

Instituto Tecnológico de Costa Rica

Escuela de Computación

Lenguajes de Programación

Profesor

Oscar Víquez Acuña

Tercer Proyecto

(Paradigma Lógico)

“Sistema de matrícula TEC”

Eliomar Rodríguez Arguedas

Sede San Carlos

14 de noviembre del 2017

Tabla de contenidos

Introducción	3
Análisis del problema	4
Análisis de Resultados	5
Conclusiones	7
Recomendaciones	8

Introducción

Esta tarea programada trata sobre la creación de una aplicación que genere horarios automáticamente, generando estos horarios respetando ciertas normas, como el choque de horarios, tipos de asignaturas, aulas disponibles y de las clases o asignaturas que posea un profesor específico.

El problema de la creación de horarios es muy viejo y conocido; ya que desde en los principios de la programación se pensaron formas de automatizar y resolver este problema eficaz y eficientemente sin necesidad de mucho esfuerzo, sin embargo, este problema varía y se adapta a las distintas épocas y eras de la tecnología, hoy en día, los problemas de horarios son enormes y poseen muchos frentes de información, por lo cual esta tarea no es sencilla.

Para lograr esta tarea, se debe implementar la aplicación en el lenguaje de programación Prolog; ya que para la realización de este proyecto se requiere de la utilización e implementación del paradigma lógico para la creación de estos horarios.

Análisis del problema

El problema de este proyecto radica en la utilización de reglas y hechos que creen el horario adecuado de acuerdo a las siguientes restricciones:

- Las asignaturas de que consta cada semestre.
- Las asignaturas a impartir por cada profesor.
- Las aulas disponibles para impartir las clases.
- Cada profesor podrá dar un máximo de 3 por semestre.
- Los posibles semestres son ocho, de primero al octavo, esto interesa para saber a cuál semestre pertenece cada asignatura.
- Los días de clase van de lunes a viernes, y las horas posibles son de 7:00 a 11:30 y de 12:30 a 16:00.

Además, los datos deben estar almacenados en una base de datos externa, los cuales serán accedados por la parte lógica del programa, luego de ser procesados por esta parte, se enviarán a la parte gráfica, donde serán mostrados al usuario. Al final se le debe generar al usuario un horario sin choques con los cursos seleccionados.

Análisis de Resultados

Descripción	Estado	Observaciones
Uso de un gestor de base de datos para almacenar la información requerida	100%	SQL server
Creación de scripts de tablas para la base de datos	100%	
Creación de dos scripts diferentes para la inserción de datos en la base de datos	100%	
Conexión entre Java, SQL server y prolog	100%	
Consulta a base de datos para obtener los datos necesarios como profesores, aulas, cursos, etc.	100%	
Creación de hechos desde Java de forma dinámica	10%	Se tuvieron varios problemas con este paso, se logró realizar el código pero después de muchos intentos no hubo resultados positivos.
Vista donde se puedan mostrar los cursos que se pueden matricular	100%	
Se cargan todos los cursos para que el usuario pueda elegir que cursos matricular	100%	
Filtro de cursos por semestre	100%	
Se logra la creación de horarios	40%	Se tiene el código, pero como no se pudo realizar la parte de crear hechos entonces no se cargaron los datos en prolog, solo en la vista.

Se muestra al usuario el horario final generado	10%	El horario generado se imprime en consola (resultado de la consulta a prolog) pero como no se pudo cargar a lógica de prolog los cursos de la base de datos solo sirve con unos datos quemados que van en el archivo, pero se demuestra que la lógica en prolog si está completa
Lógica de prolog completa (reglas, hechos)	100%	
En caso de choque de horario se le notifica al usuario	0%	No se logró llegar a esa parte lastimosamente por el mal manejo de tiempo y al final no se pudo lograr el cometido.
Manual de usuario	0%	Aunque la ventana es muy intuitiva y fácil de utilizar, no se logró realizar el manual de usuario.
Documentación externa	100%	

Conclusiones

Mediante la realización de este proyecto, se pudieron comprobar las ventajas y beneficios de ciertas características de la programación lógica en Prolog, como por ejemplo el concepto de unificación, el cual permitió asociar las variables necesarias para la creación de los horarios y poder respetar las restricciones de choques y demás; además de la gran ventaja de la utilización de hechos y predicados, los cuales permitieron la relación de las características de los objetos de cada horario y las especificaciones que existan entre ellos; además de la simplicidad y sencillez en la implementación de estructuras complejas, en este caso todos los horarios, y por último, las facilidades que posee este paradigma a la hora de realizar backtracking y recursividad en sí, lo cual lo hace un paradigma muy sencillo y potente a la vez.

Luego de haber realizado este proyecto, el nivel de conocimiento adquirido en el tema de reglas, predicados, backtracking, recursividad, bases de datos, utilización de librerías, conexiones entre distintos lenguajes de programación y otros temas relacionados se vieron notoriamente incrementados.

Para terminar, se experimentaron y recordaron diferentes técnicas del paradigma de la programación lógica, la cual es entretenida e interesante de utilizar, además, de que estos nuevos y viejos conocimientos en este apartado pueden que nos sirvan más adelante en nuestra carrera universitaria y en un futuro ambiente laboral.

Recomendaciones

Para la correcta ejecución de todo el proyecto, a continuación se adjuntan dos enlaces en donde podrá encontrar una guía de cómo configurar el proyecto con SQL server, Java y prolog.

El siguiente enlace es para poder configurar SQL server con Java

<http://acodigo.blogspot.com/2013/06/conectar-sql-server-con-java.html>

El siguiente enlace es para poder configurar Prolog con Java

<https://www.youtube.com/watch?v=9IM9yRfoFMc>