predicado “handempty”, decimos además que el bloque x está “clear” y “ontable” .

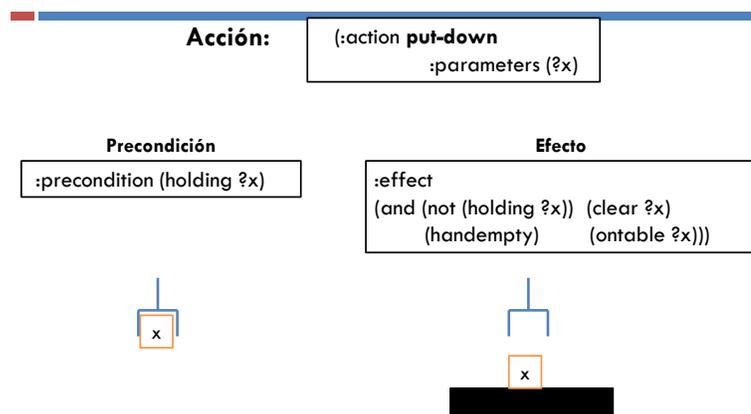


Figura 2.8: Ejemplo del mundo de los bloques- Acción Put-down y su componentes

acción: “stack” que hace referencia a “apilar un bloque”, en la figura 2.9 se observa cómo se representa esta acción, tanto en sintaxis como en imagen para ilustrar. Los componentes de esta acción son:

- *parámetros*: aquí necesita dos bloques, el que será apilado y sobre el cual se apilará.
- *precondición*: que el bloque y sobre el cual se quiere apilar, esté sin nada encima, es decir, “clear”, y el bloque x esté siendo sostenido por el brazo mecánico “holding”.
- *efectos*: la negación de toda la precondición, es decir, que el bloque x ya no está sostenido la negación de “holding”, y ya que éste se apiló en el bloque y , y ya no está “clear”. Ahora el bloque que está “clear” es el apilado, el bloque

x , también están los predicados del brazo vacío “handempty” y el predicado “on” que indica que un bloque esta en otro.

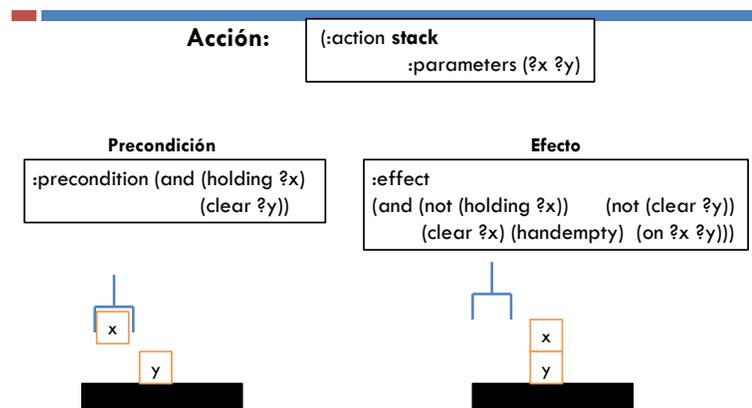


Figura 2.9: Ejemplo del mundo de los bloques- Acción stack y su componentes

acción: “unstack” esta es la acción contraria a “stack”. Se intenta quitar un bloque que esta apilado en otro, en la figura 2.10 se observa cómo se representa esta acción, tanto en sintaxis como en imagen para ilustrar. Los componentes de esta acción son:

- *parámetros*: aquí necesita dos bloques: “x” y “y”.
- *precondición*: como la acción es “desapilar” un bloque de otro, necesitamos tener algo “apilado”, así que la precondición es que el bloque “x” esté en “y” (on), que el bloque “x” no tenga nada encima (clear) y que el brazo mecánico esté disponible (handempty).
- *efectos*: el brazo mecánico sostiene al bloque “x”, “y” no tiene nada encima, se agregan la negación de que el bloque “x” no tiene nada encima (clear), la negación de que el brazo esté disponible (handempty) y la negación de que esté el bloque “x” en “y”.

PROBLEMA

Una vez que tenemos el *Dominio*, procedemos a definir el *problema*, cabe aclarar que una vez que tenemos el dominio definido, podemos modelar varios problemas, es

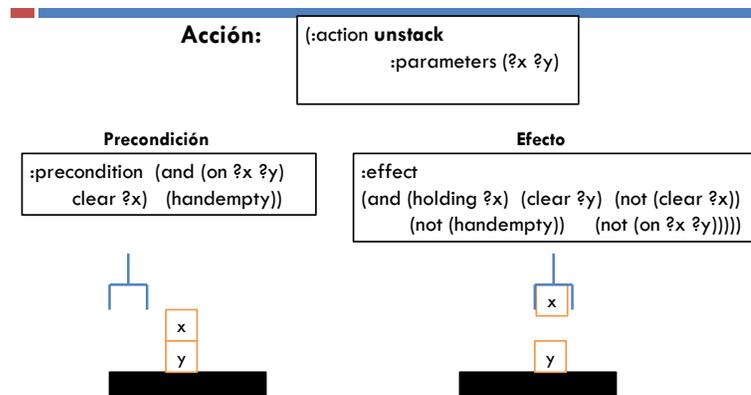


Figura 2.10: Ejemplo del mundo de los bloques- Acción unstack y sus componentes

decir, diferentes instancias o escenarios que utilizan el mismo conjunto de operadores para solucionarlos. En nuestro ejemplo el problema tiene cuatro objetos, y el objetivo es que estén apilados uno sobre el otro siguiendo el orden A, B, C, D. Esto sería un escenario, pero pudiera ser que en lugar de estar todos los bloques apilados uno encima del otro, solo se quieran dos, o en un orden distinto.

Al igual que en el dominio, el problema tiene un “nombre” como en la figura 2.11, es importante definir además, a que dominio pertenece, figura 2.12 y los objetos a modelar, en nuestro ejemplo son cuatro, véase figura 2.13

Nombre del problema

```
(define (problem BLOCKS-4-0))
```

Figura 2.11: Ejemplo del mundo de los bloques- Nombre de Problema

A que dominio pertenece

```
(:domain BLOCKS)
```

Figura 2.12: Ejemplo del mundo de los bloques- A que dominio pertenece el Problema

ESTADO INICIAL Y ESTADO FINAL

Es en el problema en donde definimos el estado inicial y el objetivo. En nuestro ejemplo de los bloques tenemos que el estado inicial es que los cuatro bloques están

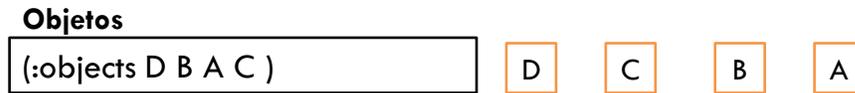


Figura 2.13: Ejemplo del mundo de los bloques- Objetos del problema

sobre la mesa (ontable), estos mismos bloques no tienen nada apilado (clear) y el brazo mecánico esta disponible. La figura 2.14 muestra como se define este estado inicial en pddl.

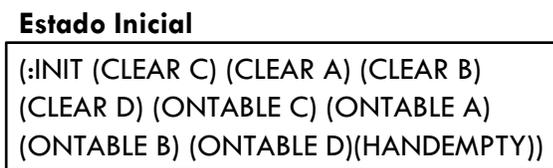


Figura 2.14: Ejemplo del mundo de los bloques- Estado Inicial definido en pddl

Asimismo el estado final que queremos modelar es que los bloques estén apilados en el siguiente orden, D sobre C, C sobre B, B sobre A. Esto lo mostramos en la figura 2.15

SOLUCIÓN

Una vez que tenemos estos dos archivos, que es nuestro *modelo* del mundo de los bloques, se procede a resolver mediante cualquier *planificador*.

La solución para este problema sería el plan mostrado en la tabla 2.1.

2.3.7 PDDL 2.1

PDDL es el lenguaje estándar para la codificación de dominios de planificación, éste fue desarrollado por Drew McDermott y fue dado a conocer en la International Planning Competition de 1998 [28]. Ya que está basado en las formulaciones Strips de problemas de planificación.

Estado Final

```
(:goal (AND (ON D C) (ON C B)
(ON B A)))
)
```

Figura 2.15: Ejemplo del mundo de los bloques- Estado Final definido en pddl

Paso 0:	Pick-up D
Paso 1:	Stack D C
Paso 2:	Pick-up B
Paso 3:	Stack B A
Paso 4:	Unstack D C
Paso 5:	Put-down D
Paso 6:	Pick-up C
Paso 7:	Stack C B
Paso 8:	Pick-up D
Paso 9:	Stack D C

Tabla 2.1: Plan del modelo del “mundo de los bloques”

Para PDDL se han desarrollado varias versiones, en donde se han ido agregando mejoras, nosotros utilizamos la versión 2.1 [17] ya que ésta tiene las características que necesitamos para modelar los planes de estudio, tales como: *strips*, *typing*, *fluent* y *durative actions*, que explicaremos durante todo este trabajo.

PDDL 2.1 ha sido diseñado para ser compatible con el fragmento de PDDL que ha sido utilizado desde 1998. Esta compatibilidad apoya el desarrollo de recursos que ayuden a establecer una base científica para el campo de la planificación de la IA. Por otra parte, El PDDL original de McDermott proporciona una base clara y bien entendida para el desarrollo e incorpora una serie de principios de diseño que consideramos que es importante conservar [17].

Se mencionan las siguientes garantías de compatibilidad con versiones anteriores:

- 1. Todos los dominios existentes PDDL (de uso común) son válidos en dominios PDDL 2.1. Esto es importante para permitir que bibliotecas existentes de problemas de referencia sigan siendo válidas.
- 2. Los planes válidos en PDDL son válidos en planes PDDL 2.1.

Una importante contribución en el desarrollo de PDDL2.1 es un medio por el que los diseñadores de dominios puedan proporcionar funciones alternativas objetivas que puedan utilizarse para juzgar el valor de un plan. El uso de números en un dominio proporciona una plataforma para medir el consumo de recursos críticos y otros parámetros. Esto es muy importante para muchas aplicaciones prácticas de planificación en el que la calidad del plan podría depender de un número de interacciones con los factores dependientes del dominio.

Una decisión de diseño en el lenguaje fue la de separar las descripciones de las acciones parametrizadas que caracterizan el comportamiento de dominio para la descripción de objetos específicos, las condiciones iniciales y los objetivos que caracterizan una instancia de un problema. Por lo tanto, un problema de planificación es creado por el emparejamiento de una descripción de un dominio con una descripción del problema. La descripción del mismo dominio se puede combinar con muchas descripciones de problemas diferentes para producir diferentes problemas de planificación en el mismo dominio. La parametrización de las acciones depende de la utilización de variables que representan términos del problema ejemplo - que se crean instancias de objetos a partir de una instancia de un problema específico cuando una acción es “ground” para su aplicación. Los pre-y post-condiciones de las acciones se expresan en proposiciones lógicas construidas a partir de los predicados y términos de argumentos (objetos de una instancia de problema) y conectivas lógicas.

Aunque el núcleo de PDDL es un formalismo STRIPS, la lengua se extiende más allá. El poder expresivo extendido incluye la capacidad de expresar un tipo de es-

estructura para los objetos en un dominio, escribir “typing” los parámetros que aparecen en las acciones y restringir los tipos de argumentos a los predicados, acciones con precondiciones negativas y efectos condicionales y el uso de la cuantificación en la expresión de ambos pre-y post-condiciones. Estas extensiones son esencialmente aquellas propuestas como ADL.

CARACTERÍSTICAS DE PDDL 2.1

Expresiones numéricas:

En PDDL 2.1 se propuso una sintaxis definitiva para las expresiones de *fluents numéricos* (fluents). Las expresiones numéricas se construyen utilizando operadores aritméticos de expresiones numéricas primitivas, los cuales son valores asociados con la tupla de los objetos de dominio por las funciones de dominio. Cabe mencionar que la sintaxis que se utiliza es prefija para todos los operadores aritméticos, incluyendo los predicados de comparación, con el fin de simplificar el análisis.

Un ejemplo que podríamos mencionar, con respecto a lo anterior, es en el caso de nuestro modelo, en donde definimos la duración de una *actividad de aprendizaje*. Ésta se definiría en el *dominio* de la manera siguiente: (DurationLA ?oa - LA). Posteriormente en el archivo *problema* le asignamos el valor de esa actividad de aprendizaje en específico (utilizando la sintaxis prefija) (= (DurationLA L1111) 25). Así, decimos que la actividad de aprendizaje L1111 tiene una duración de 25 (unidades de tiempo).

Condiciones:

Las condiciones en las expresiones numéricas son siempre las comparaciones entre pares de expresiones numéricas. Los efectos pueden hacer uso de una selección de las operaciones de asignación con el fin de actualizar los valores de las expresiones numéricas primitivas. Estos incluyen la asignación directa y asignaciones relativas (tales como aumento y disminución). Los números no se distinguen en sus posibles funciones, por lo que los valores pueden representar, por ejemplo, las cantidades de recursos, acumulación de servicios públicos, índices o contadores. En esta versión se

decidió sólo permitir funciones con valores numéricos.

Métricas del Plan:

Las métricas del plan especificarán la base sobre la cual se evaluará un plan para un problema particular. Los mismos estados iniciales y finales pueden producir planes óptimos totalmente diferentes dadas diferentes métricas del plan. Por supuesto, un planificador podría no optar por utilizar la métrica para orientar su desarrollo de una solución, sólo para evaluar una solución post hoc. Este enfoque podría dar lugar a planes sub-óptimos, e incluso de mala calidad, pero es un enfoque pragmático para el manejo de métricas, las cuales han sido ampliamente utilizadas en las competiciones.

Una de las métricas más utilizadas es el valor total de tiempo (total-time) que puede ser usado para referirse a la duración temporal de todo el plan. Otros valores deben ser construido a partir de expresiones numéricas primitivas definidas dentro de un dominio y manipulados por las acciones del dominio. Como consecuencia de ello, los indicadores del plan sólo puede expresar métricas no-temporales en PDDL2.1 utilizando expresiones numéricas en el dominio.

Cualquier expresión aritmética puede ser utilizada en la especificación de una métrica - no hay ningún requisito de que la expresión sea lineal. Es responsabilidad del diseñador de dominio de asegurarse de que la métrica del plan esté bien definida (por ejemplo, no implican una división por cero). Se muestra a continuación un ejemplo de la sintaxis de una métrica de un ejemplo de gasto de gasolina:

(: metric minimize (+ (2 (fuel-used car)) (fuel-used truck)))*

Acciones durativas:

Para manejar los trabajos recientes de planificación temporal se ha desarrollado dos formas de acción durativa que permiten la especificación sólo de formas restringidas de condiciones de tiempo y efectos en su descripción. Las dos formas son acciones durativas discretas y acciones durativas continuas.

El modelado de las relaciones temporales en una acción durativa discreta se realiza por medio de condiciones y efectos anotados temporalmente. Todas las condiciones y los efectos de las acciones durativas debe ser anotadas temporalmente. La anotación de una condición se hace explícita si la proposición asociada se debe mantener al *inicio* del intervalo (el punto en el que se aplica la acción), el *final* del intervalo (el punto en el que los efectos finales de la acción se afirman) o en el intervalo del *inicio* y el *fin* (invariable durante la duración de la acción). La anotación de un efecto se hace explícito si el efecto es inmediato (lo que sucede en el inicio del intervalo) o tardío (si sucede al final del intervalo). Ningún otro punto en el tiempo es accesible, por lo que toda la actividad discreta se realiza en los puntos identificados *inicio* (and) y *final* (end) de las acciones del plan. Si se requiere mantener un intervalo que esté abierto en ambos extremos (*start* y *end*) se utiliza la construcción *over all*.

Para ejemplificar lo anterior veamos la figura 2.16 donde se muestra cómo se utilizan las anotaciones de inicio (at start) y fin (at end) de una acción durativa en PDDL. Esta es una acción de selección de una actividad de aprendizaje de nuestro modelo, en la cual se requiere un recurso, en este caso la computadora.

```
(:durative-action CHOOSE-LA-nothasreqs
:parameters (?s - student ?oa - LA ... ?eq - resource)
:duration (= ?duration (DurationLA ?oa))
:condition (and
...
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
:effect (and
...
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1)))
```

Figura 2.16: Ejemplo de acciones durativas en PDDL

En esta acción se representa que para poder realizar la acción de selección de una LA, además de otras condiciones, se necesita que esté disponible el recurso requerido por la actividad de aprendizaje. La instrucción en PDDL es (*at start (> (quantity-resource increase ?eq) 0*)). Como efecto de la condición se tiene que al inicio la cantidad de dicho recurso disminuye en uno, es decir, el estudiante lo

está utilizando, esto en PDDL es $(at\ start\ (decrease\ (quantity-resource\ ?eq)\ 1))$. Y por último, al final de ese efecto ese recurso es liberado nuevamente, $(at\ end\ (increase\ (quantity-resource\ ?eq)\ 1))$. En la figura 2.17 se muestra de una manera gráfica lo anterior.

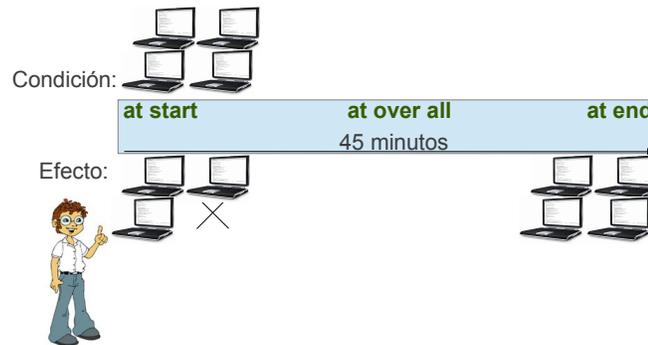


Figura 2.17: Ilustración de intervalos de tiempo de las acciones durativas. Se tiene que la acción dura 45 minutos y el recurso está asociado a una actividad de aprendizaje.

Por otro lado, si uno desea especificar que un hecho p se mantiene en el intervalo cerrado durante la duración de una acción durativa, entonces son requeridas las tres condiciones: $(at\ start\ p)$, $(over\ all\ p)$ y $(at\ end\ p)$.

SEMÁNTICA DE PDDL2.1

En PDDL 2.1 se añadió cuatro extensiones significativas con respecto a la planificación clásica y la semántica Lifschitz desarrollado para STRIPS. Estos son:

- la introducción del tiempo, de manera que los planes describen el comportamiento con respecto a una línea de tiempo real;
- relacionada con la primera extensión, el tratamiento de la concurrencia - acciones se pueden ejecutar en paralelo, que pueden conducir a planes que contienen procesos concurrentes que interactúan (aunque estos procesos se encapsulan en acciones durativas en PDDL2.1);
- una extensión para manejar valores numéricos *fluents*;

- el uso de efectos condicionales, tanto solos como en combinación con todas las extensiones anteriores.

La semántica se basa en un conocido modelo de transición de estados. Los requisitos de la semántica puede reducirse a cuatro elementos esenciales:

- 1. Para definir lo que es un *estado*. La introducción de tiempo y valores numéricos complican la definición usual de un estado como un conjunto de átomos.
- 2. Para definir cuando un Estado satisface una fórmula proposicional que representa una condición o requisito objetivo de una acción. Una extensión de la interpretación habitual de un estado como una valoración en la que un átomo es verdadero si y sólo si el átomo está en el estado (la suposición del mundo cerrado) es necesario para controlar los valores numéricos en el estado.
- 3. Para definir la transición de estado inducido por la aplicación de una acción. La regla de actualización para el estado lógico debe complementarse con una explicación de las consecuencias para la parte numérica del estado.
- 4. Para definir cuándo dos acciones se pueden aplicar simultáneamente y cómo su aplicación *concurrente* afecta a la aplicación de dichas acciones de forma individual.

2.3.8 COMPETENCIAS DE PLANIFICADORES

La Conferencia Internacional sobre Planificación y Programación Automática (ICAPS) es el principal foro para investigadores y profesionales en la planificación y programación - dos tecnologías que son fundamentales para áreas como la manufactura, sistemas de espacio, ingeniería de software, robótica, educación y entretenimiento. La conferencia ICAPS resultado de la fusión de dos conferencias bianuales, la Conferencia Internacional sobre Planificación y Programación de Inteligencia Artificial (AIPS por sus siglas en inglés de International Conference on Artificial Intel-

ligence Planning and Scheduling) y la Conferencia Europea de Planificación (ECP por sus siglas en inglés de European Conference on Planning) [25].

Los objetivos principales de ICAPS son fomentar el campo de la planificación y la programación automática mediante la organización de reuniones técnicas, incluida la conferencia anual ICAPS, a través de la organización de escuelas de verano, cursos y actividades de capacitación en varios eventos, a través de la organización de competencias de planificación y la programación, la evaluación comparativa y otros medios de promoción y evaluación del estado del arte en el campo, mediante la promoción de la participación de jóvenes científicos en el campo a través de becas y otros medios, y mediante la promoción y difusión de publicaciones, los sistemas de planificación y programación, dominios, simuladores, herramientas de software y material técnico.

Uno de los criterios para la selección de los planificadores a utilizar para resolver nuestro modelo fue el hecho de que fueron los más sobresalientes en las Competencias Internacionales de Planificación (IPC), evento bi-anual organizado por ICAPS.

A continuación, procedemos a describir los planificadores escogidos para resolver los modelos resultantes de este trabajo de Tesis.

2.3.9 PLANIFICADOR SGPLAN

SGPLAN es un planificador que quedó en el primer lugar en la IPC (International Planning Competition) del 2006 en la parte determinística de la competición [24]. SGPlan particiona un problema de planificación grande en subproblemas, cada uno con sus propios sub-objetivos, y resuelve soluciones inconsistentes de sub-objetivos utilizando la condición saddle-point extendida (ESPC del inglés extended saddle-point condition).

La partición de sub-objetivos es eficaz porque cada subproblema particionado implica un espacio de búsqueda sustancialmente menor que el del problema original. SGPlan desarrolló métodos para la detección de ordenamiento razonables en-

tre sub-objetivos, un análisis de agenda de objetivos intermedios para descomponer jerárquicamente cada subproblema, un algoritmo de reducción de espacios de búsqueda para eliminar acciones irrelevantes en subproblemas, y una estrategia para llamar al mejor planificador para resolver cada subproblema del nivel inferior [11].

SGPlan soporta PDDL2.1 (que es el que utilizamos en este proyecto), predicados derivados, y algoritmos para soportar hechos iniciados en el tiempo.

En versiones más recientes (2006) SGPlan está diseñado para resolver problemas de planificación temporal y no temporal especificado en PDDL3. El marco inherente en SGPlan consiste en la descomposición paralela, resolución de restricciones, y el solucionador de subproblemas.

Además de las variables de estado binarias y numéricas, SGPlan soporta variables de estado multi-valores que pueden ser definidos ya sea de forma explícita, utilizando objetos *fluents* o detectando implícitamente mediante la traducción a partir de una representación binaria existente. Una formulación de múltiples valores ha resultado beneficiosa para los planificadores, ya que proporciona una representación más compacta que una representación binaria de valor. De hecho, SGPlan ha demostrado éxito en la utilización de una formulación de varios valores para la solución eficaz de descomposición de problemas y optimización objetivo-utilidad. SGPLAN explota la localidad de restricciones de planificación para particionar un problema grande de planificación por medio de una descomposición paralela en subproblemas débilmente acoplados. La descomposición de particiones paralelas de un espacio de estados en subproblemas de tal manera que el espacio de estados combinado es el producto cruzado de los espacios de estado de los subproblemas. Esto lleva a un reparto de las variables en (posiblemente superpuestos) conjuntos. Las variables que se comparten entre los subconjuntos se llaman compartidas o variables complicadas, mientras que los de un subconjunto no compartido se llaman variables privadas o locales. Como resultado de la partición de las variables, las restricciones también se dividen en subconjuntos: las que implican sólo variables en un subconjunto se denominan restricciones locales, mientras que las variables que involucran en más de

un subconjunto se llaman complicadas o restricciones globales.

2.3.10 PLANIFICADOR LPG

LPG [21] es un planificador *independiente del dominio* que ha participado en la IPC (International Planning Competition), en donde fue premiado como mejor planificador automatizado en el 2003 y posteriormente en la IPC 2004 fue premiado como mejor desempeño en dominios que implican Timed Initial Literals y en la calidad del plan (satisficing planning track). Es un planificador estocástico basado en un algoritmo de *mejor primero* similar al que usa FF [23]. FF es el planificador independiente del dominio que se basa en búsquedas en el espacio de estados hacia adelante. LPG es recomendado para dominios que tienen cantidades numéricas y duraciones [13].

LPG produce planes de calidad multicriterio. El núcleo del sistema se basa en un método de búsqueda local estocástica y una representación basada en grafos llamada “Temporal Action Graphs” (TA-graphs). Dentro de las características de LPG es que soporta el lenguaje PDDL2.1 que soporta “acciones durativas” y “cantidades numéricas”.

La búsqueda local está emergiendo como un método poderoso para tratar la planificación totalmente automatizada, aunque en principio este enfoque no garantiza la generación de planes óptimos.

El esquema de búsqueda general de este planificador es Walk-plan, un procedimiento de búsqueda local estocástica similar a la conocida Walk-sat. Dos de las más importantes extensiones en este planificador son el uso de *grafos de acción temporal* (TA-graphs) y algunas nuevas técnicas para guiar el proceso de búsqueda local. En un grafo-TA los nodos de las acciones están marcados con valores temporales que estiman el tiempo más corto cuando la acción correspondiente termina, mientras que los nodos hechos son marcados con valores temporales estimando el tiempo más corto cuando el hecho correspondiente se convierte en verdadero. Un conjunto de

restricciones de orden se mantienen durante la búsqueda para manejar las acciones mutuamente excluyentes y representar las restricciones temporales implícitas en las relaciones “causales” entre las acciones y el plan actual.

La nueva heurística explota alguna información de accesibilidad para sopesar los elementos (grafo-TA) en la búsqueda del vecindario que resuelven inconsistencias seleccionadas de un grafo-TA actual. La evaluación de estos grafos-TA se basa en la estimación del número de pasos de búsqueda necesarios para alcanzar una solución (un plan válido), el makespan estimado, y la estimación del costo de ejecución. El **LPG** es un planificador incremental, en el sentido de que produce una secuencia de planes válidos cada uno de los que mejora la calidad de los anteriores. La calidad del plan se modela mediante la ejecución y costos temporales de una manera flexible (el usuario puede determinar la importancia relativa del criterio de la calidad del plan).

Finalmente, terminamos esta sección recordando que los planificadores anteriormente mencionados (SGPLAN y LPG) los utilizamos como un medio para validar nuestros modelos, y para obtener soluciones, y de esta manera tratar de analizar los límites que estos puedan tener en resolver nuestros tipos de modelos.

CAPÍTULO 3

FORMULACIÓN DEL PROBLEMA Y METODOLOGÍA

3.1 PLANTEAMIENTO DEL PROBLEMA

Hoy en día los estudiantes tienen múltiples actividades a realizar durante su trayectoria escolar. Algunas de las actividades que realizan son asistir a clases, hacer tareas, leer libros, hacer exámenes, tomar asesorías o tutorías, participar en actividades en grupo o equipo, realizar servicio social o prácticas profesionales, además de actividades extracurriculares como actividades deportivas, etc. En la figura 3.1 ilustramos algunas de estas actividades.



Figura 3.1: Diversas actividades que realiza un estudiante.

Además de lo antes mencionado, en la actualidad existe una gran diversidad de recursos disponibles (Véase figura 3.2) para llevar a cabo las actividades escolares,

por ejemplo: diapositivas, ejercicios, artículos de investigación, libros, videos, salones de clase, personal universitario, computadoras, software, equipo de laboratorio, etc.

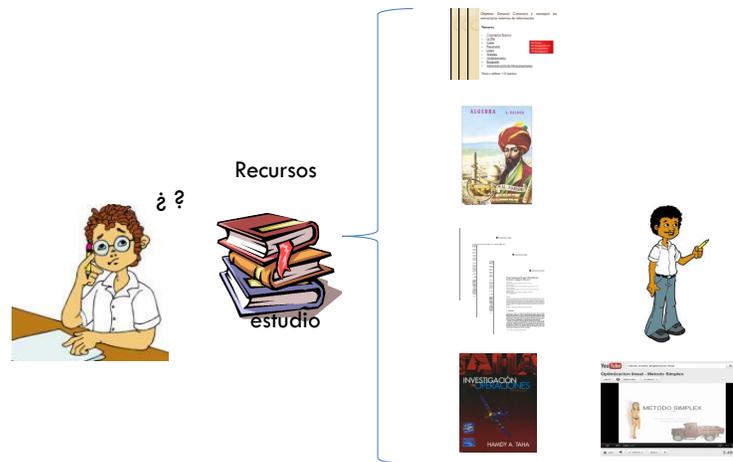


Figura 3.2: Diversidad de recursos para estudio.

El principal problema es entonces que el estudiante tiene que decidir ¿qué estudiar?, ¿cuándo? y ¿qué utilizará? para posiblemente mejorar su rendimiento académico a través del tiempo. Considerando además, que cada actividad le consume tiempo, el estudiante tiene que tomar un conjunto de decisiones que afectan, ya sea directa o indirectamente, su trayectoria escolar.

Investigaciones recientes sugieren que “la capacidad de los estudiantes para gestionar su propio estudio es una de las habilidades más importantes que los estudiantes necesitan para aprender” [33]. El alumno tiende a aprender mejor cuando las aplicaciones tecnológicas están directamente relacionadas al currículum que tienen que cubrir y cuando la aplicación está integrada en un día escolar típico [39].

Creemos que el uso de *técnicas de planificación* permitirán al estudiante obtener control sobre sus actividades académicas. Considerando la importancia de generar planes educativos personalizados, donde el estudiante pueda administrar sus actividades de estudio. Pero realizar esto de la manera tradicional no es factible. Por un lado, porque crear múltiples actividades que estén asociadas a los objetivos de aprendizaje de un plan de estudios, o de al menos una materia, de una forma manual,

sería impensable. Si además, si se quiere realizar esto para cada estudiante, sería demasiado trabajo, el cual sería innecesario porque las “condiciones” de un estudiante estarían cambiando de acuerdo a la “situación” que tenga en “ese momento”, el objetivo, etc.

La finalidad de este proyecto es la aplicación de técnicas avanzadas de Planificación en Inteligencia Artificial al ámbito educativo. La idea principal es *modelar* el programa académico de un alumno de educación a nivel superior (aunque no está limitado a este nivel), incluyendo en dicho modelo la situación actual del estudiante, considerando las restricciones de tiempo y recursos del sistema educativo, con el fin de generar planes educativos que lo guíen durante su trayectoria escolar para intentar mejorar su rendimiento académico.

3.2 PLANIFICACIÓN EN INTELIGENCIA ARTIFICIAL

Para la generación de los planes de estudio personalizados de los estudiantes, utilizamos técnicas de Planificación de Inteligencia Artificial ya que según su definición, “es el proceso de búsqueda y articulación de una secuencia de acciones que permiten alcanzar un objetivo” [38]. Para ello utilizamos el lenguaje de modelación PDDL.

Una vez estructurado el modelo, se resuelve utilizando algoritmos de planificación [36] automatizada, dando como resultado planes de acciones educativas para satisfacer un conjunto de objetivos de aprendizaje para el estudiante.

En la figura 3.3 se muestra el proceso de planificación.

3.3 CARACTERÍSTICAS DEL MODELO

Modelar un plan de estudios personalizado para cada estudiante es complejo, ya que hay que “abstraer” el proceso educativo y decidir qué factores se tomarán en

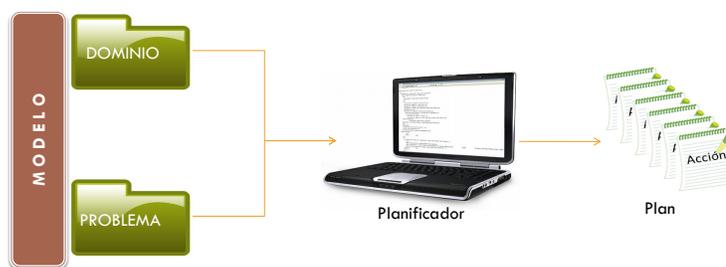


Figura 3.3: Proceso de planificación.

cuenta en dicha modelación.

Entre los factores a considerar están las “acciones” que se tomarán en cuenta en la modelación del currículum académico, considerando propiedades como el tiempo, recursos y métricas. Una de estas métricas es el “score” de una *actividad de aprendizaje*, la acumulación de éstas métricas (acumulación de calidad), para que de alguna manera, se pueda determinar el grado de aprendizaje obtenido, además de garantizar la acumulación de estas métricas lo suficiente para “aprobar” un *objetivo de aprendizaje*.

En la figura 3.4 presentamos el plan de estudios de la Carrera de Ingeniero en Mecatrónica de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León, que utilizamos para ejemplificar nuestro modelo.

Materias previas como requisito

Debido a que las “acciones” que contemplamos en la modelación del plan de estudios incluyen materias cuyas precondiciones son distintas de acuerdo a la materia de que se trate, se decidió hacer una *acción* por cada materia. Es decir, la *precondición* para poder “inscribir”, por ejemplo, la materia de MatemáticasIV, es que haya cursado y aprobado MatemáticasIII. En la figura 3.5 podemos observar estas relaciones. Además, una materia puede tener más de una materia requerida, como en el caso de FísicaII que necesita FísicaI y MatemáticasII. En cambio, para poder inscribir Matemáticas I, por ejemplo, no es necesario que haya cursado ninguna otra materia. Debido a esto, no es posible modelar una sola acción que reciba

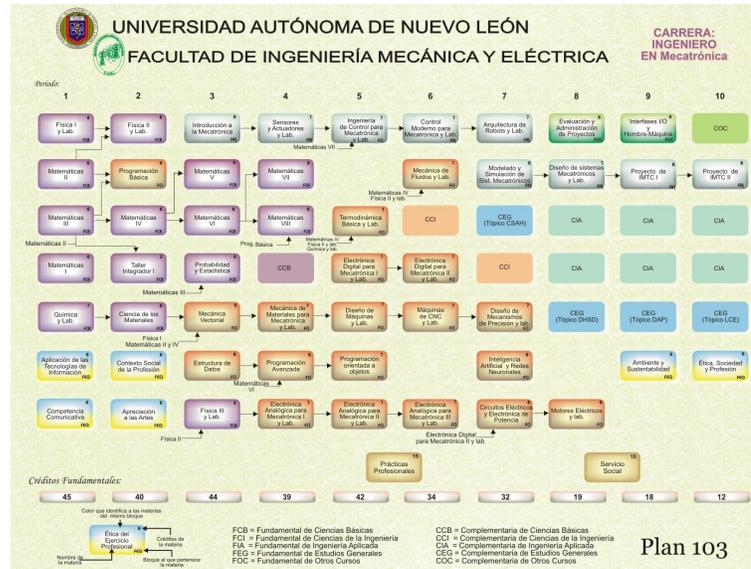


Figura 3.4: Plan de estudios de la Carrera de Mecatrónica-FIME/U.A.N.L.

todas las materias, sino una acción por cada materia.

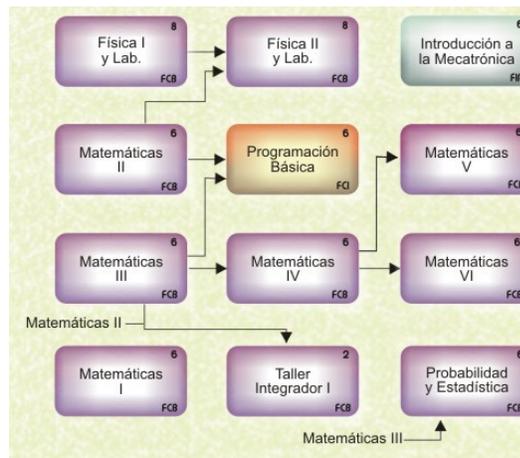


Figura 3.5: Ejemplo de materias previas como requisito para inscribir una materia en el Plan de estudios de la Carrera de Mecatrónica-FIME/U.A.N.L.

Una característica del modelo es que cada materia se divide en temas, subtemas y actividades de aprendizaje, los cuales tendrán una ponderación y duración por haberlos realizado. Para esto es necesario realizar una “acción” para cada diferente tipo de *actividad de aprendizaje* vinculada a un subtema, una acción para cada tema de la materia, y vincular cada subtema con su tema y al final calcular la “calificación”

que obtendría el estudiante por las acciones realizadas al haber cursado y aprobado dicha materia. Es decir, se ponderan las acciones para tratar de asegurar que el alumno tenga al menos, una calificación aprobatoria (en nuestro ejemplo, mayor a 70, considerando que es a nivel licenciatura).

Lo anterior nos lleva a representar dentro de la modelación de una materia la existencia de relaciones de orden y jerarquía que existen tanto en la materia en sí, como entre los temas, subtemas y actividades de aprendizaje. En la tabla 3.1 se muestran estas relaciones de orden y jerarquía contempladas en la modelación.

Requires	Esta relación de orden indica si una materia necesita de las bases de otra materia o curso, para poder realizarse.
IsPart Of	Indica la relación de jerarquía entre los temas, subtemas y actividades de aprendizaje. Así los temas <i>IsPartOf</i> del tema principal que es la materia, los subtemas <i>IsPartOf</i> de los temas y los objetos de aprendizaje <i>IsPartOf</i> de los subtemas.
IsBasedOn	Proporciona relaciones de orden o precedencia entre subtemas y temas.

Tabla 3.1: Relación de orden y jerarquía

3.3.1 REPRESENTACIÓN DE LA ESTRUCTURA DE LA MATERIA

Las materias, las podemos visualizar como una organización jerárquica, como se ve en la Figura 3.6, en donde cada una de ellas está comprendida por los temas que se deben cubrir para entender esa materia, los cuáles a su vez, se dividen en subtemas u objetivos específicos.

Estos *Objetivos de Aprendizaje* describen lo que el estudiante va a ser capaz de aprender, entender o hacer como resultado del desarrollo de un tema que involucra un conjunto de *Actividades de Aprendizaje* (LA en la figura), por medio de las cuales se

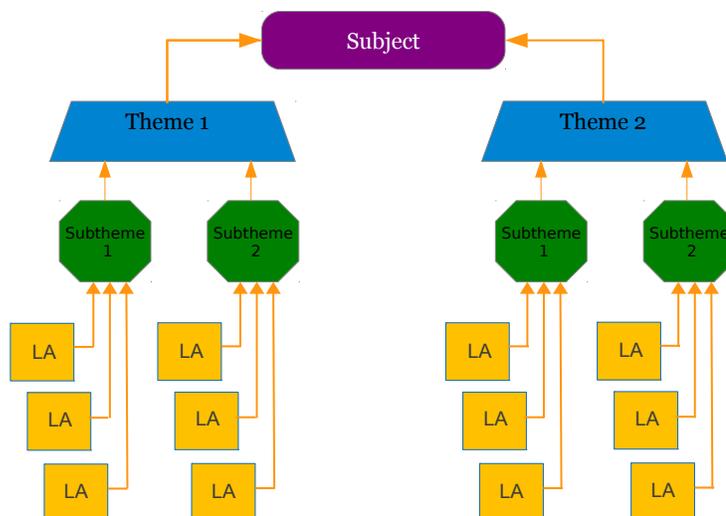


Figura 3.6: Figura de cómo es que se ve una materia del plan de estudios.

obtienen nuevos conocimientos, habilidades o aptitudes. Dentro de estas *actividades de aprendizaje* se proponen diferentes objetos de aprendizaje (LO), con un valor (score) y duración definidos.

Estos “score” definidos se van ir acumulando con la ejecución de las actividades de aprendizaje escogidas. Esto es importante, porque de acuerdo a estos valores es que ciertas partes del plan educativo se activan o desactivan, es decir, algunas actividades de aprendizaje serán seleccionadas y otras no. Más adelante en la sección de funciones de acumulación de calidad describiremos con más detalle esto.

Por otro lado, aunque en la actualidad no hay una definición clara de un objeto de aprendizaje [35]; podemos decir que, “un objeto de aprendizaje (LO) puede basarse en un texto electrónico, una simulación, un sitio web, una imagen gráfica, una película QuickTime, un applet de Java o cualquier otro recurso educativo que se puede utilizar en el aprendizaje” de un objetivo particular del currículo académico a cubrir [32]. Es decir, es la relación que existe entre recursos educativos y actividades de aprendizaje para satisfacer un objetivo de aprendizaje.

Para poder decir que se ha “aprendido” una materia se tendrán que cubrir los distintos temas que la comprenden, que a su vez tienen diferentes objetivos o sub-

temas, y éstos a su vez necesitan de actividades de aprendizaje que están comprendidas por objetos de aprendizaje, que un estudiante puede utilizar para comprender ese objetivo.

Cuando tenemos este conjunto de actividades el planificador podrá elegir, de acuerdo a la situación actual del estudiante que comprende entre otras cosas, su objetivo (que en nuestro ejemplo es “aprobar” esa materia), materias previamente aprobadas y la métrica a optimizar (en nuestro ejemplo la métrica será minimizar el tiempo).

En la figura 3.7 se representa una materia (Subject) que tiene varios temas a cubrir (Themes) en donde cada tema tiene subtemas (subtheme) y éstos a su vez tienen diversas actividades de aprendizaje (LA).

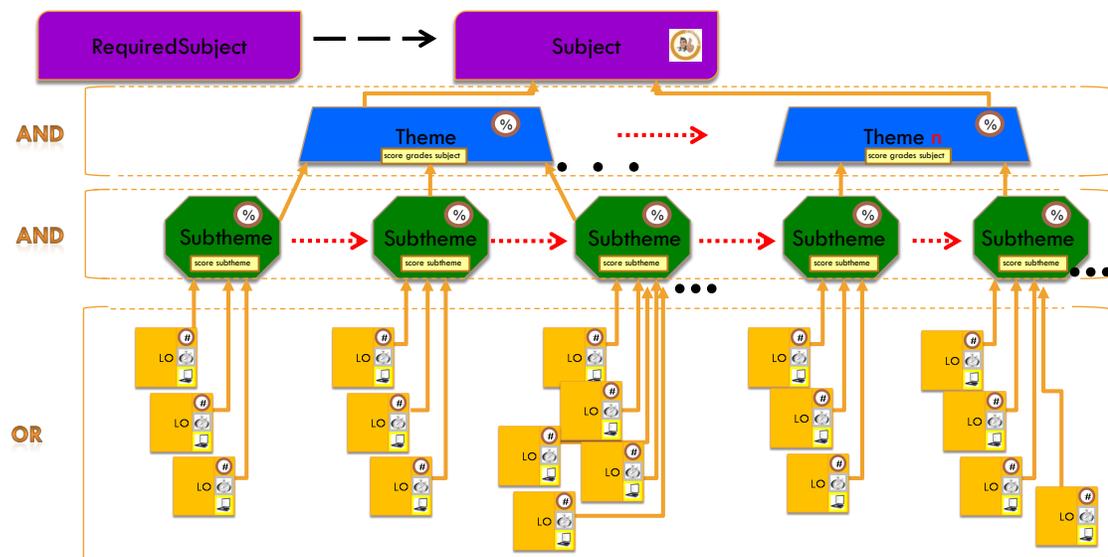


Figura 3.7: Estructura de una materia del plan de estudios

Si observamos la figura están las palabras “AND” y “OR” éstas son para indicar que tanto los temas como los subtemas (que tienen un “AND”) tienen que ser abordados por el estudiante, es decir, el estudiante no podrá omitir ningún tema, ni ningún subtema. Pero en el caso de las *actividades de aprendizaje* (LA) que hay para cada subtema, no necesariamente tiene que ser así, es decir, el planificador seleccionará entre ellos, de acuerdo a características del estado inicial del estudiante

tales como: objetivo, materias aprobadas anteriormente, actividades de aprendizaje ya realizados, métrica a optimizar, etc., proveyendo al estudiante diversas actividades con recursos asociados y duración diferentes.

Como mencionamos anteriormente, para poder “inscribir” una materia, puede existir la restricción de haber “aprobado” una materia anterior, ya sea porque es una materia sucesiva, porque es la base para esa materia, etc. Esto es representado en la siguiente figura 3.8 con RequiredSubject.

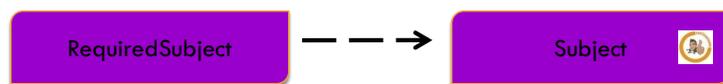


Figura 3.8: Dependencia de otra materia. RequiredSubject

Asimismo las relaciones de jerarquía y orden descritas anteriormente y mostradas en la figura de la estructura de la materias son representadas con distintos tipos de flechas en la figura 3.9.



Figura 3.9: Flechas para diferenciar los distintos tipos de dependencias.

Los temas pueden tener distinta *ponderación* en una materia, y esto es igual para los subtemas comprendidos en dichos temas, lo anterior es para dar flexibilidad al instructor o maestro de clase que clasifique en orden de importancia los temas, además de poder agregar asesorías o trabajos adicionales que se consideren necesarios para fortalecer las actividades de aprendizaje. Estas ponderaciones son representados como números decimales de dos dígitos y la suma del conjunto de temas debe ser igual a uno. Asimismo para los subtemas, con la diferencia de que la suma del conjunto de subtemas por tema sea uno.

Asimismo, las actividades de aprendizaje (LA) están comprendidas por los objetos de aprendizaje (LO), los cuales tienen características como la *duración* (en

unidades de tiempo) que se necesita para realizarlo. Un *score*, que en nuestro caso significa el valor asignado, para tratar de alguna manera de medir el aprendizaje del estudiante. Este valor (score) en este modelo es un número entero positivo no mayor a cien. Y por último, un *recurso* asociado a ese LO, pudiendo ser ese recurso: una computadora, un libro, un artículo de investigación, etc. Ver figura 3.10



Figura 3.10: Ejemplificación de un LO

3.3.2 RESTRICCIONES DE TIEMPO

Para considerar restricciones de tiempo en la modelación, las “acciones” se modelan con una duración en cada una de ellas, específicamente en las *actividades de aprendizaje* (LA).

En las actividades de aprendizaje se considera el tiempo, (en unidades de tiempo) de acuerdo a lo que el experto considera que es la duración del *objeto de aprendizaje*. Un supuesto en nuestra modelación es que la duración de un LA tiene una relación directa con el “valor” que tiene el realizar esa LA. Es decir, a mayor tiempo, mayor “valor” y viceversa.

3.3.3 USO DE RECURSOS EDUCATIVOS

Para considerar los recursos educativos, en cada *actividad de aprendizaje* está asociado un recurso para desempeñar esa actividad, tales recursos pueden ser: computadoras, libros, artículos de investigación, etc., que serán definidos en el *problema* del modelo. Especificamos los tipos de recursos educativos a modelar, junto con una

cantidad de ellos, ya que sabemos que los recursos son limitados. En donde tenemos que cada LA tiene asociado un recurso. En caso de que se quisiera modelar un LA sin recursos se establece un recurso ficticio “not-has-resource” en el *problema*.

En el caso de que se quisiera modelar un recurso “ilimitado” el modelo permite esta flexibilidad, ya que solo se necesita definir un número lo suficientemente grande en la cantidad de ese recurso en el *problema*.

3.3.4 FUNCIONES DE ACUMULACIÓN DE CALIDAD

Una característica importante en nuestro modelo es la *acumulación de calidad*, le decimos así a la acumulación de score de cada *actividad de aprendizaje* comprendida en los subtemas. Cada subtema y tema se manejan con ponderaciones. Es importante remarcar que no es necesario realizar todas las LA de un subtema, salvo que la acumulación de calidad no sea lo suficiente como para decir que un alumno ha “aprobado” ese subtema del tema de esa materia.

Para detallar un poco más esto, decimos que cada *actividad de aprendizaje* que se selecciona tiene una puntuación (score) la cual se acumula para cada subtema, que a su vez se acumula para cada tema de la materia, dando como resultado la “calificación” que se espera que el estudiante obtenga al realizar dichas actividades. Es importante remarcar que la acumulación de calidad no es una simple sumatoria de los *objetos de aprendizaje*, ya que cada tema y subtema podrían tener diferente ponderación, de manera que al multiplicar el valor asignado a cada LO por la ponderación otorgada al subtema de un tema en específico, se obtendrá la “calificación”; por esta razón necesitamos una jerarquía de actividades para poder acumular correctamente los resultados por materia.

Hemos seleccionado este tipo de acumulación de calidad en las acciones debido a que creemos que muchos de los cursos actuales tienen este tipo de acumulación, además de no ser una acumulación trivial. Nosotros ejemplificamos tal vez, la forma más compleja de la propiedad de acumulación de score. Aún así, nuestros modelos

son suficientemente flexibles que si quisieramos cambiar la función de acumulación sería posible, por ejemplo para considerar clases de scores como A, A+, B, B+, C, etc. Esta parte podría ser definida por el usuario (que en este caso sería el maestro) de acuerdo al tipo de ponderación que el quiera considerar.

A continuación se define formalmente la métrica de acumulación de calidad que diseñamos en nuestro modelo:

Donde:

D = Conjunto de actividades de aprendizaje seleccionados

β_i = Conjunto de subtemas por tema

α_j = Conjunto de actividades por subtema

i = tema

j = subtema

k = actividad

d_k = valor o score de la actividad de aprendizaje

f_i = Ponderación del tema

e_j = Ponderación del subtema

Lo que nos dá como resultado la siguiente ecuación de acumulación de calidad (o scores):

$$\sum_{i \in I} \sum_{j \in \beta_i} \sum_{k \in D \in \alpha_j} d_k e_j f_i \quad (3.1)$$

A continuación se muestra un ejemplo de cómo es que se acumula esta calidad en la materia de Estructura de Datos. En la tabla 3.2 se muestran los temas que comprenden la materia de Estructura de Datos del tercer semestre de la carrera de Ingeniería en mecatrónica. Utilizaremos esta materia para ejemplificar cómo es

Materia	Temas
Estructura de datos	Conceptos básicos Pilas Colas Recursión Listas Árboles Algoritmos de ordenamiento

Tabla 3.2: Temas de la materia **Estructura de Datos** del tercer semestre del plan de estudios de Ing. en mecatrónica. Seleccionamos solo dos temas: Conceptos básicos y Algoritmos de ordenamiento para nuestro ejemplo

que se acumula la calidad en una materia considerando solo dos temas: Conceptos-básicos y Algoritmos-de-ordenamiento de los siete que la comprenden. En la tabla 3.3 se muestran los temas con sus subtemas. A manera de acortar el ejemplo, de estos temas se modelarán solo dos subtemas de cada uno, *Arreglos* y *Conjuntos* del tema *Conceptos-básicos* e *Inserción-directa* y *Método-Quicksort* del tema *Algoritmos-de-ordenamiento*.

En la figura 3.11 se muestra la materia de estructura de datos y los dos temas y subtemas seleccionados para ejemplificar la acumulación de calidad. De la misma manera se sugieren 3 actividades de aprendizaje para cada subtema con distintos recursos, duración y score (valores de cada LA) diferentes.

En la tabla 3.4 se muestran las actividades de aprendizaje de los Subtemas seleccionados. En el modelo, específicamente en el archivo problema, indicamos que el *objetivo* del alumno es aprobar esa materia y que la métrica a optimizar es *el tiempo total*, es decir, *minimizar el total-time*. La solución arrojada por el planificador, es decir el plan educativo, es que solo necesitamos realizar dos LAs de cada subtema. En la figura 3.12 por ejemplo, para el tema *Conceptos-básicos* se seleccionan LA-Arr01

Tema	Subtemas
Conceptos Básicos	<ul style="list-style-type: none"> -Definición del término de estructura -Definición y especificación -Arreglos -Registros -Conjuntos
Algoritmos de ordenamiento	<ul style="list-style-type: none"> -Antecedentes general -Ordenación por intercambio directo -Ordenación por inserción directa -Ordenación por el método shell -Ordenación por el método Quicksort -Ordenación por intercalación

Tabla 3.3: Descripción de temas y subtemas de la materia de **Estructura de Datos**.

Se seleccionaron dos subtemas por tema. Arreglos y Conjuntos del tema Conceptos básicos. Inserción directa y método Quicksort del tema Algoritmos de ordenamiento

y LA-Arr03 del Subtema *Arreglos* y LA-Con01 y LACon02 del subtema *Conjuntos*. Asimismo, para el tema *Algoritmos-de-ordenamiento* se seleccionan LA-Dir01 y LA-Dir02 del Subtema *Inserción-directa* y LA-Qui01 y LA-Qui03 del subtema *Método-Quicksort*.

La modelación que se utiliza para la acumulación de calidad será por subtema en una métrica llamada “score”, esto a su vez es por cada tema. Se toma el valor del LA multiplicado por el porcentaje del subtema al que pertenece. Como ejemplo tomemos el subtema *Arreglos* del tema *Conceptos-básicos* de la figura 3.13 (le hemos puesto nombre a las actividades de aprendizaje a manera de que no sean confundidas).

Se puede observar que el subtema Arreglos tiene un valor de 0.50, y el LA-Arr01 tiene un valor de 40. Suponemos que el estudiante no ha realizado ninguna actividad, ni aprobado nada de esta materia, esto es especificado en el *estado inicial*

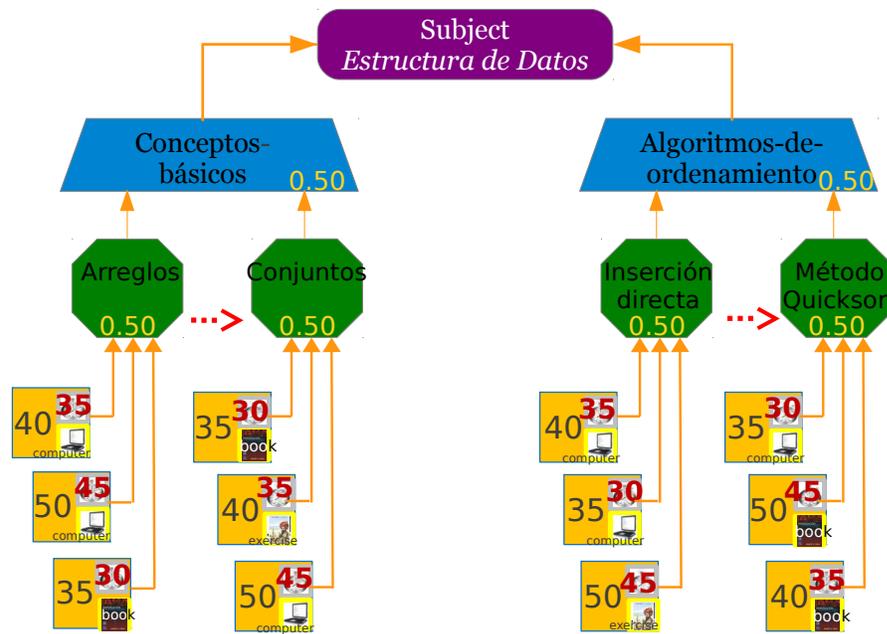


Figura 3.11: Ejemplo de la materia de Estructura de datos con los temas, subtemas y actividades de aprendizaje.

del estudiante. Entonces el score acumulado inicial = 0. Figura 3.14.

Así tenemos que:

$$ScoreSubtemaArreglosstudent1 = (value_LA - Arr01 * percentSubtemaArreglos)$$

Es decir:

$$40 * .50 = 20$$

Ahora tenemos que el score del subtema *Arreglos* del tema *Conceptos-básicos* de esta materia, (de este estudiante) tiene acumulado un valor de 20.

En el mismo subtema (*Arreglos*), se seleccionó el LA-Arr03 con el valor de 35. Hay que tener en cuenta que el score acumulado ya no es 0, sino 20 del anterior LA seleccionado.

tomando en cuenta la misma ecuación de:

$$ScoreSubtemaArreglosstudent1 = (value_LA * percentSubtemaArreglos)$$

Subtemas	Nombres de Actividades de aprendizaje
Arreglos	LA-Arr01 LA-Arr02 LA-Arr03
Conjuntos	LA-Con01 LA-Con02 LA-Con03
Inserción-directa	LA-Dir01 LA-Dir02 LA-Dir03
Método-Quicksort	LA-Qui01 LA-Qui02 LA-Qui03

Tabla 3.4: Descripción de actividades de aprendizaje para cada subtema. En donde el subtema Arreglos y Conjuntos pertenece al tema Conceptos básicos. Inserción directa y método Quicksort al tema Algoritmos de ordenamiento

tenemos:

$$35 * .50 = 17.50$$

Ahora el valor del Score del Subtema *Arreglos* es:

$$20 + 17.50 = 37.50$$

Ahora tenemos que el score del subtema *Arreglos* del tema *Conceptos-básicos* de esta materia, (de este estudiante) tiene acumulado un valor de 37.50. Si vemos la figura 3.15 se muestra la acumulación del subtema Arreglos, luego de manera similar ocurre con el subtema Conjuntos, del mismo tema.

Si hacemos la misma operación con los subtemas Inserción-directa y Método-Quicksort del tema *Algoritmos-de-ordenamiento* tenemos que cada subtema ten-

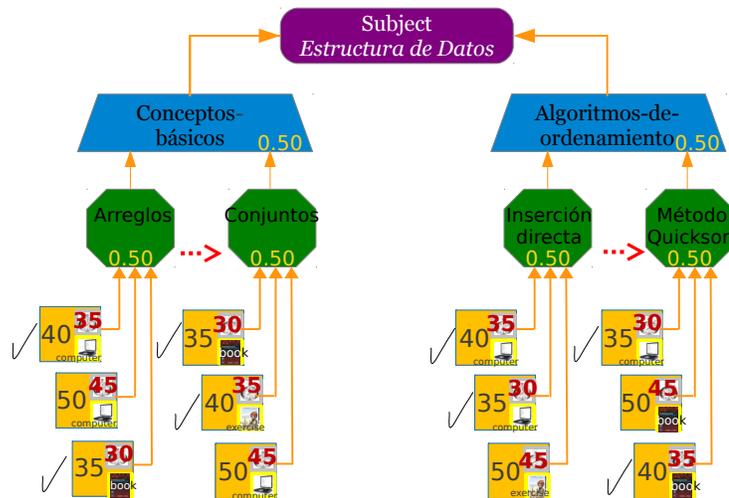


Figura 3.12: Selección de las actividades de aprendizaje por cada subtema.

drá acumulado un valor (que en este ejemplo es el mismo para cada subtema) que posteriormente será evaluado a nivel del tema. Véase figura 3.16

Si se observa en la figura anterior cada subtema tiene un valor acumulado. Dichos valores se establecen como restricciones a evaluar en la acción definida para cada tema en el modelado. Decimos que el valor acumulado en el subtema debe ser mayor a la calificación considerada como mínima (70 en el ejemplo).

Tenemos que:

$$ScoreSubthemeArreglos \geq mingrade * percentSubtemaArreglos$$

Es decir:

$$70 * .50 = 35$$

así que se cumple la condición de:

$$37.50 \geq 35$$

Lo anterior es para evaluar cada subtema, en un tema determinado. Esta restricción es la que utilizamos para de alguna manera “obligar” al algoritmo de planificación a realizar las actividades necesarias para aprobar cada subtema de los temas

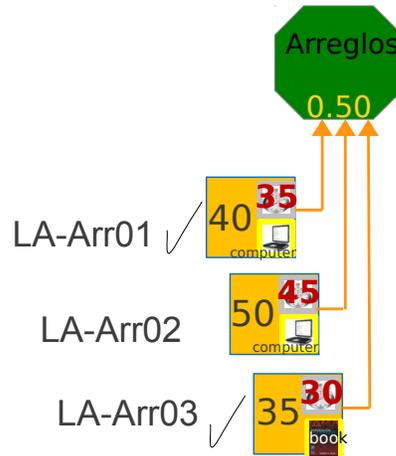


Figura 3.13: Tema Conceptos Básicos.

Temas	Subtemas	Actual
Conceptos	Arreglos	0
Basicos	Conjuntos	0
Algoritmo de ordenamiento	Inserción-directa	0
	Método-Quicksort	0

Figura 3.14: Tabla de acumulación actual de score.Inicial

de la materia.

Finalmente, para dar una “calificación” a la materia se realiza una sumatoria del producto de los scores de los subtemas por la ponderación del tema al que pertenece, es decir

$$GradesEstructuraDatosstudent1 = ScoreSubthemeArreglos_student1 * percentTemaConceptosBasicos$$

En la figura 3.17 decimos que para el tema *Conceptos-básicos* tenemos que inicialmente tiene cero, luego hacemos el cálculo descrito anteriormente y tenemos que ahora tiene 18.50:

$$37.50 * .50 = 18.50$$

Esto es para el subtema *Arreglos*, luego hacemos lo mismo para el subtema

Temas	Subtemas	Actual
Conceptos	Arreglos	20
Basicos	Conjuntos	0
Algoritmo de ordenamiento	Inserción-directa	0
	Método-Quicksort	0

Selección de LA-Arr01.

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	0
Algoritmo de ordenamiento	Inserción-directa	0
	Método-Quicksort	0

Selección de LA-Arr03.

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	17.5
Algoritmo de ordenamiento	Inserción-directa	0
	Método-Quicksort	0

Selección de LA-Con01

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	37.5
Algoritmo de ordenamiento	Inserción-directa	0
	Método-Quicksort	0

Selección de LA-Con02.

Figura 3.15: Acumulación de subtemas: Arreglos y Conjuntos, del tema Conceptos Básicos

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	37.5
Algoritmo de ordenamiento	Inserción-directa	20
	Método-Quicksort	0

Selección de LA-Dir01

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	37.5
Algoritmo de ordenamiento	Inserción-directa	37.5
	Método-Quicksort	0

Selección de LA-Dir02.

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	37.5
Algoritmo de ordenamiento	Inserción-directa	37.5
	Método-Quicksort	17.5

Selección de LA-Qui01

Temas	Subtemas	Actual
Conceptos	Arreglos	37.5
Basicos	Conjuntos	37.5
Algoritmo de ordenamiento	Inserción-directa	37.5
	Método-Quicksort	37.5

Selección de LA-Qui03

Figura 3.16: Acumulación de subtemas: Inserción-directa y Método-Quicksort, del tema Algoritmos-de-ordenamiento

Conjuntos del mismo tema (Conceptos-básicos), y tenemos:

$$37.50 * .50 = 18.50$$

Ahora *Grades EstructuraDatos* tiene: 37 únicamente de los subtemas del tema *Conceptos-básicos*.

Si hacemos esto por cada uno de los siguientes subtemas, *Inserción-directa* y *Método-Quicksort* del tema *Algoritmos-de-ordenamiento* (que en nuestro ejemplo cada subtema tiene un valor de 18.50 cada uno), tenemos que la “calificación” final para el estudiante en esa materia es de 74.

Con lo anterior, cumplimos con la restricción impuesta para aprobar una materia que es que la calificación sea mayor que la calificación mínima.

$$GradesEstructuraDatos \geq mingrade$$

es decir, se cumpla:

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	0
	Algoritmo de ordenamiento	0
Evaluación inicial=0.		

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	18.5
	Algoritmo de ordenamiento	0
Evaluación del valor del Subtema Arreglos.		

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	37
	Algoritmo de ordenamiento	0
Evaluación del valor del Subtema Conjuntos.		

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	37
	Algoritmo de ordenamiento	18.5
Evaluación del valor del Subtema Inserción directa.		

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	37
	Algoritmo de ordenamiento	37
Evaluación del valor del Subtema Método Quicksort.		

Materia	Temas	Eval.
Estructura de Datos	Conceptos Basicos	37
	Algoritmo de ordenamiento	37
Calificación obtenida por la materia de ED.		75
Suma de evaluaciones.		

Figura 3.17: Formación de la calificación. Cada tema suma el conjunto de subtemas que tiene y los multiplica por la ponderación que tiene ese tema. Al final la suma de las evaluaciones por tema, conforman dicha “calificación”.

$$74 \geq 70.$$

Podemos ver que esta “calificación” cumple con las expectativas de “aprobar la materia”, en el “menor tiempo”, que era el objetivo del estudiante.

3.4 TRANSFORMACIÓN DEL PROBLEMA EDUCATIVO A UN MODELO PDDL

La *Planificación* en Inteligencia Artificial utiliza un lenguaje de modelación el cuál es llamado *PDDL*(acrónimo de Planning Domain Definition Language), que traducido es Lenguaje de Definición de Dominio de Planificación.

El modelo consta de dos fases: la generación del *archivo de dominio* y la generación del *archivo de problema*. Esto facilita el uso de un dominio para múltiples problemas. Ésta es una de las cualidades de la modelación en este lenguaje de planificación, ya que una vez definido un dominio, no importa qué estudiante se quiera representar, ni su situación actual, que para nosotros es su estado inicial, ya que el dominio se podrá utilizar para cualquier estudiante.

3.4.1 REQUERIMIENTOS EN PDDL DE NUESTRO MODELO

Los requerimientos que necesitamos para realizar nuestro modelo comprendido por los archivos de *dominio* y *problema* son: *Types*, *Durative Actions*, *Numeric Fluents*, *Strips*, y *Equality*. En la tabla 3.5 se observan estos requerimientos.

Requerimiento	Definición
:strips	Estilo básico STRIPS de añadir y borrar. Estructura de precondiciones y efectos en cada acción.
:typing	Permite los nombres de tipo en las declaraciones de las variables.
:fluent	Permite la definición de funciones y el uso de efectos utilizando los operadores de asignación y las precondiciones aritméticas.
:durative actions	Permite aplicar acciones durativas. Tenga en cuenta que esto no implica: fluents.
:equality	Soporte = como se incorporó en el predicado.

Tabla 3.5: Características de PDDL que utilizamos en nuestro modelo y las cuáles son soportadas por los planificadores seleccionados

3.4.2 DOMINIO

El archivo de *dominio* modela todas las acciones que un estudiante puede realizar, este modelo es general e independiente del alumno, es decir, no se basa en un estudiante en específico. Aquí, se modelan las materias del currículo educativo que un estudiante debe cursar, junto con sus restricciones de orden y jerarquía, entre otras características.

PREDICADOS

Para manipular nuestro modelo, PDDL utiliza *predicados* que son relaciones entre los objetos que conforman nuestro dominio para representar hechos del mundo real. Dichas relaciones son con las materias, estudiantes, temas, subtemas, actividades de aprendizaje. Según [27] son propiedades de los objetos que nos interesan, las cuales pueden ser verdaderas o falsas.

A continuación se muestra un listado de los predicados utilizados para representar las propiedades de las materias, temas, subtemas y LA.

1. (*available-subject ?subj - subject ?s - student*) indica si una materia está disponible para ser inscrita. Esto es útil para no dejar que un estudiante de primer semestre pudiera tener la opción de inscribir una materia de quinto semestre, cuando ésta no tenga ninguna materia anterior de requisito.
2. (*free ?s - student*) indica que el estudiante está disponible. Algunas cuestiones a considerar es que pudiera estar suspendido, dado de baja, etc.
3. (*pass-degree ?subj - subject ?s - student*) nos sirve para indicar si un alumno aprobó una materia específica. Además de que si quisiera inscribir una materia que tenga de prerrequisito esta materia aprobada, lo pueda hacer.
4. (*enrollment ?s - student ?subj - subject*) indica si el alumno quedó inscrito en una materia. Una vez inscrito, podrá realizar cualquier actividad relacionada a esa materia. Un estudiante no podrá realizar absolutamente nada, si no está inscrito en la materia.
5. (*done-Theme ?t - Theme ?subj -subject ?s - student*) indica si se aprobó el tema de una materia específica. Esto es necesario no solo para hacer verdadero este hecho, sino que además nos ayuda en el orden entre los temas, ya que no se podrá decir que se aprobó el tema 5 por ejemplo, si no se ha “hecho” el tema 1. Otra cualidad es que para asegurar que todos los temas se han realizado, se

realizó una acción que comprende el conjunto de estos predicados, de acuerdo a la cantidad de temas que tenga la materia.

6. (*done-subject-LA ?subj - subject ?s - student*) Una vez que se han realizado todos los temas, es decir, se realizaron todas las actividades de aprendizaje de tal manera que se han cubierto todos los temas de esa materia, se indica con este predicado tal situación.
7. (*not-done-LA ?oa - LA ?subj - subject ?s - student*) nos indica que una LA no ha sido realizada, de tal manera que permitirá hacerla, es decir, es una restricción para proteger al modelo de que el planificador seleccione dos veces la misma actividad de aprendizaje para un subtema determinado.
8. (*not-approved ?subj - subject ?s - student*) es una restricción para no permitir que un alumno que haya cursado y aprobado una materia, vuelva a poder inscribirla.
9. (*KindResourceLO ?oa - LA ?eq - resource*) es para asociar el tipo de recurso que se utilizará en la actividad de aprendizaje, pudiendo ser una computadora, un libro, un artículo científico, etc., es decir, para realizar la actividad “ejercicio ?oa” se necesita el recurso “libro ?eq”. Un supuesto en este trabajo es que cada actividad tiene asociado un recurso, en caso de que no sea así en el archivo problema se crea un recurso ficticio llamado “not-has-resource”.
10. (*done ?oa - LA*) nos indica que se ha realizado esta actividad de aprendizaje.

Clasificaciones de actividades de aprendizaje (LA).

11. (*not-has-reqs ?oa - LA*) indico que para poder realizar esta LA no es necesario haber realizado ninguna otra actividad. Esto generalmente es aplicable para todas las LA del subtema 1 del tema 1 de cualquier materia, ya que no necesitan de haber aprobado nada antes para poder ejecutarse.

12. (*has-reqs ?oa - LA ?req - LO*) para poder realizar este tipo de LA existe la restricción de haber realizado otro LA ya sea anterior o alguno en particular, si es que tuvieran alguna secuencia, como por ejemplo, en el tema 1 la actividad fue leer el capítulo 1 del libro x y para el tema 2 es leer el capítulo 2 del mismo libro, entonces si por alguna razón, el planificador escoge la actividad del tema 2 de leer el cap2 de ese libro, entonces tendrá que incluir en el plan la actividad del tema 1 de leer el cap1, o de lo contrario, no seleccionará ninguna de las dos actividades. Esta restricción junto con otras funciones y predicados nos ayudan a crear otros escenarios, descritos más adelante.
13. (*has-multiple-reqs ?oa - LA ?req - LO*) similar al anterior, solo que esta restricción es para indicar que requiere más de un LA.

Relaciones de Jerarquía.

14. (*isPartOfSubtheme ?oa - LA ?subt - subtheme*) Indica que un LA determinado forma parte de un subtema. Sirve además para diferenciar cada LA que es parte de un subtema, que a su vez es parte de un tema, de una materia.
15. (*isPartOfTheme ?subt - subtheme ?t - Theme*) Indica que un subtema forma parte de un tema determinado. Sirve además para diferenciar un subtema de cualquier otro en un tema.
16. (*isPartOfSubject ?t - Theme ?subj - subject*) Indica que un tema forma parte de una materia determinada. Sirve además para diferenciar un tema de cualquier otro en una materia.

FUNCIONES

Nos permite utilizar expresiones numéricas las cuales se construyen utilizando operadores aritméticos de expresiones numéricas primitivas. Cabe mencionar que la sintaxis que se utiliza es prefija para todos los operadores aritméticos, incluyendo los

predicados de comparación, con el fin de simplificar el análisis. Los números no se distinguen en sus posibles funciones, por lo que los valores pueden representar, por ejemplo, las cantidades de recursos, acumulación de scores, % de temas, tiempo, contadores. Esto es importante y es una de las bondades de estas funciones numéricas, ya que pueden representar cualquier valor definido por el usuario.

Una característica de las funciones en el modelado es que una vez definidas en el *dominio* podemos asignarle cualquier valor en el archivo problema, lo cual facilita la modelación, cuando se trata de expresiones numéricas.

A continuación se muestra un listado de las funciones utilizadas para representar los diversos valores necesarios en nuestra modelación:

1. (*credits-subject ?subj - subject*) indica los créditos que tiene una materia en su plan de estudios. Este valor es definido por el usuario en el *problema*.
2. (*total-credits-subject-gain ?s - student*) es la cantidad de créditos acumulados que tiene un estudiante a lo largo de su carrera, es decir, la suma de los créditos que tiene cada materia que ha aprobado. Esta métrica se va acumulando para cada estudiante, sin embargo, puede ser inicializada en cualquier valor que represente los créditos de materias previamente aprobadas por el estudiante.
3. (*available-credits ?s - student*) es una cantidad que se establece para describir la cantidad de créditos que tiene “disponible” un estudiante para poder inscribir materias, por ejemplo, si se establece esta función en 48, y cada materia tiene 8 créditos, el estudiante podrá inscribir 6 materias. Esto se establece para que un estudiante no pueda inscribir más materias. Esto es definido en el *problema*.
4. (*score ?req - LO ?s - student*) indica el “score” de un subtema. Esta función es el resultado de la multiplicación del valor que tiene un LA seleccionado por el porcentaje del subtema al que pertenece por ejemplo, el LA 1 del ejemplo presentado en la figura 3.18 tiene un valor de 30 puntos y el subtema 1 al que pertenece tiene un porcentaje del 50 %, entonces, $30 * .50 = 15$, el score del

subtema 1 tiene un valor de 15. Recordemos que esta función se irá acumulando a medida que se realicen las diferentes actividades de aprendizaje de un subtema, de manera que asegure una calificación mínima para el subtema.

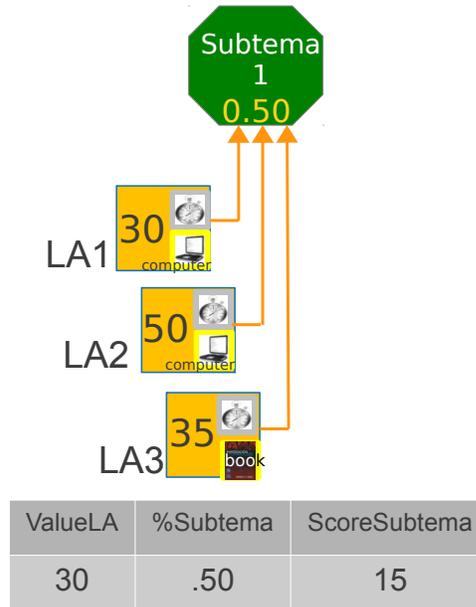


Figura 3.18: Ejemplo de la función de score

5. (*grades ?subj - subject ?s - student*) indica la “calificación” final de una materia, al haberse realizado todas las actividades de aprendizaje, esta función se incrementa en la evaluación de cada tema con el resultado de los score de cada subtema que pertenecen a ese tema.
6. (*quantity-resource ?eq - resource*) indica la cantidad de recursos que se tiene disponibles, es decir, 40 computadoras, 30 libros, etc, esto es definido en el problema.
7. (*valueLA ?oa - LA*) es el “valor” que se establece a una actividad de aprendizaje, representando de alguna manera el grado de aprendizaje que se obtendría al realizar dicha actividad del subtema al que pertenece. Nosotros definimos estos valores asignados a los LA en números enteros positivos no mayor 100. Tomando en cuenta que estos valores es importante que la suma de los que

pertenecen a un subtema en específico sean al menos mayor que la calificación definida como mínima.

8. (*percentSubtheme ?req - LO*) indica el porcentaje que tiene un subtema dentro de un tema, ya que como habíamos mencionado, cada subtema de un tema en específico tiene una ponderación. Esto nos sirve también, para manipular la “acumulación de calidad”, es decir la calificación que se irá “formando” como resultado de la realización de las actividades de aprendizaje en cada subtema, de cada tema en una materia. Con el fin de normalizar los datos, la suma de las ponderaciones de todos los subtemas de un tema debe ser igual a uno.
9. (*percentTheme ?t - Theme*) indica el porcentaje que tiene un tema determinado en una materia. Al igual que en los subtemas, cada tema tiene asignado una ponderación. La suma de las ponderaciones de todos los temas que pertenecen a una materia determinada debe ser igual a uno.
10. (*mingrade ?subj - subject*) indica la calificación mínima indicada como aceptable para que un estudiante apruebe una materia. Esto es definido por el usuario en el *problema*. En nuestra modelación asignamos el valor de 70, ya que en la licenciatura esa es la calificación mínima de aprobado, pero pudiera ser 80 (si fuera el caso de la calificación mínima de maestría), u otro valor.
11. (*DurationLA ?oa - LA*) es el tiempo que se establece de duración para realizar una actividad de aprendizaje. Por ejemplo, realizar la lectura de unas diapositivas tiene un tiempo de 20 minutos. Este tiempo es definido por el usuario, y puede estar o no vinculado con el valor asignado a cada LA. En nuestra modelación tenemos la suposición de que el valor de un LA y la duración de ésta tienen una relación directa, es decir a mayor tiempo, mayor valor del LA.
12. (*maxgrade-subtheme ?subt - subtheme*) indica la calificación máxima por subtema. Esto nos ayuda limitar por “arriba” la cantidad de LA a realizar, es decir, que el plan no arroje más LA de las que un alumno necesita para aprobar una materia. Nosotros utilizamos esta métrica para restringir al modelo

a no realizar demasiadas LA, de manera que la suma del valor de los LA no excedan la cantidad definida como máxima. En nuestra modelación esta función la establecimos con el valor de 100. Sin embargo pudiera tener 150 o 200 u otro valor, si el usuario no quiere restringir tanto el modelo.

13. (*amount-in-subtheme* - LA) contiene la cantidad acumulada de score, como si se contemplara una calificación parcial, que se necesita de un subtema para poder realizar cualquier otra actividad de un subtema posterior. Esta cantidad puede ser definida por el usuario en el *problema*. Nosotros estuvimos manejando solo dos valores para nuestro ejemplo: el cero y el 70. Sin embargo una de las ventajas de manejar funciones numéricas en los modelos es que pueden representar cualquier valor.

Para ejemplificar mejor esta función, diremos que para poder realizar cualquier actividad del subtema 5, por ejemplo, se necesita que el subtema anterior, es decir, el subtema 4 cumpla con alguna de las siguientes opciones (de acuerdo a lo que queremos modelar): a) se haya realizado al menos un LA de ese subtema, si es así se define la función *amount-in-subtheme* en 0. b) se hayan realizado todos los LA necesarios para obtener la calificación mínima. Como tenemos como calificación mínima el 70, entonces definimos el valor de la función *amount-in-subtheme* en 70. En la figura 3.19 se puede apreciar estas dos opciones de requerimientos a subtemas. En la experimentación solo se contempló la opción a.

Es importante resaltar que en todas las funciones mencionadas anteriormente sus valores son flexibles, es decir, es muy fácil para nosotros representar el hecho de que la calificación mínima de un estudiante es de 80 (como en maestría) en lugar de 70. Muchas de estas funciones tienen esta característica, ya que lo que nosotros estamos representando es el concepto, pero a fin de cuentas el valor es conforme a las necesidades del usuario. Esto sería como una preferencia que es capaz de capturar nuestro modelo.

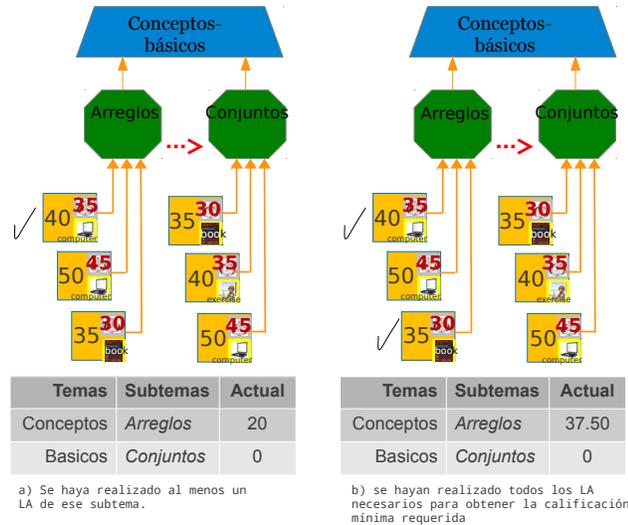


Figura 3.19: Gráfica que ilustra cuando para realizar cualquier LA de un subtema se necesita ya sea que haya realizado al menos un LA (opción a) o que haya completado todas las LA necesarias en el subtema requerido.

ACCIONES

Las acciones son las formas de cambiar el estado [4]. Las acciones son básicamente funciones de transición que nos permiten cambiar el estado del mundo. Para su especificación se necesita: el *nombre de la acción*; la *condición previa* que es una conjunción de átomos (literales positivas) que dice qué debe existir en el mundo antes de poder aplicar la acción, además de funciones numéricas para ciertas acciones de nuestro modelo, y el *efecto de una acción* que es una conjunción de literales (positivas o negativas) que dice de qué manera cambia la situación (el estado del mundo) al aplicar el operador.

A continuación se mencionan las acciones que contemplamos en nuestro modelo:

- (: *durative-action enroll-subject - Materia*).

Esta acción permite “inscribir” una materia. *Materia*. Por ejemplo, la acción de *inscribir-FisicaII*, a diferencia de la acción *inscribir-FisicaI*, nece-

sita agregar en las precondiciones la restricción de haber *aprobado* FísicaI y Matemáticas II. En la figura 3.20 se muestra la acción de inscribir la materia FisicaII.

```
(:durative-action enroll-subject_FisicaII
:parameters (?s - student)
:duration (= ?duration 1)
:condition (and
  (at start (available-subject FisicaII ?s))
  (at start (not-approved FisicaII ?s))
  (at start (<(credits-subject FisicaII)(available-credits ?s)))
  (at start (pass-degree FisicaI))
  (at start (pass-degree MatematicasII))
)
:effect (and
  (at start (enrollment ?s FisicaII))
  (at start (decrease (available-credits ?s)(credits-subject FisicaII)))
  (at start (not (available-subject FisicaII ?s)))
)
)
```

Figura 3.20: Acción de inscribir la materia de Física II en PDDL.

En esta acción pueden verse las precondiciones necesarias para que dicha acción pueda ser aplicable. Entre sus precondiciones están: que al inicio de la duración de la acción que haya disponibilidad de la materia, no haberla aprobado anteriormente, que la cantidad de créditos que tiene esa materia no exceda la cantidad de créditos que tiene el estudiante disponibles por semestre (una forma de control en la cantidad de materias a inscribir). Además de esto, sólo en el caso de materias que necesitan de haber aprobado otra(s) materia(s), se incluye la precondición de haber aprobada dicha(s) materia(s).

Tiene como efecto que: el estudiante quede “inscrito” en esa materia; se calcula una métrica para controlar la cantidad de materias que puede inscribir e indica que la cantidad de créditos disponibles decrece de acuerdo a la cantidad de créditos que tiene al cursar esa materia; y se realiza la negación de que la materia este *disponible* para ese estudiante, de manera que no pueda volver a “inscribirla”.

Manejamos 3 acciones para representar los distintos escenarios que pudieran presentarse al momento de querer realizar las actividades

de aprendizaje:

- (*: durative-action CHOOSE-LA-nothasreqs*).

Esta acción nos permite modelar las actividades de aprendizaje que no tienen ningún prerrequisito para poder ejecutarse, es decir, no necesitan que alguna actividad de aprendizaje se haya realizado antes. Como ejemplo, podemos mencionar las LA del Subtema 1 del Tema 1 de la materia de Física, si consideramos que los temas y subtemas mantienen un orden de realización, podemos observar que no necesitan de ninguna otra actividad de aprendizaje previa.

Entre sus precondiciones existen: que el estudiante esté disponible (libre) para poder realizar esta LA; que el estudiante esté inscrito en la materia; que el estudiante no haya realizada esta misma LA anteriormente; se establece una relación para identificar que un LA pertenece a un subtema, y éste pertenece a un tema, que a su vez pertenece a una materia determinada con los predicados *isPartOf*; otra precondición es el tipo de recurso que tiene esa LA; que la cantidad que se establezca en el modelo de ese recurso sea mayor a cero, es decir, que los recursos estén disponibles y no se esté utilizando por algún otro estudiante; que la calificación máxima por subtema supere el valor que tiene el LA, esto nos sirve como control, para no realizar más LA de los necesarios; y por último el predicado que indica que esa LA no tiene prerrequisitos.

Los efectos de realizar esta acción son: al inicio de esta acción (porque recordemos que tiene duración) el estudiante no está disponible (*not(free ?s)*); asimismo, la cantidad del recurso utilizado para esta LA disminuye (porque se está utilizando). Como efecto al término de la acción se vuelve a “liberar” el recurso utilizado; asimismo el estudiante ahora está disponible (libre); se establece que el estudiante ya realizó esta LA, aquí ponemos atención a dos cosas, se niega el predicado de la precondición (*not-done-LA*) para que no vuelva a entrar a esta acción (para la misma LA) y se establece el predicado de (*done ?oa*) para que indique que ya está realizado este LA, en caso de que algún otro LA necesitara de éste, como veremos en las otras acciones que modelamos para las

actividades de aprendizaje.

Características:

- Esta acción se utiliza para representar a todas las LA que cumplan con esta condición de no tener prerrequisitos.
- Es una acción durativa, en donde su duración depende de la duración de la actividad de aprendizaje modelada.
- Sus parámetros son variables, de manera que puede recibir cualquier actividad de aprendizaje que cumpla con las precondiciones. Esto puede ser así, ya que incluye predicados que aseguran que un LA pertenezca a un subtema, y ese subtema pertenezca a un tema y que ese tema pertenezca a una materia determinada y no ninguna otra.

2)

□ (*: durative-action CHOOSE-LA-hasreqsSubtheme*).

Esta acción nos permite modelar las actividades de aprendizaje que requieren o dependen para poder ejecutarse, de la cantidad que un subtema en específico, ya sea anterior o no, tenga acumulado de *score*. Es decir, necesitan que algunas actividades de aprendizaje se hayan realizado en un subtema determinado, de manera que hayan *acumulado calidad* en estos subtemas. Como ejemplo, podemos mencionar las LA del Subtema 3 del Tema 1 de la materia de Física, si consideramos que los temas y subtemas mantienen un orden de realización, podemos observar que para realizar cualquier LA del subtema 3 necesito haber realizado otra(s) actividad(es) de aprendizaje de los subtemas 1 y 2. Para modelar esto, se establece un orden en los subtemas inmediatos anteriores, en este ejemplo, sería el subtema 2, pero podría darse el caso, de que se quiere modelar que un subtema determinado necesita que un subtema aunque no sea inmediatamente anterior, se realice. Para representar los diferentes valores que podría tener el subtema requerido se deja como variable la cantidad o valor que se requiere en el subtema, esto para dar flexibilidad a las siguientes situaciones:

a)

El score del subtema sea mayor a cero. Esto indica que para poder realizar cualquier LA de un subtema, el estudiante necesita haber realizado “al menos” una LA del subtema requerido.

b)

El score del subtema sea mayor a la calificación mínima (70). Esto indica que para poder realizar cualquier LA de un subtema, el estudiante necesita haber aprobado el subtema requerido, es decir, ya completó todo el subtema requerido.

Las precondiciones de esta acción son: similares a la acción anterior, con la diferencia de que en lugar del predicado (*not-has-reqs*) que indicaba que no tenía prerrequisitos, esta acción tiene el predicado (*has-reqs*) y la precondición que indica que se hayan realizado actividades de aprendizaje del subtema requerido, de manera que haya acumulado calidad en ese subtema. A continuación se muestra esta precondición:

$$(> (score \ ?req \ ?s)(* (amount-in-subtheme \ ?oa)(percentSubtheme \ ?req)))$$

Los efectos de realizar esta acción son: iguales a la acción anterior, ya que entre otras cosas, se indica que ese LA se ha realizado.

Características:

- Esta acción se utiliza para representar a todas las LA que cumplan con esta condición de tener prerrequisitos en los subtemas.
- Es una acción durativa, en donde su duración depende de la duración de la actividad de aprendizaje modelada.
- Sus parámetros son variables, de manera que puede recibir cualquier actividad de aprendizaje que cumpla con las precondiciones.
- Para representar los diferentes valores que podría tener el subtema requerido se deja como variable la cantidad o valor que se requiere en el subtema.

3)

□ (*: durative-action CHOOSE-LA-hasreqsLA*).

Esta acción nos permite modelar las actividades de aprendizaje que requieren o dependen para poder ejecutarse, de que se haya realizado una actividad de aprendizaje en específico. Es decir, necesitan que alguna actividad de aprendizaje se haya realizado ya sea en su mismo subtema o cualquier otro subtema anterior. Como ejemplo, supongamos que tenemos una secuencia de LA en un determinado tema, en donde estas actividades de aprendizaje tienen en común un recurso como un libro y en el subtema 1 tenemos un LA que es “leer capítulo 1”, en el subtema 2 del mismo tema, está el LA “leer capítulo 2”, entonces existe una secuencia entre estos dos LA, así que se establece una dependencia, de que para realizar el LA de “leer el Capítulo 2” se debió haber realizado el LA de “leer capítulo 1”.

Las precondiciones de esta acción son: similares a la acción anterior, con la diferencia de que en lugar de la precondición que indica que se hayan realizado actividades de aprendizaje del subtema requerido, se agrega el predicado (*done ?req*) mencionado anteriormente, que indica que se haya realizado el LA que se requiere, anteriormente.

Los efectos de realizar esta acción son: iguales a la acción anterior, ya que entre otras cosas, se indica que ese LA se ha realizado.

Características:

- Esta acción se utiliza para representar a todas las LA que cumplan con esta condición de tener prerequisites de otro LA anterior, aunque no necesariamente inmediatos.
- Es una acción durativa, en donde su duración depende de la duración de la actividad de aprendizaje modelada.
- Sus parámetros son variables, de manera que puede recibir cualquier actividad de aprendizaje que cumpla con las precondiciones.

- Se observa que el mismo predicado que se ha estado utilizando para indicar que un LA se ha realizado, se utiliza como precondition en esta acción para representar la dependencia hacia otro LA realizado anteriormente (aunque no necesariamente el anterior inmediato).

4)

□ (*durative-action CHOOSE-LA-hasreqs-multipleLA2*).

Esta acción nos permite modelar las actividades de aprendizaje que requieren o dependen para poder ejecutarse, de que se hayan realizado otras actividades de aprendizaje anteriormente. Es decir, necesitan que algunas actividades de aprendizaje se hayan realizado ya sea en su mismo subtema o cualquier otro subtema anterior. Esta acción a diferencia de la anterior, es la cantidad de LA que se requieren.

Las condiciones de esta acción son: similares a la acción anterior, con la diferencia de que se agrega el predicado (*done ?req*) para cada LA requerido; se establece el predicado (*has-multiple-reqs ?oa ?req1*) en lugar del predicado (*has-reqs ?oa ?req*), y al igual que con el predicado (*done ?req*) se agrega para cada LA requerido. En esta precondition si tenemos 2 LA requeridos es importante agregar el predicado (*not (= ?req1 ?req2)*) en donde nos indica que los LA requeridos son diferentes.

Los efectos de realizar esta acción son: iguales a la acción anterior, ya que entre otras cosas, se indica que ese LA se ha realizado.

Características:

- Esta acción se utiliza para representar a todas las LA que cumplan con esta condición de tener prerrequisitos de otros LA anteriores, aunque no necesariamente inmediatos.
- Es una acción durativa, en donde su duración depende de la duración de la actividad de aprendizaje modelada.

- Sus parámetros son variables, de manera que puede recibir cualquier actividad de aprendizaje que cumpla con las precondiciones.
- Se observa que el mismo predicado que se ha estado utilizando para indicar que un LA se ha realizado, se utiliza como precondición en esta acción para representar la dependencia hacia otro LA realizado anteriormente (aunque no necesariamente el anterior inmediato).
- Esta acción es para representar que un LA requiere de DOS LA anteriores, si se quisiera representar que ese LA requiere de más esa cantidad, entonces se tendrá que modelar otra acción similar que represente esa cantidad de LA necesarias, es decir si un LA requiere de otros 3 LA entonces se modelará otra acción con otro nombre, similar tal vez, en donde en lugar de tener el número 2 en el nombre sea el 3, y se agregarán los predicados *(done ?req1)* y *(has-multiple-reqs ?oa ?req1)* tantas veces sea la cantidad de LA requeridos. Como ejemplo, tenemos un LA que requiere de otros 3 LA anteriores. Modelamos esta acción con el nombre: CHOSE-LA-hasreqs-multipleLA3, en sus parámetros ahora tiene 3 LA; req1, req2, req3. En la precondición además de los predicados “normales” de esta acción se agregan los siguientes predicados por cada LA requerido:

(done ?req1)

(done ?req2)

(done ?req3)

además de:

(has-multiple-reqs ?oa ?req1)

(has-multiple-reqs ?oa ?req2)

(has-multiple-reqs ?oa ?req3)

5)

En el caso en que un LA requiera de un subtema (con score mayor a cero ó a 70) y al mismo tiempo, requiera otro LA, esto será un OR, ya que no

podrá entrar a varias acciones un mismo LA, o entra a la acción de *CHOOSE-LA-has-reqsSubtheme* ó a la acción de *CHOOSE-LA-has-reqsLA* ó a ninguna de estas dos y el planificador decide seleccionar otras LA que cumplan con el score necesario para aprobar ese subtema.

Acciones para los temas:

Como cada materia de un plan de estudios tiene una determinada cantidad de temas a cubrir, y estos temas son diferentes para cualquier otra materia, nosotros representamos en nuestro modelo una *acción* para cada tema. Es decir, no podemos decir que todas las materias tienen la misma cantidad de temas, ni que dichos temas son iguales para cada materia. De manera que es necesario representar una acción para cada tema que forma parte de una materia a modelar.

Estas acciones son diferentes a las anteriores, ya que no son acciones que tenga que realizar el estudiante, sino que forman parte del sistema y tienen la finalidad de evaluar que los subtemas que lo comprenden están completos, es decir, si tienen la suficiente *acumulación de calidad*. Esto quiere decir, que es en este tipo de acciones donde los diseñadores podrían modificar la forma como se acumula la calidad, o el tipo de acumulación de score.

□ (*: durative-action PASS-Theme-Tema1100 - Materia*).

Esta acción es para asegurar que se hayan realizado los subtemas comprendidos en el tema. Tiene como único parámetro al estudiante, ya que los subtemas tendrán que ser declarados explícitamente en dicha acción, a esto se le llama “ground”. Sus precondiciones son: que el estudiante esté inscrito en la materia; que el *score* del subtema sea mayor o igual a la calificación mínima como aprobatoria (multiplicada por el porcentaje del subtema). Esta precondición se repite para cada uno de los subtemas que comprenden ese tema. Como efecto tenemos que: la función que acumula la calificación total de la materia se incrementa con el score del subtema multiplicado por el porcentaje del

tema, esta precondition se repite para cada uno de los subtemas; y por último se indica que ese Tema se ha realizado.

Como ejemplo, tenemos que la materia Química tiene 5 temas: Tema1, Tema2, Tema3... Supongamos que el tema 1 tiene solo 2 subtemas: Subtema1, Subtema2.

la acción del tema 1 sería:

(: durative-action PASS-Theme-Tema1-QuímicaI)

...

:condition (and

...

(at start (\geq (score Subtema1 ?s)((mingrade QuímicaI)(percentSubtheme Subtema1))))*

(at start (\geq (score Subtema2 ?s)((mingrade QuímicaI)(percentSubtheme Subtema2))))*

:effect (and

(at start (increase (grades QuímicaI ?s) ((score Subtema1 ?s)(percentTheme Tema1))))*

(at start (increase (grades QuímicaI ?s) ((score Subtema2 ?s)(percentTheme Tema1))))*

Características de esta acción:

- Como ya se había mencionado se realiza una acción por cada tema de la materia a modelar.
- Dentro de cada acción de tema, se repiten las condiciones por cada subtema que comprenda ese tema.
- Como decimos que tenemos relaciones de orden entre los temas, no se evalúa el tema 2 de una materia, si antes no se evaluó el tema 1. Para representar esto se agrega la precondition *(done-Theme Tema - Materia)* a la acción del tema para indicarle que necesita de la evaluación de un

tema anterior. Siguiendo el ejemplo anterior, para el Tema2 de Química sería:

(: durative-action PASS-Theme-Tema2-QuimicaI)

...

:condition (and

(at start (done-Theme Tema1 QuímicaI ?s))

...

para el Tema3 sería:

(: durative-action PASS-Theme-Tema3-QuimicaI)

...

:condition (and

(at start (done-Theme Tema2 QuímicaI ?s))

...

Actividad de aprendizaje obligatoria

Si se quisiera representar que una actividad de aprendizaje es obligatoria en un tema, como por ejemplo, un examen, la opción en nuestro modelo es que se agregue la siguiente precondition al tema en el que es obligatorio.

(at end (done LAObligatorio Materia ?s))

Acción por materia

□ *(: durative-action PASS-Materia)*

Esta acción al igual que las acciones de temas, es una acción propia del sistema que no realizará el estudiante. Se modela una acción por cada materia lo cual nos ayuda a asegurar que se hayan realizado todos los temas de la materia. La precondition es que el alumno esté inscrito en la materia; que se hayan realizado los temas que lo comprenden, esto se verifica con el predicado *(done-Theme)*, efecto de las acciones por cada tema.

Este predicado se repite por cada tema que comprenda la materia:

(at start (done-Theme Tema1 Quimica1 ?s))

(at start (done-Theme Tema1 Quimica1 ?s))

...

Como efecto se declaran realizados todos los LA de esa materia con el predicado: *(done-subject-LA Materia ?s)*

Acción para evaluar si cualquier materia se aprueba

- *(: durative-action take-subject-pass)* Esta acción es propia del modelo y es utilizada para “evaluar si cualquier materia es aprobada”. No es necesario realizar una acción por cada materia debido a que esta acción solo evalúa que se hayan realizado todas las acciones necesarias de manera que el estudiante haya aprobado dicha materia. Las precondiciones para ejecutar esta acción son: que el alumno esté inscrito en esa materia; que haya realizado todas las actividades de aprendizaje necesarias para aprobar la materia, eso es representado con el predicado *(done-subject-LA Materia ?s)* ya que para que este predicado sea verdadero tuvieron que haberse realizado otras acciones (anteriores a ésta); y finalmente, que la calificación que obtuvo en esa materia (que se fue acumulando en acciones anteriores) es suficiente para aprobarla. Lo que da como efecto que: el alumno aprueba la materia; se niega que la materia está disponible, es decir, no está disponible ahora; se niega el predicado que indicaba que no estaba aprobada la materia, es decir, negamos que no está aprobada, entonces ahora está aprobada. Este tipo de acciones dependen mucho del diseñador del modelo, ya que podría tener algún otra regla, nosotros incrementamos, es decir, sumamos la cantidad de créditos obtenidos con el crédito de la materia.

REPRESENTACIÓN DE LAS ACCIONES MODELADAS

A continuación se muestra en la figura 3.21 la representación de todas las acciones contempladas en nuestro modelo, los predicados que se utilizan y las funciones numéricas para representar las métricas, y cómo es que éstos se relacionan con las acciones en sus precondiciones y efectos. Los rectángulos azules representan las acciones, los verdes los predicados, los morados las métricas. Las líneas son para representar las relaciones que existen entre las acciones, los predicados y las funciones, aquellas líneas que están “antes” de la acción son sus precondiciones, asimismo las líneas que están después de la acción su(s) efectos. Es importante notar que los efectos de una acción sirven como precondiciones para otras acciones.

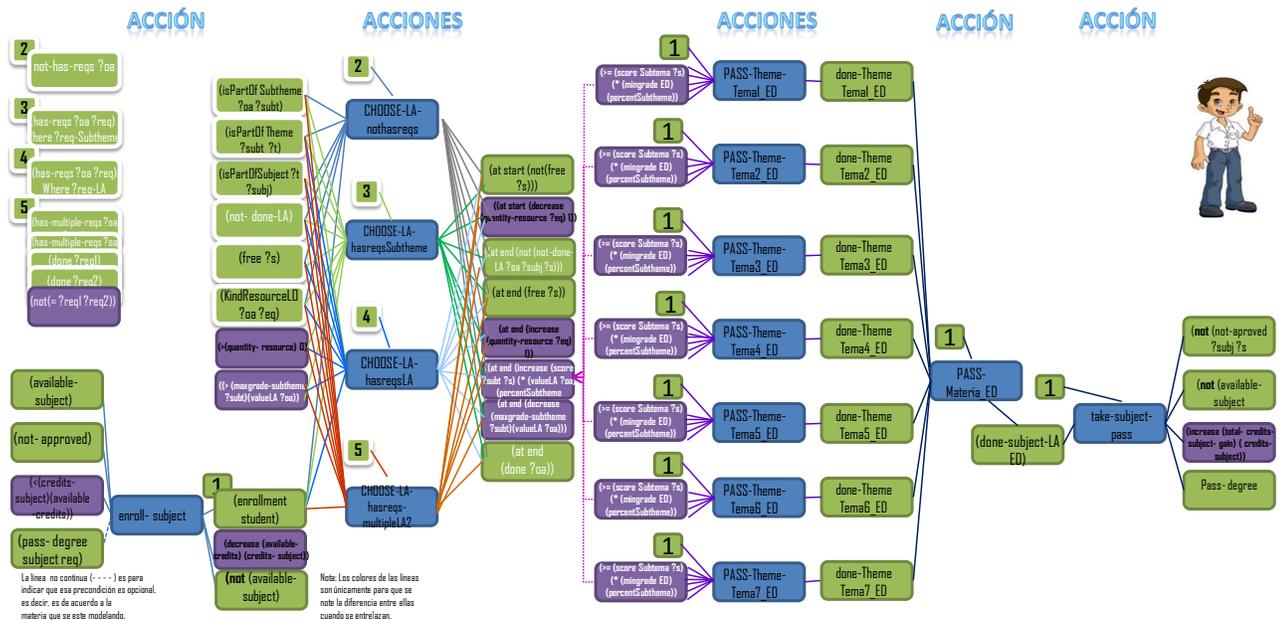


Figura 3.21: Representación de las acciones modeladas.

Se puede observar que dentro de los rectángulos los textos son negros o blancos, esto se hizo a propósito, ya que el texto blanco nos indica que ese predicado o función representa a un conjunto de ellos, es decir, para el predicado *(not-done-LA)*

habrá tantos predicados como actividades de aprendizaje estén modeladas. Si hay 15 LA habrá 15 predicados (*not-done-LA*).

Para representar aquellos predicados que deben definirse en el archivo problema, en la imagen el rectángulo tiene orilla blanca. Y solo para mantener un orden en la imagen y no empalmar acciones se utilizan los cuadrados pequeños que son como etiquetas representando a los predicados previamente etiquetados.

La imagen anterior nos sirve para visualizar el diseño de la estructura del modelo. Es importante aclarar que no está representando la manera en como lo resuelve algún planificador, sino cómo están estructuradas sus acciones, así como su relación con los predicados y funciones ya sea como precondiciones o efectos.

3.4.3 PROBLEMA

En el archivo *problema* es en donde se especifican los *objetos* a modelar (objects), en este caso es el estudiante; la situación actual del estudiante o dicho en términos de planificación, estado inicial (init), es decir, que materias tiene aprobadas, cuantos créditos tiene, si tiene alguna actividad de aprendizaje ya realizada, etc.; los objetivos (goal) tales como “aprobar la materia X ”, “realizar el tema Y ”, etc.; y por último, las métricas a optimizar, como el tiempo, la calificación, etc.

Las materias, temas, subtemas y actividades de aprendizaje, así como los recursos, también son objetos, pero éstos se definen en el dominio como “constantes”.

OBJETOS

Los objetos son las cosas en el mundo educativo que nos interesa.

Entre los objetos están:

1. En el *problema*:

a) *student*.- Hace referencia al estudiante a modelar

2. En el *dominio*:

- a) *subject*.- Es la materia.
- b) *subtheme*.- Subtema.
- c) *LA*.- Actividad de aprendizaje.
- d) *resource*.- Recursos.

ESTADO INICIAL

El estado inicial es el estado del mundo en el que se empieza. En nuestro modelo es la situación actual del estudiante. Para definir un estado inicial hacemos uso de los predicados que establecimos en el dominio. Se asume un “mundo cerrado”, es decir, cualquier predicado que no se mencione en el estado inicial se asume que no existe.

En el estado inicial (*init*) se especifican:

- La disponibilidad del estudiante.
- Las materias disponibles para inscribir.
- Materias no aprobadas por el estudiante.
- Actividades de aprendizaje no realizados de cada materia por estudiante.
- La especificación de cual actividad de aprendizaje pertenece a que subtema.
- La especificación de cual subtema pertenece a que tema.
- La especificación de cual tema pertenece a que materia.
- La especificación de que recurso pertenece a cada actividad de aprendizaje.
- La especificación de cuáles actividades de aprendizaje no tienen requisitos.

- La especificación de que una actividad de aprendizaje tiene requisitos de otro LA.
- La especificación de que una actividad de aprendizaje tiene como requisitos varios LA.
- La especificación de que una actividad de aprendizaje tiene como requisito un subtema, y aquí se especifica también la cantidad acumulada de score que se requiere del subtema. Nosotros la definimos con el valor cero, si solo deseamos que haya empezado alguna LA de ese subtema, o con un valor de 70 si lo que deseamos es que lo haya completado antes de realizar cualquier LA (Considerando que estamos tomando el 70 como calificación mínima. Estos valores son definidos por el usuario.

Se especifica además las cantidades iniciales de:

- Créditos que tiene actualmente este estudiante.
- La suma de las calificaciones actualmente aprobadas.
- La calificación (grades) de la materia de ese estudiante.
- Créditos que tiene cada materia.
- Calificación mínima de aprobado por materia.
- Cantidad de créditos disponibles por alumno para inscribir.
- El score de las métricas de cada subtema que lleva el estudiante. Actualmente las modelamos inicializadas en cero, para indicar que no ha realizado nada de esa materia.
- Cantidad de cada recurso que existe.
- El Valor de cada *actividad de aprendizaje*.
- El porcentaje (puntaje) de cada Subtema.

- El porcentaje de cada tema.
- La duración de cada actividad de aprendizaje.
- La calificación máxima por subtema. Nosotros definimos esta calificación máxima por subtema con el valor 100.

En el mundo real todas estas especificaciones dadas en el problema, serían la entrada a nuestros modelos a partir del usuario, LMS, o cualquier otro mecanismo de entrada de datos. Considerando que estos modelos propuestos forman parte de un sistema educativo completo, el cual sería el encargado de proporcionar toda esta información.

Para esta tesis, nosotros utilizamos un programa generador de instancias, tomando datos de un archivo de entrada, donde solo estuvimos variando la cantidad de materias y las clases, por motivo de experimentación.

ESTADO FINAL - OBJETIVO

Especificación de Objetivo: Las cosas que queremos sean verdad. AL igual que en el *estado inicial*, en el *objetivo* se establecen los *predicados* que queremos sean verdad.

Básicamente cualquier función numérica o predicado puede ser utilizado como objetivo. Además, se pueden modelar varios objetivos a la vez, como por ejemplo, aprobar varias materias. Algunos ejemplos de objetivo que se pueden modelar son:

- Que apruebe una materia específica.
Ejemplo: *(pass-degree FisicaI student1)*
- Que apruebe un tema específico.
Ejemplo: *(done-Theme Tema1200 FisicaI student1)*
- Que realice una actividad de aprendizaje en específico.
Ejemplo: *(done L1111)*

- Que se haya realizado al menos un LO de un subtema en específico.
Ejemplo: ($>$ (*score Subtema1110 student1*) 0)
- Que realice los LO de ese subtema asegurando una cantidad de promedio en ese subtema. Aquí el valor puede cambiar de acuerdo al valor considerado como mínimo por el usuario. En nuestro caso, el valor asignado es 70 como calificación mínima. ($>$ (*score Subtema1120 student1*) 70)
- Si se quiere realizar TODOS los LO de ese subtema en específico. En la modelación propuesta, esto no es intuitivo, de manera que se tiene que realizar un cálculo adicional, en donde se sumen los valores de los LO. Es decir tenemos 3 actividades de aprendizaje con valor de 30, 40 y 50 cada uno, así que realizamos una sumatoria de sus valores y tenemos: $30 + 40 + 50 = 120$
($>$ (*score Subtema1110 student1*) 120)
- Que la calificación de una materia sea mayor a cero, es decir que ya haya aprobado algún tema.
Ejemplo: ($>$ (*grades FisicaI student1*) 0)
- Que la calificación de una materia sea mayor a una cierta cantidad, como 70, 80, etc.
Ejemplo: ($>$ (*grades FisicaI student1*) 70)
- Que el alumno quede inscrito en una materia.
Ejemplo: (*enrollment student1 FisicaI*)
- Que el alumno realice los LA de todos los subtemas de una materia.
Ejemplo: (*done-subject-los FisicaI student1*)

Si se quieren modelar varios objetivos tales como aprobar varias materias, se tendría que especificar de la siguiente manera:

(*:goal (and*
(*pass-degree FisicaI student1*)
(*pass-degree MatematicasI student1*)

(pass-degree QuimicaI student1)
))

MÉTRICAS A OPTIMIZAR

Cualquier función que se haya declarado en el *dominio* se puede utilizar como métricas a optimizar, incluyendo el tiempo total del plan.

Las métricas a optimizar que pudieran modelarse son:

- Minimizar el tiempo total.

Ejemplo: *(:metric minimize (total-time))*

- Maximizar el total de créditos obtenidos.

Ejemplo: *(:metric maximize (total-credits-subject-gain student1))*

- Maximizar la calificación del estudiante.

Ejemplo: *(:metric maximize (grades FisicaI student1))*

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En esta sección se describe la manera como fue llevada la experimentación. La intención es generar los dominios y problemas de planificación de una manera automática para observar el funcionamiento de los planificadores al momento de tratar de solucionar dichos modelos generados. La creación de los modelos se realizó de manera aleatoria respetando ciertos criterios mencionados más adelante.

Para las pruebas se modeló un solo estudiante. Se asume que los recursos asociados a las actividades de aprendizaje son suficientes. El objetivo definido en el problema es que apruebe la(s) materia(s) que sean definidas en el dominio. La métrica a optimizar es el “total-time”, es decir, minimizar el tiempo total del plan. Aunque hay que aclarar que no es lo único que puede modelarse, ya que una vez teniendo el dominio, en el archivo problema podrían modelarse más “situaciones”, como la cantidad de estudiantes, otro objetivo, otra métrica, etc.

A continuación se describe el desarrollo de la experimentación:

4.1 GENERADOR DE INSTANCIAS

Para generar los dominios y problemas de planificación de manera automática, se realizó un programa en el lenguaje C. Este generador toma los datos de un archivo de entrada en el cual se varían la cantidad de materias y la clase (pequeña, mediana y grande) descrita más adelante. Posteriormente por medio de un script se manda

llamar a los planificadores seleccionados, en nuestro caso son SGPLAN y LPG, para resolver los dominios y problemas y si encuentran solución generar los planes.

Para los requerimientos tenemos que:

-Únicamente las actividades de aprendizaje que pertenecen al primer subtema del primer tema de una materia no tendrán ningún requerimiento. Es decir, siendo del primer subtema de la materia no puede depender de un subtema previo, porque no existe.

-Todas las demás actividades de aprendizaje, que no sean del primer subtema de cada materia, tendrán dependencia a un subtema anterior, siempre y cuando pertenezcan a una misma materia. Aquí tenemos dos tipos de orden entre los subtemas: que sea al subtema inmediato anterior ó a cualquier otro subtema del conjunto de subtemas anteriores a él. En la experimentación estos datos quedaron fijos.

-De las actividades de aprendizaje (que no son del primer subtema) un porcentaje de ellas tienen dependencia a otra actividad de aprendizaje. Variando la cantidad de actividades de aprendizaje de acuerdo a la clase que se está modelando.

-Asimismo para aquellas actividades de aprendizaje que tienen dependencia a múltiples actividades de aprendizaje. En nuestro caso solo se modeló a dos actividades anteriores.

4.1.1 CREACIÓN DE TIPOS DE MODELOS Y CLASES.

En la tabla 4.1 se muestra los dos tipos de modelos que se crearon: con requerimientos y sin ellos. Llamamos requerimientos a las relaciones de precedencia que existen en el modelo entre los subtemas y actividades de aprendizaje. Estos a su vez se clasifican en 3 clases: pequeña, mediana y grande. Las clases se refieren a la cantidad de objetos a modelar. La clase pequeña comprende de 1 a 3 temas por materia, en donde a su vez cada tema tiene entre 1 y 3 subtemas cada uno. Y finalmente cada subtema tiene entre 1 y 3 actividades de aprendizaje. Teniendo en

el peor de los casos para cada materia: 27 actividades de aprendizaje para la clase pequeña, 64 para la clase mediana y 343 para la clase grande.

Tipos	Clases	<i>Tamaños</i>
Con requerimientos	Pequeña	1-3 Temas, Subtemas y Actividades de aprendizaje
	Mediana	3-5 Temas, Subtemas y Actividades de aprendizaje
	Grande	5-7 Temas, Subtemas y Actividades de aprendizaje
Sin requerimientos	Pequeña	1-3 Temas, Subtemas y Actividades de aprendizaje
	Mediana	3-5 Temas, Subtemas y Actividades de aprendizaje
	Grande	5-7 Temas, Subtemas y Actividades de aprendizaje

Tabla 4.1: Descripción de los tipos de modelos y sus clases

4.1.2 ARCHIVO DE ENTRADA PARA EL GENERADOR DE INSTANCIAS

Se realizó un archivo de entrada, en donde vienen las especificaciones del modelo a interpretar en los dominios/problemas de planificación. De este archivo de entrada tomamos los datos que irán variando para correr nuestra experimentación.

Los datos que varían del archivo de entrada son:

- La cantidad de materias a modelar. Modelando a partir de 1 a 5 materias.
- Probabilidad que define la cantidad de temas, subtemas y actividades de aprendizaje a ser de determinada clase. Es decir, si queremos modelar una clase grande, se le da una probabilidad arriba del 90% a temas, otra probabilidad igual a subtemas y después a las actividades de aprendizaje.

En este archivo de entrada tenemos datos que son necesarios para la modelación y los cuáles se mantuvieron fijos durante la experimentación: Probabilidad de tener una materia previa. Al momento de modelar una cierta cantidad de materias cual es la probabilidad de esa materia tenga una materia requerida. Tipos y cantidad de recursos. Probabilidad de que una actividad de aprendizaje

sea obligatoria en un determinado tema de la materia. Probabilidad de orden entre los subtemas, es decir, manejamos dos tipos de orden entre subtemas: (1) a el subtema inmediato anterior o (2) al conjunto de subtemas anteriores al inmediato anterior.

4.2 DISEÑO DE EXPERIMENTOS

Se generaron 30 dominios y problemas de planificación para cada materia modelada variando las clases y si tiene o no requerimientos. En la tabla 4.2 se muestra a detalle cada subgrupo generado.

Con requerimientos	Sin requerimientos
1 materia clase pequeña	1 materia clase pequeña
2 materia clase pequeña	2 materia clase pequeña
3 materia clase pequeña	3 materia clase pequeña
4 materia clase pequeña	4 materia clase pequeña
5 materia clase pequeña	5 materia clase pequeña
1 materia clase mediana	1 materia mediana pequeña
2 materia clase mediana	2 materia mediana pequeña
3 materia clase mediana	3 materia mediana pequeña
4 materia clase mediana	4 materia mediana pequeña
5 materia clase mediana	5 materia mediana pequeña
1 materia clase grande	1 materia grande pequeña
2 materia clase grande	2 materia grande pequeña
3 materia clase grande	3 materia grande pequeña
4 materia clase grande	4 materia grande pequeña
5 materia clase grande	5 materia grande pequeña

Tabla 4.2: Subgrupos de dominios y problemas generados

4.3 PRUEBAS Y RESULTADOS

Para la realización de las pruebas a los dominios y problemas de planificación generados, que en realidad es nuestro modelo, se llama a los dos planificadores SGPLAN y LPG para dar una solución. Para simplificar, a los dominios y problemas que conforma nuestro modelo les llamaremos simplemente problemas.

A manera de no confundir cuando un problema no es resuelto por algún planificador, una vez obtenidos los resultados, a estos problemas se les asignó un tiempo de “resolución” mayor al tiempo más alto para encontrar solución. En las gráficas a estos problemas no resueltos se les representa con un punto negro.

4.3.1 MODELOS CON REQUERIMIENTOS

En la figura 4.1 se muestran 6 gráficas con los resultados de la clase pequeña con 1, 2, 3, 4 y 5 materias, todas con requerimientos. En la gráfica de una 1 materia se observa el tiempo en que tardó cada planificador en resolver los problemas. Cómo los tiempos en encontrar una solución por SGPLAN son pequeños, en la gráfica se pueden observar apenas encima del tiempo 0. De manera que se observa que SGPLAN tuvo un mejor rendimiento que LPG para encontrar solución. Los puntos negros que están en el tiempo 400 son aquellos problemas no resueltos, en su mayoría por LPG. Algo similar ocurre con 2, 3, 4 y 5 materias. Con la diferencia que los tiempos en 3, 4 y 5 materias se elevan y la cantidad de problemas no resueltos por ambos planificadores aumenta. En la última gráfica de este conjuntos se muestra el porcentaje de problemas resueltos por cada planificador clasificándolos por la cantidad de materias modeladas. Se observa que a medida que aumenta la cantidad de materias el porcentaje de problemas resueltos por ambos planificadores disminuye. Aún así, SGPLAN tiene un mayor porcentaje por encima de LPG de problemas resueltos para esta clase pequeña con requerimientos.

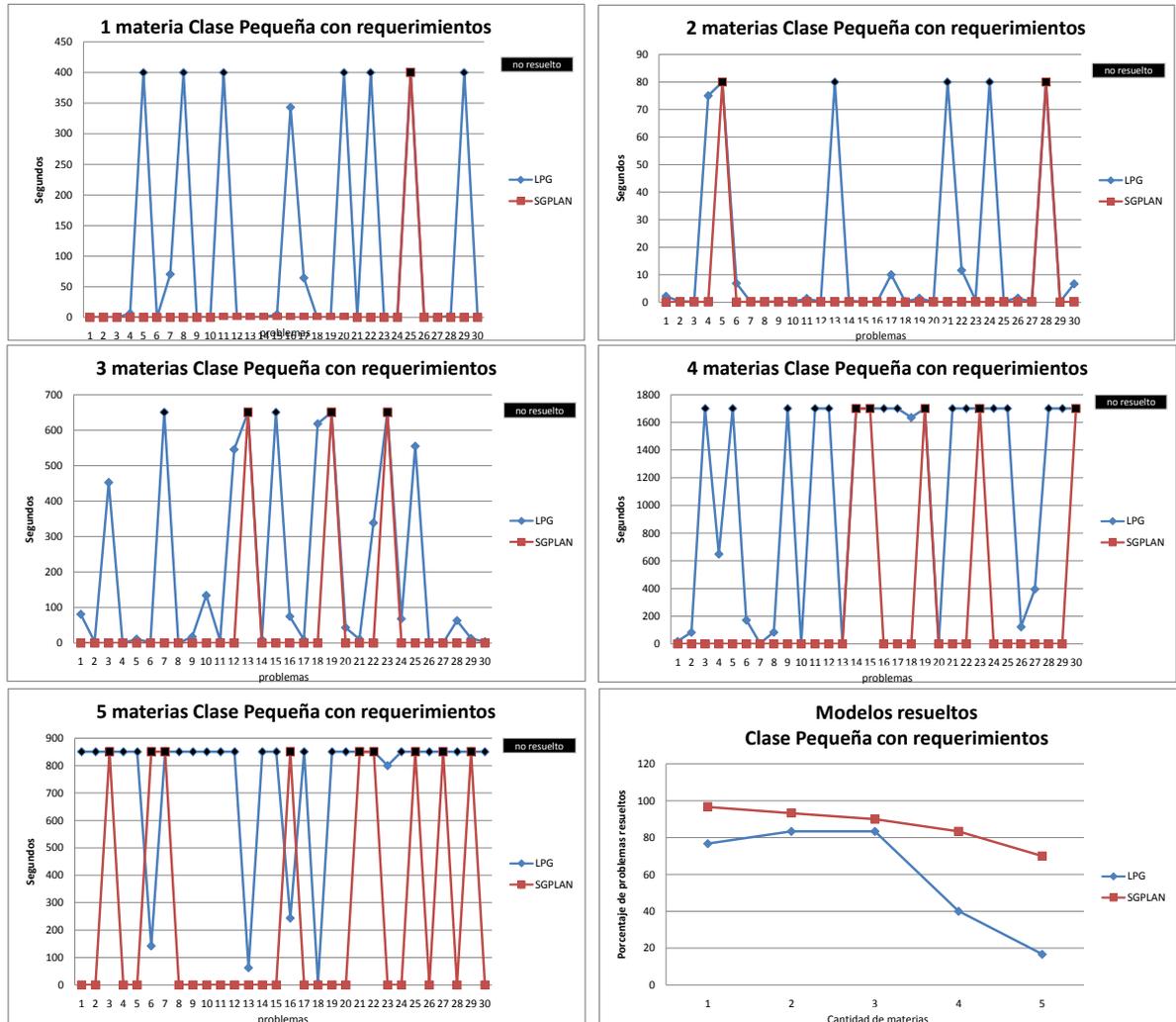


Figura 4.1: Comparación de resultados con los planificadores SGPLAN y LPG en la clase pequeña con requerimientos.

Los resultados para la clase mediana con requerimientos se muestran en la figura 4.2. En la gráfica de 1 materia se observa que el tiempo en encontrar una solución a los problemas es pequeño en ambos planificadores. En la gráfica de 2 y 3 materias el tiempo se dispara y se observa que la cantidad de problemas no resueltos por ambos planificadores aumenta en relación con la gráfica de 1 materia. Para las gráficas de 4 y 5 materias aumenta drásticamente la cantidad de problemas no resueltos por ambos planificadores. Aunque no puede ser apreciado en estas gráficas el planificador LPG está “debajo” de la marca de SGPLAN. Se observa que a medida que aumenta

la cantidad de materias ambos planificadores muestran problemas al tratar de resolverlos. En la última gráfica se muestra el concentrado de los problemas resueltos por ambos planificadores para esta clase mediana con requerimientos en donde se observa que para esta clase el porcentaje de problemas resueltos disminuyó abajo de 10 por ciento para ambos planificadores.

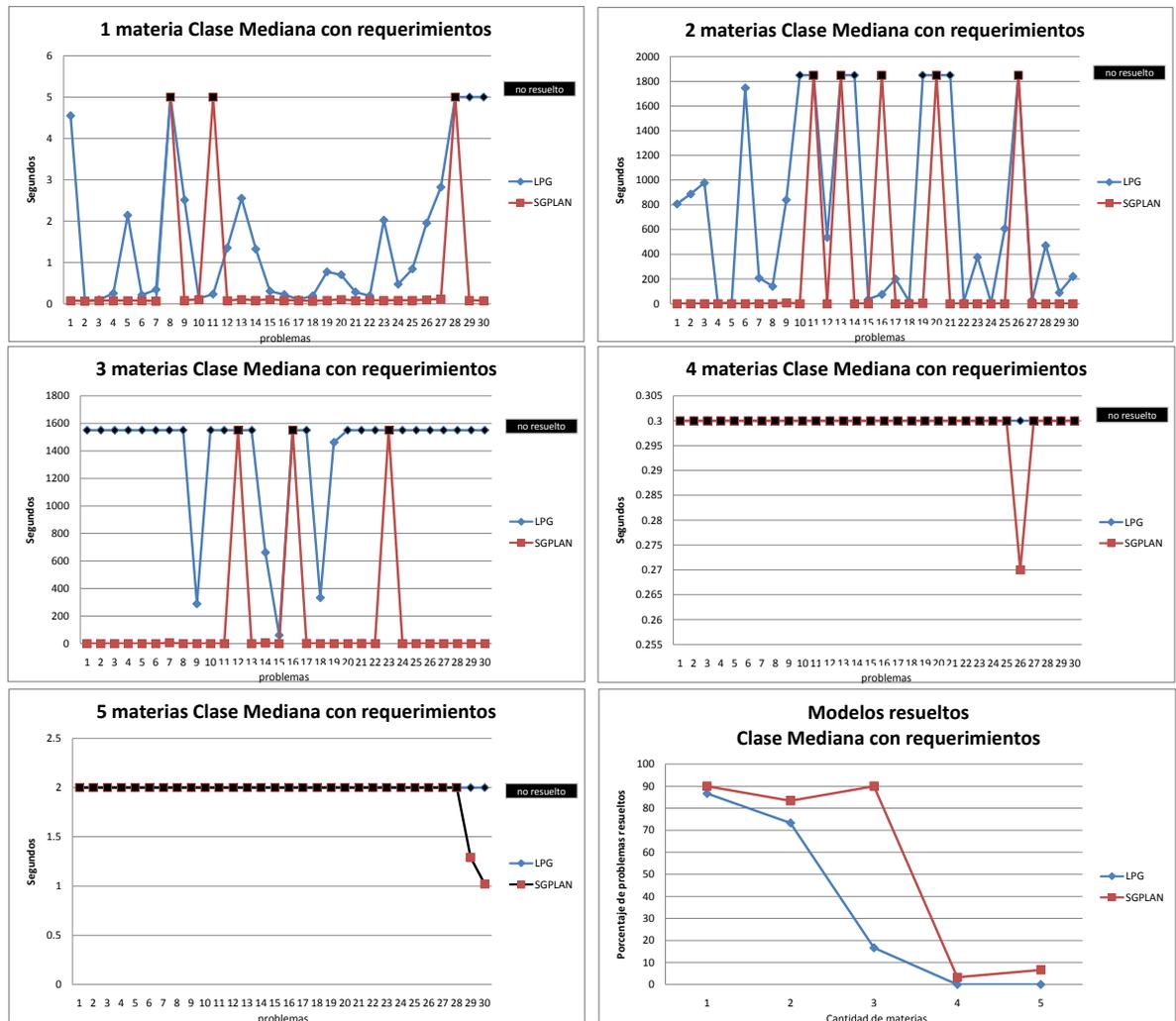


Figura 4.2: Comparación de resultados con los planificadores SGPLAN y LPG en la clase mediana con requerimientos.

Para la clase grande con requerimientos los resultados se muestran en las gráficas de la figura 4.3. En la gráfica de 1 materia se SGPLAN puede resolver la mayoría de los problemas en un tiempo pequeño, en comparación con LPG. En la gráfica de

2 materias se observa que tiempo que tardan los planificadores en encontrar una solución aumenta al igual que el número de problemas no resueltos. En la gráfica de 3 materias se observa algo similar, aunque los tiempos son menores y la cantidad de problemas resueltos por LPG es casi nula. En la gráfica de 4 y 5 materias ambos planificadores no encuentran solución. En la última gráfica se muestra el porcentaje de problemas resueltos por ambos planificadores, en donde LPG a partir de 2 materias no pudo encontrar soluciones. SGPLAN se mantuvo con un buen porcentaje de problemas resueltos solo en las primeras 3 materias, ya que a partir de 4 y 5 materias no fue capaz de encontrar solución.

4.3.2 MODELOS SIN REQUERIMIENTOS

Para generar los modelos sin requerimientos, de manera que pudieran ser comparados, se tomaron los modelos generados con requerimientos y se modificaron los predicados para establecer que ninguna actividad de aprendizaje tenía requerimientos.

Los resultados de la clase pequeña con 1, 2, 3, 4 y 5 materias, todas sin requerimientos se muestran en la figura 4.4. En la gráfica de 1 materia se observa que SGPLAN mantiene tiempos bajos en encontrar soluciones en comparación con LPG, esto mismo se observa en la de 2 materias, con la diferencia que el tiempo para encontrar solución por parte de LPG disminuye. A partir de 3, 4 y 5 materias se observa un incremento en la cantidad de problemas no resueltos, pero SGPLAN se mantiene rápido en encontrar soluciones. En la última gráfica se muestra el porcentaje de problemas resueltos por cada planificador clasificándolos por la cantidad de materias modeladas. Se observa que a medida que aumenta la cantidad de materias el porcentaje de problemas resueltos por ambos planificadores disminuye. Aún así, SGPLAN mantiene un porcentaje alto de problemas resueltos. LPG por su parte, a partir de 4 y 5 materias el porcentaje de problemas resueltos oscila entre el 18 y 40 %.

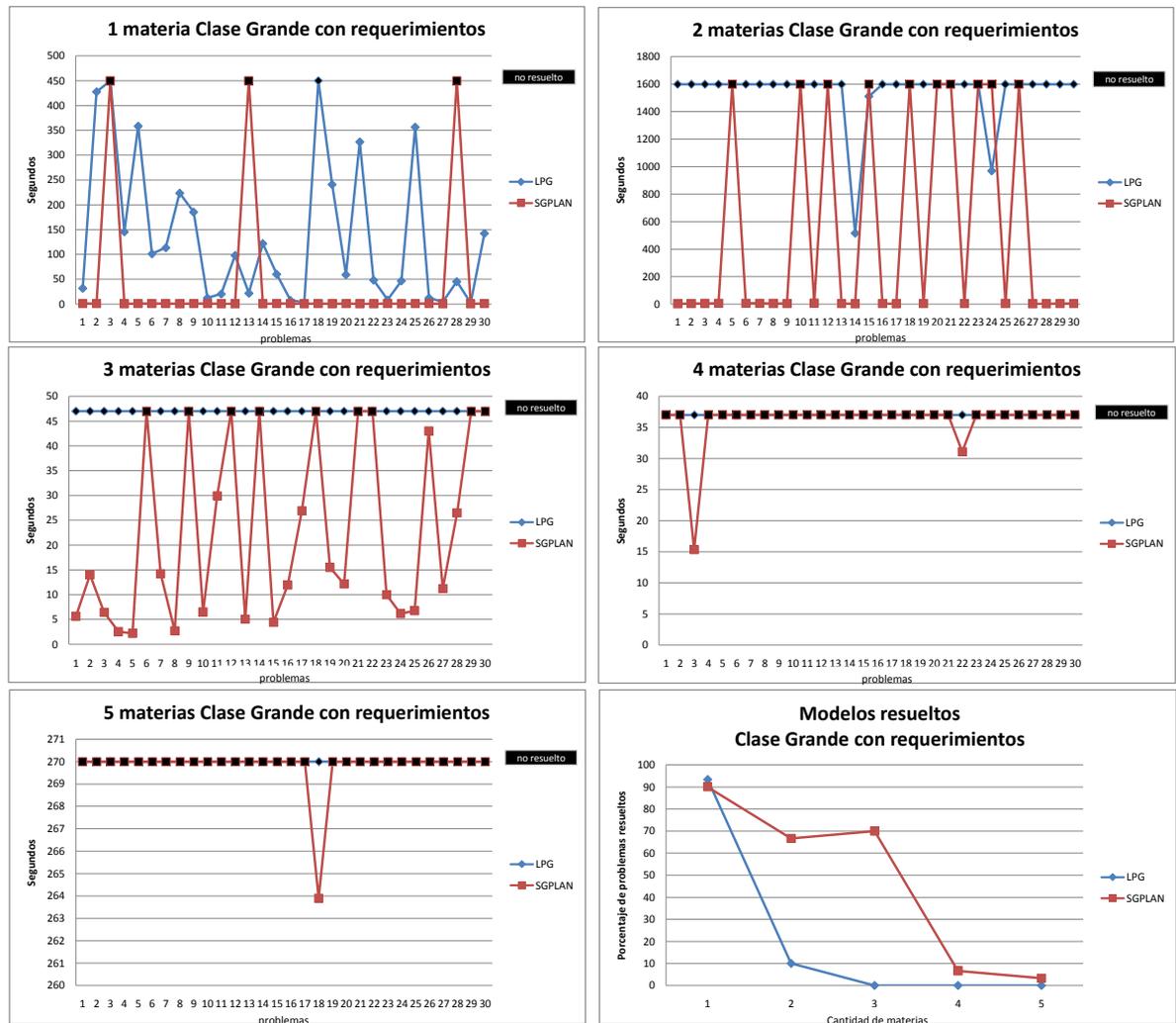


Figura 4.3: Comparación de resultados con los planificadores SGPLAN y LPG en la clase grande con requerimientos.

Los resultados para la clase mediana sin requerimientos se muestran en la figura 4.5. Para 1 materia se observa que el tiempo es pequeño para ambos planificadores, sobresaliendo SGPLAN. A partir de 2, 3 y 4 materias los tiempos en encontrar solución aumentan para LPG, así como la cantidad de problemas no resueltos; SGPLAN se mantiene con tiempos bajos y aumenta ligeramente la cantidad de problemas no resueltos. Para 5 materias LPG no pudo encontrar ningunas solución, SGPLAN aumenta la cantidad de problemas no resueltos en comparación con 1,2,3,y materias anteriores. En la última gráfica de este concentrado, se muestran el porcentaje de

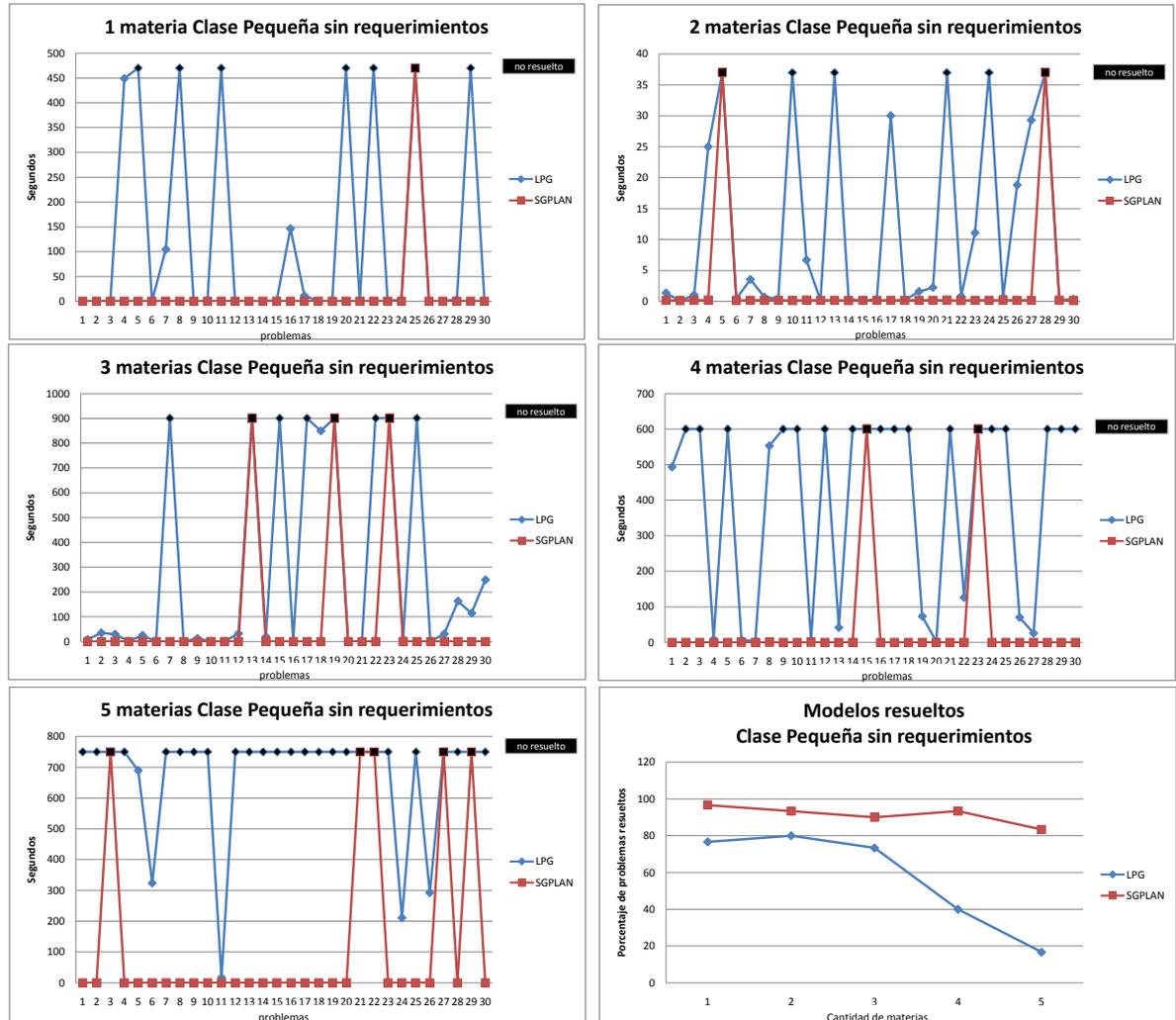


Figura 4.4: Comparación de resultados con los planificadores SGPLAN y LPG en la clase pequeña sin requerimientos.

problemas resueltos por ambos planificadores para esta clase mediana sin requerimientos. Se observa que SGPLAN mantiene un alto porcentaje de problemas resueltos, aunque decreció un poco para 5 materias. LPG por su parte, disminuyó arriba del 50% de problemas resueltos al pasar de 2 a 3 materias. Resolviendo menos del 5% de problemas de 4 y 5 materias.

Para la clase grande sin requerimientos los resultados se muestran en la figura 4.6. Para 1 materia SGPLAN mantiene tiempos bajos en encontrar solución y LPG no pasó los 200 segundos en encontrar soluciones. Sin embargo, ambos planifi-

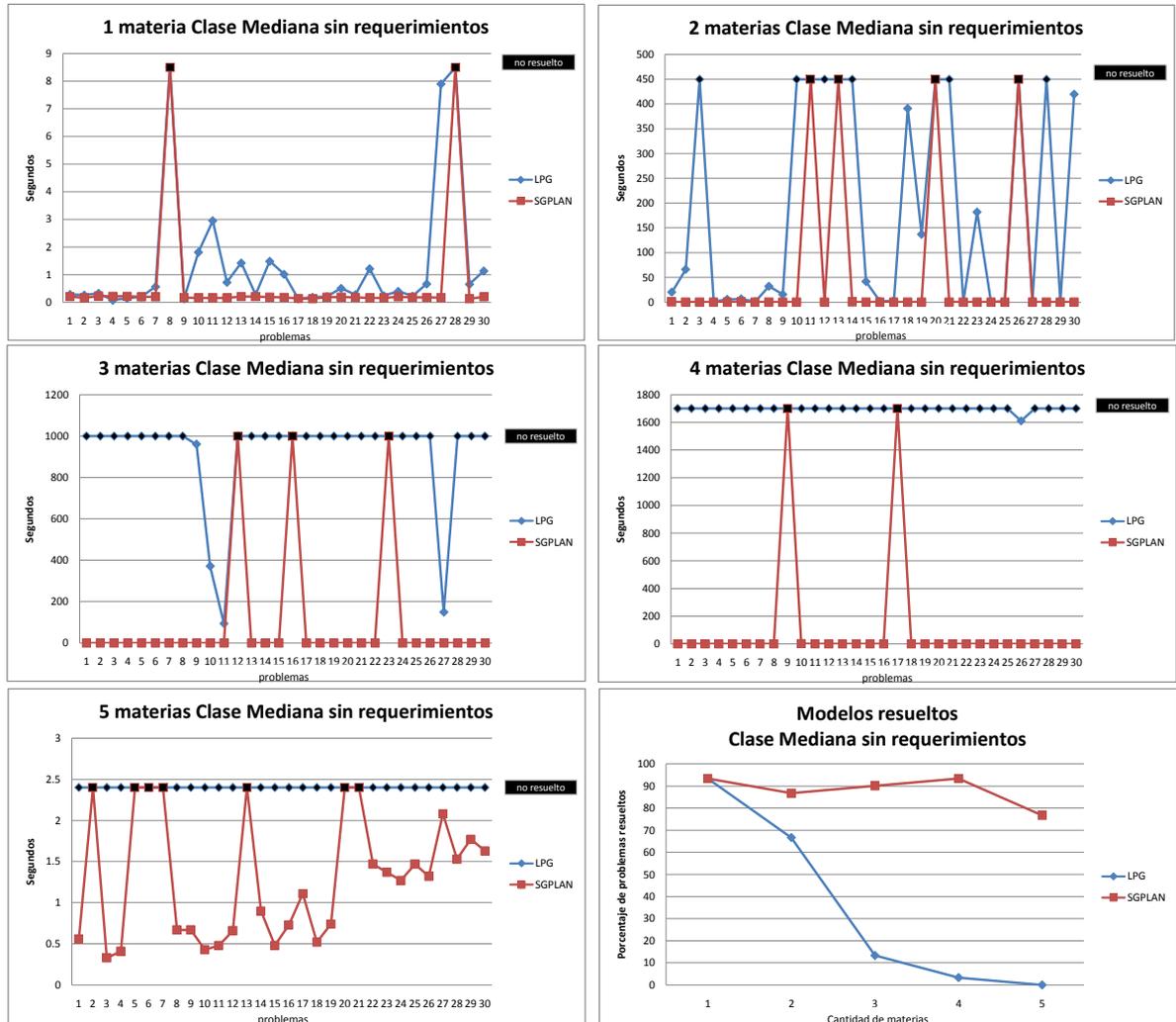


Figura 4.5: Comparación de resultados con los planificadores SGPLAN y LPG en la clase mediana sin requerimientos.

cadore en dos problemas tuvieron un tiempo de solución alto. A partir de 2 materias la cantidad de problemas no resueltos por LPG aumenta. Para 2 y 3 materias SGPLAN mantiene un tiempo bajo en encontrar soluciones, sin embargo, aumenta la cantidad de problemas no resueltos. Para 5 materias, en comparación con 4 materias, SGPLAN aunque los tiempos de solución aumentan, disminuye la cantidad de problemas no resueltos. En la última gráfica se muestra el porcentaje de problemas resueltos por ambos planificadores, en donde LPG sufre una disminución del 80 % de problemas no resueltos al pasar de 1 a 2 materias, y no pudiendo resolver a partir

de 3 materias. SGPLAN se mantuvo arriba del 80 % de problemas resueltos en las primeras 3 materias, teniendo una disminución de poco más del 20 % al pasar a 4 materias. Se observa además, que SGPLAN en 5 materias aumentó la cantidad de problemas resueltos en comparación con 4 materias. Para esta situación en particular no se identifica si el aumento es con respecto a alguna característica de los modelos generados o del planificador, ya que a medida que aumenta el tamaño de materias a modelar va disminuyendo la cantidad de problemas resueltos, pero en particular la de 4 materias tiene una disminución mayor.

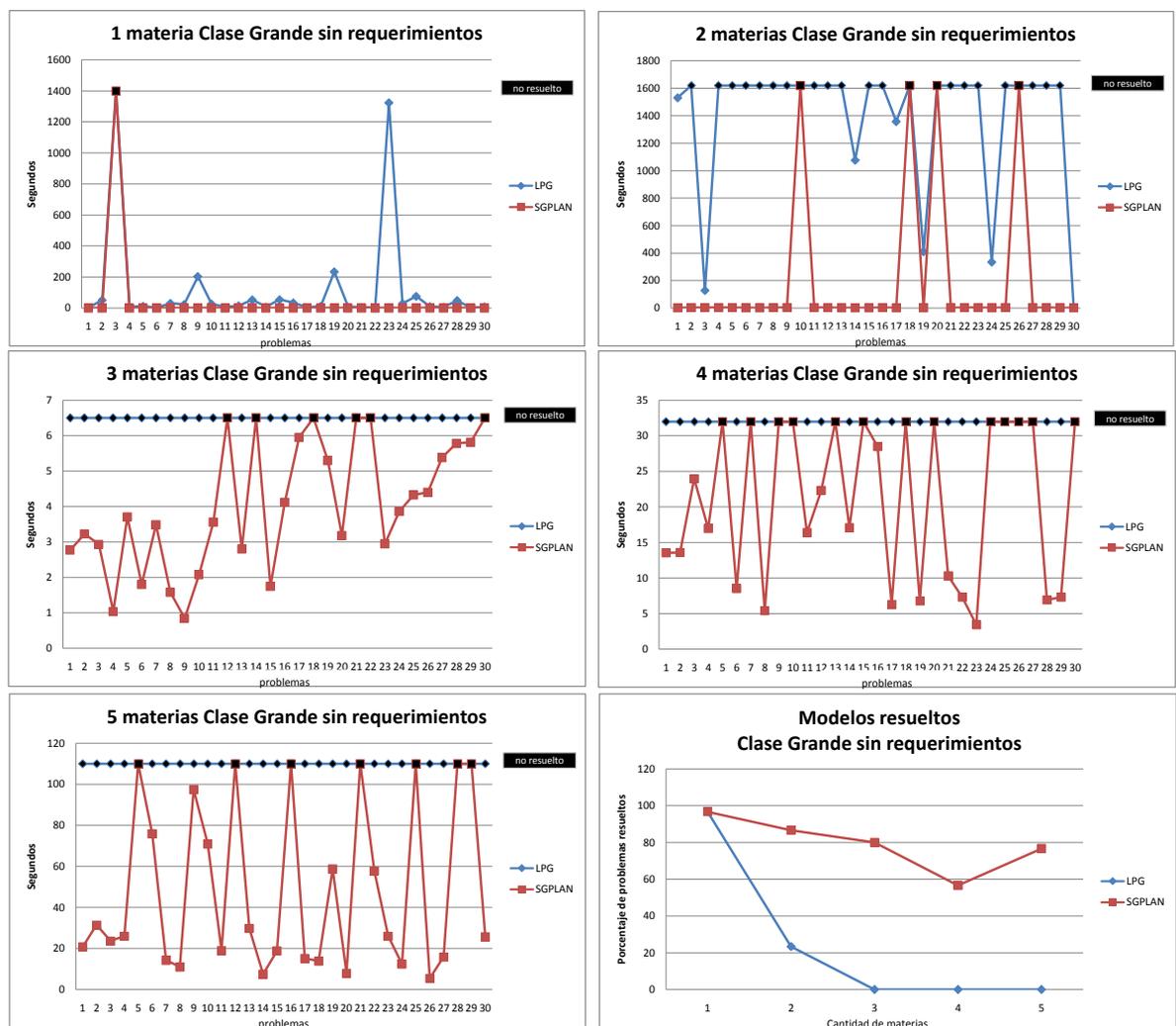


Figura 4.6: Comparación de resultados con los planificadores SGPLAN y LPG en la clase grande sin requerimientos.

CONCENTRADO DE PROBLEMAS RESUELTOS POR AMBOS PLANIFICADORES

Como hemos venido observando a lo largo de las pruebas realizadas, a medida que aumenta la cantidad de materias a modelar (escalabilidad), los planificadores parecen no encontrar solución. En la figura 4.7 se muestra los resultados generales de ambos tipos de modelos, sus clases y la cantidad de materias modeladas.

CLASE PEQUEÑA

En la clase pequeña ya sea con o sin requerimientos se observa una disminución del porcentaje de problemas resueltos por ambos planificadores. En 1, 2 y 3 materias ya sea con o sin requerimientos SGPLAN se mantuvo constante en el porcentaje de problemas resueltos, que fueron arriba del 90%. Sin embargo para 4 y 5 materias si existe una diferencia significativa para SGPLAN cuando son con requerimientos y sin ellos ya que aumentó alrededor de 10% de problemas resueltos sin requerimientos en comparación con los modelos con requerimientos.

LPG para 1 y 2 materias con y sin requerimientos se mantuvo arriba del 80% de problemas resueltos. Para 3 materias disminuyó un 10% de problemas resueltos sin requerimientos en comparación de con requerimientos. Al pasar de 3 a 4 materias hubo una disminución del 43% de problemas resueltos en ambos tipos de modelos (con o sin requerimientos) y una disminución de 23% al pasar de 4 a 5 materias. Con respecto a esto, se observa que para 4 y 5 materias LPG se mantuvo constante con y sin requerimientos. Para esta clase se observa que no afecta a LPG si los modelos tienen o no requerimientos.

CLASE MEDIANA

Para la clase mediana con o sin requerimientos SGPLAN presenta para 1,2 y 3 materias modeladas arriba del 80% de problemas resueltos. Sin embargo en 4 y 5 materias con requerimientos el porcentaje de problemas resueltos disminuye a menos de 10% de problemas resueltos. No así para esas mismas materias sin requerimientos, ya que presentan un aumento significativo al pasar de modelos con

Clase	Materias	Con requerimientos		Sin requerimientos	
		%LPG	%SGPLAN	%LPG	%SGPLAN
Pequeña	1	77	97	77	97
	2	83	93	80	93
	3	83	90	73	90
	4	40	83	40	93
	5	17	70	17	83
Mediana	1	87	90	93	93
	2	73	83	67	87
	3	17	90	13	90
	4	0	3	3	93
	5	0	7	0	77
Grande	1	93	90	97	97
	2	10	67	23	87
	3	0	70	0	80
	4	0	7	0	57
	5	0	3	0	77

Figura 4.7: Concentrado de problemas resueltos por ambos planificadores. Modelando con o sin requerimientos

requerimientos a sin ellos. Teniendo un desempeño de 77% en 5 materias sin requerimientos en comparación con el 7% de 5 materias con requerimientos. Y un 93% de problemas resueltos para 4 materias sin requerimientos contra 3% de problemas con requerimientos.

LPG por otro lado parece no afectarle que los modelos sean con o sin requerimientos, ya que solo en 1 materia con requerimientos aumento un 6% de problemas resueltos en sin requerimientos. Y al contrario en 2 materias, disminuyo 6% al pasar a sin requerimientos y un 4% en 3 materias. En esta clase mediana se observa que al tener 3 materias modeladas para ambos tipos de modelos (con o sin restricciones) ya que disminuye un 50% de los problemas resueltos en comparación con 2 materias. En 5 materias para esta clase mediana el porcentaje de problemas resueltos de los dos tipos, es casi nulo para este planificador.

CLASE GRANDE

En la clase grande para 1 materia con o sin requerimientos el desempeño de ambos planificadores es superior al 90%. Para el caso de SGPLAN a pesar de que

disminuyó un 23 % al pasar de 1 a 2 materias con requerimientos, al pasar de 2 a 3 materias con requerimientos aumentó un 3 % de problemas resueltos. Ahora, si se compara con los modelos sin requerimientos, para 2 materias aumenta el porcentaje en un 20 % de problemas resueltos, y un 10 % en 3 materias (sin requerimientos). Para 4 y 5 materias SGPLAN resuelve menos del 10 % con requerimientos. Se observa en estas materias que sin requerimientos aumenta el porcentaje de problemas resueltos a 57 % en 4 materias y 77 % en 5 materias. Se concluye que para esta clase grande en SGPLAN si es significativo el aumento de problemas resueltos cuando es sin requerimientos.

Para el caso de LPG en la clase grande solo resuelve arriba del 90 % de problemas resueltos cuando es 1 materia, ya que cuando son 2 materias el porcentaje de problemas resueltos disminuye un 80 % en el caso de modelos con requerimientos y 74 % en modelos sin requerimientos. Se observa que ambos tipos de modelos (con o sin requerimientos) a partir de 3 materias no puede resolver ninguno.

Para concluir con este capítulo mencionamos que el hecho de no utilizar requerimientos en los modelos realmente influye en el desempeño de SGPLAN, esto tiene sentido, si asumimos que un problema sin requerimientos podría en teoría ser más fácil de resolver. Así que para SGPLAN la interconectividad (es decir modelos con requerimientos) es un factor crítico para la resolución de los problemas.

En el caso de LPG el factor que domina inmediatamente es el tamaño del problema (la cantidad de materias modeladas) ya que a medida que la cantidad de éstas aumenta no encuentra solución.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones obtenidas de la experimentación con los modelos generados en PDDL, compuestos por los archivos de dominio y problema. Así como si fueron o no resueltos por medio de los planificadores SGPLAN y LPG. Después se describirán las recomendaciones que se tienen de esta investigación así como el trabajo futuro.

5.1 CONCLUSIONES

En este trabajo se definió el problema de generación de planes educativos para estudiantes tomando en cuenta restricciones de tiempo en las actividades de aprendizaje y asignación de recursos. Se ha diseñado un modelo de planificación en PDDL que captura las propiedades de tiempo en la duración de las actividades, recursos asociados a los LO, estudiantes, la acumulación de score en las actividades de aprendizaje seleccionadas (funciones de acumulación de calidad), las restricciones entre materias, temas, subtemas y actividades de aprendizaje con respecto a sus relaciones jerárquicas y de orden.

Es importante remarcar que todas estas características en su conjunto no han sido modeladas en los trabajos encontrados. Además de esto, los modelos son flexibles, puesto que algunas de las funciones de acumulación de calidad o los valores que se usan en ellas, son completamente modificables de acuerdo a la necesidad del usuario.

Estos modelos para efectos de la experimentación, fueron creados de una manera aleatoria, a través de un generador codificado en C, el cual toma de un archivo de entrada, la cantidad de materias a modelar y la clase a la cual pertenece, que puede ser pequeña, mediana o grande. Esto para los modelos con requerimientos. Para aquellos modelos que no tienen requerimientos, se modificaron los modelos con requerimientos generados por el programa, haciendo que todas las actividades de aprendizaje no tengan requerimientos.

Se corrieron estos modelos para su solución con los planificadores SGPLAN y LPG utilizando un script para cada clase variando la cantidad de materias modeladas y diferenciando entre si tienen o no requerimientos.

De los resultados obtenidos en el capítulo anterior se concluye que la cantidad de materias modeladas, es decir la escalabilidad, es significativa para que dichos planificadores puedan encontrar una solución. Ya que a medida que aumenta la cantidad de materias a modelar disminuye el porcentaje de problemas resueltos por ambos planificadores, es decir, parecen no encontrar solución. Esta disminución de problemas resueltos es más perceptible en el caso del planificador LPG, ya que en la clase pequeña resolvió eficientemente las primeras 3 materias, en la clase mediana solo las primeras dos y en la clase grande solo 1 materia. Esto en ambos tipos de modelos con o sin requerimientos.

Además de esto, se observa, en el planificador SGPLAN que si los modelos no tienen requerimientos en las actividades de aprendizaje es más fácil obtener solución a ellos. Esto es comprensible, ya que estaría manejando menos restricciones.

En resumen: El factor que impacta en el desempeño de la resolución de problemas para el planificador SGPLAN es la interconectividad, aún por encima de la escalabilidad. Para el planificador LPG el factor que domina inmediatamente es el tamaño del problema, es decir, la escalabilidad, ya que a medida que la cantidad de materias aumenta independientemente de que el modelo tenga o no requerimientos, no encuentra solución.

Como conclusión decimos que nuestro modelo está desarrollado apropiadamente, considerando las necesidades que se modelaron, como duración en las actividades, dependencias de precedencia, métricas acumuladas en varias acciones, etc. Sin embargo, los planificadores presentan ciertas limitantes para darles solución de una manera eficiente en las clases grandes con requerimientos, que son las que podríamos decir, se acercan más a la realidad, ya que no tenemos una materia con un solo tema o tres, como es el caso de las clases pequeñas, las cuales fueron resueltas por ambos planificadores y de una manera rápida.

5.2 RECOMENDACIONES

El modelo propuesto contempla un estilo de evaluación en base a 100. Aún así, el modelo puede ser flexible con respecto a la forma de evaluar puesto que algunas de las funciones de acumulación de calidad o los valores que se usan en ellas, son completamente modificables.

Nosotros realizamos un programa en C que genera los modelos automáticamente, variando únicamente algunos datos de entrada, tales como las clases pequeña, mediana o grande y la cantidad de materias. Sería conveniente para el estudiante y asimismo para el responsable de la materia, tener una herramienta que facilite la visualización tanto de los datos de entrada como los planes arrojados como solución dada de los planificadores. Es decir, crear una herramienta de modelado que permita al instructor crear los cursos de la materia a través de componentes gráficos y formularios de entrada que sean fáciles de utilizar.

Esta herramienta deberá servir como intermediario entre los estudiantes y responsables de la materia, sin la necesidad de que alguno de ellos conozca la modelación en PDDL, sino que los datos sean interpretados, se genere el modelo y sean corridos en los planificadores para que den la solución a estos modelos. Mostrando los resultados de una manera gráfica.

Otras características que no contemplamos en nuestra modelación y que sería interesante analizar son: la sincronización o colaboración de una acción entre varios estudiantes. Adaptación de los diseños de aprendizaje, es decir, los planes arrojados como resultado del proceso de planificación, como resultado de evaluaciones intermedias durante el desarrollo de un curso. Utilización de un tiempo límite para el estudiante al realizar sus actividades de aprendizaje, ya que solo estamos modelando si tiene o no disponibilidad. Analizar si es conveniente incluir el desempeño del estudiante, de manera que algunas actividades de aprendizaje estén restringidas por este desempeño, considerando la “dificultad” que pueda tener ese LA de manera que esto afecte su selección. Analizar la conveniencia de incluir estilos de aprendizaje vinculados a las actividades de aprendizaje. Establecer un límite de tiempo en los planes, ya que por lo general los objetivos de las materias deben cubrirse en un determinado período de tiempo, como por ejemplo, un semestre. Incluir ventanas de tiempo a los modelos, para que además de que se tenga la secuencia de acciones a realizar, se especifique el día u hora.

5.3 TRABAJO FUTURO

A pesar de los avances logrados, hay características que deben contemplarse para enriquecer aún más nuestro modelo, pero antes de poder pensar en extenderlo, se tendría que considerar si existen planificadores que puedan dar solución a dichos modelos.

Dados los resultados obtenidos con respecto a los dos planificadores seleccionados, los cuales presentan problemas en la resolución de los modelos generados, sería adecuado investigar los algoritmos de planificación de manera que se puedan ampliar al manejo de más información, manejando métricas y operaciones matemáticas más eficaces. Se propone extender los algoritmos de planificación actuales de manera que sean capaces de dar solución a un modelo más complejo.

Los factores a analizar en estos algoritmos de planificación serían la escalabil-

idad, es decir que puedan soportar mayor cantidad de materias, al menos las que se consideran en promedio para un semestre y la interconectividad, haciendo referencia a los requerimientos de las actividades de aprendizaje.

En el caso de la escalabilidad el planificador LPG presentó problemas, ya que a medida que crecía la cantidad de materias modeladas no pudo encontrar solución, sin embargo, parece no haberle afectado la interconectividad entre las actividades de aprendizaje. Lo anterior necesita una investigación minuciosa en el algoritmo de planificación utilizado en este planificador, sin embargo, creemos que la interconectividad no le afectó ya que LPG considera a todas sus acciones como excluyentes. En el caso de SGPLAN el factor crítico fué la interconectividad, como SGPLAN divide los problemas en subproblemas para resolver sub-objetivos, creemos que pudiera presentarse más problema al momento de unir estos subproblemas y darse cuenta que existe relación entre ellos. Aún así, esto tiene que ser analizado a detalle.

Viendo la importancia que tiene el tratar con los algoritmos de planificación para poder resolver de una manera eficiente este modelo propuesto y posteriormente poderlo extender a un modelo que contemple más propiedades, se considera el trabajo de análisis de los algoritmos de planificación como un trabajo futuro de esta tesis.

Además de lo anterior, se considera como trabajo futuro analizar la independencia entre las materias, es decir, aunque se esté modelando más y más materias, cada materia por lo regular es independiente de la otra, a menos que tenga algún requerimiento a nivel materia, pero en sí los temas, subtemas y actividades de aprendizaje, así como sus requerimientos son entre objetos de la misma materia. Se analizaría además el impacto que tendría el resolver cada materia de una forma independiente y después unir los planes para todas las materias.

Con respecto a la extensión del modelo se puede contemplar un trabajo alternativo con áreas de pedagogía y psicología para adaptar los objetos de estudio a los diferentes perfiles de aprendizaje para intentar garantizar que el alumno tenga acceso a materiales de estudio apropiados a su personalidad y estilo.

Otra posible extensión es la inclusión de ventanas de tiempo en nuestro modelo, ya que no solo se estaría dando una secuencia de actividades a los estudiantes, sino también se le indicaría el tiempo en que lo tiene que realizar.

Además de lo anterior, sería conveniente la modificación del modelo para que contemple los tiempos de prescripción, es decir, que los tiempos vayan sincronizados con el inicio y fin de un período de tiempo, como por ejemplo, un semestre, para que el alumno tenga la oportunidad de acreditar en el tiempo que marca la institución.

APÉNDICE A

GLOSARIO

A.1 TÉRMINOS EDUCATIVOS

Asesoría.- Consulta que brinda un profesor especializado en una disciplina (asesor) fuera de sus horas de docencia para resolver dudas o preguntas sobre temas específicos que domina. Además pueden ser incluidas la dirección de tesis u otras actividades académicas como prácticas profesionales y servicio social. En la modalidad abierta o a distancia, es el servicio en la que un estudiante distante recibe orientación por parte de un experto en la materia o contenido en relación a: estrategias de estudio, realización de trabajos, contenidos, problemas, o dificultades en las experiencias de aprendizaje.

Curso.- Unidad educativa en la que se ofrece un conjunto estructurado de conocimientos teóricos y/o prácticos. Para acreditarla se debe alcanzar un nivel aprobatorio en una evaluación. v. asignatura

Curso en línea.- Evento académico utilizando la Internet o por medio del web

Currículo (latín: curriculum).- Conjunto de asignaturas, actividades, experiencias de aprendizaje y métodos de enseñanza y otros medios para alcanzar los objetivos del programa educativo. v. Plan de estudios.

Currículo (académico).- El término currículo se refiere al conjunto de ob-

jetivos, contenidos, criterios metodológicos y técnicas de evaluación que orientan la actividad académica (enseñanza y aprendizaje) ¿cómo enseñar?, ¿cuándo enseñar? y ¿qué, cómo y cuándo evaluar? El currículo permite planificar las actividades académicas de forma general, ya que lo específico viene determinado por los planes y programas de estudio (que no son lo mismo que el currículo). Mediante la construcción curricular la institución plasma su concepción de educación. De esta manera, el currículo permite la previsión de las cosas que hemos de hacer para posibilitar la formación de los educandos

Para la estructuración del currículo (que es diferente en niveles básico, medio y superior), las autoridades académicas, planificadores escolares, docentes y demás involucrados (pudiendo ser alumnos, egresados, empleadores, etc.) deben tomar en cuenta lo siguiente: Lo que se debe enseñar y lo que los alumnos deben aprender; Lo que se debe enseñar y aprender y lo que realmente se enseña y aprende; es decir, lo ideal y lo real; Encontrar solución a estos pequeños malentendidos que se crean debido a que no somos capaces de ver más allá de lo que nuestros ojos nos enseñan.

Diseñador instruccional. Experto en estrategias de aprendizaje; con visión amplia e integral, que selecciona los medios, materiales y orienta el acto educativo en eventos a distancia.

Diseño curricular.- Planeación de la estructura que tendrá el plan de estudios atendiendo a las necesidades del estudiante para una formación integral y al desarrollo del campo disciplinar.

Educación Superior.- Niveles académicos posteriores al nivel 4 de la clasificación internacional de UNESCO, comprende los niveles 5 (licenciatura o pregrado) y 6 (especialidad, maestría o doctorado, o sea el posgrado).

Licenciatura (v. carrera).- Primer grado académico de la educación superior cuyo antecedente obligatorio es el bachillerato o equivalente, dirigido a formar actitudes, aptitudes, habilidades, métodos de trabajo y conocimientos relativos al ejercicio de una profesión.

Modelo educativo.- Forma en que la institución educativa organiza las actividades y funciones que sustentan el proceso de Enseñanza - Aprendizaje.

Plan de estudios.- Documento que contiene el planteamiento de las acciones específicas de aprendizaje contempladas en un programa educativo, consta de justificación, lista de signaturas y otras acciones específicas de aprendizaje, perfil de egreso, perfil de ingreso e integración vertical y horizontal de las asignaturas.- Conjunto ordenado de asignaturas, prácticas, estudios y otras actividades de enseñanza y aprendizaje que determinan el contenido de un programa educativo y que se deben cumplir para obtener, en un centro de educación superior, el título o grado correspondiente.

Trayectoria escolar.- Desempeño de un estudiante o conjunto de estudiantes (cohorte), durante su estancia en una institución de educación hasta su egreso.

Tutoría.- Forma de relación pedagógica que se establece mediante diversas actividades entre un profesor (docente-tutor) y el estudiante que se le asigne (tutorado). Es un proceso dinámico institucional de acciones sistemáticas que complementa a la docencia, ofrece atención en forma personalizada a los estudiantes para contribuir a su formación integral y mejorar la calidad educativa.

A.2 GLOSARIO DE ACRÓNIMOS

TIC.- Tecnologías de la Información y la Comunicación.

IEEE.- Institute of Electrical and Electronics Engineers.

LOM.- Learning Object Metadata. Metadatos de objetos de aprendizaje.

ADL.- Advanced Distributed Learning.

SCORM.- Shareable Content Object Reference Model. Modelo de referencia para contenidos compatibles.

LMS.- Learning Management System. Sistemas Gestores de aprendizaje.

AICC.- Aviation Industry CBT [Computer-Based Training] Committee.

IMS.- Instructional Management System Project.

ARIADNE.- Alliance of Remote Instruction Authoring and Distribution Networks for Europe.

LO.- Learning Object. Objeto de aprendizaje

APÉNDICE B

PDDL

B.1 HISTORIA DEL PDDL

PDDL está basado en el lenguaje LISP (acrónimo de LIStProcessing), el cual es el segundo lenguaje de programación de alto nivel más viejo de extenso uso hoy en día.

El elemento fundamental en Lisp es la lista, en el sentido más amplio del término, pues tanto los datos como los programas son listas (de ahí viene su nombre). Las listas en LISP están delimitadas por paréntesis (como en PDDL).

Lisp sigue una filosofía de tratamiento no-destructivo de los parámetros, de modo que la mayoría de las funciones devuelven una lista resultado de efectuar alguna transformación sobre la que recibieron, pero sin alterar esta última.

Uno de los motivos por los que Lisp es especialmente adecuado para la IA es el hecho de que el código y los datos tengan el mismo tratamiento (como listas); esto hace especialmente sencillo escribir programas capaces de escribir otros programas según las circunstancias.

B.2 STRIPS

El lenguaje STRIPS es el lenguaje básico en el que está desarrollado PDDL. Se le nombró así por el programa de planificación pionero conocido como STanford Research Institute Problem Solver (solucionador de problemas del Instituto de Investigación de Stanford). El nombre STRIPS plantea dos problemas. Primero, la organización ya no utiliza el nombre “Stanford”, sino que actualmente se le conoce como SRI International. Segundo, el programa es lo que actualmente se denomina planificador, no un solucionador de problemas; pero en 1970, cuando se le diseñó, todavía no se establecía la diferencia. Si bien el planificador STRIPS fue reemplazado desde ya hace mucho tiempo, el lenguaje STRIPS empleado para especificar acciones resulta invaluable y son muchas las variantes tipo “STRIPS” que se han diseñado.

APÉNDICE C

MODELO EN PDDL

C.1 MODELO DE 1 MATERIA DE CLASE PEQUEÑA CON REQUERIMIENTOS

C.1.1 DOMINIO

(define (domain degree)

(:requirements :durative-actions :typing :fluents :equality)

(:types student resource - object

subject Theme subtheme LA - LO)

(:constants

Material1 - subject

Tema1

Tema2

Tema3 - Theme

Subtema1

Subtema2

Subtema3

Subtema4

Subtema5

Subtema6

Subtema7

Subtema8

Subtema9 - subtheme

LA1

LA2

LA3

LA4

LA5

LA6

LA7

LA8

LA9

LA10

LA11

LA12

LA13

LA14

LA15

LA16

LA17

LA18 - LA

rec0

rec1

rec2

rec3 - resource

)

(:predicates

(available-subject ?subj - subject ?s - student)

(free ?s - student)

(pass-degree ?subj - subject ?s - student)

(enrollment ?s - student ?subj - subject)

(done-Theme ?t - Theme ?subj -subject ?s - student)

(done-subject-LA ?subj - subject ?s - student)

(not-done-LA ?oa - LA ?subj - subject ?s - student)

(not-approved ?subj - subject ?s - student)

(isPartOfSubtheme ?oa - LA ?subt - subtheme)

(isPartOfTheme ?subt - subtheme ?t - Theme)

(isPartOfSubject ?t - Theme ?subj - subject)

(KindResourceLO ?oa - LA ?eq - resource)

(not-has-reqs ?oa - LA)

(has-reqs ?oa - LA ?req - LO)

(has-multiple-reqs ?oa - LA ?req - LO)

(done ?oa - LA)

)

(:functions

(credits-subject ?subj - subject)

(total-credits-subject-gain ?s - student)

(available-credits ?s - student)

(score ?req - LO ?s - student)

(quantity-resource ?eq - resource)

(valueLA ?oa - LA)

(percentSubtheme ?req - LO)

```
(percentTheme ?t - Theme)
(mingrade ?subj - subject)
(DurationLA ?oa - LA)
(maxgrade-subtheme ?subt - subtheme)
(amount-in-subtheme ?oa - LA)
)
```

```
(:durative-action enroll-subject_ Material1
:parameters (?s - student)
:duration (= ?duration 1)
:condition (and
(at start (available-subject Material1 ?s))
(at start (not-approved Material1 ?s))
(at start (<(credits-subject Material1)(available-credits ?s)))
)
:effect (and
(at start (enrollment ?s Material1))
(at start (decrease (available-credits ?s)(credits-subject Material1)))
(at start (not (available-subject Material1 ?s)))
)
)
```

```
(:durative-action CHOOSE-LA-nothasreqs
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq
- resource)
:duration (= ?duration (DurationLA ?oa))
:condition (and
(at start (free ?s))
(at start (enrollment ?s ?subj)))
```

```

(at start (not-done-LA ?oa ?subj ?s))
(at start (isPartOfSubtheme ?oa ?subt))
(at start (isPartOfTheme ?subt ?t))
(at start (isPartOfSubject ?t ?subj))
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
(at start (not-has-reqs ?oa))
(at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
(at start (not(free ?s)))
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1))
(at end (not (not-done-LA ?oa ?subj ?s)))
(at end (increase (score ?subt ?s) (* (valueLA ?oa) (percentSubtheme ?subt))))
(at end (free ?s))
(at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
(at end (done ?oa))
)
)

```

```

(:durative-action CHOOSE-LA-hasreqsSubtheme
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq
- resource ?req - LO)
:duration (= ?duration (DurationLA ?oa))
:condition (and
(at start (free ?s))
(at start (enrollment ?s ?subj))
(at start (not-done-LA ?oa ?subj ?s))
(at start (isPartOfSubtheme ?oa ?subt))

```

```

(at start (isPartOfTheme ?subt ?t))
(at start (isPartOfSubject ?t ?subj))
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
(at start (has-reqs ?oa ?req))
(at start (> (score ?req ?s) (* (amount-in-subtheme ?oa) (percentSubtheme ?req))))
(at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
(at start (not(free ?s)))
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1))
(at end (not (not-done-LA ?oa ?subj ?s)))
(at end (increase (score ?subt ?s) (* (valueLA ?oa) (percentSubtheme ?subt))))
(at end (free ?s))
(at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
(at end (done ?oa))
)
)

(:durative-action CHOOSE-LA-hasreqsLA
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq
- resource ?req - LA)
:duration (= ?duration (DurationLA ?oa))
:condition (and
(at start (free ?s))
(at start (enrollment ?s ?subj))
(at start (not-done-LA ?oa ?subj ?s))
(at start (isPartOfSubtheme ?oa ?subt))
(at start (isPartOfTheme ?subt ?t))

```

```

(at start (isPartOfSubject ?t ?subj))
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
(at start (has-reqs ?oa ?req))
(at start (done ?req))
(at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
(at start (not(free ?s)))
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1))
(at end (not (not-done-LA ?oa ?subj ?s)))
(at end (increase (score ?subt ?s) (* (valueLA ?oa) (percentSubtheme ?subt))))
(at end (free ?s))
(at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
(at end (done ?oa))
)
)

```

```

(:durative-action CHOOSE-LA-hasreqs-multipleLA2
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq
- resource ?req1 - LA ?req2 - LA)
:duration (= ?duration (DurationLA ?oa))
:condition (and
(at start (not(= ?req1 ?req2)))
(at start (free ?s))
(at start (enrollment ?s ?subj))
(at start (not-done-LA ?oa ?subj ?s))
(at start (isPartOfSubtheme ?oa ?subt))
(at start (isPartOfTheme ?subt ?t))

```

```

(at start (isPartOfSubject ?t ?subj))
(at start (KindResourceLO ?oa ?eq))
(at start (> (quantity-resource ?eq) 0))
(at start (done ?req1))
(at start (done ?req2))
(at start (has-multiple-reqs ?oa ?req1))
(at start (has-multiple-reqs ?oa ?req2))
(at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
(at start (not(free ?s)))
(at start (decrease (quantity-resource ?eq) 1))
(at end (increase (quantity-resource ?eq) 1))
(at end (not (not-done-LA ?oa ?subj ?s)))
(at end (increase (score ?subt ?s) (* (valueLA ?oa) (percentSubtheme ?subt))))
(at end (free ?s))
(at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
(at end (done ?oa))
)
)

```

```

(:durative-action PASS-Theme-Tema1_ Material
:parameters (?s - student)
:duration (= ?duration 1)
:condition (and
(at start (enrollment ?s Material1))
(at start (>= (score Subtema1 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema1))))
(at start (>= (score Subtema2 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema2))))

```

```

(at start (>= (score Subtema3 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema3))))
)
:effect (and
(at start (done-Theme Tema1 Material1 ?s))
)
)

```

```

(:durative-action PASS-Theme-Tema2_ Material1
:parameters (?s - student)
:duration (= ?duration 1)
:condition (and
(at start (enrollment ?s Material1))
(at start (>= (score Subtema4 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema4))))
(at start (>= (score Subtema5 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema5))))
(at start (>= (score Subtema6 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema6))))
(at start (done-Theme Tema1 Material1 ?s))
)
:effect (and
(at start (done-Theme Tema2 Material1 ?s))
)
)

```

```

(:durative-action PASS-Theme-Tema3_ Material1
:parameters (?s - student)
:duration (= ?duration 1)

```

```

:condition (and
(at start (enrollment ?s Material1))
(at start (>= (score Subtema7 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema7))))
(at start (>= (score Subtema8 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema8))))
(at start (>= (score Subtema9 ?s)(* (mingrade Material1) (percentSubtheme Sub-
tema9))))
(at start (done-Theme Tema2 Material1 ?s))
)
:effect (and
(at start (done-Theme Tema3 Material1 ?s))
)
)

(:durative-action PASS-Material1
:parameters (?s - student)
:duration (= ?duration 1)
:condition (and
(at start (enrollment ?s Material1))
(at start (done-Theme Tema1 Material1 ?s))
(at start (done-Theme Tema2 Material1 ?s))
(at start (done-Theme Tema3 Material1 ?s))
)
:effect
(at start (done-subject-LA Material1 ?s))
)

(:durative-action take-subject-pass

```

```

:parameters (?s - student ?subj - subject)
:duration (= ?duration 1)
:condition (and
  (at start (enrollment ?s ?subj))
  (at start (done-subject-LA ?subj ?s))
)
:effect (and
  (at start (not (not-approved ?subj ?s)))
  (at start (increase (total-credits-subject-gain ?s) (credits-subject ?subj)))
  (at start (not (available-subject ?subj ?s)))
  (at end (pass-degree ?subj ?s))
)
)
)
)
)

```

C.1.2 PROBLEMA

```

(define (problem degree-example)
  (:domain degree)
  (:objects
    student1 - student
  )

  (:init
    (free student1)
    (= (total-credits-subject-gain student1) 0)
    (= (available-credits student1) 48)
    (available-subject Material1 student1)
    (= (credits-subject Material1) 7)
  )
)

```

(= (mingrade Material1) 70)
(not-approved Material1 student1)
(not-done-LA LA1 Material1 student1)
(= (score Subtema1 student1) 0)
(not-done-LA LA2 Material1 student1)
(= (score Subtema2 student1) 0)
(not-done-LA LA3 Material1 student1)
(not-done-LA LA4 Material1 student1)
(= (score Subtema3 student1) 0)
(not-done-LA LA5 Material1 student1)
(not-done-LA LA6 Material1 student1)
(not-done-LA LA7 Material1 student1)
(not-done-LA LA8 Material1 student1)
(not-done-LA LA9 Material1 student1)
(not-done-LA LA10 Material1 student1)
(= (score Subtema4 student1) 0)
(not-done-LA LA11 Material1 student1)
(not-done-LA LA12 Material1 student1)
(= (score Subtema5 student1) 0)
(not-done-LA LA13 Material1 student1)
(= (score Subtema6 student1) 0)
(not-done-LA LA14 Material1 student1)
(= (score Subtema7 student1) 0)
(not-done-LA LA15 Material1 student1)
(= (score Subtema8 student1) 0)
(not-done-LA LA16 Material1 student1)
(= (score Subtema9 student1) 0)
(not-done-LA LA17 Material1 student1)
(not-done-LA LA18 Material1 student1)

(= (quantity-resource rec0) 100000)

(= (quantity-resource rec1) 20)

(= (quantity-resource rec2) 30)

(= (quantity-resource rec3) 40)

(= (valueLA LA1) 71)

(= (valueLA LA2) 50)

(= (valueLA LA3) 45)

(= (valueLA LA4) 70)

(= (valueLA LA5) 80)

(= (valueLA LA6) 71)

(= (valueLA LA7) 71)

(= (valueLA LA8) 50)

(= (valueLA LA9) 45)

(= (valueLA LA10) 44)

(= (valueLA LA11) 70)

(= (valueLA LA12) 50)

(= (valueLA LA13) 70)

(= (valueLA LA14) 30)

(= (valueLA LA15) 40)

(= (valueLA LA16) 71)

(= (valueLA LA17) 70)

(= (valueLA LA18) 71)

(= (DurationLA LA1) 67)

(= (DurationLA LA2) 53)

(= (DurationLA LA3) 51)

(= (DurationLA LA4) 73)

(= (DurationLA LA5) 83)

(= (DurationLA LA6) 69)
(= (DurationLA LA7) 73)
(= (DurationLA LA8) 54)
(= (DurationLA LA9) 43)
(= (DurationLA LA10) 48)
(= (DurationLA LA11) 67)
(= (DurationLA LA12) 57)
(= (DurationLA LA13) 74)
(= (DurationLA LA14) 31)
(= (DurationLA LA15) 48)
(= (DurationLA LA16) 78)
(= (DurationLA LA17) 71)
(= (DurationLA LA18) 72)

(= (percentSubtheme Subtema1) 0.24)
(= (percentSubtheme Subtema2) 0.43)
(= (percentSubtheme Subtema3) 0.33)
(= (percentSubtheme Subtema4) 0.49)
(= (percentSubtheme Subtema5) 0.21)
(= (percentSubtheme Subtema6) 0.30)
(= (percentSubtheme Subtema7) 0.27)
(= (percentSubtheme Subtema8) 0.29)
(= (percentSubtheme Subtema9) 0.44)

(= (percentTheme Tema1) 0.35)
(= (percentTheme Tema2) 0.36)
(= (percentTheme Tema3) 0.29)

(isPartOfSubtheme LA1 Subtema1)

(isPartOfSubtheme LA2 Subtema1)
(isPartOfSubtheme LA3 Subtema1)
(isPartOfSubtheme LA4 Subtema2)
(isPartOfSubtheme LA5 Subtema2)
(isPartOfSubtheme LA6 Subtema3)
(isPartOfSubtheme LA7 Subtema4)
(isPartOfSubtheme LA8 Subtema5)
(isPartOfSubtheme LA9 Subtema5)
(isPartOfSubtheme LA10 Subtema6)
(isPartOfSubtheme LA11 Subtema6)
(isPartOfSubtheme LA12 Subtema6)
(isPartOfSubtheme LA13 Subtema7)
(isPartOfSubtheme LA14 Subtema7)
(isPartOfSubtheme LA15 Subtema7)
(isPartOfSubtheme LA16 Subtema8)
(isPartOfSubtheme LA17 Subtema8)
(isPartOfSubtheme LA18 Subtema9)

(isPartOfTheme Subtema1 Tema1)
(isPartOfTheme Subtema2 Tema1)
(isPartOfTheme Subtema3 Tema1)
(isPartOfTheme Subtema4 Tema2)
(isPartOfTheme Subtema5 Tema2)
(isPartOfTheme Subtema6 Tema2)
(isPartOfTheme Subtema7 Tema3)
(isPartOfTheme Subtema8 Tema3)
(isPartOfTheme Subtema9 Tema3)

(isPartOfSubject Tema1 Material1)

(isPartOfSubject Tema2 Material1)

(isPartOfSubject Tema3 Material1)

(KindResourceLO LA1 rec1)

(KindResourceLO LA2 rec1)

(KindResourceLO LA3 rec3)

(KindResourceLO LA4 rec2)

(KindResourceLO LA5 rec1)

(KindResourceLO LA6 rec1)

(KindResourceLO LA7 rec2)

(KindResourceLO LA8 rec3)

(KindResourceLO LA9 rec2)

(KindResourceLO LA10 rec1)

(KindResourceLO LA11 rec2)

(KindResourceLO LA12 rec2)

(KindResourceLO LA13 rec3)

(KindResourceLO LA14 rec1)

(KindResourceLO LA15 rec3)

(KindResourceLO LA16 rec2)

(KindResourceLO LA17 rec2)

(KindResourceLO LA18 rec2)

(not-has-reqs LA1)

(not-has-reqs LA2)

(not-has-reqs LA3)

(has-reqs LA4 Subtema1)

(= (amount-in-subtheme LA4) 0)

(has-reqs LA5 Subtema1)

(= (amount-in-subtheme LA5) 0)

(has-reqs LA6 Subtema2)
(= (amount-in-subtheme LA6) 0)
(has-reqs LA7 Subtema3)
(= (amount-in-subtheme LA7) 0)
(has-reqs LA8 Subtema4)
(= (amount-in-subtheme LA8) 0)
(has-reqs LA9 Subtema4)
(= (amount-in-subtheme LA9) 0)
(has-reqs LA10 Subtema5)
(= (amount-in-subtheme LA10) 0)
(has-reqs LA11 Subtema5)
(= (amount-in-subtheme LA11) 0)
(has-reqs LA12 Subtema5)
(= (amount-in-subtheme LA12) 0)
(has-reqs LA13 Subtema6)
(= (amount-in-subtheme LA13) 0)
(has-reqs LA14 Subtema6)
(= (amount-in-subtheme LA14) 0)
(has-reqs LA15 Subtema6)
(= (amount-in-subtheme LA15) 0)
(has-reqs LA16 Subtema7)
(= (amount-in-subtheme LA16) 0)
(has-reqs LA17 Subtema7)
(= (amount-in-subtheme LA17) 0)
(has-reqs LA18 Subtema8)
(= (amount-in-subtheme LA18) 0)

(has-reqs LA6 LA1)
(has-reqs LA8 LA4)
(has-reqs LA12 LA6)

```
(has-reqs LA14 LA7)
```

```
(= (maxgrade-subtheme Subtema1) 100)  
(= (maxgrade-subtheme Subtema2) 100)  
(= (maxgrade-subtheme Subtema3) 100)  
(= (maxgrade-subtheme Subtema4) 100)  
(= (maxgrade-subtheme Subtema5) 100)  
(= (maxgrade-subtheme Subtema6) 100)  
(= (maxgrade-subtheme Subtema7) 100)  
(= (maxgrade-subtheme Subtema8) 100)  
(= (maxgrade-subtheme Subtema9) 100)  
)
```

```
(:goal (and  
(pass-degree Material1 student1)  
)  
)
```

```
(:metric minimize (total-time))  
)
```

BIBLIOGRAFÍA

- [1] *Introducción a la Metodología Didáctica. Formación Profesional Ocupacional Ebook*, MAD-Eduforma, 2006.
- [2] *Diseño Instruccional En Un Entorno de Aprendizaje Abierto*, Universidad Tecnológica de Pereira, 2007.
- [3] BARCHINO, R., J. M. GUTIÉRREZ y S. OTÓN, «An example of Learning Management System», *IADIS / International Association for Development of the Information Society*, 2005.
- [4] BERZAL, F., «PDDL», recurso libre, disponible en <http://elvex.ugr.es/decsai/iaio/slides/P9>
- [5] BISCAY, C. E., «Los estándares de e-learning», recurso libre, disponible en <http://www.palermo.edu/ingenieria/downloads/CyT5/CYT506.pdf>, 2006.
- [6] BOHL, O., «The sharable content object reference model (SCORM) - a critical review», en *Computers in Education, 2002. Proceedings. International Conference on*.
- [7] BONET, B. y H. GEFNER, «Planning as heuristic search», *Artificial Intelligence*, 2001.
- [8] CABERO, J., «Bases pedagógicas del e-learning», *Revista de Universidad y Sociedad del Conocimiento (RUSC) [artículo en línea]*, **3**(1), págs. 01–10, 2006.

- [9] CASTILLO, L., L. MORALES, A. GONZÁLEZ-FERRER, J. FDEZ-OLIVARES, D. BORRAJO y E. ONAINDÍA, «Automatic generation of temporal planning domains for e-learning problems», *J. of Scheduling*, **13**(4), págs. 347–362, agosto 2010, URL <http://dx.doi.org/10.1007/s10951-009-0140-x>.
- [10] CASTILLO, L., L. MORALES, A. GONZÁLEZ-FERRER, J. FERNÁNDEZ-OLIVARES y O. GARCÍA-PÉREZ, «Current Topics in Artificial Intelligence», capítulo Knowledge Engineering and Planning for the Automated Synthesis of Customized Learning Designs, Springer-Verlag, Berlin, Heidelberg, págs. 40–49, 2007, URL http://dx.doi.org/10.1007/978-3-540-75271-4_5.
- [11] CHEN, Y., C.-W. HSU y B. W.WAH, «SGPlan: Subgoal Partitioning and Resolution in Planning», *American Association for Artificial Intelligence*, 2004.
- [12] COMMITTEE, ©. I. L. T. S., «WG12: Learning Object Metadata», recurso libre, disponible en <http://ltsc.ieee.org/wg12/>, 2005.
- [13] DE MADRID-ESCUELA POLITÉCNICA SUPERIOR, U. C. I., «Manual de usuario PLTool», recurso libre, disponible en <http://www.plg.inf.uc3m.es/software/pltool/manual-pdf/manual-es.pdf>, 2007.
- [14] DUART, J. M., «Estrategias en la introducción y uso del e-Learning en educación superior», *Educación Médica versión impresa ISSN 1575-1813*, **9**(2), págs. 15–22, 2006.
- [15] DUQUE MÉNDEZ, N. D., C. JIMÉNEZ RAMÍREZ y J. A. GUZMÁN LUNA, «IA Planning for automatic generation of customized virtual courses», *IOS Press*, 2005.
- [16] DYNARSKI, M., L. CLARKE, B. COBB, J. FINN, R. RUMBERGER y J. SMINK, «Dropout Prevention», *National Center for Education Evaluation and regional Assistance. Institute of Education Sciences. U.S. Department of Education*, 2000.

- [17] FOX, M. y D. LONG, «PDDL 2.1: An Extension to PDDL for Expressing Temporal», *Journal of Artificial Intelligence Research*, 2003.
- [18] GARRIDO, A. y E. ONAINDIA, «On the Application of Planning and Scheduling Techniques to E-learning», *IEA/AIE'10 Proceedings of the 23rd International Conference on Industrial Engineering and other Applications of applied Intelligent System*, 2010.
- [19] GARRIDO, A., E. ONAINDIA, L. MORALES, L. CASTILLO, S. FERNÁNDEZ y D. BORRAJO, «Modeling E-Learning Activities in Automated Planning», recurso libre, disponible en <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.9985>, 2012.
- [20] GARRIDO, A., E. ONAINDIA y O. SAPENA, «Automated planning for personalised course composition», *Advanced Learning Technologies, ICALT*, 2009.
- [21] GEREVINI, A., A. SAETTI y I. SERINA, «Planning through Stochastic Local Search and Temporal Action Graphs», *Journal of Artificial Intelligence Research (JAIR)*, 2003.
- [22] HERNÁNDEZ, E., «Unidades de aprendizaje, una propuesta de complemento a los objetos de aprendizaje», *Teoría de la Educación: Educación y Cultura en la Sociedad de la Información*, 2005.
- [23] HOFFMANN, J. y B. NEBEL, «The FF planning system: fast plan generation through heuristic search», *Journal of Artificial Intelligence Research*, 2001.
- [24] HSU, C.-W. y B. W. WAH, «The SGPlan Planning System in IPC-6», *Association for the Advancement of Artificial Intelligence*, 2008.
- [25] ICAPS, «Home Page ICAPS», recurso libre, disponible en <http://ipc.icaps-conference.org/>, 2013.
- [26] KOPER, R., *Learning Design*, Springer Berlin Heidelberg, 2005.

- [27] MALTE, H., «An Introduction to PDDL», recurso libre, disponible en <http://www.cs.toronto.edu/~sheila/2542/w09/A1/introtopddl2.pdf>, 2009.
- [28] McDERMOTT, D. M., «The 1998 AI planning systems competition», *AI magazine*, 2000.
- [29] MORALES, L., «Automatic Generation of Learning Designs: Two Planning Approaches Comparison», recurso libre, disponible en <http://buhoz.net/public/lluvia/documents/Seminario2009.pdf>, 2009.
- [30] MORALES REYNAGA, L. C., *Generación Automática de Diseños de Aprendizaje: Diferentes Enfoques de Planificación*, Tesis de Maestría, Universidad de Granada, 2008.
- [31] MORALES REYNAGA, L. C., *Generación Automática de Diseños de Aprendizaje: Diferentes Enfoques de Planificación*, Tesis Doctoral, Universidad de Granada, 2011.
- [32] NCGREAL, R., «Learning objects: A practical definition», *International Journal of Instructional Technology and Distance Learning (IJITDL)*, 2004.
- [33] PASHLER, H., P. BAIN, B. BOTTGE, A. GRAESSER, K. KOEDINGER, M. MCDANIEL y J. METCALFE, «Organizing Instruction And Study to Improve Student Learning, A Practice Guide», *National Center for Education Research. U.S. Department of Education*, 2007.
- [34] PETCH, J., «Instructional Management Systems (IMS)», recurso libre, disponible en <http://www.ncgia.ucsb.edu/ige98/report/ims.html>, 2012.
- [35] POLSANI, P. R., «Use and Abuse of Reusable Learning Objects», *Journal of Digital Information*, 2003.
- [36] RUSSELL, S. y P. NORVIG, *Inteligencia Artificial. Un enfoque Moderno*, primera edición, Prentice Hall Hispanoamerica, S.A., México, DF, 1996.

-
- [37] RUSSELL, S. y P. NORVIG, *Artificial Intelligence: A Modern Approach*, segunda edición, Prentice Hall, Estados Unidos de América, 2003.
- [38] RUSSELL, S. y P. NORVIG, *Inteligencia Artificial. Un enfoque Moderno*, segunda edición, Pearson Educación, Madrid, 2004.
- [39] SMITH, G. E. y S. THRONE, *Differentiating Instruction with Technology in k-5 Classrooms*, primera edición, International Society for Technology in Education (ISTE), Estados Unidos de America, 2007.
- [40] VARAS, M. L., «Repositorios de Objetos de Aprendizaje», recurso libre, disponible en <http://cvonline.uaeh.edu.mx/Cursos/Especialidad/TecnologiaEducativaG13/Modulo4/unida>
- [41] WINSTON, P. y B. HORN, *LISP*, tercera edición, Addison-Wesley, Estados Unidos de América, 1989.
- [42] WORKS, P., «Estándares E-Learning», recurso libre, disponible en <http://internet-educativa.pbworks.com/w/page/20082240/Estrning>, 2008.

FICHA AUTOBIOGRÁFICA

Cristina Maya Padrón

Candidato para el grado de Maestro en Ingeniería
con especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

MODELACIÓN DE PLANES DE ESTUDIO USANDO
TÉCNICAS AVANZADAS DE PLANIFICACIÓN

Licenciada en Informática Administrativa por la Facultad de Contaduría Pública y Administración de la Universidad Autónoma de Nuevo León, egresada en el año 2008 con Reconocimiento “Mención Honorífica” por desempeño académico sobresaliente.