

## Capítulo V.

### Generación de sugerencias

Para la generación de sugerencias para la asistencia dinámica del Jefe de Departamento es necesaria la generación masiva de combinaciones entre los diferentes elementos que conforman una sección (materia, profesor, hora de inicio, hora de fin y salón) discriminando posteriormente (o de ser posible, anteriormente) las combinaciones inválidas a través de reglas preestablecidas. Como es posible intuir a través de lo dicho anteriormente, este tipo de procesamiento suele ser altamente intensivo y poco eficaz por lo que se han desarrollado muchos sistemas que usan “Programación Lógica” para resolver este tipo de problemas.

La programación lógica se refiere a la creación de un programa basado en fórmulas matemáticas que representan las condiciones a cumplir como función objetivo, lo cual es un paradigma completamente diferente a la programación típica. “Esto significa que el programador no necesita escribir un programa que resuelva un problema, sino especifica cómo debe de ser la solución [a través de reglas y condiciones] y el sistema de inferencias del propio sistema [...] intentará encontrar la manera de resolver el problema y [dará la] solución en sí. Esto se realiza a través de *reglas* y *hechos*. Los hechos son datos de entrada y las reglas se pueden usar para derivar más hechos y así encontrar la solución al problema” (Koch, 2000)

De los múltiples sistemas que han sido desarrollados para ser usados con la programación lógica, se eligió DLV (DataLog with Disjunction) como plataforma para desarrollar esta parte del sistema. DLV es un “sistema deductivo de base de datos basado en programación lógica disyuntiva” (Bihlmeyer, 2006) y es un programa libre que se

encuentra en constante desarrollo desde 1997, lo cual le ha dado bastante estabilidad y aceptación en la comunidad científica.

Como fue mencionado anteriormente, la programación lógica requiere de hechos y reglas, siendo las últimas las que se enumerarán a continuación dando una explicación muy sencilla de entender para todos:

### **5.1 Reglas generadas a partir de las elecciones del usuario:**

1. El curso deberá ser impartido el número de días en la semana que el usuario elija.
  - a. Si el usuario elije 2 días, de preferencia deberán ser martes y jueves
  - b. Si el usuario elije 3 días, de preferencia deberán ser lunes, miércoles y viernes
  - c. Si el usuario elije 4 días, de preferencia deberán ser de lunes a jueves
2. La duración diaria del curso por deberá ser la indicada por el usuario.
3. La materia de la que se conforma la sección deberá ser la que el usuario elija.
4. El profesor que imparta la sección deberá ser la que el usuario elija si alguno fue indicado, de lo contrario se le dará preferencia a los profesores que hayan solicitado impartir la materia.
5. El salón en el que se imparta la sección deberá ser el que el usuario elija si alguno fue indicado, de lo contrario, se le dará preferencia a los salones que estén asociados con la carrera a la que pertenece la materia.

### **5.2 Reglas inherentes al sistema de sugerencias:**

1. Los días en los que sean impartidas las secciones deberán de ser entre lunes y viernes únicamente.

2. El horario del curso deberá empezar a una hora exacta o treinta minutos después de la hora.
3. Dado un horario de inicio y de fin de una sección, el profesor que la imparta no deberá tener otro curso con el cual se sobrepongan los horarios para los días de la sección en cuestión.
4. Dado un horario de inicio y fin de la sección, no deberá haber otra sección que se imparta en el mismo salón cuyos horarios se sobrepongan.
5. Si el usuario no eligió ningún profesor en particular para impartir la sección, se deberá dar prioridad a los profesores que indicaron querer impartir la materia.
6. Se debe intentar cumplir con las preferencias de horario que establezca el profesor para impartir sus cursos.
7. La sección no deberá ser impartida en el mismo horario que otra materia del mismo semestre del mismo plan de estudio a menos que exista otra sección disponible para los alumnos.
8. La sección deberá ser impartida todos los días en el mismo salón, mismo horario y por el mismo profesor.

### **5.3 Catalogo de hechos**

Un catálogo de hechos (también conocido como *datalog*) es simplemente un archivo de texto que contiene la lista de hechos que utiliza DLV para encontrar soluciones (también conocidas como *modelos*). Se puede entender el catálogo de hechos como la base de datos que contiene toda la información necesaria para crear combinaciones válidas y ser convertidas de regreso a las entidades manejadas a través de todo el sistema (como Profesores, Salones, Materias, Salones, etc.).

Aunque con DLV es posible recuperar información directamente de la base de datos (a través de ODBC), en este proyecto se decidió mantener la conexión a la base de datos como exclusiva del Modelo de Datos, por lo cual, se le solicita a Hibernate listados de los objetos almacenados en la base de datos, de los cuales se extrae la información mínima necesaria para crear un archivo de texto plano con el siguiente formato de tuplas:

<u>Información</u>	<u>Tuplas</u>
Lista de profesores	profesor( Profesor ).
Lista de salones	salon( Salon ).
Lista de planes de estudio	plan( Plan, Semestre, Materia ).
	seccion_materia( Seccion, Materia ).
	seccion_profesor( Seccion, Profesor ).
	seccion_salon( Seccion, Salon ).
Lista de secciones	seccion_inicio( Seccion, Hora_inicio ).
	seccion_fin( Seccion, Hora_fin ).
	seccion_dias( Seccion, Dia ).
Lista de preferencias de profesores por materias	profesor_materia_preferente( Profesor, Materia ).
Lista de preferencias de horarios de profesor	profesor_horario( Profesor, Hora_inicio, Hora_fin, Dia ).

Es importante mencionar que *salón*, *profesor*, *plan\_de\_estudios* y *seccion* son representaciones numéricas (IDs) de las entidades en la base de datos real. *Hora\_inicio* y *hora\_fin* son horas transformadas a minutos y divididas entre 5 (siendo 5 minutos el mínimo común denominador de horarios) para facilitar las operaciones matemáticas que se tienen que realizar sobre los mismos en DLV, y finalmente los días son números enteros del 1 al 5 representando de lunes a viernes.

Una vez que DLV tiene toda la información necesaria en el *datalog* y todas las reglas definidas, es posible realizar la búsqueda de una solución. No se explicará el funcionamiento interno de DLV porque eso va más allá del propósito de este documento,

pero se puede resumir todo el proceso en una transformación de las reglas y hechos en representaciones manipulables por el motor de generación de modelos que se encarga de finalmente devolver modelos estables (si los hubiese).

Observaciones importantes:

- El *datalog* se construye automáticamente en el momento en el que el usuario solicita sugerencias para reflejar el estado actual del sistema.
- El *datalog* sólo contiene información de los planes de estudio, profesores y salones relacionados con la carrera a la que pertenece la materia elegida por el usuario ya que las sugerencias sólo podrán ser conformadas por entidades relacionadas a la carrera.
- El *datalog* contiene información de todos los cursos ya existentes en el sistema para poder asegurar que se tiene toda la información de salones, horarios y profesores involucrados.

## 5.4 Modelos Estables

Los modelos resultantes o soluciones, si existe alguno, serán obtenidos de tuplas formadas de la siguiente manera: *seccion( Materia, Profesor, Salón, Hora\_inicio, Hora\_fin)* y *dias(N)*, donde todas las variables mantendrán los mismos significados que en el *datalog*.

Es importante mencionar que DLV es capaz de encontrar un número casi infinito de soluciones a un problema (si los tuviese), y tratándose de un prototipo de software el que fue diseñado, se permitirá al usuario determinar el número máximo de modelos esperados (de 1 a 30) o todos los encontrados por DLV. Lo recomendable es que el usuario nunca exceda de 30 soluciones ya que el tiempo de procesamiento para todos los modelos puede ser exagerado si existen muchas soluciones.

## 5.5 Estructura del sistema en DLV

El programa escrito para DLV fue dividido en los siguientes archivos con el propósito de facilitar su comprensión, facilitar el mantenimiento del código y facilitar su extensibilidad:

- *Datalog.dlv:*

Este archivo contiene toda la información necesaria almacenada en la base de datos. El contenido de este archivo fue explicado previamente.

- *Dias.dlv:*

Este archivo contiene las reglas usadas para determinar qué días de la semana debe de ser impartida la sección, así como reglas de preferencia a partir de la elección del número total de días a la semana que debe planearse el curso.

- *Horario.dlv*

Este archivo define el horario de la sección que se desea crear; la cual debe tener una duración especificada por el usuario. También en este archivo se define que el horario de inicio debe de ser a la hora o media hora después.

- *Planes.dlv*

Este archivo asegura el cumplimiento de las reglas referentes a los planes de estudio y que las materias estén disponibles para todos los estudiantes cursando sus respectivos semestre.

- *Profesor.dlv*

Este archivo asigna un profesor a cada modelo generado a partir de las preferencias de los profesores (en caso de existir) relacionadas con la materia del curso que se desea