

**INSTITUTO POLITECNICO NACIONAL**

---

---

**CENTRO DE INVESTIGACION EN COMPUTACION**

**GENERACION DE HORARIOS MEDIANTE SISTEMAS  
BASADOS EN CONOCIMIENTO**

**TESIS**

**QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS DE LA COMPUTACION  
PRESENTA:**

**BORIS ANTONIO ARANDA BENITEZ**

**DIRECTOR: DR. ALVARO DE ALBORNOZ BUENO  
CODIRECTOR: M. en C. JOSÉ RAFAEL CEN ZUBIETA**

México, D.F.

Octubre del 2007



## ÍNDICE

<b>RESUMEN</b>	i
<b>ABSTRAC</b>	ii
<b>ÍNDICE</b>	iii
<b>LISTA DE FIGURAS</b>	v
<b>CAPÍTULO 1. INTRODUCCIÓN</b>	
1.1 Antecedentes	1
1.2 Planteamiento del problema	2
1.3 Objetivo general	4
1.4 Objetivos específicos	5
1.5 Justificación	5
1.5.1 Justificación teórica	5
1.5.2 Justificación metodológica	5
1.5.3 Justificación práctica	6
1.6 Alcances y limitaciones	7
<b>CAPÍTULO 2 MARCO TEÓRICO</b>	
2.1 El estado del arte	9
2.1.1 Antecedentes	9
2.1.2 Algunos métodos para resolver el Problema de la Generación de Horarios	9
2.1.2.1 Algoritmos Genéticos	12
2.1.2.2 Programación Lógica de Restricciones	12
2.1.2.3 Redes Neuronales	13
2.1.2.4 Recocido Simulado	13
2.1.2.5 Operadores Recombinatorios Especializados	13
2.1.2.6 El Sistema Experto basado en reglas	14
2.1.3 Observaciones realizadas a algunos paquetes generadores de horarios	15
2.2 Los horarios	16
2.2.1 El problema de la elaboración de horarios	17
2.3 El proceso de la elaboración de horarios	22

### **CAPÍTULO 3 TÉCNICAS DE INTELIGENCIA ARTIFICIAL UTILIZADAS PARA LA GENERACIÓN DE HORARIOS**

3.1 Sistemas Expertos	24
3.1.1 Función de un Sistema Experto	25
3.1.2 Arquitectura de Sistemas Expertos	25
3.1.3 La base de conocimiento	26
3.1.4 Motor de inferencias	27
3.1.5 ¿Por qué utilizar un Sistema Experto?.	27
3.1.6 Construcción de prototipos	28
3.2 Algoritmos genéticos	29
3.2.1 Antecedentes	29
3.2.2 Definición de Algoritmo Genético	30
3.2.3 Ventajas y desventajas	31
3.2.4 ¿Cómo saber si es posible usar un Algoritmo Genético?	31
3.2.5 Parámetros que deben ser especificados en el programa	32
3.3 Programación Lógica de Restricciones...(PLR).	34
3.3.1 Antecedentes de la PLR	34
3.3.2 Restricciones	35
3.3.3 Sistemas con restricciones	37
3.3.4 Referencia histórica de las restricciones	37
3.3.5 Satisfacción de las restricciones	38
3.3.6 Resolución de restricciones	38
3.3.7 Organización de proyectos	38

### **CAPÍTULO 4 ANÁLISIS Y DISEÑO**

4.1 Recopilación de la información	41
4.2 Análisis de la información	46
4.3 Solución definida	48
4.4 Requerimientos del sistema	49
4.5 Diseño del sistema	52
4.5.1 Mapa conceptual del proyecto	52
4.5.2 Mapas conceptuales de los módulos	54
4.5.3 Diagramas de contexto	62
4.5.4 Diagramas entidad relación	67
4.5.5 Diccionario de la base de datos	73
4.5.6 Lista de funciones o programas	85
4.5.7 Diagrama de arquitectura (menús)	87

### **CAPÍTULO 5 IMPLEMENTACIÓN**

5.1 Pantallas del sistema	89
---------------------------	----

**CAPÍTULO 6 PRUEBAS Y RESULTADOS**

6.1 Pruebas del sistema	114
6.2 Pruebas del sistema	115
6.2.1 Prueba 1: Área de menús	115
6.2.2 Prueba 2: Validación de entradas	118
6.2.3 Prueba 3: Despliegado de resultados	120
6.2.3.1 Módulo generación de grupos por abrir	120
6.2.3.2 Módulo generación de horarios	121
6.2.3.3 Módulo asignación de aula	122
6.2.3.4 Módulo selección de catedráticos	127

**CAPÍTULO 7 CONCLUSIONES Y TRABAJOS FUTUROS**

7.1 Conclusiones	131
7.2 Trabajos futuros	132

<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>133</b>
-----------------------------------	------------

## LISTA DE FIGURAS

Figura	Nombre	Página
3.1	Arquitectura de un Sistema Experto	26
3.2	Pseudocódigo del Algoritmo Genético simple	33
4.1	Mapa conceptual del proyecto	52
4.2	Diagrama de secuencia	54
4.3	Sistema Experto generador de grupos	55
4.4	Diagrama del módulo Programación Lógica de Restricciones	57
4.5	Diagrama del módulo Algoritmo Genético	58
4.6	Diagrama de Sistema Experto para sugerir profesor a asignatura	60
4.7	Diagrama de Sistema Experto para sugerir asignaturas para catedrático	61
4.8	Diagrama de contexto del proyecto	64
4.9	Diagrama de funciones fundamentales (entradas)	65
4.10	Diagrama de funciones fundamentales (salidas)	66
4.11	Diagrama entidad relación del módulo administrativo	67
4.12	Diagrama E-R del módulo para determinar y ajustar número de grupos	68
4.13	Diagrama E-R del módulo para asignar aulas (Algoritmo Genético)	69
4.14	Diagrama E-R del módulo para asignar horario (PLR)	70
4.15	Diagrama E-R del módulo designa mejor catedrático (Sistema Experto)	71
4.16	Diagrama E-R del módulo selecciona mejor materia (Sistema Experto)	72
5.1	Portada inicial del proyecto	88
5.2	Acerca del sistema	89
5.3	Menú principal del proyecto	89
5.4	Menú grupos por abrir	90
5.5	Materias	90
5.6	Consulta materias	91
5.7	Pantalla servicios escolares	91
5.8	Pantalla consulta servicios escolares	92
5.9	Pantalla criterios por materia	92
5.10	Pantalla consulta criterios por materia	93
5.11	Pantalla carreras	93
5.12	Pantalla criterios	94
5.13	Pantalla consulta criterios	94
5.14	Pantalla reglas	95
5.15	Pantalla consulta reglas	95
5.16	Pantalla ajustes	96

5.17	Pantalla consulta ajustes	96
5.18	Pantalla criterios por regla	97
5.19	Pantalla consulta criterios por regla	97
5.20	Pantalla cálculo de ajustes	98
5.21	Pantalla explicación de la inferencia	98
5.22	Pantalla inicial del módulo generación de horarios	99
5.23	Pantalla principal del módulo generación de horarios	99
5.24	Pantalla principal del menú Algoritmos Genéticos	100
5.25	Pantalla de administración de grupos	100
5.26	Pantalla de consulta de administración de grupos	101
5.27	Pantalla del control de aulas	101
5.28	Pantalla de consultas del control de aulas	102
5.29	Pantalla para generación de población inicial	102
5.30	Pantalla de evaluación de versiones	103
5.31	Pantalla del proceso de selección	103
5.32	Pantalla del proceso de clonación	104
5.33	Pantalla de mutación	104
5.34	Pantalla del proceso de cruzamiento	105
5.35	Horarios finales	105
5.36	Pantalla menú selección de catedráticos	106
5.37	Pantalla de perfiles	106
5.38	Pantalla de consulta perfiles	107
5.39	Pantalla perfiles de las materias	107
5.40	Pantalla consulta perfiles de las materias	108
5.41	Pantalla captura datos de catedráticos	108
5.42	Pantalla consulta datos de catedráticos	109
5.43	Pantalla captura perfil de catedráticos	109
5.44	Pantalla consulta perfil de catedráticos	110
5.45	Pantalla muestra inferencia catedráticos para materia	110
5.46	Pantalla explicación del porqué un maestro puede dar una materia	111
5.47	Pantalla materias que puede impartir un profesor	111
5.48	Pantalla de explicación del porqué un profesor puede dar una materia	112
6.1	Portada inicial del proyecto	115
6.2	Menú principal del proyecto	116
6.3	Menú grupos por abrir	116
6.4	Pantalla menú selección de catedráticos	117
6.5	Pantalla principal del menú Algoritmos Genéticos	117
6.6	Pantalla de captura de datos de las Materias	118
6.7	Pantalla de captura de datos de Criterios por Materia	119
6.8	Pantalla de captura de datos de Carreras	119
6.9	Pantalla cálculo de ajustes	120

6.10	Pantalla inicial del módulo generación de horarios	121
6.11	Pantalla principal del módulo generación de horarios	122
6.12	Pantalla del control de aulas	123
6.13	Pantalla de consultas del control de aulas	123
6.14	Pantalla de genera población inicial	124
6.15	Pantalla de evaluación de versiones	124
6.16	Pantalla del proceso de selección	125
6.17	Pantalla del proceso de clonación	125
6.18	Pantalla de mutación	126
6.19	Pantalla del proceso de cruzamiento	126
6.20	Horarios finales	127
6.21	Pantalla captura perfil de catedráticos	128
6.22	Pantalla muestra inferencia catedráticos para materia	128
6.23	Pantalla explicación porque un maestro puede dar una materia	129
6.24	Pantalla materias que puede impartir un profesor	129
6.25	Pantalla de explicación porque un profesor puede dar una materia	130

## **RESUMEN**

En este estudio se aborda al problema NP-completo de la elaboración de horarios en una escuela de nivel superior. Se definen los elementos de un conjunto de Sistemas Expertos, un módulo de Programación Lógica de Restricciones y un módulo de Algoritmos Genéticos para solucionar el problema.

Comenzando con la determinación de grupos por abrir, después la asignación de horario a los grupos, posteriormente sigue la asignación de aula a los grupos y terminando con la propuesta de profesor para las asignaturas, se logra generar un conjunto de soluciones aceptables de horarios sin problemas que satisfacen las restricciones naturales que imponen las entidades involucradas (catedráticos, aulas, asignaturas y los grupos) en el proceso.



## ABSTRACT

This research deals about the Timetabling NP-complete problem applied at the Instituto Tecnológico de Zacatepec. In this document the elements of three Experts System, a constraints logic programming module and a Genetics Algorithms module to solve the problem, are presented

Starting with a process to determine how many groups should be open the next semester, after this, continue the **timetabling assignation** to the groups, the next process is the **classroom assignation** to the groups and the final module **propose a teacher for the groups**, all this make a set of acceptable solutions without timetabling problems where the natural constraints are satisfied.

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 Antecedentes

#### ¿Qué es un horario?

**“El horario completo bien puede gobernar las vidas de miles de alumnos y cientos de profesores para los 200 días del año escolar, semestre a semestre, hora tras hora. Como una poderosa herramienta puede fácilmente crear o deshacer situaciones de enseñanza aprendizaje.”** [Johnson, 1980].

En la educación superior la historia siempre es la misma, la proporción de tiempo dictada por el horario para los alumnos y profesores es generalmente muy baja. A pesar de esto, las dificultades en la creación de los horarios son grandes. Existen varios departamentos y carreras cada uno con sus propias ideas de cómo y cuándo se deben programar sus cursos. El número de estudiantes puede variar considerablemente de año en año. Similarmente, las preferencias de los alumnos a favor o en contra de las materias pueden variar dramáticamente de un semestre a otro. Los estudiantes pueden tomar cursos de un departamento o de varios de ellos, aun más, pueden seleccionar materias de otras carreras. La reciente desaparición de las restricciones de créditos para cursar las asignaturas complica más la situación porque amplía el rango de materias que puede tomar un alumno. Los horarios aparecen en varias formas: matutinos, vespertinos, mixtos y todo el día. Los horarios escolares indican cuando un maestro y un alumno deben ir a cada clase. Todas las escuelas tienen un horario de cursos que dirige lo que cada estudiante y catedrático debe hacer en determinado momento. Esta tesis versa específicamente sobre el problema de construir horarios para Institutos Tecnológicos en México.

La planeación de los horarios es un trabajo que se realiza semestralmente antes de iniciar el siguiente periodo escolar, tomando en cuenta la demanda de grupos que se ofrecerán y asignando a estos grupos: los alumnos, el aula, el catedrático y el horario correspondiente, mediante información que se obtiene de varias maneras.

El principal objetivo de este proyecto es dar una solución aproximada (aceptable) al problema de la generación de horarios de una escuela de nivel superior que se resume así:

“Dado un conjunto de departamentos y carreras, asignaturas con prerrequisitos, grupos, catedráticos con especialidad y horario definidos, aulas, laboratorios y alumnos con sus preferencias de materias; el problema es construir un condensado de horarios factible que satisfaga todas las restricciones fuertes y minimice las restricciones medias y suaves”

Las restricciones fuertes son restricciones de espacio y tiempo tal como planear dos cursos en un aula al mismo tiempo. Las restricciones medias y suaves son las preferencias de maestros y estudiantes que deberán ser satisfechas si es posible. [Elmohamed, 1998]

El problema de la generación de horarios ha sido tratado por muchos investigadores utilizando métodos que van desde la investigación de operaciones hasta disciplinas de la Inteligencia Artificial como son Algoritmos Genéticos, Recocido Simulado, Redes Neuronales, etc.

## 1.2 Planteamiento del problema

La generación de horarios es una forma particular del problema de asignación de recursos. En el caso más general, existe una serie de objetos (recursos) y una serie de procesos que necesitan ser realizados por cada uno de esos objetos. En un contexto educacional, cada estudiante debe asistir a un número predeterminado de clases y el profesor debe impartirlas durante la semana en un aula predeterminada; el estudiante, el profesor y el aula pasan aquí a ser el objeto y las clases el proceso.

### La complejidad del problema de la generación de horarios

La generación de horarios es un problema de optimización combinatorio multi-dimensional no Euclidiano con múltiples restricciones y por lo tanto difícil de resolver. [Elmohamed, 1998]

El problema de la generación de horarios está agrupado dentro de los considerados de complejidad **NP-completos**, es decir, que no pueden ser resueltos en tiempo polinomial usando un algoritmo determinístico. No se conoce un algoritmo determinístico de tiempo polinomial para resolver alguno de esos problemas y la mayoría de la gente cree que dichos algoritmos no existen [Karp, 1972].

La generación de horarios es la asignación de lapsos de tiempo a un conjunto de eventos, sujeto a las restricciones en esas asignaciones. Según Elmohamed [Elmohamed, 1998] el problema NP-completo de la elaboración de horarios es un problema para la

satisfacción de restricciones que puede ser definido de la siguiente manera:

Para una escuela dada con:

$N_p$  profesores con sus horarios y turnos,  
 $N_q$  materias con su temario y objetivos,  
 $N_x$  aulas y laboratorios con su capacidad,  
 $N_s$  estudiantes,

es requerido programar  $N_t$  parejas de:

*alumno-materia,*  
*hora-materia,*  
*maestro-materia,*  
*aula-materia*

una vez definidas las parejas anteriores se deben programar elementos de la forma *materia-maestro-hora-aula*

En un límite de  $N_t$  lapsos de tiempo para producir una programación factible. En una programación factible no se puede encontrar a un profesor, grupo o estudiante en más de un lugar al mismo tiempo y un aula no puede ser asignada con más de una clase a la vez o más estudiantes de su capacidad o un profesor programado fuera de su horario normal o con más horas frente a grupo de las reglamentarias, además de un conjunto adicional de restricciones.

### **Importancia del problema**

La administración de los recursos comprende la apertura de los grupos para inscribir a los alumnos el próximo semestre, la asignación de carga académica a los catedráticos, la utilización de aulas y laboratorios de química, producción, cómputo, electromecánica, aulas de dibujo y demás instalaciones del Instituto Tecnológico de Zacatepec (ITZ).

Cuando por errores imputables al creador de los horarios se hizo una mala planeación, se derivan los siguientes problemas:

- Alumnos que se inscriben con una carga de créditos menor a la mínima por falta de grupos suficientes.
- Alumnos que deben estar en el ITZ desde las 7:00 hrs. hasta las 20:00 hrs. varios días a la semana por abrir grupos en horarios muy separados.
- Alumnos que pierden la secuencia de materias seriadas por falta de cupo.
- Alumnos que deben permanecer 1 ó 2 semestres más en el ITZ por no prever suficientes grupos para semestres terminales.
- Catedráticos que no imparten el mínimo reglamentario de horas frente a grupo por no crearles grupos suficientes.
- Catedráticos a los que se les avisa 1 día antes del inicio de clases que impartirán una

asignatura nueva.

- Asignaturas que son impartidas por profesores que no cumplen el perfil deseado en esa asignatura.
- Choque de grupos en las aulas.
- Choque de grupos en laboratorios.
- Falta de laboratorio a algunos grupos.
- Deficiente utilización de las aulas.

La creación de un sistema que evite de manera automática caer en errores como los mencionados NO ES TAREA TRIVIAL, intentar resolver este problema involucra la elaboración de un “sistema inteligente” capaz de tomar decisiones, aprender, inferir basándose en hechos, generar nueva información y a partir de ella nuevamente volver a decidir. El sistema que se elabore debe tener la capacidad para deducir e inducir de una manera combinada.

### **Desventajas del sistema actual**

La mayoría de los problemas anteriores son ocasionados por alguno o varios de los siguientes motivos:

- Todo el proceso de cálculo y administración de los recursos es manual.
- El proceso de elaboración coincide con el cierre de semestre actual y como las personas responsables también son catedráticos deben combinar la generación de horarios con las evaluaciones finales, revisión de proyectos y entrega de calificaciones de sus grupos a los alumnos.
- La elaboración de los horarios es un proceso que se lleva semanas de labor diaria.
- Requiere la atención total de los coordinadores durante todo el tiempo de elaboración.
- Al no tener la información automatizada se tienen pérdidas de tiempo en la toma de decisiones como por ejemplo; en el préstamo de aulas no utilizadas o en la forma de compartir catedráticos.
- Se tienen fuertes restricciones debido a los horarios de los catedráticos.
- Hay fuertes dificultades cuando se deben negociar cambios de horarios porque el proceso implica mover asignaturas, buscar aulas, dejar paquetes de materias por semestre, etc.
- El estado de ánimo del coordinador influye en la elaboración de los horarios.

## **1.3 Objetivo general**

Generación de una solución aproximada del condensado de horarios para una escuela de nivel superior que satisfaga las restricciones ocasionadas por los recursos involucrados mediante la elaboración de un sistema híbrido (experto, genético y lógico restrictivo).

## 1.4 Objetivos específicos

Se tiene el diseño e implementación de un sistema que:

- Proporciona una solución aproximada aceptable al problema de la generación de horarios.
- Permite a los coordinadores delegar a la computadora la asignación de recursos.
- Ahorra tiempo en la generación de los horarios.
- Reduce el trabajo manual.
- Agiliza la toma de decisiones.
- Reduce el tiempo de elaboración de horarios.
- Facilita la modificación de información generada.
- Aumenta la eficiencia en las negociaciones entre jefes y coordinadores.
- Lleva al Instituto Tecnológico de Zacatepec (ITZ) a la modernización en este proceso.

## 1.5 Justificación del sistema

### 1.5.1 Justificación teórica

La utilización de la Inteligencia Artificial (IA) para la solución de problemas de la vida cotidiana ha sido una estrategia utilizada cuando los sistemas tradicionales de computación no proveen mecanismos suficientes para garantizar una solución adecuada, su utilización para la generación de horarios (creación de grupos, asignación de hora, asignación de aula, determinación de catedrático) presenta una alternativa ideal para comprobarlo.

### 1.5.2 Justificación metodológica

La utilización de un grupo de Sistemas Expertos que realice la asignación de los recursos para la generación de grupos y le proponga una materia determinada a un profesor según el perfil académico de ambos, reúne las características necesarias para dar la solución elegante y efectiva al problema planteado anteriormente porque:

- Se requiere del conocimiento de un experto para plantear el funcionamiento del sistema.
- El conocimiento se alimentará a una base de conocimientos que conste de varias partes.
- Para la generación de grupos de una asignatura dada, el sistema debe “razonar” para determinar cuantos de éstos se deben abrir para ofertarlos a los alumnos.

- En la asignación de nuevas materias a los catedráticos el sistema debe “determinar” quién es el más idóneo para impartirla de acuerdo al perfil de ambos: materia y maestro.

La Programación Lógica de Restricciones (PLR) es una de las técnicas que maneja de manera natural información sometida a ciertas restricciones y fue seleccionada para seguir trabajando en la resolución del problema porque:

- Los horarios deben satisfacer ciertas restricciones como por ejemplo: no se deben generar mas grupos de los necesarios, no puede haber dos materias con el mismo grupo, no deben existir grupos del mismo paquete a la misma hora, etc.
- Cada paquete de un semestre es un objeto que por sí mismo posee ciertas cualidades y restricciones por ejemplo: no debe tener horas sin asignatura entre la hora inicial y final, existen materias que se deben impartir de una hora diaria otras deben ser impartidas de 2 horas diarias, etc.
- La solución presentada por la PLR inicia con datos aleatorios de un horario, después se verifica si las restricciones se cumplen o no, si se cumplen se guarda el dato, si no, se genera un nuevo dato aleatorio.

Los Algoritmos Genéticos utilizados aquí para asignar aula a los grupos cumplen con la metodología de la Computación Evolutiva porque:

- Se tienen datos iniciales de los grupos (materia-hora).
- Se genera una población inicial que son  $n$  versiones de un condensado de horarios sin aula.
- Esta población inicial es evaluada, se seleccionan los condensados más adecuados y se eliminan las versiones menos adecuadas.
- Se realizan cruzamientos, mutaciones, clonaciones y se crea la nueva generación de horarios.
- Se repite el proceso hasta llegar a una utilización eficiente de las aulas.

Por lo anterior se pueden observar las ventajas de utilizar un sistema inteligente sobre uno convencional para solucionar el problema planteado.

Una vez definido el modelo para solucionar el problema de la asignación de horarios para las asignaturas, aulas y catedráticos de una institución de educación superior, será fácil extrapolarlo para solucionar algunos problemas referentes a la planeación de horarios de otro tipo de escuelas de nivel medio superior.

### 1.5.3 Justificación práctica

La creación de un sistema que resuelva el problema de la elaboración de horarios y genere automáticamente la programación de grupos, aulas y maestros sin violar las

restricciones fuertes, minimizando las restricciones medias y suaves, optimizando la utilización de las aulas, cumpliendo con las restricciones de los profesores, asignando profesores adecuados para impartir las clases, traerá como consecuencia condiciones para mejorar el proceso de enseñanza aprendizaje.

## 1.6 Alcances y limitaciones

### Alcances.

El sistema generará un espacio de soluciones finito que satisfaga las restricciones fuertes, medias y suaves optimizando el uso de los recursos disponibles, se puede decir que el sistema optimizará los recursos cuando los horarios generados tengan las características mencionadas más adelante.

Esta tesis documenta la creación de un sistema basado en el conocimiento para solucionar el problema de la generación de horarios en el Instituto Tecnológico de Zacatepec que permite:

- Generar grupos suficientes para que todos los estudiantes se inscriban al menos con la carga mínima reglamentaria.
- Que los alumnos tengan horarios adecuados a sus necesidades y preferencias.
- Que todos los grupos tengan capacidad suficiente para todos los estudiantes que deseen inscribirse.
- Que los profesores impartan materias de acuerdo a su perfil profesional.
- Evitar los choques de grupos en aulas y laboratorios.
- Que todos los grupos tengan aula asignada desde el inicio del semestre.
- La utilización óptima de las aulas y laboratorios.
- La correcta asignación de aulas y laboratorios a los cursos.
- Realizar de manera automática la asignación de los recursos.
- Determinar el número de grupos que se deben abrir.
- Llenar las aulas con grupos sin permitir choque de horarios.
- Imprimir la lista de grupos ofertados para el semestre siguiente.
- Cuando exista el grupo de una materia sin maestro, determinar cual es el mejor para impartirla.

Tener un Sistema terminado que contenga los siguientes módulos

- ✓ Sistema administrativo para control de horarios.
- ✓ Sistema Experto para determinar la cantidad de grupos por abrir.
- ✓ Sistema Experto para sugerir materias a un catedrático.
- ✓ Sistema Experto para seleccionar los mejores catedráticos para impartir una asignatura.



- ✓ Módulo de Programación Lógica de Restricciones para generar los horarios a los paquetes de materias por semestre.
- ✓ Módulo de Algoritmo Genético para asignar aula a los grupos.

Cada módulo tendrá:

- Interfaces de usuario amigables.
- Tiempos de ejecución rápidos (independientemente de la máquina objetivo).
- Reglas de inferencia para generar resultados (sólo los Sistemas Expertos).
- Búsquedas heurísticas o aleatorias según el caso.
- Procesos para realizar inducción y deducción (sólo los Sistemas Expertos).
- Manejo de incertidumbre (sólo los Sistemas Expertos).
- Un módulo de explicaciones en lenguaje natural.

### **Limitaciones.**

El sistema está diseñado para generar horarios de una escuela de nivel superior donde el coordinador de cada carrera es el responsable de la generación de sus horarios, controlar sus profesores, impartir cierto número de materias y utilizar cierta cantidad de aulas y laboratorios.

Trabaja exclusivamente en lenguaje español.

Las reglas de la base de conocimiento tienen un formato predeterminado por una sintaxis definida para el sistema.

El sistema está elaborado para trabajar en microcomputadoras compatibles con IBM.

No maneja los horarios individuales de los alumnos en el proceso de inscripción, esta actividad deberá ser realizada por los sistemas del departamento de servicios escolares.

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2.1 El estado del arte

##### 2.1.1 Antecedentes

El problema de la generación de horarios ha sido atacado de varias maneras a nivel mundial, inclusive existe un grupo internacional que convoca al congreso llamado PATATxx (*Practice and Theory of Automated Timetabling*) donde xx es el año del congreso en cuestión.

La ASAP (*Automated scheduling, optimization and planing group*) es otro organismo cuyo fin principal es conducir investigación en modelos, heurísticas y algoritmos para la producción automática de soluciones de alta calidad a una gran variedad de problemas de planeación y optimización del mundo real, incluyendo elaboración de horarios, logística, asignación de espacio y mantenimiento de inventarios.

##### 2.1.2 Algunos métodos para resolver el Problema de la Generación de Horarios

La generación de horarios académicos ha sido acometida por muchos investigadores, básicamente dentro del área de la Investigación de Operaciones.

###### 2.1.2.1 Investigación de Operaciones

El algoritmo para resolver problemas de asignación se denomina Húngaro, ya que fueron dos matemáticos Húngaros, König (1916) y Egervary (1931), los que aportaron las teorías que sirven de base a este método [Daellenbach, 1987].

Este problema se presenta en diversos casos de toma de decisiones. Los problemas típicos de asignación implican asignar tareas a máquinas, trabajadores a tareas y proyectos, personal de ventas a territorios de ventas, contratos a licitaciones, horarios de aulas disponibles a horarios de maestros disponibles, enfermos a camas disponibles del hospital, cierto número de empleados a cierta cantidad de puestos en una empresa, aviones a destinos aéreos, entre otros.

Una característica importante de los problemas de asignación es que se asigna un trabajador, una tarea,..., a una sola máquina, proyecto,...En particular se busca el conjunto de asignaciones que optimice el objetivo planteado, tal como minimizar costos, minimizar tiempo o maximizar utilidad.

La formulación del problema general de asignación es:

$$\text{MIN } W = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij}$$

Sujeta a:

$$\sum_{j=1}^n X_{ij} = 1 \quad i = 1, \dots, m$$

$$\sum_{i=1}^m X_{ij} = 1 \quad j = 1, \dots, n$$

$$X_{ij} \geq 0 \quad i = 1, \dots, m; \quad j = 1, \dots, n$$

Corresponde a un caso especial del problema del transporte en el cual las variables  $X_{ij}$  sólo pueden tomar el valor 0 ó 1; tomar el valor 1 si el origen  $i$  se hace corresponder al destino  $j$  y 0 en caso contrario.

### **El Algoritmo Húngaro**

Este algoritmo se usa para resolver problemas de minimización, ya que es más eficaz que el empleado para resolver el problema del transporte por el alto grado de degeneración que pueden presentar los problemas de asignación. Las fases para la aplicación del método Húngaro son:

**Paso 1:** Encontrar primero el elemento más pequeño en cada fila de la matriz de costos  $m \times m$ ; se debe construir una nueva matriz al restar de cada costo el costo mínimo de cada fila; encontrar para esta nueva matriz, el costo mínimo en cada columna. A continuación se debe construir una nueva matriz (denominada matriz de costos reducidos) al restar de cada costo el costo mínimo de su columna.

**Paso 2:** (En algunos pocos textos este paso se atribuye a Flood). Consiste en trazar el número mínimo de líneas (horizontales o verticales o ambas únicamente de esas maneras) que se requieren para cubrir todos los ceros en la matriz de costos reducidos; si se necesitan  $m$  líneas para cubrir todos los ceros, se tiene una solución óptima entre los ceros cubiertos de la matriz. Si se requieren menos de  $m$  líneas para cubrir todos los ceros, se debe continuar con el paso 3. El número de líneas para cubrir los ceros es igual a la cantidad de asignaciones que hasta ese momento se pueden realizar.

**Paso 3:** Encontrar el menor elemento diferente de cero (llamado  $k$ ) en la matriz de costos reducidos, que no está cubierto por las líneas dibujadas en el paso 2; a continuación se debe restar  $k$  de cada elemento no cubierto de la matriz de costos reducidos y sumar  $k$  a cada elemento de la matriz de costos reducidos cubierto por dos líneas (intersecciones). Por último se debe regresar al paso 2.

#### **Notas:**

1. Para resolver un problema de asignación en el cual la meta es maximizar la función objetivo, se debe multiplicar la matriz de ganancias por menos uno (-1) y resolver el problema como uno de minimización.
2. Si el número de filas y de columnas en la matriz de costos son diferentes, el problema de asignación está desbalanceado. El método Húngaro puede proporcionar una solución incorrecta si el problema no está balanceado; debido a lo anterior, se debe balancear primero cualquier problema de asignación (añadiendo filas o columnas ficticias) antes de resolverlo mediante el método Húngaro.
3. En un problema grande, puede resultar difícil obtener el mínimo número de filas necesarias para cubrir todos los ceros en la matriz de costos actual. Se puede demostrar que si se necesitan  $j$  líneas para cubrir todos los ceros, entonces se pueden asignar solamente  $j$  trabajos a un costo cero en la matriz actual; esto explica por qué termina cuando se necesitan  $m$  líneas.

También han aparecido métodos no convencionales que atacan el problema desde diferentes ángulos. Los más usuales son:

- ✓ Algoritmos Genéticos
- ✓ Redes Neuronales
- ✓ Recocido Simulado
- ✓ Búsquedas Tabú
- ✓ Sistemas Expertos
- ✓ Programación Lógica de Restricciones
- ✓ Grafos Coloreados
- ✓ Operadores Recombinatorios Especializados

### 2.1.2.2 Algoritmos Genéticos

Los Algoritmos Genéticos han sido utilizados satisfactoriamente para planear exámenes y cursos en cierto número de casos. [Corne, 1993] utiliza una buena aproximación tradicional donde cada gene representa la hora en la cual se presentará un examen o curso. Con operadores de cruzamiento y mutación como los mencionados anteriormente, agrega sustanciosos ahorros de tiempo al método de Corne. [Paechter, 1994] da una aproximación diferente donde el gene de cada examen no solo especifica “el cuando” sino que también busca un nuevo horario del examen si después del cruzamiento el curso esta causando un conflicto. Si el examen no puede ser programado en alguna hora entonces no se agrega a la planeación con la finalidad de no tener horarios infactibles.

[Burke,1994] incorpora dominios de conocimiento específico para asegurar que ninguna de las soluciones candidato es no factible. En particular una representación directa se utilizada que incluye no sólo el cuándo sino también el dónde un curso o examen deberán ser tomados permitiendo al programador de horarios detener la ejecución del algoritmo en cualquier punto y tener ya una serie de soluciones de buenos horarios factibles.

### 2.1.2.3 Programación Lógica de Restricciones

Varios autores han presentado técnicas de Programación Lógica de Restricciones para resolver problemas de asignación de horarios pero éste sigue siendo un nuevo enfoque. También han aparecido algunos usando PLR para la planeación de clase-maestro.[Yoshikawa, 1994]

Kang y White presentaron una buena aproximación estándar de PLR, programando cursos sin permitir ningun conflicto [Kang y White, 1994]. Ellos usaron una técnica de reversa equivalente al *backtracking*, con lo cual, cuando un curso no puede ser

asignado, uno similar es desasignado y el nuevo es colocado en su lugar. Esto previene una búsqueda exhaustiva completa. Esto también introduce el concepto de un delimitador de tamaño que establece un máximo en el tamaño del aula que puede usar un curso en particular. Las restricciones son priorizadas en cinco diferentes categorías, donde la categoría cinco debe ser liberada antes que la categoría cuatro y así con las demás. Este procedimiento fue utilizado para programar exitosamente los cursos de la Universidad de Ottawa.

También se ha utilizado la técnica de la Programación Lógica de Restricciones para hacer los horarios de una facultad utilizando una regla heurística para seleccionar que curso asignar primero [Fahrion y Dollanski, 1992].

#### **2.1.2.4 Redes Neuronales**

De las distintas categorías de las Redes Neuronales Artificiales, la más apropiada para resolver el problema es la familia de las Redes Neuronales de reacción basadas en neuronas multiestado con ecuaciones de teoría de campos con significado controlando las dinámicas. Las ecuaciones anteriores son usadas en lugar de los procedimientos para actualización de demanda estocásticos que han sido usados por otros esquemas para evitar quedar atorados en circuitos locales mínimos [Carrasco, 2004].

#### **2.1.2.5 Recocido Simulado**

Recocido Simulado se refiere al uso de una técnica de optimización combinatoria o discreta. Divide el problema en grupos del mismo tiempo de clase los cuales son especificados por cada departamento académico basados en la habilidad de los instructores. Funciona así:

1. El proceso asigna aulas procesando un grupo a la vez empezando con el primero.
2. El proceso continúa hasta que todos los grupos son procesados.
3. Cuando una solución es obtenida dividiendo el problema en conjuntos de horario, una mutación especial es generada con grupos traslapados. Esto podría generar una mala asignación de aula, para minimizar esto se utiliza el siguiente paso.
4. Los grupos son procesados basados en el número de aulas, este número de aulas es optimizado y se termina con las horas de clase con el mínimo de paquetes a esa hora.
5. La implementación del programa genera una base de datos con la nueva asignación de salones para el próximo semestre [Abramson, 1991].

#### **2.1.2.6 Operadores Recombinatorios Especializados**

Los operadores de cruzamiento de los Algoritmos Genéticos producen buenas

combinaciones de genes para ser pasados entre los diferentes miembros de una población. En los Algoritmos Genéticos estándar es un procedimiento similar a escoger los primeros  $n$  genes del primer padre y los últimos del segundo padre, este procedimiento no es útil en situaciones donde existen fuertes restricciones sobre la forma de los miembros de la población, aquí un operador violará esos requerimientos. El operador de recombinación heurístico híbrido trabaja así:

1. Los grupos no programados del periodo anterior son pasados al algoritmo.
2. Primero selecciona aquellos grupos programados en el periodo  $i$  en los padres 1 y 2.
3. Los grupos extra son seleccionados de aquellos programados en el periodo  $i$  ya sea del padre 1 o del padre 2 o de los dejados del periodo  $i - 1$ .
4. Cualquiera de los grupos no programados es pasado al periodo  $i+1$ .
5. Sólo los grupos que no crean conflicto con aquellos ya programados para este periodo pueden ser seleccionados [Wilke, 2002].

#### 2.1.2.7 El Sistema Experto Basado en Reglas

Los Sistemas Basados en Reglas proporcionan un buen punto de referencia para saber que tan buenos son otros métodos en comparación a esta técnica estándar. Varias de estas reglas son usadas para proporcionar sensibles elecciones para los movimientos en el algoritmo de Recocido Simulado en lugar de seleccionar cambios de manera aleatoria, se ha usado este sistema como un preprocesador para generar una buena aproximación inicial y darla al Recocido Simulado.

El Sistema Experto Basado en Reglas consiste en un número de reglas (o heurísticas) y recursión convencional para ayudar en la generación de asignación de clases. Las estructuras de datos o componentes del sistema son:

- Una matriz de distancias entre los departamentos académicos y los edificios que serán utilizados.
- Estructura de los datos de las materias para cada grupo programado en el campo.
- Estructura para las aulas (independientemente del tipo de aula) involucradas en el proceso.
- Horarios para cada periodo de tiempo, guardando cuales están disponibles y cuales ya fueron ocupadas.
- La estructura de los departamentos.
- Las preferencias de los estudiantes indicando las clases requeridas independientemente del semestre que sean las materias [Gervás, 1999].

### 2.1.3 Observaciones realizadas a algunos paquetes generadores de horarios

**Nombre : DeMISt**

**Descripción:** Puede describirse como un paquete que realiza bien las operaciones de casa (la propia universidad que lo creó), pero salió tan costoso como si hubieran comprado un paquete [Demist, 2000].

**Proveedor :** DeMISt University.

**KnowTIS**

Es un proyecto iniciado por la universidad de Leeds. En la literatura aparece como un paquete que es mejor para la generación de Grandes Horarios para una universidad que para un departamento. Utiliza el beneficio de gran cantidad de información centralizada para la optimización de recursos. [Schedul, 2000]

Leeds University.

**STARS**

STARS(Space, Time and Resources Management) Igual que el anterior corre mejor en la elaboración de horarios de buen tamaño, que en una aplicación departamental De Montfort University.

**Meeting maker / Office Talk**

Es un generador de clases y acciones diarias, es bueno cuando no es útil el horario de los alumnos, ya que proporciona únicamente los espacios para las clases [Gervás, 1999] Varios proveedores.

**GP-UNTIS**

Es un paquete australiano que funciona perfectamente para los requerimientos de las escuelas secundarias  
Timetabling Services UK.

**Syllabus Plus**

Es una buena opción para uso departamental, cumple los requerimientos de una universidad para la generación de horarios, no trabaja a nivel de estudiantes, pero si lo hace con grupos de ellos, no es factible para la generación de Grandes Horarios Scientia LTD.

**Facility Timetabler**

Es una buena opción para uso departamental y para Grandes Horarios tiene interfaz con los sistemas de alumnos y permite la creación de horarios individuales, permite el uso



centralizado de las aulas para una buena optimización.

Facility Systems.

### **Mimosa for Windows**

Es un software general de planificación de la enseñanza y de programación de horarios. La aplicación utiliza todas las propiedades de Windows eficientemente, es rápida, tiene gran capacidad y lo más importante: es tan flexible que puede aplicarse en todo el tipo de escuelas, desde la escuela primaria a las universidades, desde las escuelas vocacionales a las academias militares.

Se podría decir que este producto cubriría todas las necesidades de la institución pero han existido algunos inconvenientes ya que no cubre con las necesidades del departamento, tan solo da ayuda al usuario para que pueda generar sus horarios, además el tener tantos botones en pantalla (200) podrían confundir al usuario y realizar horarios erróneos y no se resolvería el problema [Nielsoft, 2000].

Mimosa Software Oy Finlandia.

### **GestHor BSP**

Es un producto informático para resolver problemas de horarios escolares en escuelas básicas, escuelas secundarias y escuelas profesionales.

Tómese en cuenta que la mayoría de los sistemas mencionados se enfoca solo a la asignación aula – materia, y no abarca lo que es asignación materia – profesor.

## **2.2 Los horarios.**

Existe mucha literatura referente a los horarios con una gran variedad de subtítulos y diferentes problemas. Una gran variedad de algoritmos han sido propuestos, empleando metodologías y heurísticas de búsqueda, produciendo horarios completos o sólo ayudando durante el proceso. Una de las grandes preguntas para la elaboración de horarios es que si éstos pueden ser generales, es decir ¿se puede encontrar un algoritmo o método, suficientemente general, que produzca buenos horarios para todos los problemas y para todas las restricciones? La alternativa, desde luego, es continuar desarrollando algoritmos especializados aislados cada vez que un problema de horarios aparece.

Al trabajar con una solución para el problema de los horarios se deben contestar las siguientes preguntas. ¿Todas las escuelas de nivel superior elaboran los horarios de la misma manera? ¿Son grandes o pequeñas las diferencias en la forma de elaborar los horarios? ¿Qué tan necesaria es la ayuda de la computadora? ¿Sólo se requiere un paquete general para la elaboración de los horarios? ¿Qué funcionalidad debe tener

dicho paquete? ¿Qué tan grande y complejo es el problema de elaboración de horarios? ¿Qué propiedades debe poseer un horario aceptable para que una institución lo use?

Se han realizado algunas encuestas para conocer si las escuelas de nivel superior tienen sistemas computarizados para la elaboración de horarios, si no los tienen, si los necesitaban y qué características debería poseer. El estudio concluyó que existe una gran demanda para tal sistema, que se requiere un alto grado de asistencia automatizada y que la mayoría de los elaboradores de horarios no estaban muy contentos con sus sistemas actuales [Comm y Mathaisel, 1988].

[Dowland y Lim, 1982] consideraron el problema de la elaboración de horarios pero proporcionaron alternativas en la encuesta de la utilidad de sistemas automatizados. [Junginger, 1986] establece que “Aunque los horarios producidos requieren modificación, las escuelas están contentas con los resultados. El tiempo ahorrado fue considerable, arriba del 75%”. Dowland, por otro lado, concluye que “Es muy difícil juzgar el éxito actual de los paquetes para la elaboración de horarios, pero el uso muy limitado de ellos sugiere que sus ventajas son limitadas”. Claramente ningún diseñador de sistemas puede garantizar la idea de que su producto será bienvenido con los brazos abiertos.

[Pendlebury, 1986] encuestó a las universidades politécnicas británicas con respecto a la elaboración de horarios de cursos. Doce de las treinta que contestaron la encuesta utilizan elaboración de horarios por computadora, (siete a gran escala y cuatro lo hacen en pequeña escala). Es interesante notar que Pendlebury encontró que la mayoría de los cursos se basan en clases que son muy independientes unas de otras y en la mayoría de las instituciones encuestadas los departamentos individuales son responsables de producir sus propios horarios.

### **2.2.1 El problema de la elaboración de horarios**

La elaboración de horarios es el acto de programar cursos a períodos de tiempo, asignar un aula a esos cursos y asignar un catedrático a dicho curso.

A menudo se hacen afirmaciones acerca de la naturaleza del problema que pueden permitir encontrar una solución más eficientemente. Sin embargo, cuando se trata de construir un sistema general, se está limitado a que las afirmaciones son el mínimo común denominador de todos los posibles escenarios.

Una información implícita que ha sido hecha a menudo es la independencia entre unidades académicas separadas. Algunas universidades programan preguntando a sus departamentos los horarios de sus propios cursos. Una unión central de dichos horarios es entonces utilizada para asegurar que todas las restricciones relevantes son cumplidas. En muchos casos esta es la forma normal en que es producido el horario de los cursos,

aunque algunos departamentos desarrollan su propio *software* para la programación de cursos.

En varias universidades los alumnos pueden tomar algunos de sus cursos en departamentos diferentes al suyo. Obviamente la independencia no debe ser asumida generalmente debido a que puede haber diferencia en la programación de los cursos de cada departamento. En el Tecnológico de Zacatepec, por ejemplo, la probabilidad promedio de que un curso tenga conflicto con uno de otro departamento es 1/650, mientras que la probabilidad de que dos cursos del mismo departamento tengan conflicto es 2/250, notándose una gran diferencia en el resultado.

Dificultades encontradas por los responsables de la generación de horarios

Algunas dificultades experimentadas por los departamentos y facultades son mencionadas aquí:

- Los programas académicos se hacen cada vez más flexibles (los estudiantes seleccionan qué asignaturas tomar, cuándo, dónde y con quién).
- Algunos departamentos tienen asignaturas para ofrecer pero no tienen muchos alumnos.
- Los espacios para ofrecer las materias son cada vez menos y el problema se agrava cuando los alumnos no quieren asistir en las tardes o los sábados.
- El *staff* de profesores no siempre está disponible para impartir clases por múltiples razones.
- Los alumnos pueden solicitar los grupos del próximo semestre desde antes de que inicie, pero al llegar el momento de inscribirse no lo hacen.
- Los departamentos están ansiosos por ofrecer una gran variedad de grupos teniendo los recursos, pero esto puede ocasionar problemas.
- Existe una gran variedad de reglas al momento de asignar las aulas (pertenencia a las carreras, prioridades de profesores, etc.).
- Como no hay un costo asociado a la utilización de un aula, y no hay sanciones por reservarla y no utilizarla, los departamentos académicos no tienen un buen motivo para hacer una correcta utilización del espacio.
- La falta de recursos ocasiona el mayor de los problemas porque obliga a la optimización en la planeación.

## **Cursos y estudiantes**

El problema de la elaboración de horarios está relacionado esencialmente con el número de eventos que requieren ser programados, en este caso los cursos para los estudiantes. Claramente el tamaño del problema varía entre los diferentes centros de estudio, desde pequeños problemas con alrededor de 50 a 100 cursos, hasta los 500 o 1000 cursos que

programar. Esta variación en el tamaño afectará el tiempo requerido para resolver el problema. Un problema pequeño puede ser resuelto casi de manera óptima mientras que en uno grande, se deben usar búsquedas heurísticas u otras formas de acotar el espacio de búsqueda de una solución en un periodo de tiempo razonable.

Se debe esperar que el número de cursos y el número de alumnos estén correlacionados. El número de cursos ofrecidos y desde luego programados debe ser calculado por extrapolación del promedio de 5 o 6 cursos por alumno y con cupo promedio de 30 alumnos por curso, obviamente algunos alumnos se inscribirán en 7 u 8 cursos por semestre, mientras que los cursos tendrán inscritos entre 20 y 40 alumnos. Los alumnos dependiendo del semestre en que se encuentran inscritos, deben matricularse a ciertos cursos específicos para ese semestre, dichos cursos deben programarse de manera que los alumnos tengan la menor cantidad de tiempo entre el primer y el último curso del día permitiéndoles con esto realizar actividades complementarias.

En el Tecnológico de Zacatepec, para una carrera con 900 alumnos se requieren ofertar aproximadamente entre 140 y 160 cursos, además los estudiantes pueden matricularse en cursos ofrecidos por otros departamentos (esto amplía el rango de posibilidades para los alumnos).

### **Los cursos y los departamentos**

El promedio de los cursos ofrecidos por cada departamento varía considerablemente entre las diferentes escuelas de nivel superior. La razón de esta variabilidad puede ser vista como un indicador de la naturaleza de la complejidad de la planeación de horarios.

Cada departamento ofrece cursos a los estudiantes de las carreras a las que da soporte directo, pero además debe ofrecer cursos para todas las otras carreras de la institución que lo requieran, esto es, si un departamento X solicita la apertura de determinado número de cursos a un departamento Y, el departamento Y debe ofertar los cursos solicitados por X para los alumnos matriculados en el departamento X, aunado a esto, algunos cursos ofertados por el departamento Y tienen compatibilidad con cursos ofrecidos por X y por lo tanto los estudiantes de Y y X pueden ingresar indistintamente en cualquiera de los cursos.

### **Los cursos, los semestres y las horas**

Una de las afirmaciones que se pueden hacer en las instituciones de nivel superior es que los cursos generalmente inician durante cierto periodo de tiempo y pueden tener duración semestral (la gran mayoría), trimestral y sin fechas predeterminadas (algunos posgrados por ejemplo). En general los cursos empiezan a las 7:00 A.M. y terminan, dependiendo del lugar a las 20:00 o a las 21:00 horas, de lunes a sábado. Aunque esto permite tener un amplio margen de horas para programar los cursos, esto hace también

aun más difícil la elaboración de los horarios porque aumenta el rango de posibles combinaciones donde programar cursos.

Los cursos se pueden programar en periodos de tiempo muy variados, aunque la mayoría de los programadores de horarios siguen los siguientes patrones:

- Curso diario de 1 hora.
- Curso de dos horas en días saltados (lunes y miércoles, jueves y viernes, miércoles y viernes).
- Curso de dos horas diarias 3 días por semana (lunes, miércoles y viernes), (lunes, martes y jueves) o (martes, jueves y viernes).
- Cursos de cuatro horas en un solo día por semana.
- En el peor de los casos se programan cursos con la combinación de algunos de los anteriores.

Dependiendo del curso algunos tienen duración de cuatro horas teóricas semanales, otros combinan cuatro horas de teoría y dos de práctica y algunos, excepcionalmente, son programados con cuatro horas de práctica y dos de teoría. Las horas de práctica generalmente se realizan en laboratorios o prácticas de campo, agregando esto un nuevo parámetro para la complejidad de la elaboración de horarios.

### **Los cursos y las aulas**

Las aulas son el espacio físico donde se impartirán los cursos, pueden ser salones de clase, laboratorios, salones de exposición, aulas para aplicación de exámenes, aulas para actividades deportivas, musicales o recreativas, cada aula tiene capacidad para que cierto número definido de estudiantes reciba sin problema unidades de enseñanza aprendizaje.

En una encuesta realizada por la Universidad de Nottingham [Burke y Elliman, 1998] se definió que la programación de los cursos en las aulas es uno de los mayores problemas a los que se enfrentan los responsables de la elaboración de horarios. Esto proviene de dos problemas mayores. El primer problema es la escasez de aulas disponibles ya que en algunas de las escuelas de nivel superior el crecimiento de la población estudiantil ha sido mayor que el crecimiento del número de aulas disponibles para la enseñanza. El segundo problema ocurre cuando se programa un curso en más de un aula o salón de clase.

La ocupación de las aulas es variable de acuerdo al departamento para el que sea asignada, algunos prefieren utilizarlas al máximo en el turno matutino, otros en el turno vespertino tienen ocupación total, algunos las utilizan exhaustivamente de lunes a jueves, unos departamentos las llenan los viernes y sábados y otros tienen que utilizarlas todo el día de lunes a sábado, inclusive solicitan prestadas a los otros departamentos aquellos espacios que no son utilizados en algún día y hora a la semana.

Para el Tecnológico de Zacatepec el recurso llamado las aulas debe ser optimizado en una buena planeación de horarios, esto es, deben ser utilizadas al máximo todos los días de la semana, desde las 7:00 hasta las 21:00 de lunes viernes y los sábados hasta las 15:00 horas.

### **Tipos de cursos**

Una de las cosas a tomarse en cuenta durante la elaboración de los horarios es el tipo de curso a programar:

- Teóricos como los cursos matemáticos, cursos de administración, etc.
- Teórico prácticos como los cursos de electrónica, sistemas digitales.
- Lenguajes de programación.
- Con prácticas de campo que deben salir al aire libre.
- Con prácticas de química.

Otra consideración es el tipo de aula requerida por los diferentes tipos de cursos:

- cursos que requieren medios audiovisuales simples (pizarrón y gis),
- algunos cursos adicionalmente requieren proyectores de acetatos y salas oscuras,
- cursos que requieren equipo de cómputo, equipo de proyección, equipo de video y televisión para el desarrollo de la cátedra etc.,
- cursos que requieren laboratorios con computadoras,
- que requieren equipos de medición electrónica, generadores de señales, enchufes eléctricos y
- cursos que requieren mesas, tomas eléctricas y equipos para conexión de redes.

### **Los cursos y los catedráticos**

Definitivamente ésta es una de las principales consideraciones que tienen en cuenta los programadores de horarios, la asignación de los cursos a los catedráticos, porque es un proceso que no puede realizarse de manera aleatoria, es un proceso delicado porque de esto depende la efectividad del proceso de enseñanza–aprendizaje.

Los catedráticos tienen características individuales que los hacen únicos independientemente de la profesión que hayan estudiado. Tienen intereses diferentes, métodos diferentes, áreas de preferencia diferentes, perfil profesiográfico distinto, tienen actividades extralaborales variadas así que:

¿Cómo asignarle un curso a un catedrático para que logre un óptimo resultado en el

proceso de enseñanza-aprendizaje?

¿Cómo garantizar que el curso asignado a un profesor será bien impartido?

¿Cómo saber que el profesor está capacitado para ofrecer un buen curso?

¿Cuáles son los requisitos que debe reunir un profesor para impartir cierto curso?

¿Dependen sólo de la profesión los cursos que puede impartir?

¿Cómo conocer de antemano las preferencias de los profesores?

¿El nivel de estudio de los profesores les capacita para impartir cualquier curso?

¿Deben los profesores ser guiados a una especialidad por los programadores de horarios?

Las respuestas a todas estas preguntas incorporadas en un sistema generador de horarios

¿permitirán hacer una buena asignación de catedrático a un curso?

### **El condensado de horarios**

Es un documento emitido cada semestre por los responsables de la elaboración de los horarios que contiene la información requerida por los alumnos, jefes de departamento y catedráticos para conocer la distribución de las materias, los grupos, las aulas, las horas y los catedráticos que impartirán los cursos, en otras palabras, el condensado de materias es el documento final de los HORARIOS que marcará las acciones de todos el semestre actual.

## **2.3 El proceso de la elaboración de horarios**

El proceso de la elaboración de horarios abarca cada acción que es requerida para crear el horario final, completo y correcto de los datos iniciales. Esto incluye recibir, validar y dar formato a los datos, generar la versión inicial, hacer las correcciones y luego producir el condensado final de horarios. El proceso de la generación de horarios abarca:

- la generación del dato de cuantos cursos se van a ofertar
- asignación de horario a esos cursos
- proporcionar aulas a los cursos
- designar profesores para los cursos definidos

la parte difícil de esto es generar los horarios con los recursos disponibles en cada departamento, cada etapa tiene sus restricciones (fuertes y suaves), los recursos no son inagotables.

### **El uso de la computadora**

En una encuesta realizada por la Universidad de Nottingham [Burke y Elliman, 1998] referente a la forma de elaborar horarios en las demás universidades se encontró que un

58% de ellas utilizan la computadora en alguna etapa del proceso, de éstas, el 21% tiene la facilidad de realizar la programación, aunque generalmente requieren captura manual de información y conocimientos previos del problema particular del proceso. El 1% tienen paquetes comerciales y el 14% tienen paquetes hechos en casa para generar horarios y el 2% aseguran que están desarrollando o comprando un paquete.

Dado el tamaño de los horarios de dichas universidades es sorprendente que muchas no tengan alguna forma de automatización. Esto se debe, en la mayoría de los casos, a que los horarios no cambian significativamente de un año a otro. Sin embargo, de las universidades que no programan horarios por computadora, la mitad usa el horario del año pasado y la otra mitad empiezan desde cero cada vez. Cuando no se usa el horario del año pasado, generalmente toma 4 semanas elaborar uno nuevo. En la mayoría de los casos donde se usa la computadora, generalmente no se toma en cuenta el horario del año anterior.

### **Causas de cambio de los horarios año tras año**

Para que un horario pueda ser usado año tras año, debe ser lo suficientemente robusto para que los cambios que ocurran en los datos de entrada produzcan sólo una pequeña alteración o modificación en el condensado final de horarios. El uso del condensado del año pasado depende en la cantidad de cambios que deben hacerse y esto a su vez depende de cuanto cambie la cantidad de grupos que deben ofertarse para cada asignatura y esto depende de la cantidad de alumnos que pueden cursar una asignatura, ahora bien, la cantidad de alumnos está en función de los índices de aprobación y reprobación del periodo, de los alumnos que por alguna razón no cursaron una asignatura y de la cantidad de grupos ofertados el periodo pasado.

Si cambia la cantidad de horas frente a grupo asignada a un departamento entonces cambiará el condensado de horarios con respecto al del año pasado, porque eso afecta directamente en la cantidad de cursos que podrá ofertarse este periodo, este es uno de los recursos más variables que influyen para hacer cambios en los condensados año tras año.

La redistribución de aulas y laboratorios (que no se realiza a menudo) llega también a influir en los cambios.

El cambio de horario matutino a vespertino (o viceversa) solicitado por un catedrático que es el especialista en impartir cierto curso ocasiona los cambios del condensado de horarios del año anterior al nuevo.



## CAPÍTULO 3

### TÉCNICAS DE INTELIGENCIA ARTIFICIAL UTILIZADAS PARA LA GENERACIÓN DE HORARIOS

#### 3.1 Sistemas Expertos (SE)

Bajo el término de Sistemas Expertos se entiende un nuevo tipo de *software* que imita el comportamiento de un experto humano en la solución de un problema. Un SE puede almacenar conocimientos de expertos para un campo determinado y solucionar un problema mediante razonamiento lógico de conclusiones.

Los **Sistemas Expertos** son sistemas que buscan modelar el conocimiento y procedimientos usados por un experto humano para resolver un problema determinado[Sánchez, 1999].

¿Un Sistema Experto utiliza técnicas de la Inteligencia Artificial (I.A.)?

Los Sistemas Expertos son uno de los puntos que componen las investigaciones en el campo de la I.A. Un sistema de computadoras que trabaje con técnicas de IA deberá estar en situación de combinar información de forma “inteligente”, alcanzar conclusiones y justificarlas (al igual que el resultado final). Los Sistemas Expertos son una expresión de los sistemas basados en el conocimiento. Con la aplicación de técnicas de Inteligencia Artificial finaliza la transición del procesamiento de datos al procesamiento de conocimientos.

Un Sistema Experto es un sistema informático que simula el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción de un experto humano en una determinada rama o dominio. Estas características le permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, aprender de la experiencia y los datos existentes, comunicarse con humanos, explicar el porqué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior.

"La incorporación dentro de un sistema de computadoras de un componente basado en el conocimiento, correspondiente a una habilidad experta, de tal forma que el sistema pueda ofrecer asesoramiento inteligente o tomar una decisión inteligente sobre una función del proceso. Una característica adicional deseable, que muchos consideran fundamental, es la capacidad del sistema, si se le solicita, de justificar su propia línea de razonamiento de un modo directamente inteligible para el interrogador. El estilo adoptado para alcanzar estas características es la programación basada en reglas" [Waterman, 1986].

### 3.1.1 Función de un Sistema Experto

La función de un Sistema Experto es aportar soluciones a problemas de un dominio específico, imitando la forma de razonamiento de los seres humanos es decir, el Sistema Experto debe ser capaz de mostrar soluciones inteligentes. [Giarratamo, 1998]

Acceder a los conocimientos adquiridos por experiencia es lo más difícil, ya que los expertos, al igual que otras personas, apenas los reconocen como tales. Son buscados con mucho esfuerzo y cuidado siendo descubiertos de uno en uno, poco a poco.

#### Ventajas del Sistema Experto.

<b>Experto Humano.</b>	<b>Experto Artificial.</b>
Mortal	Permanente
Difícil de documentar.	Fácil de documentar.
Difícil de transferir.	Fácil de transferir.
Impredecible.	Consistente.
Más caro.	Menos caro.

#### Ventajas del Experto Humano.

<b>Experto Humano.</b>	<b>Experto Artificial.</b>
Creativo.	No creativo.
Adaptable.	Requiere mantenimiento.
Experiencia sensorial.	Entrada simbólica.
Conocimiento de sentido común.	Conocimiento técnico.

### 3.1.2 Arquitectura de Sistemas Expertos [Gines, 1986] [Giarratamo, 1998]

La arquitectura general de un SE (Fig. 3.1) está compuesta fundamentalmente de:

- **Base de Conocimiento:** Representa el conocimiento adquirido del experto humano incluyendo experiencia, sentido común e intuición.
- **Motor de Inferencia:** Decide que se debe aplicar, que se debe hacer, resuelve conflictos nuevos, hace uso de la base del conocimiento. Proporciona mecanismos de razonamiento, búsqueda, aprendizaje, etc.

- **Interfaz Usuario:** Facilita la comunicación entre el Sistema Experto y el usuario que realiza las consultas.
- **Subsistema de Adquisición:** Mejor conocido como la Interfaz de Experto, permite ingresar el conocimiento a la Base de Conocimientos por parte del humano experto en algún dominio de conocimiento.
- **Subsistema de explicaciones:** Permite dar a conocer los resultados obtenidos por el Sistema Experto.

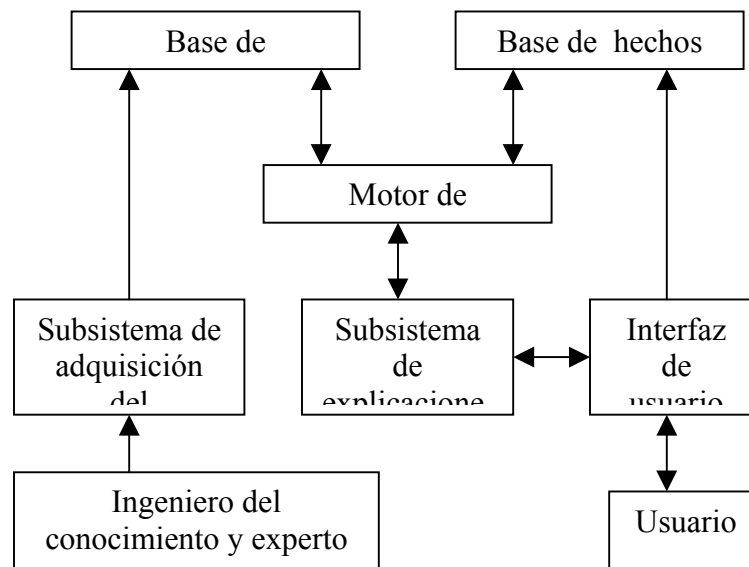


Figura 3.1 Arquitectura de un Sistema Experto

### 3.1.3 La base de conocimiento

El enfoque más común para representar el dominio del conocimiento que se requiere para un SE son las reglas de producción. Estas pueden ser referidas como reglas “*situación-acción*” o “*if-then*”. De esta forma, con frecuencia una base de conocimientos está fraguada principalmente de reglas, las cuales son invocadas por un casamiento de patrones con las particularidades de las tareas circunstanciales que van apareciendo en la base de hechos.

Las reglas en una base de conocimientos representan el dominio de hechos y la heurística de un buen juicio de acciones a tomar cuando uno llega a situaciones específicas. La fuerza de un SE yace en el conocimiento específico del dominio del problema. Casi todos los sistemas existentes basados en reglas contienen cientos o miles de ellas y generalmente se obtienen de entrevistas con expertos durante un tiempo largo. En cualquier SE las reglas se conectan una a otra por ligas de asociación para formar

redes de reglas. Una vez que han sido ensambladas tales redes, entonces se tiene una representación de un *cuero de conocimientos* que es sustancial.

Generalmente, un experto tiene muchas reglas de criterio empíricas por lo que el soporte suele ser incompleto para la evidencia que se dispone. En tales casos una técnica es la de vincular valores numéricos a cada regla para indicar el grado de certidumbre que ésta tiene. En la operación de SE los valores de certidumbre se combinan unos a otros a través de la red, considerando también la certidumbre de los datos de entrada del problema, para llegar finalmente a un valor de certidumbre de la solución final.

### 3.1.4 El motor de inferencias

El sistema experto modela el proceso de razonamiento humano con un módulo conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos. Contrasta los hechos particulares de la base de hechos con el conocimiento contenido en la base de conocimientos para obtener conclusiones acerca del problema.

El modelo de problema - solución y sus métodos organizan y controlan las etapas que se van tomando para la resolución de problemas. Un modelo muy poderoso y común es el de encadenar las reglas `` *if-then-else*" con el fin de formar alguna línea de razonamiento. Si el encadenamiento se inicia a partir de un conjunto de condiciones y se mueve hacia alguna conclusión, el método se llama encadenamiento hacia adelante y, en su caso contrario, el encadenamiento es hacia atrás.

Un problema de encadenamiento hacia adelante, sin una apropiada heurística de poda, derivará muchas soluciones de las cuales no todas son necesarias. En el caso de encadenamiento hacia atrás, es necesario también trabajar con una heurística apropiada, con la finalidad de poder manejar submetas conjuntivas. En general, para atacar el problema de conjunción, es necesario encontrar un caso en donde todas las submetas interactuantes se satisfagan. Pero con frecuencia una búsqueda de tal naturaleza nos da una explosión combinatoria de posibilidades. Es por esta razón, que el dominio apropiado de la heurística y su esquema inferencial deberán ser encontrados para cada tipo de problemas, para que de esta manera se tenga un SE eficaz y eficiente.

### 3.1.5 ¿Por qué utilizar un Sistema Experto? [Pluss, 1999] [Sanchez, 1999]

1. Con la ayuda de un Sistema Experto, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento formal especializado".
2. Los Sistemas Expertos pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos.
3. Los Sistemas Expertos razonan pero basándose en un conocimiento adquirido y no tienen sitio para la subjetividad.

4. El uso de Sistemas Expertos es especialmente recomendado en las siguientes situaciones:
- Cuando los expertos humanos en una determinada materia son escasos.
  - En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.
  - Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

### 3.1.6 Construcción de prototipos [Turban, 1995]

En el desarrollo de Sistemas Expertos se plantean dos importantes riesgos:

1- No existen implementaciones similares que puedan servir de orientación al encargado del desarrollo en la casi totalidad de los casos.

2- En muchos puntos, los requisitos necesarios están esbozados con muy poca precisión.

El diseño y la especificación requieren una temprana determinación de la interfaz del *software* y de la funcionalidad de los componentes. En el desarrollo de Sistemas Expertos deben alterarse a menudo durante y también después de su implementación, ya que los requisitos se han ido configurando y han obtenido mayor precisión, o porque se ha descubierto que deben iniciarse otras vías de solución. Durante el desarrollo, resulta más apropiado empezar con implementaciones tipo *test* para encontrar el camino hacia una solución definitiva y para hacerlas coincidir con las necesidades del usuario.

## 3.2 Algoritmos Genéticos

### 3.2.1 Antecedentes [Medina, 2004] [Goldberg, 1989] [Davis, 1991]

El **Algoritmo Genético** es una técnica de búsqueda basada en la teoría de la **evolución de Darwin** que ha cobrado tremenda popularidad en todo el mundo durante los últimos años.

Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven al

adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los **genes** de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables se transmiten a sus descendientes cuando éste se reproduce sexualmente.

En 1859, Darwin publica su libro **«El origen de las especies»**. El libro causó una gran polémica en el mundo científico por las revolucionarias teorías en él contenidas: que las especies evolucionan acorde al medio, para adaptarse a éste. Con ello, el universo pasaba de ser una creación de Dios, estática y perfecta desde su inicio, a un conjunto de individuos en constante competición y evolución para poder perpetuar su especie en el tiempo. La existencia de una especie pasa así a ser algo dinámico; las especies se crean, evolucionan y desaparecen si no se adaptan. Para cada especie animal, la naturaleza proponía un crudo filtro: sólo los mejores, los más aptos, los que mejor se adapten al medio conseguían sobrevivir lo suficiente para llegar a la madurez, y encontrar una pareja para perpetuar sus aptitudes que le hacían más apto.

La informática ve aquí un claro proceso de optimización. Se toman los individuos mejor adaptados (mejores soluciones temporales), se cruzan (se mezclan), generando nuevos individuos (nuevas soluciones) que contendrán parte del código genético (información) de sus dos antecesores, y, por lo tanto, aunque el nuevo individuo no tenga que estar forzosamente mejor adaptado, de hecho puede que ni la probabilidad de que el nuevo individuo generado esté mejor adaptado que los padres sea alta, el promedio de la adaptación de toda la población si mejora, ya que tienden a perpetuarse y extenderse las mejores características, y a extinguirse las poco beneficiosas o perjudiciales. En los Algoritmos Genéticos se crea una nueva abstracción -la población- cuya función de coste mejorará globalmente, por lo que puede que se encuentre algún individuo con mejores características.

Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a fines de los años 60 desarrolló una técnica que permitió incorporarla a un programa. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "Planes Reproductivos", pero se hizo popular bajo el nombre "Algoritmo Genético" tras la publicación de su libro en 1975.

En esa universidad, Holland impartía un curso titulado Teoría de sistemas adaptativos. Dentro de este curso, y con una participación activa por parte de sus estudiantes, fue donde se crearon las ideas que más tarde se convertirían en los Algoritmos Genéticos. Los objetivos de su investigación fueron dos:

- ❖ Imitar los procesos adaptativos de los sistemas naturales, y
- ❖ Diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Unos 15 años más adelante, David Goldberg, actual delfín de los Algoritmos Genéticos, conoció a Holland, y se convirtió en su estudiante. Golberg era un ingeniero industrial trabajando en diseño de tuberías y fue uno de los primeros que trató de aplicar los Algoritmos Genéticos a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complejo, Goldberg consiguió lo que quería, escribiendo un Algoritmo Genético en una computadora personal Apple II. Estas y otras aplicaciones creadas por estudiantes de Holland convirtieron a los Algoritmos Genéticos en un campo con base suficiente como para celebrar la primera conferencia en 1985, ICGA'85. Tal conferencia se sigue celebrando bianualmente [Beasley, 1993]

Una definición bastante completa de un Algoritmo Genético es la propuesta por John Koza: "es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud " [Koza, 1992].

### 3.2.2 Definición de A.G.

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas [Goldberg, 1989]

Un **Algoritmo Genético** consiste en una función matemática que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación.

Los **Algoritmos Genéticos** presentan ventajas sobre otros algoritmos que los hacen preferibles para determinado tipo de aplicaciones. Algunas de las características de los Algoritmos Genéticos son:

- Son algoritmos estocásticos. Dos ejecuciones distintas pueden dar dos soluciones distintas. Esto es útil por el hecho de que hay gran cantidad de isómeros que

corresponden a soluciones válidas, por lo que es interesante que distintas ejecuciones puedan dar isómeros distintos.

- Son algoritmos de búsqueda múltiple, luego dan varias soluciones. Aunque habitualmente la energía de los individuos de la población final es similar, los individuos suelen ser distintos entre sí. Con el modelo de paralelización empleado - genético multipoblacional- la probabilidad de obtener muchas soluciones distintas es más alta todavía.
- Son los algoritmos que hacen una barrida mayor al subespacio de posibles soluciones válidas. De hecho, se considera que, de todos los algoritmos de optimización estocásticos, los Algoritmos Genéticos son de los más exploratorios disponibles.

### 3.2.3 Ventajas y Desventajas [Bull, 1993]

- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se usan para problemas de optimización de una función objetivo- resultan menos afectados por los óptimos locales (falsas soluciones) que las técnicas tradicionales.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.
- La mayoría de las otras técnicas de optimización son también estocásticas.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, número de generaciones, etc.-.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

### 3.2.4 ¿Cómo saber si es posible usar un Algoritmo Genético?

La aplicación más común de los **Algoritmos Genéticos** ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas pudieran ser apropiados para la técnica, y se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla:

- Su **espacio de búsqueda** (i.e., sus posibles soluciones) debe estar delimitado dentro de un cierto rango.
- Debe poderse definir una **función de aptitud** que indique que tan buena o mala es una cierta respuesta.
- Las **soluciones deben codificarse** de una forma que resulte relativamente fácil de implementar en la computadora.



### **Decisiones para implementar un Algoritmo Genético [Michalewicz, 1992]**

Las decisiones que hay que tomar para implementar un Algoritmo Genético son:

- Criterio de codificación. Cómo se va a almacenar la información en el genoma.
- Criterio de tratamiento de individuos no factibles. Cómo se van a tratar a los individuos que no cumplan las restricciones.
- Criterio de inicialización. Cómo se va a construir la población inicial.
- Criterio de parada. Determina cuándo el algoritmo ha llegado a una solución aceptable.
- Función de adaptación. Corresponde a la función de costo de la investigación operativa tradicional.
- Operadores genéticos. Se emplean para determinar cómo va a ser la nueva generación. Básicamente son los operadores de cruce y mutación, aunque pueden ser empleados otros adicionales -muerte, aborto, envejecimiento...-. Tanto cruce como mutación pueden ser realizados de muchas formas distintas.
- Criterios de reemplazo. Los criterios que determinan quiénes se van a cruzar. No tienen que ser obligatoriamente los mismos que los criterios de selección de los padres.
- Parámetros de funcionamiento. Determinados parámetros que, sin poder ser englobados en ninguno de los anteriores, son fundamentales para el funcionamiento de un Algoritmo Genético. Es el caso, por ejemplo, del tamaño de la población, la probabilidad de la aplicación de los operadores genéticos, etc.

#### **3.2.5 Parámetros que deben ser especificados en el programa**

**Nº de generaciones:** cantidad de veces que se produce una nueva generación de individuos, utilizando las reglas anteriormente explicadas. Por lógica, cuantas más generaciones se produzcan, se obtendrán mejores individuos.

**Nº de individuos por generación:** en este ejemplo, la cantidad de individuos de una generación a otra es constante, no varía.

**Nº de entrenamientos:** cantidad de veces que se lleva a cabo un entrenamiento entre diferentes generaciones de individuos.

```

BEGIN /* Algoritmo Genético Simple */
  Generar una población inicial.
  Computar la función de evaluación de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generación */
      FOR Tamaño población/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generación,
          para el cruce (probabilidad de selección proporcional
          a la función de evaluación del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la función de evaluación de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generación.
        END
      IF la población ha convergido THEN
        Terminado := TRUE
      END
    END
  END
END

```

Fig. 3.2 Pseudocódigo del Algoritmo Genético simple

*Los Algoritmos Genéticos son una poderosa herramienta de propósito general de optimización que modela los principios de la evolución. [Burke, 1994].*

El proceso es el siguiente:

1. Una población *random* de horarios factibles es creada. Cada horario es evaluado de acuerdo a un conjunto de criterios (por ejemplo el tamaño del horario, cuantos grupos son colocados a la misma hora y en el mismo salón, cuantos salones hay vacíos).
2. Los horarios son seleccionados de manera aleatoria para ser las bases de la próxima generación. Los buenos horarios son más deseables que los malos. En las bases de la evaluación de estos miembros de la población (horarios factibles) se escogen los padres de la próxima generación de horarios. Evaluando numéricamente durante el proceso de selección buscando los mejores horarios, los peores horarios son eliminados al mismo tiempo que la búsqueda se dirige hacia las áreas más promisorias del espacio de soluciones
3. El Operador Mutación se aplica generalmente a un solo miembro de la población e involucra el cambio aleatorio de la hora o aula de algunos grupos, después almacena el nuevo horario, siempre manteniendo un horario factible. Esto agrega

- un elemento que permite también introducir material genético útil que ha sido perdido en otros horarios en las primeras generaciones
4. El Operador de Cruzamiento toma una pareja de horarios y los combina de alguna manera para producir uno o dos nuevos horarios, esto se hace generalmente de forma aleatoria seleccionando algunos grupos de la mañana de un miembro de la población y los algunos grupos de la tarde de otro miembro para producir un nuevo horario, cualquier grupo que no pueda ser colocado de esta manera, se manda al periodo siguiente disponible.
  5. La nueva población es generada. El proceso será repetido hasta que una buena solución sea generada.

### 3.3 Programación Lógica de Restricciones (PLR)

#### 3.3.1 Antecedentes de la PLR

La Programación Lógica de Restricciones (PLR) es una técnica que tiene mucho en común con los algoritmos de asignación heurística. Dechter describe así el problema de la satisfacción de restricciones. “Un problema de satisfacción de restricciones (PSR) es representado por una red de restricciones, consistente en un conjunto de  $n$  variables,  $X_1, \dots, X_n$ ; con sus respectivos valores de dominio,  $R_1, \dots, R_n$  y un conjunto de restricciones. Una restricción  $C_i (X_{i_1}, \dots, X_{i_j})$  es un subconjunto del producto cartesiano  $R_{i_1} \times \dots \times R_{i_j}$ , formado por todas las tuplas de valores para un subconjunto  $(X_{i_1}, \dots, X_{i_j})$  de las variables que son compatibles unas con otras” [Dechter, 1990].

Una solución de la red es una asignación de valores a todas las variables de tal forma que todas las restricciones son satisfechas” Esta es una representación más poderosa que el modelo de los Grafos Coloreados porque puede representar restricciones más complejas. El problema de los grafos coloreados es equivalente a un PLR binario en la cual cada restricción sólo tiene dos variables.

Este modelo se resuelve tratando de asignar valores a cada una de las variables en cuestión y entonces regresando si se encuentra en un punto muerto y quitando restricciones si se descubre que no se puede encontrar solución. El orden en el que las variables son seleccionadas y el orden de los valores que son asignados a las variables puede ser alterado para tratar aumentar la velocidad de búsqueda.

La PLR es un método que permite el diseño rápido de prototipos de resolución de problemas complejos. Las principales aportaciones en esta línea se centran en el desarrollo de programas que incluyen conjuntos de restricciones adaptadas a una determinada familia de problemas, así como al diseño formal e implementación de modelos de interpretación en paralelo con el fin de mejorar la eficiencia del proceso

deductivo. El objetivo final es disponer de sistemas de resolución implementados para apoyar la creación de sistemas donde la explosión combinatoria de los elementos de un problema hace muy difícil la búsqueda de una solución en un tiempo razonable [Farion, 1992].

En esta sección se proporciona una breve introducción de la Programación Lógica con Restricciones y sus aplicaciones dentro del marco general de la programación lógica. Se trata éste de un campo que ha tenido una notable expansión debido a su potencial aplicación práctica, dado que el análisis de las restricciones permite mejorar considerablemente la eficiencia del cálculo de determinadas funciones.

### 3.3.2 Restricciones

En primer lugar, se empieza por contestar la pregunta: "¿Qué es una restricción?". Esta cuestión se puede responder tan brevemente como escribe Roman Bartak[1998]:

*Una restricción es simplemente una relación lógica entre varias variables desconocidas, tomando cada una un valor dentro de un dominio determinado (p.e.,  $A+B=C$ , -el círculo está dentro del cuadrado-, etc...).*

Así las restricciones aparecen de forma natural en todos los campos del conocimiento humano. "La suma de los ángulos de un triángulo es 180 grados" es un ejemplo de restricción que aparece en el mundo real. De hecho, las restricciones son el medio natural en el que las personas expresamos los problemas en todas las áreas.

Con la explicación dada, se ve de forma natural que las restricciones gozan de varias propiedades interesantes:

- las restricciones pueden especificar información parcial, es decir, en las restricciones no hace falta que se den de forma única los valores de sus variables,
- las variables son no direccionales, es decir, en una restricción típica sobre dos variables (X,Y), se puede inferir una restricción sobre X, dada una restricción sobre Y y viceversa,
- las restricciones son declarativas, es decir, se puede especificar que relación deben mantener sin especificar el procedimiento computacional que implemente esa relación,
- las restricciones son aditivas, es decir el orden de imposición de las restricciones no importa, todo lo que importa al fin y al cabo es la conjunción de restricciones y su efecto,
- las restricciones no suelen ser independientes, normalmente restricciones dentro del conjunto total de restricciones comparte variables.

Así, siguiendo esta idea, el objetivo de la Programación Lógica con Restricciones, es sustituir el proceso de unificación por la resolución de restricciones sobre un dominio

particular de interés. Con esta visión, se considera que el procesador de restricciones debe encargarse de comprobar la consistencia del conjunto de restricciones y, dentro de lo posible, reducirlo a alguna forma normal.

Según Elmohamed [Elmohamed, 1998] las restricciones pueden ser divididas en tres tipos: fuertes, medias y suaves.

**Restricciones fuertes.** Son todas aquellas que deben ser cumplidas obligatoriamente, el costo de no cumplir estas restricciones es sumamente alto y por lo tanto en un horario factible su costo asociado por incumplimiento debe ser cero. Por ejemplo (en un problema real se pueden encontrar de 10 a 15 restricciones fuertes):

- Dos o más clases impartidas por un profesor no se pueden programar a la misma hora.
- Una clase teórica y su laboratorio no se pueden programar a la misma hora en el mismo día.
- Una clase no se puede programar en un aula a menos que su capacidad sea mayor al número de alumnos inscritos.

**Restricciones medias.** Estas restricciones no se refieren a conflictos de tiempo, pero tienen asociado un alto costo a ellas, sin embargo este costo no es tan alto como el de las restricciones fuertes. En el condensado de horarios final el costo de estas restricciones es minimizado y es deseable reducirlo a cero. Por ejemplo (en un problema real se pueden tener entre 20 y 25 restricciones medias):

- Para cada estudiante el balancear las secciones Lun/Mie/Vie y Mar/Jue
- Balancear o distribuir y equilibrar las horas de clase en la semana
- Las clases pueden requerir periodos de tiempo contiguos
- Las clases se pueden agrupar con otras clases formando paquetes
- Los horarios particulares deben tener la menor cantidad de “hoyos” posible; esto es que las clases deben ser consecutivas unas después de otras

**Restricciones suaves.** Estas restricciones no tienen nada que ver con conflictos de tiempo y tienen un bajo costo asociado a ellas. También, en el condensado de horarios final, este costo es minimizado pero no necesariamente debe ser cero. Por ejemplo (se pueden tener entre 30 y 40 restricciones suaves)

- Balancear la inscripción en cursos de varios paquetes.
- Especificar la hora de comida y otros periodos de descanso.
- Los catedráticos pueden requerir periodos de tiempo sin clase.
- No inscribir alumnos-atletas en cursos que tengan conflicto con sus horas de práctica deportiva, etc.

### 3.3.3 Sistemas con Restricciones

Una manera más formal de definir las restricciones es considerarlas como una fórmula de primer orden interpretada en una estructura, para así tener en cuenta la semántica particular del sistema de restricciones. Esta semántica declarativa es adecuada cuando

existe *a priori* una estructura que se adecua al sistema de restricciones (p.e. Los reales para las restricciones aritméticas), pero esto no funciona demasiado bien para los sistemas de restricciones más prácticos (p.e. Dominios finitos). Obviamente, no se puede direccionar ningún cálculo operacional relacionado con el propio procesador de las restricciones [Elmohamed, 2001].

### 3.3.4 Referencia Histórica de las Restricciones

Las restricciones como un área de trabajo que combina investigación de un gran número de campos, incluyendo Inteligencia Artificial, Lenguajes de Programación, Cálculo Simbólico y Lógica Computacional. Las restricciones sobre redes y los problemas de satisfabilidad con restricciones empezaron a estudiarse por los investigadores de Inteligencia Artificial en los años 70.

La programación con restricciones ha sido aplicada con notable éxito en numerosos campos. Los trabajos en esta área, se remontan a la investigación de Inteligencia Artificial y Representación Gráfica por Computadora, por los años 60 y 70. En la última década, sin embargo, ha empezado a darse una mayor conciencia de que las restricciones proporcionan las bases para un potente acercamiento a la programación, modelado y resolución de problemas y que los diferentes esfuerzos por explotar estas ideas pueden ser unidos bajo el mismo prisma conceptual y práctico: la programación con restricciones.

Entre los dominios tratados por los sistemas de Programación Lógica con Restricciones, los que más ampliamente son investigados son, sin duda, los dominios finitos, los números reales y racionales con restricciones aritméticas y los *booleanos*. De hecho, 3 de los principales lenguajes de Programación Lógica con Restricciones son un claro ejemplo de esta tendencia: CHIP, que proporciona restricciones sobre dominios finitos, racionales y *booleanos*, PrologIII, que incluye racionales, *booleanos* y listas, y PLR(R), que maneja restricciones sobre reales.

En la actualidad, se pueden observar dos ramas principales en los avances en la programación con restricciones que parten de diferente base y por tanto, usan diferente metodología para resolver las restricciones. Estas dos ramas son:

- Satisfacción de Restricciones
- Resolución de Restricciones

### 3.3.5 Satisfacción de Restricciones

La satisfacción de restricciones parte de los avances en Inteligencia Artificial (problemas combinatorios) y los Gráficos por computadora (sistema de SKETCHPAD[1963], expresando coherencia geométrica en el caso de análisis de escenas).

### 3.3.6 Resolución de Restricciones

La resolución de restricciones difiere de la satisfacción de restricciones porque usa variables con dominios infinitos. Además, las restricciones individuales son más complicadas, por ejemplo, ecuaciones no lineales. En consecuencia los algoritmos de resolución de restricciones usan métodos algebraicos y numéricos en vez de combinaciones y búsqueda. Sin embargo, existe metodología para discretizar los dominios infinitos en un número finito de componentes y entonces, se aplican las técnicas de Satisfacción de Restricciones.

La programación con restricciones ha sido aplicada con éxito en numerosos campos. Las aplicaciones recientes incluyen [Detcher, 1990]:

- Gráficos por Computadora,
- Procesamiento del Lenguaje Natural (construcción de analizadores eficientes),
- Sistemas de Bases de Datos (para reforzar y/o mantener la consistencia de los datos),
- Problemas de Investigación Operativa (como la optimización de problemas),
- Biología Molecular (secuenciación de ADN),
- Aplicaciones de Empresa,
- Ingeniería Eléctrica (localización de fallos),
- Diseño de Circuitos (para computar configuraciones),
- Planificación y organización
- Configuración
- Juegos
- Configuración de horarios
- Criptoaritmética
- Coloreado de Grafos
- Razonamiento de hipótesis químicas

En esta sección se va a dedicar una parte a cada una las aplicaciones más estudiadas.

### 3.3.7 Organización de Proyectos

El problema de la Organización es el proceso de asignar actividades a recursos en el tiempo. En el caso general de organización es un proceso de tomar decisiones, donde se pueden tomar dos tipos de decisiones:

- Decisiones de situar temporalmente: "Cuándo debería empezar cada actividad".
- Decisiones de asignación de recursos: "Con qué recursos debería ejecutarse cada actividad".

Los problemas de organización y de asignación de recursos están basados en restricciones. Diferentes restricciones pueden afectar a la organización: la duración de

una actividad, las fechas de contrato, restricciones precedentes, tiempos de transferencia o de inicio de la actividad, la capacidad de los recursos y el reparto de recursos.

La relajación de las restricciones principales caracteriza la calidad de los problemas de organización. Estas restricciones son las relacionadas con fechas de entrega, productividad, periodicidad de los cambios de herramientas, niveles de inventario, margen de tiempo, etc. Algunas de estas restricciones pueden ser combinadas en la función de evaluación para la optimización.

Las aplicaciones industriales de las técnicas de resolución de restricciones a los problemas del tipo de asignación de tareas incluye[Detcher, 1990]:

- Organización de la producción.
- Organización y planificación del transporte y la logística.
- Administración del tráfico.
- Diseño de horarios de recursos y de hombres.
- Asignación de plantilla incluyendo aplicaciones de ingeniería de organización.
- Problemas de "empaquetado".
- Configuración de máquinas.

Claramente, hay muchos problemas relacionados en el área de la organización; en la práctica para los sistemas comerciales e industriales reales, los límites entre unos y otros no están claros. Es difícil encontrar algún problema real que se ajuste a un modelo preconcebido sin tener que descartar partes del problema y aproximar otras, así que hay que tomar estos modelos con amplio margen de aproximación.

Otra complicación es el tema de la optimización. En algunos casos el usuario está satisfecho sólo con tener una manera automática de tener la respuesta, pero la mayoría quieren tener al menos una buena respuesta, y algunos el mejor resultado posible. Pero debido a la gran complejidad combinatoria de la mayoría de los problemas reales de organización, una optimización real es muy difícil, así que en la mayoría de los casos se usa la programación con restricciones en combinación con otras técnicas tales como Algoritmos Genéticos para dar soluciones casi óptimas en un tiempo razonable.

Normalmente los problemas de organización se presentan usando variables con restricciones fuertes y restricciones suaves. Estas pueden ser agrupadas en la abstracción del problema de organización. Así, típicamente, el tiempo de comienzo y fin y la duración de cada actividad se representan por variables con restricciones. Los dominios de estas variables son el rango de tiempo de interés en la organización requerida.



# CAPÍTULO 4

## ANÁLISIS Y DISEÑO

### 4.1 Recopilación de información

- Documentos recopilados
  - Catálogo de asignaturas, prerrequisitos y semestre
  - Catálogo de aulas y su capacidad
  - Catálogo de catedráticos, horas disponibles
  - Mapa reticular de la carrera
  - Reporte de grupos abiertos en periodo actual
  - Reporte de alumnos por grupo en periodo actual
  - Reporte de alumnos no inscritos por materia y semestre
  - Reporte de simulación (si todos aprobaran sus cursos)
  - Reporte del índice de reprobación del semestre anterior
  - Encuesta aplicada para determinar los conocimientos requeridos para impartir una materia
  - Encuesta aplicada para obtener los conocimientos de cada catedrático
  - Árbol de conocimiento realizado para determinar los grupos por abrir de cada asignatura
  - Árbol de conocimiento del perfil de las asignaturas
  - Árbol de conocimiento del perfil de los catedráticos
  - Árbol de conocimiento de los criterios de apertura de materia
  
- El proceso de la elaboración de los horarios
  1. Elaboración de los reportes estadísticos del periodo actual (grupos abiertos, alumnos por grupo, alumnos no inscritos por materia, simulación, índice de reprobación)
  2. Cálculo de la estimación de alumnos por materia del próximo periodo
  3. Cálculo del número de grupos por materia requeridos para el próximo periodo
  4. Ajuste de los grupos por materia requeridos para el próximo periodo
  5. Formación de los grupos en paquetes por semestre
  6. Asignación de hora y aula a los grupos

7. Selección de catedrático al grupo y viceversa
8. Ajustes en hora, aula y catedrático por grupo
9. Elaboración del reporte condensado para publicación de horarios
10. Elaboración de los reportes de grupos sin maestro, maestros con horas disponibles, grupos sin aula

Para elaborar los horarios de los grupos y maestros un semestre determinado se tienen en consideración los siguientes elementos:

- El estadístico de Servicios Escolares (SE)  
El departamento de SE cuenta con un sistema que registra todo el historial académico de los alumnos inscritos en el ITZ, las materias cursadas y sus calificaciones, las diferentes carreras con su retícula de asignaturas y sus prerrequisitos, los grupos del semestre actual, la cantidad de alumnos en cada grupo de una asignatura, los índices de reprobación de semestres anteriores, las materias que puede cursar un alumno el próximo semestre, etcétera. Toda esta información plasmada en reportes diferentes es la base para la generación de los horarios del próximo semestre.
- Los grupos requeridos  
Sobre la base de operaciones aplicadas a la información estadística se obtiene la cantidad de grupos requeridos para abrirse el próximo semestre, este número de grupos es la cantidad óptima con la que se podrán inscribir los alumnos sin tener problemas de falta de espacio, el alterar disminuyendo o aumentando el número de grupos requeridos puede ocasionar alumnos con carga crediticia menor a la reglamentaria o que existan grupos con menos de 10 alumnos respectivamente.
- Las aulas  
El ITZ cuenta con 52 aulas repartidas entre las 8 carreras, a cada coordinador de carrera se le asigna determinado número de aulas para su administración, así que al asignar el aula a cada grupo debe cuidar la correcta distribución de los grupos en los horarios correspondientes y que la capacidad de las aulas no rebase el número de alumnos inscritos
- Los maestros y departamentos  
Cada departamento esta formado por un jefe y los catedráticos que forman la academia de cada carrera, los profesores pueden ser de tiempo completo,  $\frac{3}{4}$  y  $\frac{1}{2}$  tiempo o profesores de asignatura, cada uno tiene un horario predeterminado en turno matutino o vespertino, los catedráticos de un departamento cubren las asignaturas de su carrera y apoyan a otras carreras a cubrir las materias que tienen el perfil de esa academia, los catedráticos son uno de los recursos mas difíciles de controlar

Restricciones propias de los profesores:

- Materia no tiene maestro.
- Maestro tiene horas disponibles.

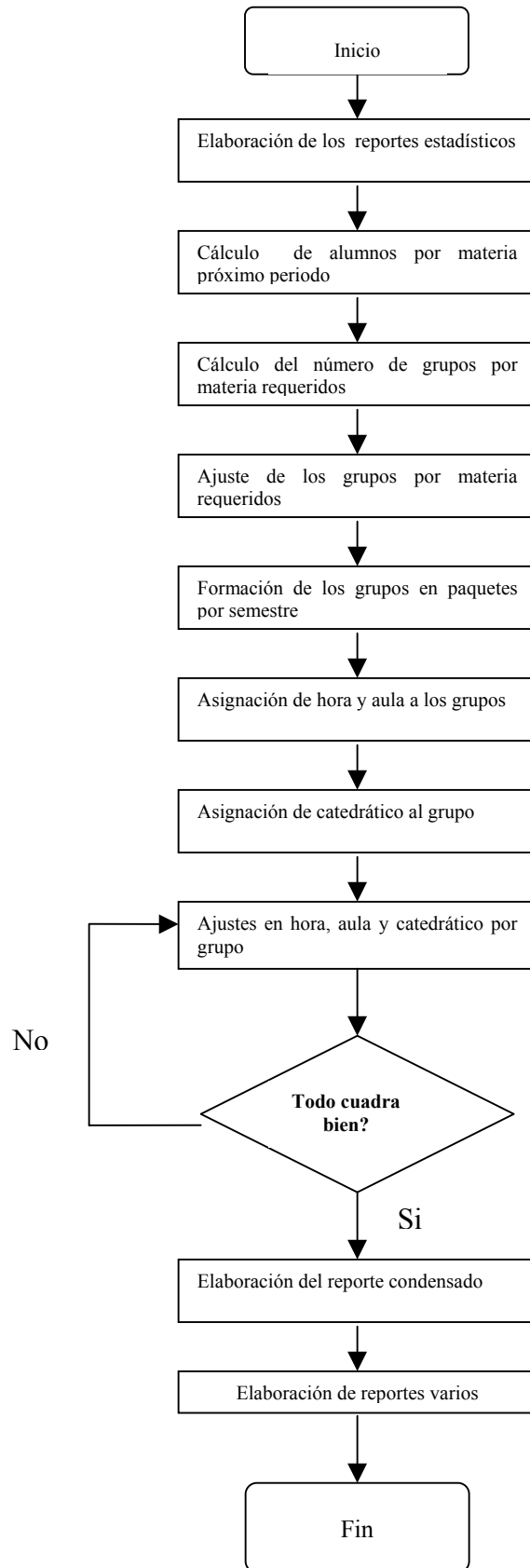
- Maestro tiene materias preferencia.
  - Maestro ha impartido materia.
  - Maestro solicita materia.
  - Maestro tiene restricciones de número de materias a impartir.
  - Restricción no pueden darse más de tres materias diferentes.
  - Maestro tiene una profesión.
  - Maestro tiene un perfil.
  - Maestro tiene un área definida.
  - Maestro prefiere materias especialidad.
  - Maestro prefiere materias semestres iniciales.
- Las materias o asignaturas  
En cada carrera se tiene un mapa reticular que tiene las materias de cada semestre: el nombre, los créditos, las horas de teoría y práctica, las ligas con los prerrequisitos, las materias de la especialidad seleccionada, etc.

Restricciones propias de las asignaturas:

- Materia de 1° semestre.
- Materia de 2° semestre.
- Materia de 3° semestre.
- Materia de 4° semestre.
- Materia de 5° semestre.
- Materia de 6° semestre.
- Materia de 7° semestre.
- Materia de 8° semestre.
- Materia de 9° semestre.
- Materia de especialidad.
- Materia tiene prerrequisito.
- Materia es de básicas.
- Materia requiere conocimientos de Base de Datos.
- Materia requiere conocimientos de Electrónica.
- Materia la puede impartir cualquier maestro.

- Materia requiere conocimientos de Redes.
  - Materia requiere ser impartida por ingeniero.
  - Materia tiene complejidad alta.
  - Materia debe ser impartida por catedrático con grado de maestría.
  - Materia requiere conocimientos de Arquitectura de Computadoras.
  - Materia no tiene cadenas.
  - Materia de Administración.
  - Materia de Ingeniería Industrial.
  - Materia requiere conocimientos de Álgebra.
  - Materia requiere conocimientos de Cálculo Diferencial e Integral.
  - Materia requiere conocimientos de Química Inorgánica y Orgánica.
  - Materia requiere una elaboración de proyectos.
- Las carreras  
Las 8 carreras que se imparten en el ITZ son ordenadas por número de alumnos:
    1. Ingeniería en Sistemas Computacionales
    2. Licenciatura en Informática
    3. Licenciatura en Administración
    4. Ingeniería Civil
    5. Ingeniería Electromecánica
    6. Ingeniería Industrial
    7. Ingeniería Química
    8. Ingeniería Bioquímica
  - Los horarios de grupo  
Una vez determinados los grupos requeridos para el próximo semestre se procede a la asignación de horario de acuerdo al número de horas teórico-prácticas, se le asigna aula y un catedrático responsable, se fija también un número de alumnos y el tiempo de laboratorio en caso de ser necesario.

Diagrama de flujo del proceso



## 4.2 Análisis de la información

- **Catálogo de asignaturas, prerrequisitos y semestre**

Función: Dar a conocer las asignaturas de una carrera, los prerrequisitos de las materias y el semestre al que pertenece una asignatura, los créditos mínimos que debe tener un alumno para cursar la asignatura

Origen: Reporte emitido por el departamento de Servicios Escolares

Campos del reporte: nombre carrera, nombre asignaturas, créditos mínimos, materias prerrequisito, semestre al que pertenece la asignatura

- **Catálogo de aulas y su capacidad**

Función: Dar a conocer la nomenclatura y capacidad de las aulas y laboratorios del ITZ

Origen: Departamento de Planeación

Campos del reporte: Nombre del edificio, número de aula, capacidad

- **Catálogo de catedráticos, horas disponibles**

Función: Dar a conocer los catedráticos por departamento y sus horas disponibles frente a grupo

Origen: Departamentos académicos

Campos del reporte: Nombre del departamento, clave de profesor, nombre del profesor, cantidad de horas frente a grupo que debe tener

- **Mapa reticular de la carrera**

Función: Mostrar el mapa reticular de una carrera, gráficamente muestra las asignaturas por semestre, sus requisitos, fórmula crediticia, etc.

Origen: División de Estudios Profesionales

Campos del reporte: Nombre y clave de la carrera, nombre y clave de las asignaturas, fórmula crediticia (número de horas teóricas, número de horas prácticas, créditos de la asignatura), las materias prerrequisito de otras, materias de especialidad, semestre de la asignatura, número de créditos necesarios para poder cursar una asignatura.

- **Reporte de grupos abiertos en periodo actual**

Función: Muestra los cursos que son ofertados en el periodo actual

Origen: Departamento de Servicios Escolares

Campos del reporte: Nombre de la carrera, nombre de las asignaturas, grupo de los cursos, cantidad de alumnos inscritos, nombre del catedrático que lo imparte

- **Reporte de alumnos por grupo en periodo actual**

Función: Muestra la cantidad de alumnos inscritos por semestre de una carrera

Origen: Departamento de Servicios Escolares

Campos del reporte: Nombre de la carrera, semestres del 1 al 13, número de alumnos inscritos por semestre

- **Reporte de alumnos no inscritos por materia y semestre**

Función: Muestra los alumnos que pudieron inscribirse a una materia pero que por alguna razón no lo hicieron

Origen: Departamento de Servicios Escolares

Campos del reporte: Nombre de la carrera, nombre de la materia, por semestre número de alumnos que pudieron inscribirse y no lo hicieron

- **Reporte de simulación (si todos aprobaran sus cursos)**

Función: Este reporte, simula que todos los alumnos inscritos en los cursos, acreditan el curso, entonces cuenta cuantos estudiantes pueden cursar todas las asignaturas como si hubieran aprobado todos sus cursos actuales

Origen: Departamento de Servicios Escolares

Campos del reporte: Nombre de la carrera, nombre de la asignatura y por semestre, el número de alumnos que pueden cursar una materia

- **Reporte del índice de reprobación del semestre anterior**

Función: Dar a los coordinadores los índices de aprobación y reprobación de los cursos del periodo anterior

Origen: Departamento de Servicios Escolares

Campos del reporte: Nombre de la carrera, nombre de la asignatura, total de alumnos aprobados, total de alumnos reprobados del periodo anterior.

- **Encuesta de conocimientos requeridos para impartir asignatura**

Función: Obtener los datos de los conocimientos que debe poseer un catedrático que quiera impartir una asignatura.

Origen: Academia de cada departamento.

Campos del reporte: Nombre de la asignatura, objetivo de la asignatura, carrera, lista de conocimientos requeridos para impartir la asignatura, carrera adecuada del profesor que imparta la asignatura.

- **Encuesta de conocimientos requeridos para perfil de profesores**

Función: Obtener los datos de los conocimientos que debe poseer un catedrático

Origen: Academia de cada departamento

Campos del reporte: Nombre del catedrático, carrera del catedrático, conocimientos en áreas particulares de cada asignatura.

### **4.3 Solución definida**

Resolver el problema iniciando con la metodología “divide y vencerás” el problema general se dividirá en 6 problemas particulares

- a) Sistema de Información para control de horarios.**
- b) Sistema Experto para definición y generación de grupos por abrir.**
- c) Módulo Programación Lógica de Restricciones para asignar horario a los grupos.**
- d) Módulo Algoritmos Genéticos para asignar aula a los grupos ya definidos.**
- e) Sistema Experto para definir catedrático para asignatura.**
- f) Sistema Experto para asignar asignatura a catedrático.**

**Debido a la naturaleza y características de la información todos los sistemas serán realizados iniciando desde cero la recopilación de información, el análisis, el diseño y la programación de cada módulo del sistema, programado en Visual Basic y Bases de Datos en Access.**



**La principal ventaja de ésta opción es que el sistema está hecho a la medida de las necesidades de la institución, poniendo las restricciones en el formato que se tienen actualmente, la desventaja es que se debe desarrollar TODO el sistema.**

#### **4.4 Requerimientos del sistema**

Los Motores de Inferencias deben ser diseñados con las siguientes características:

- Búsqueda de primero en profundidad, en ocasiones cuando se interactúe con el usuario este tipo de búsqueda será utilizado para recorrer toda la base de conocimiento.
- Búsqueda heurística en la que el usuario deberá dar datos iniciales de por donde empezar la búsqueda de alguna información.
- Tomar información de la Base de Conocimiento y guardarla en las bases de datos, y también inferir la información y guardarla en las bases de datos para su posterior tratamiento e impresión.
- Seleccionar la mejor opción y presentarla al usuario para los casos en que el usuario decida cambiar algún dato, la máquina deberá presentar una opción y saber indicar el porque la “eligió”.
- Dar otras soluciones además de la óptima, teniendo porcentajes de aceptación para cada solución y permitiendo la búsqueda de maestros a impartir materias, aulas para algún grupo, o cantidad de grupos a abrir.
- Dar resultados en lenguaje natural de modo que el usuario entienda claramente lo que la máquina le comunica.
- Trabajar interactivamente con el usuario para que éste pueda cuestionar el porque asignó un maestro, abrió grupos, asignó un aula, etcétera.
- Tener capacidad para el uso del paradigma basado en analogías “Si un maestro ya dio la materia en algún semestre entonces puede darla en este semestre”, o bien “Si este paquete de materias está completo, entonces repite todo el paquete este semestre”.

**Interfaz de Usuario :** recibe y entrega información interactuando con el usuario, es decir, el usuario puede:

- 1) alimentar hechos,
- 2) solicitar resultados y reportes,
- 3) cuestionar cómo se obtuvieron ciertas conclusiones.

**La interfaz de usuario debe tener las características siguientes:**

- Comunicarse en el lenguaje del usuario sea un humano u otro dispositivo.
- Recibir hechos del usuario y almacenarlos en la base de hechos.
- Presentar de manera amigable los objetivos a buscar e inferencias en pantalla.
- Dar de manera clara la explicación de la inferencia realizada.

- Dar resultados por escrito en informes.
- Proporcionar al usuario la facilidad de alimentar la base de conocimientos.
- Dar al usuario la facultad de corregir las inferencias.
- Si el usuario es inexperto, facilitar la ayuda necesaria.

### **Selección del lenguaje de programación**

Visual Basic 6.0 proporciona a los programadores facilidad para desarrollar aplicaciones complejas en poco tiempo (comparado con lo que cuesta programar en Visual C++, por ejemplo).

El sistema necesita interfaces amigables al usuario con ventanas y botones emergentes en cualquier momento y Visual Basic 6.0 es una herramienta de diseño de aplicaciones para Windows, en la que éstas se desarrollan en una gran parte a partir del diseño de una interfaz gráfica (pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc). En una aplicación Visual Basic 6.0 el programa está formado por una parte de código puro y otras partes asociadas a los objetos que forman la interfaz gráfica. Visual Basic 6.0 permite la programación de sus ventanas, botones y componentes de manera más fácil que Java con sus *Applets*.

En la parte del sistema de control de horarios que es un sistema de información común, los programas que llenan de información las bases de datos son muy parecidos unos con otros, al hacer los procesos de altas, bajas, modificaciones, consultas y validaciones se utilizan las ventajas que Visual Basic 6.0 ofrece de la Programación Orientada a Objetos (POO), ésta es una forma de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización de código. Las características fundamentales de la POO son abstracción, encapsulamiento, herencia y polimorfismo.

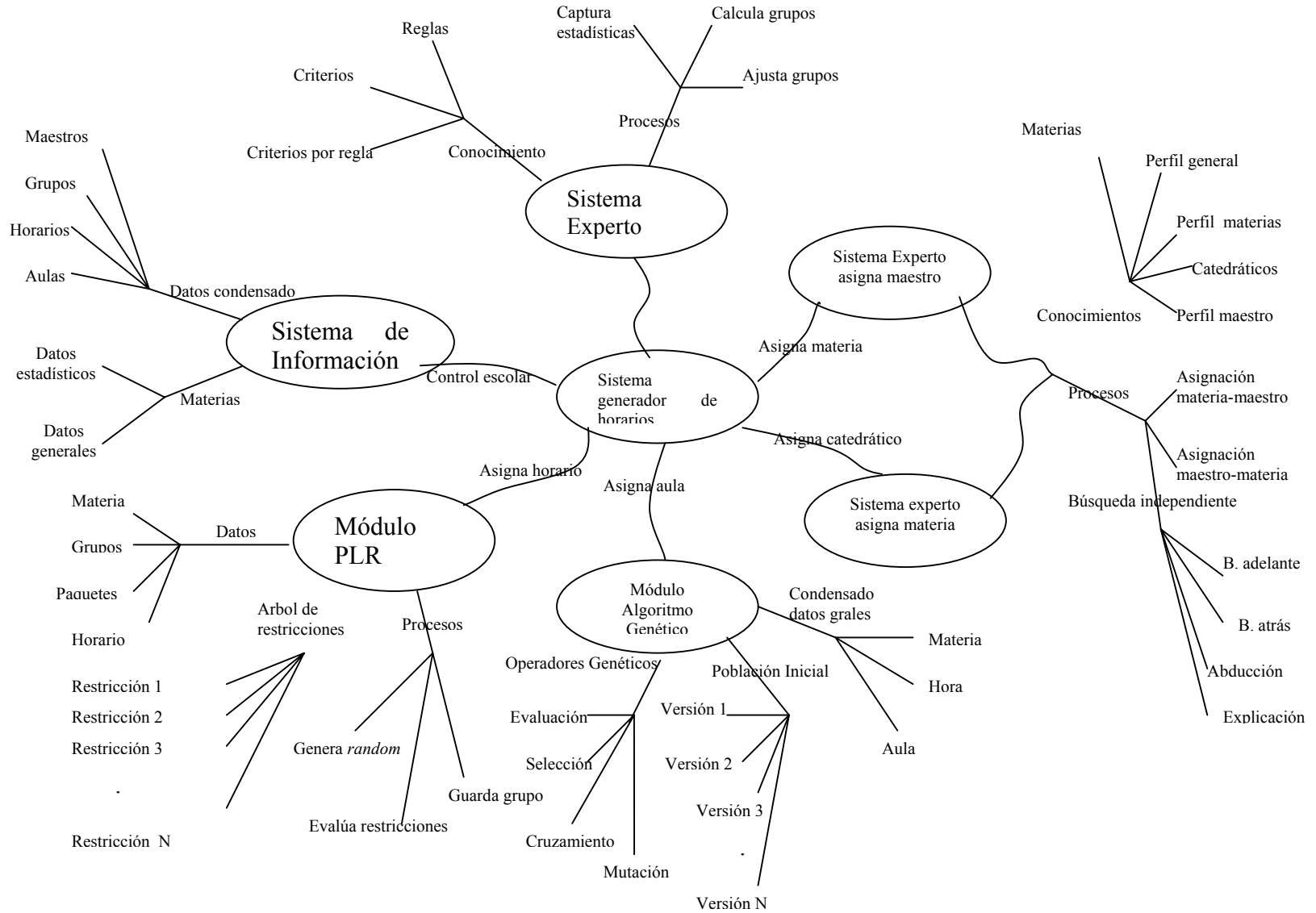
Visual Basic 6.0 soporta la abstracción, la encapsulación, el polimorfismo y la reutilización de código. Los objetos de Visual Basic están encapsulados; es decir, contienen su propio código y sus propios datos.

### **Manejador de Base de datos para el almacenamiento de la información**

El manejador de base de datos a utilizar será Microsoft Access debido a que los usuarios que alimentan la base de conocimientos son expertos en diferentes áreas del conocimiento y tiene conocimientos limitados de los diferentes manejadores y están más familiarizados con Microsoft Access. Además de que Access es transportable fácilmente.

Además la cantidad y tipo de información que se almacena en las tablas no requiere manejadores de bases de datos potentes o con características especiales.

### 4.5 MAPA CONCEPTUAL DEL PROYECTO





En la Figura 4.1 se muestra el mapa conceptual del proyecto. La burbuja central representa el sistema generador de horarios y cada rama representa uno de los seis módulos que lo componen, las burbujas de cada rama contienen a su vez varias ramificaciones en las que se muestran de manera gráfica los componentes más representativos de los procesos utilizados, la información que manejan, el conocimiento que utilizan y las restricciones involucradas.

El módulo Servicios Escolares contiene información necesaria para la administración de los horarios en general y proporciona los datos para el funcionamiento de los demás módulos.

El módulo Sistema Experto Creación de Grupos (SECG) muestra el conocimiento y procesos utilizados para definir la cantidad de grupos a ofertar a partir de la información estadística proporcionada por servicios escolares, más adelante en el documento se explica a detalle su funcionamiento.

En la Figura 4.1 se aprecia que los módulos Sistema Experto Asigna Catedrático(SEAC) y Sistema Experto Asigna Materia(SEAM) comparten las bases de conocimiento porque el conocimiento de un SE es la base de hechos del otro, en ambos casos los procesos difieren solamente en la información que utilizan y en los resultados que producen, nótese que existe un proceso que es “búsqueda independiente” que puede catalogarse como un Sistema Experto adicional que sirve para sugerir las asignaturas que puede impartir una persona que no está dentro de las bases de datos de maestros y por lo tanto su perfil tampoco estará capturado, en esta caso los hechos (su perfil de conocimientos) es capturado en el momento de la consulta.

El módulo de Algoritmos Genéticos (AG) muestra los datos que recibe, como se genera una población inicial, el tratamiento que se le aplica por medio de los operadores genéticos y la información que se genera.

Por último en la Figura 4.1 se nota también el módulo donde se utiliza la Programación Lógica de Restricciones que contiene 3 ramas donde una representa los datos recibidos y la información generada, otra rama denota un árbol de restricciones que deben ser satisfechas por los paquetes de horarios, la última rama denota los procesos que se llevan a cabo, debe notarse que existe un proceso aleatorio que genera cada grupo, un proceso que evalúa el grupo contra las restricciones y uno más que guarda los grupos que cumplieron con todas.

En este mapa conceptual no se muestra en que orden se ejecuta cada módulo del proyecto, la Figura 4.2 ilustra la secuencia de ejecución de los módulos.

La Figura 4.2 muestra el diagrama con la secuencia en que se van ejecutando los procesos, los datos que ingresan a cada módulo y la información que genera.

### 4.5.2 MAPAS CONCEPTUALES DE LOS MÓDULOS

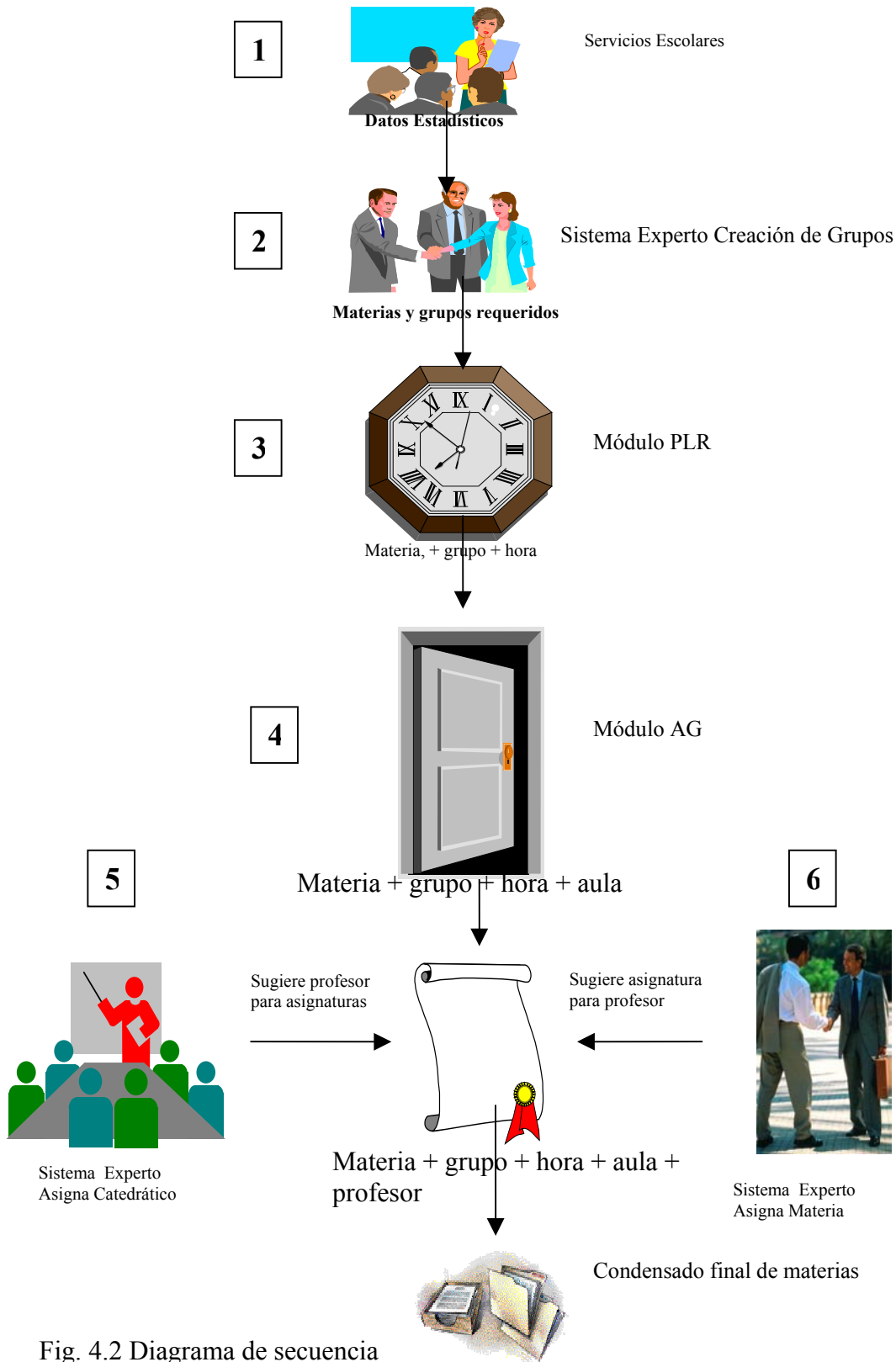


Fig. 4.2 Diagrama de secuencia

Como puede verse todo inicia cuando Servicios Escolares envía la información a la División de Estudios Profesionales (donde “habitan” los coordinadores) quienes son los responsables de la elaboración de los horarios.

Un primer proceso captura la información que proviene de Servicios Escolares, el segundo módulo define cuantos grupos deben programarse, el tercer proceso asigna la hora a cada grupo, el cuarto módulo es el proceso que asigna las aulas, los procesos quinto y sexto sirven para sugerir qué catedrático es el más adecuado para cada grupo y como resultado final se tiene el condensado final de horarios para el siguiente semestre.

En la Figura 4.3 se muestran los componentes del primer Sistema Experto utilizado para definir la cantidad de grupos que deben ofertarse el próximo semestre, en la parte superior izquierda de la figura esta el coordinador de la carrera quien es el encargado de cargar el conocimiento a las Bases de Conocimiento utilizando la interfaz de experto, es la persona indicada para definir los criterios que tiene cada materia, define las reglas para la agrupación de los criterios y define en cada regla que tipo de ajuste se realiza.

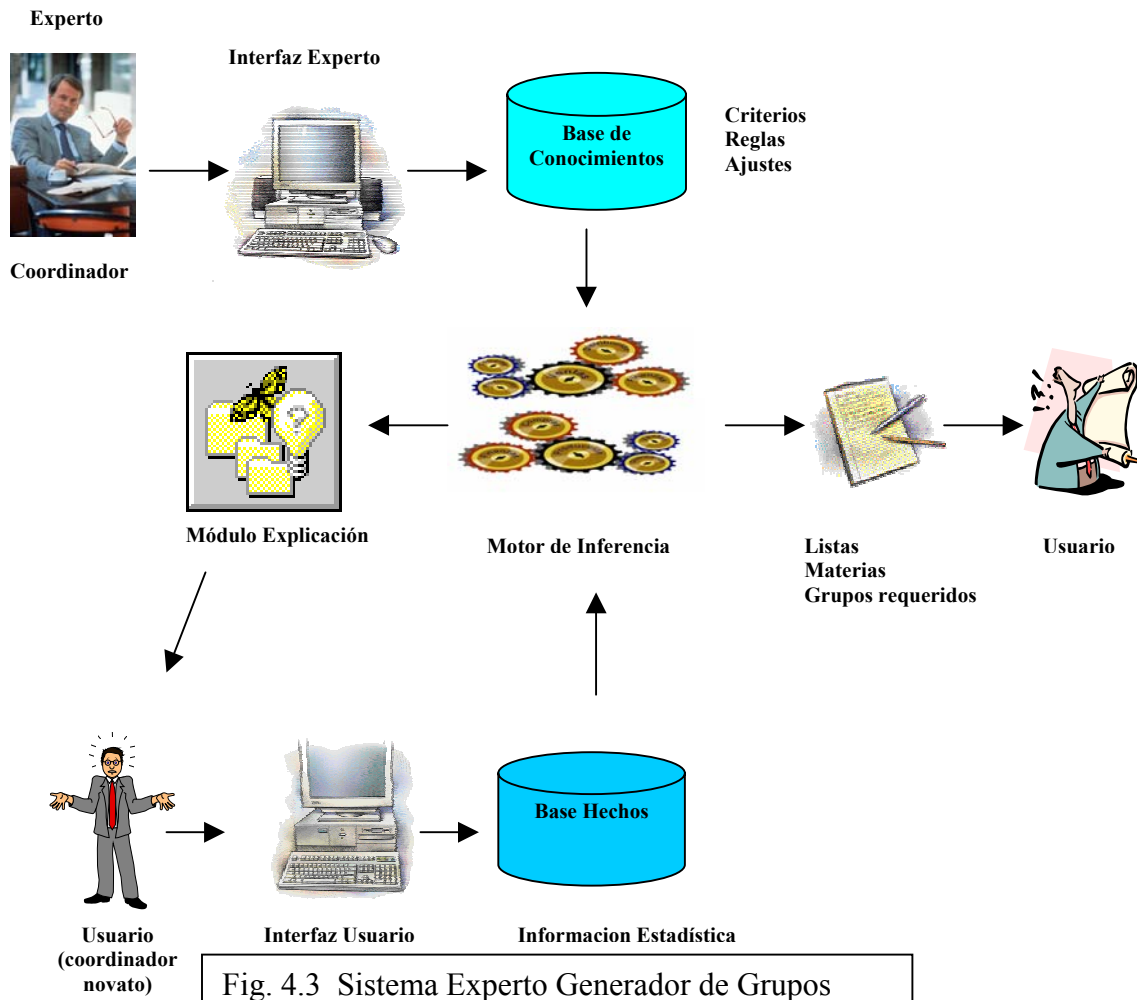


Fig. 4.3 Sistema Experto Generador de Grupos

El usuario en la parte inferior de la figura puede ser el coordinador o su asistente y se encarga de capturar la información estadística proporcionada por servicios escolares que son los datos de alumnos inscritos actualmente en cada materia, el índice de reprobación, los alumnos inscritos en la materias prerrequisito, una simulación de los alumnos probables para cursar esa materia si todos los inscritos aprueban todo y los alumnos que pueden cursar las materias pero no se inscribieron.

Con los datos anteriores se realiza la siguiente operación:

$$\begin{aligned} & (\text{Inscritos en materia} * \text{índice de reprobación de materia}) / 100 + \\ & \text{Alumnos no inscritos} + \\ & \text{Alumnos en prerrequisito} - (\text{alumnos en prerrequisito} * \text{índice de reprobación}) / 100 + \\ & (\text{Simulación} - \text{alumnos en prerrequisito} - \text{no inscritos}) \\ & = \text{Alumnos del próximo semestre} \end{aligned}$$

Este último dato (alumnos del próximo semestre) proporciona la cantidad de alumnos por materia que pueden inscribirse en los grupos según los estadísticos de Servicios Escolares, pero no contempla preferencias de alumnos, conveniencia de inscripción, etc.

Ofrecer los grupos necesarios para la inscripción de alumnos tomando el dato calculado ocasiona que al terminar las inscripciones existan grupos vacíos, grupos sobresaturados, alumnos con pocas materias y varios problemas mas.

La finalidad de este Sistema Experto es realizar un **ajuste** al dato calculado anteriormente para incluir en su cálculo todas aquellas características y experiencia del coordinador para minimizar los problemas mencionados en el párrafo anterior.

Los ajustes pueden ser de varios tipos:

- Ajuste drástico hacia abajo. Se aplica una reducción del 50% al dato calculado.
- Ajuste hacia arriba. Se agrega un grupo más al dato calculado.
- Ajuste hacia abajo. Se resta un grupo al dato calculado.
- Ajuste numérico. Se ajusta utilizando la regla del redondeo de decimales
- Ajuste drástico hacia arriba. Se suman 2 ó 3 grupos al dato calculado

Cuando utilizar una u otra regla está definido por los criterios que cada materia contiene naturalmente y que con el paso de los semestres, puede variar (aunque también pueden pasar años sin sufrir ningún cambio).

La Figura 4.4 describe la utilización de la técnica de la Inteligencia Artificial llamada Programación Lógica de Restricciones para generar el condensado de grupos a partir de una lista de materias que contiene solamente el nombre y la cantidad de grupos que se deben abrir de una materia.



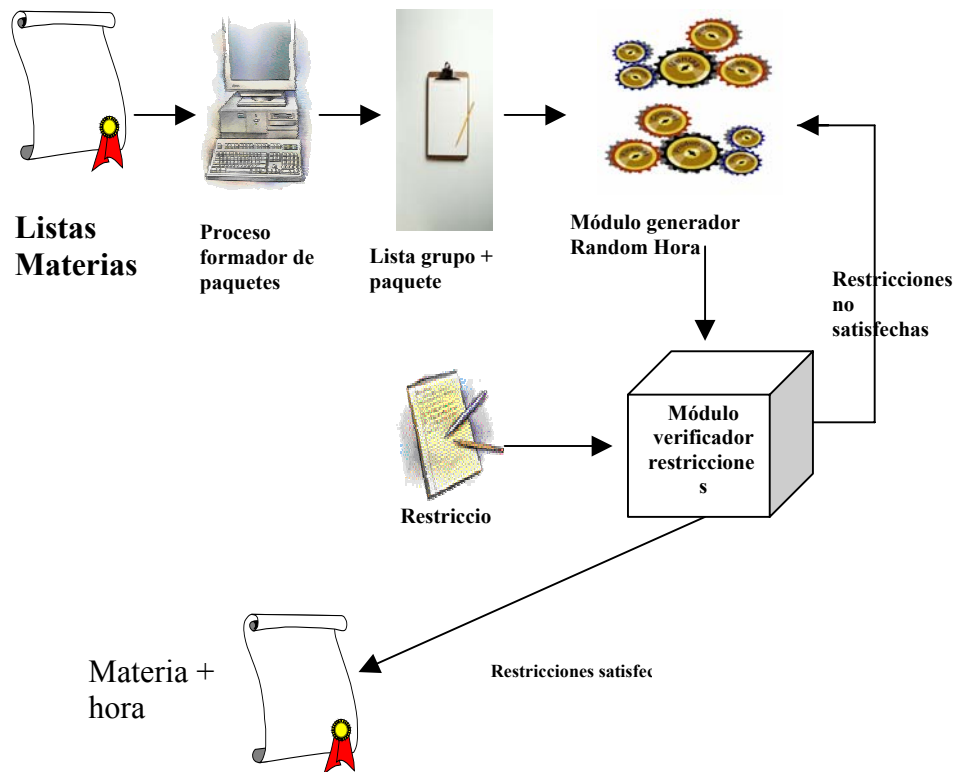


Fig. 4.4 Diagrama del módulo Programación Lógica de Restricciones

Existe un proceso inicial que forma los “paquetes” de grupos, esto es, las materias se ordenan por el semestre al que corresponden (generalmente son 6 materias por semestre), después para cada semestre se generan los grupos correspondientes, si por ejemplo se deben hacer 3 grupos de la materia X, se generan tres registros de la materia X y cada registro se etiqueta:

- Materia X grupo AA,
- Materia X grupo AB y
- Materia X grupo AC

Una vez que todas las materias han sido generadas y “empaquetadas” se pasan al **generador de horario** en bloques de un paquete a la vez. Este proceso funciona asignando un horario aleatorio a cada grupo, después se verifica que cada grupo cumpla con todas las restricciones que son del tipo: grupo no “choca” con otro del paquete, grupo proporciona horario continuo al paquete, grupo tiene horario de 2 horas por día, grupo no deja horas vacías en paquete, grupo cumple con todas sus horas asignadas, etc.

La Figura 4.5 muestra el diseño del Algoritmo Genético utilizado para asignar aula a los grupos que provienen del módulo anterior (PLR) y recibe como datos de entrada un condensado de horarios (materia + grupo + hora).

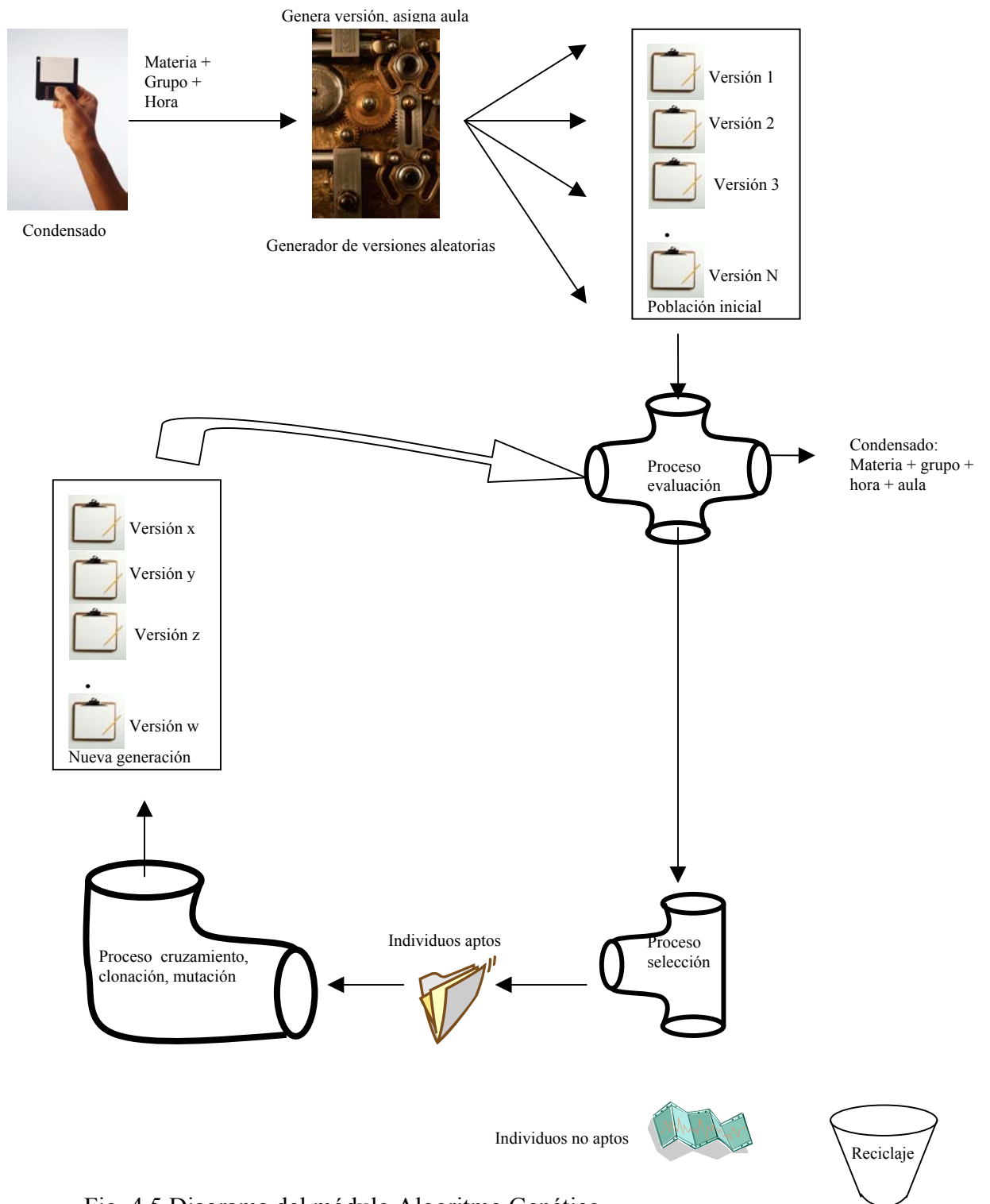


Fig. 4.5 Diagrama del módulo Algoritmo Genético

Esta rama de la Inteligencia Artificial es utilizada generalmente para procesos de optimización y en esta ocasión se utiliza para asignar las aulas al condensado de materias.

Al inicio se reciben como datos generales el condensado de materias. La población inicial se obtiene copiando  $N$  veces ( **$N$  versiones**) este condensado y agregando a cada registro el número de versión y un aula asignada de manera aleatoria, de esta manera se obtienen  $N$  versiones diferentes del condensado original.

Cada versión es entonces evaluada contando el número de choques en aula y la cantidad de espacios vacíos por aula, a cada versión se le genera un vector para guardar la evaluación (cromosoma de evaluación) y de esta manera se mide el grado de aptitud que tiene cada versión.

Las versiones que tienen menos choques y menos espacios vacíos se consideran con un **grado de aptitud alto** y son las mejores, al contrario las que tienen muchos choques son consideradas de aptitud baja, no sirven para la reproducción y son desechadas.

Una vez seleccionadas las mejores versiones se procede a la aplicación de los operadores genéticos, el primero que se aplica es el cruzamiento y consiste en elegir dos versiones de alta aptitud, **tomar segmentos seleccionados heurísticamente** de ellas y formar un nuevo individuo (una nueva versión), se repite este proceso  $Y$  veces con las diferentes versiones hasta tener **una nueva generación** de  $N$  condensados de horarios.

Repetir este proceso de generación, evaluación y selección durante  $Z$  generaciones aplicando cuando se necesite la clonación que es un proceso de copiar completamente una versión (con alto grado de aptitud) a la nueva generación.

Al principio la convergencia hacia horarios sin choques en aula es rápida, pero al paso de las generaciones disminuye, entonces se debe usar la mutación que es un proceso de acceder manualmente los registros de alguna versión  $Z_i$  en la generación  $Z$  y corregir algún aula al final después de varias generaciones, selecciones, cruzamientos, clonaciones y mutaciones se obtiene un condensado de aulas libres de choques y con pocos espacios vacíos.

En la Figura 4.6 se muestran los componentes del Sistema Experto utilizado para sugerir cuales son los catedráticos que mejor se desempeñarían para impartir una asignatura, tomando en cuenta los conocimientos que tiene un profesor (adquiridos a lo largo de su carrera) y cuales son los conocimientos necesarios para impartir la cátedra de alguna asignatura.

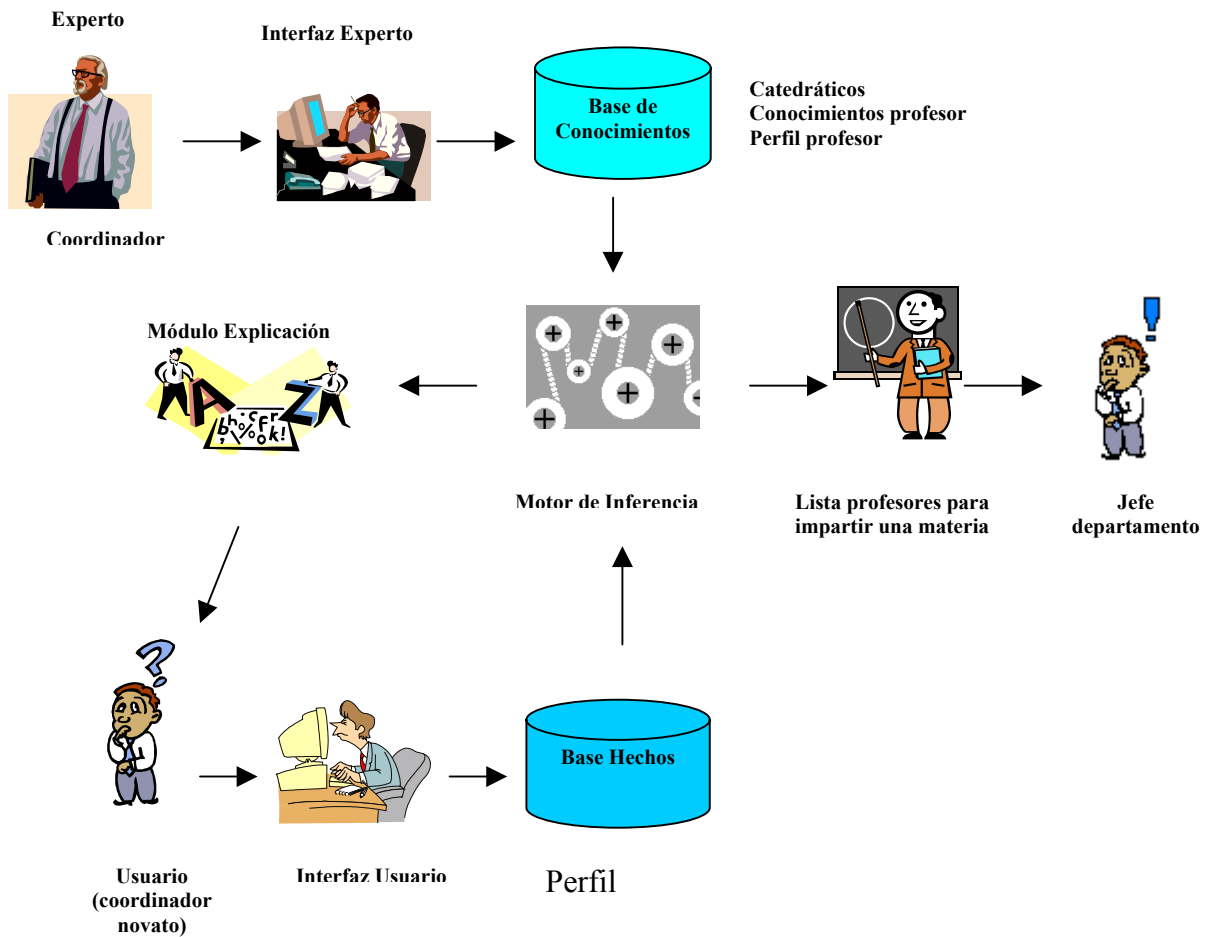


Fig. 4.6 Diagrama de Sistema Experto para sugerir profesor a asignatura

Todo inicia haciendo una encuesta a los profesores acerca del perfil y los conocimientos necesarios para poder impartir una asignatura, el objetivo de la misma y los principales temas que abarca, todo lo anterior se captura en la base de hechos como el perfil de la materia.

Posteriormente se hace otra encuesta a los profesores pidiéndoles que seleccionen cuales son los conocimientos y el grado de dominio que tienen de los diferentes tópicos de las asignaturas, esto se convierte en el perfil del catedrático. Por último con un proceso de inducción se determina para cada materia cual es el grado de certeza con el cual puede un profesor impartir una asignatura.

Nótese en la Figura 4.7 que el coordinador captura el conocimiento referente a las asignaturas (objetivo, temas, características necesarias para impartirla), y este es almacenado en la Base de Conocimientos.

Por otro lado, un profesor debe llenar una encuesta en la que indique cuales son los temas que domina y en que porcentaje lo hace (esto es el perfil del catedrático), esta información se almacena en la base de hechos.--

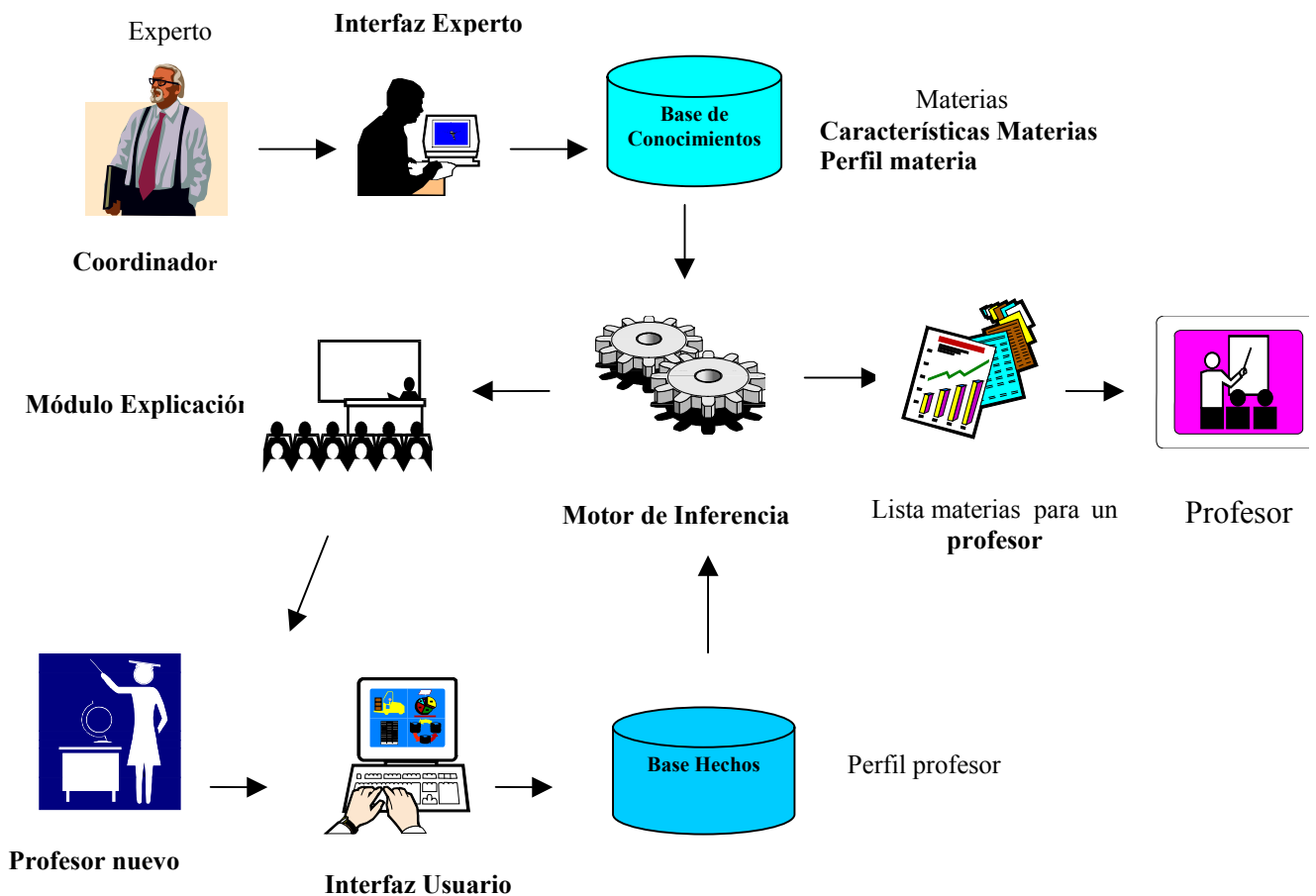


Fig. 4.7 Diagrama de Sistema Experto para sugerir asignaturas para catedrático

El Motor de Inferencia hace los *matches* respectivos y genera una lista de las materias que puede impartir un profesor y con que grado de certeza está capacitado para hacerlo.

Por último el módulo de explicación informa el porqué dio esa inferencia mostrando las características de materia y profesor que hicieron match.

### 4.5.3 Diagrama de contexto

En la Figura 4.8 se detallan los elementos que intervienen aportando y recibiendo información para el funcionamiento del sistema.

Los departamentos académicos aportan al sistema una lista con los catedráticos que laboran ahí, incluye los **datos personales de los profesores**, número de tarjeta, nombre, carrera, etc. Los coordinadores de carrera son los responsables de la elaboración de los horarios, estos funcionarios son la pieza fundamental de este proceso porque aportan sus conocimientos para:

- Definir el ajuste al dato estadístico proporcionado por Servicios Escolares
- Definir las restricciones para la creación de los paquetes de grupos
- Definir las restricciones para asignar hora a cada grupo
- La distribución de los grupos en las aulas
- Asignar la asignatura que mejor puede impartir un catedrático
- Asignarle a una asignatura al profesor que mejor puede impartirla

El departamento de Servicios Escolares es el iniciador del proceso de la elaboración de horarios, aporta los **datos estadísticos** obtenidos de la situación escolar actual de los alumnos, cada alumno está inscrito en alguna de las 8 carreras que se ofrecen en el Instituto, está cursando entre 5 y 8 asignaturas por semestre, en cada periodo acredita o reprueba algunas materias. En cada carrera se tiene un mapa reticular que incluye aproximadamente 50 asignaturas, con prerrequisitos, créditos, horas en aula, horas en laboratorio, orden por semestre, etc.

El departamento División de Estudios Profesionales (donde están adscritos los coordinadores de carrera) es el responsable de distribuir las aulas disponibles del Instituto entre los distintos coordinadores, **las aulas** son un recurso escaso y por lo tanto valioso, de esta distribución depende en gran medida la holgura o estrechez que tienen los coordinadores para asignar las aulas a los grupos creados.

Los profesores son otro de los recursos valiosos de una institución, son los responsables de transmitir sus conocimientos a los alumnos, cada catedrático posee conocimientos de diversas áreas, tópicos y ramas del saber humano, el **perfil del catedrático** está en relación directa con el dominio que tiene de cada una de ellas y varía desde un conocimiento superficial hasta el dominio total de un área.

Los generadores *random* de los módulos PLR y AG proporcionan la información necesaria generada de manera aleatoria para el funcionamiento de esos módulos.

A los coordinadores y alumnos se les proporciona un condensado de horarios, los alumnos conocen los grupos a los que se pueden inscribir, las horas las aulas y los catedráticos que las impartirán.

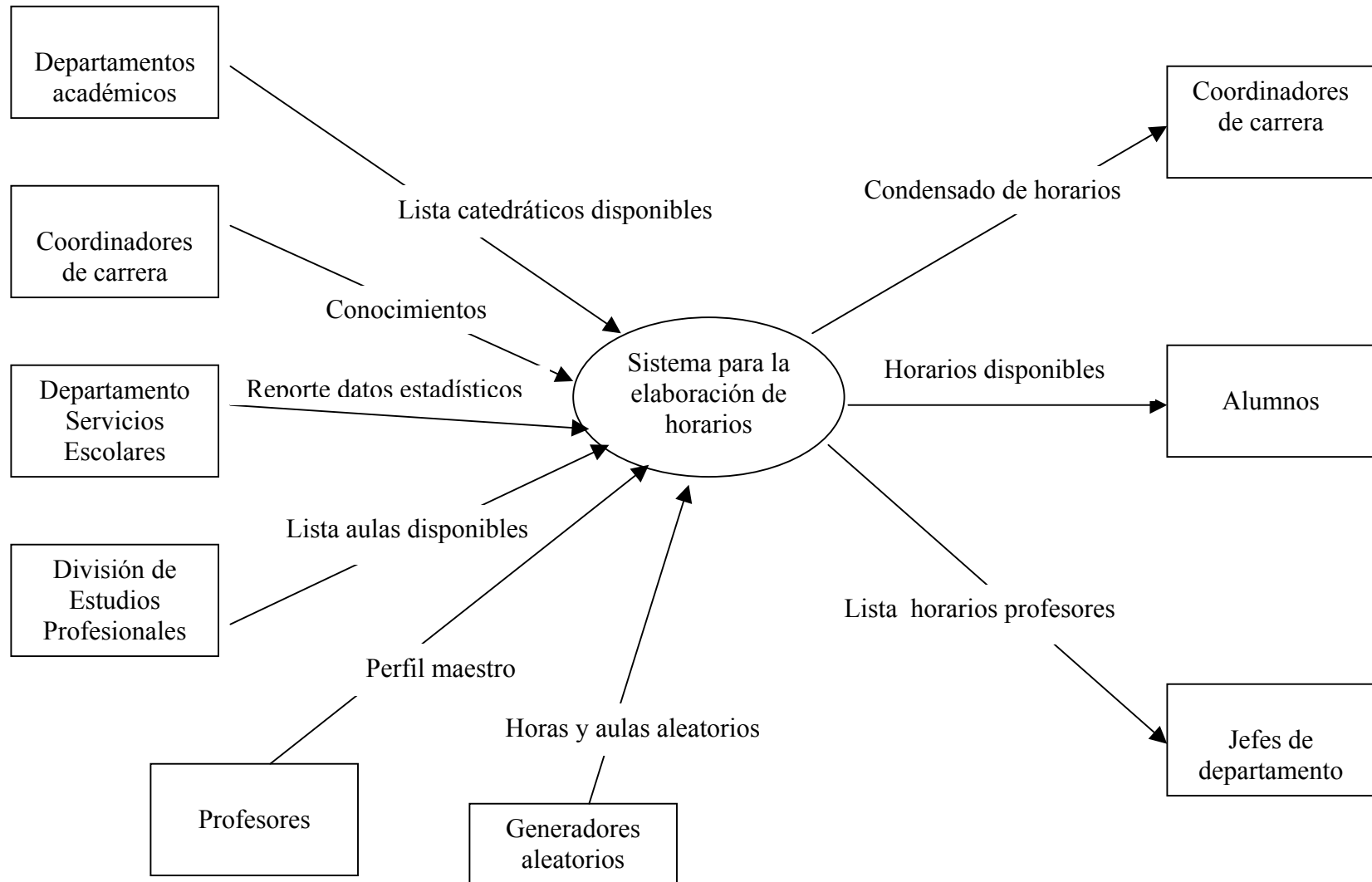


Fig. 4.8 Diagrama de contexto del proyecto

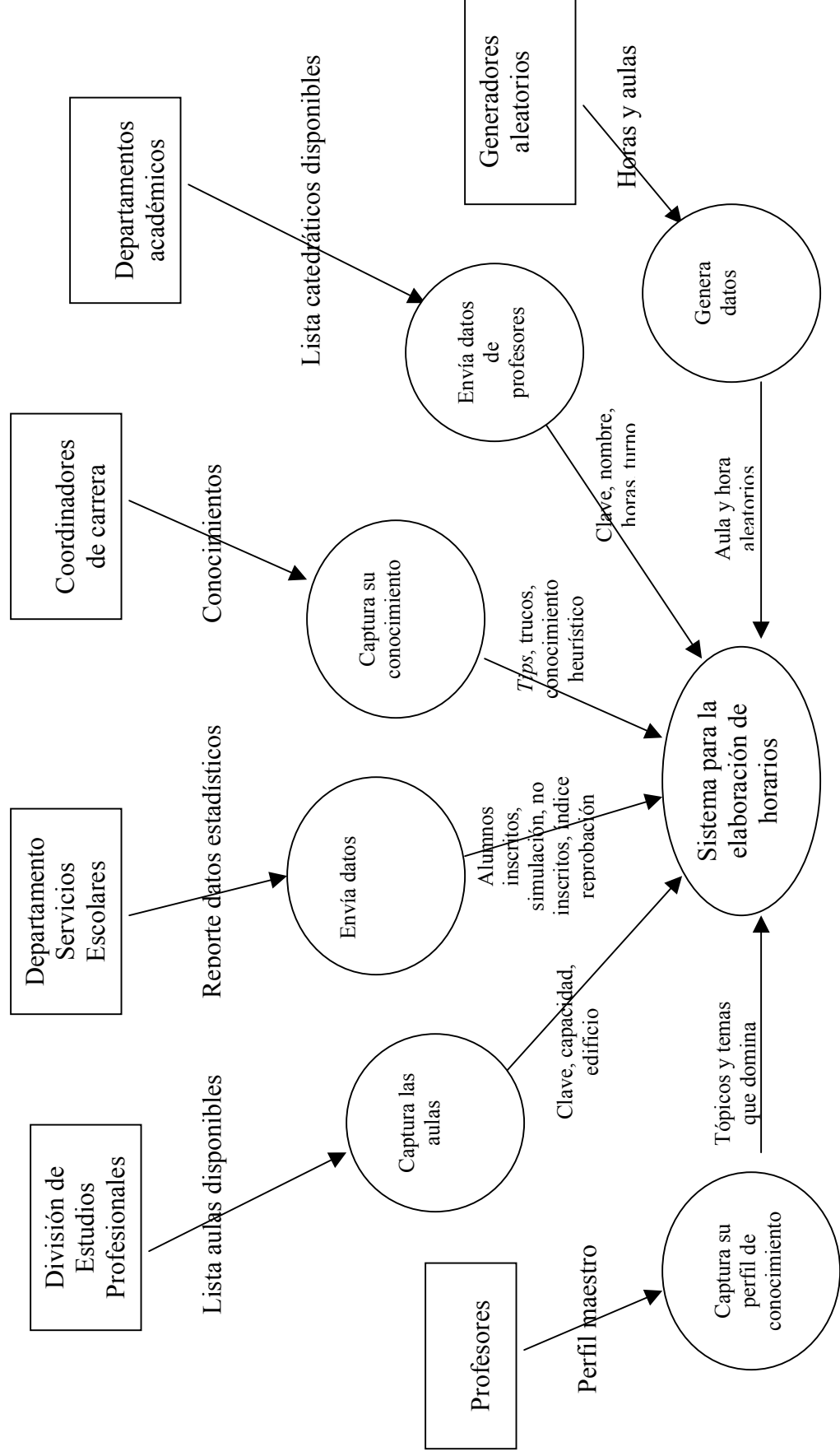


Fig. 4.9 Diagrama de funciones fundamentales (entradas)



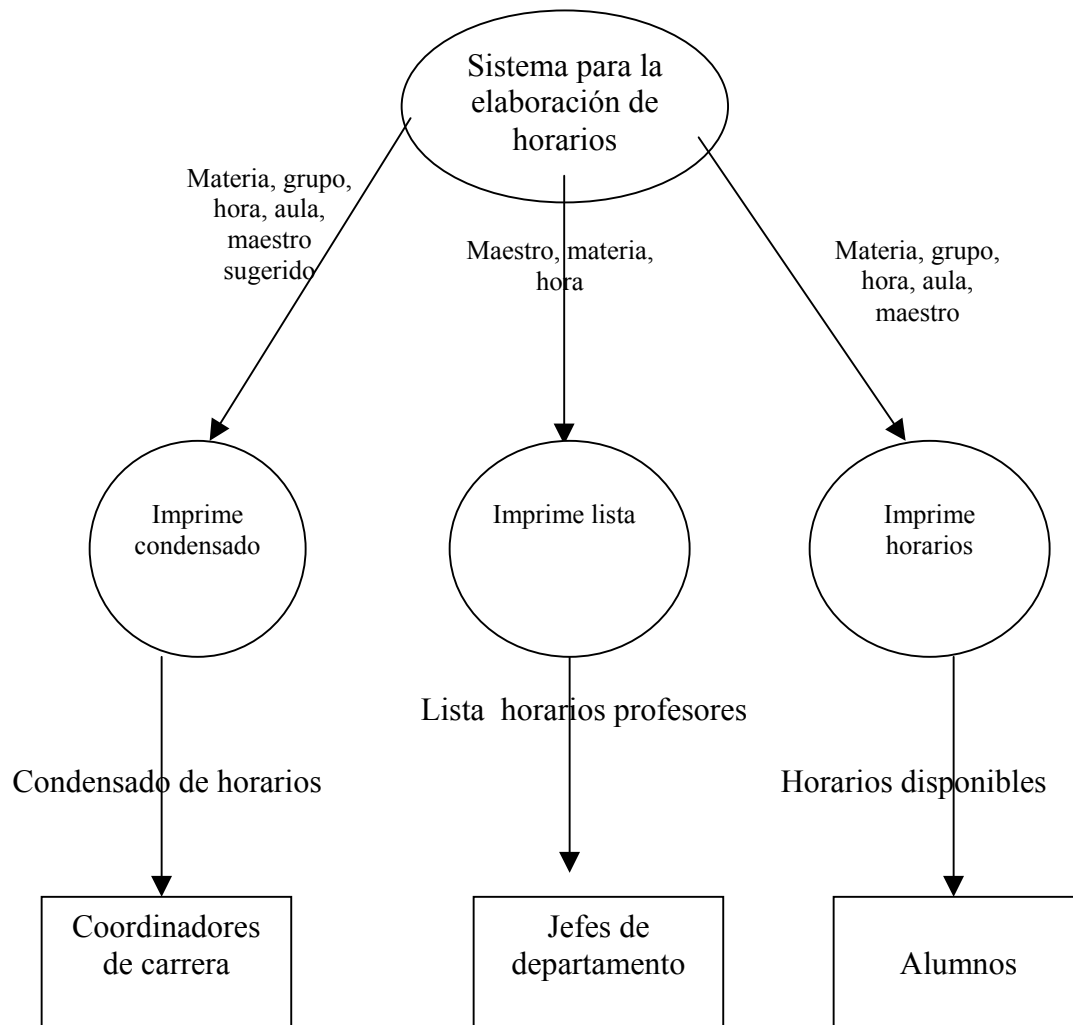


Fig. 4.10 Diagrama de funciones fundamentales (salidas)

### 4.5.4 Diagrama entidad - relación

NOTA: Los siguientes diagramas muestran las tablas utilizadas por cada módulo, para mayor claridad se repitieron las tablas necesarias para representar cada parte del sistema pero en la base de datos no existe tal duplicación.

Este diagrama contiene las tablas del módulo administrativo de los horarios, en un principio éstas eran las únicas tablas de un sistema de control de horarios

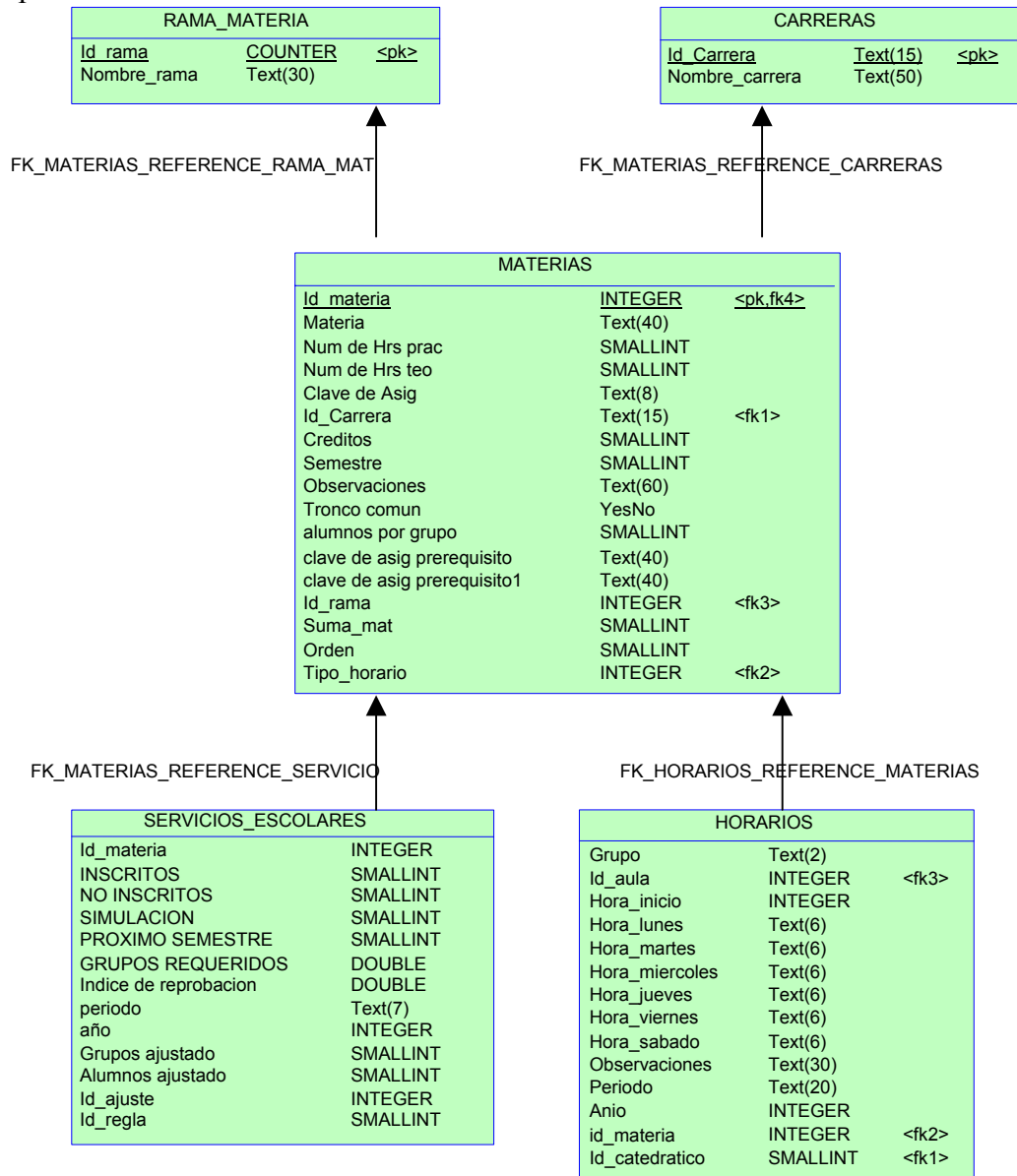


Fig. 4.11 Diagrama entidad – relación del módulo administrativo

Este diagrama muestra las tablas donde se almacena el conocimiento de los coordinadores para que el Sistema Experto del módulo sugiera y aplique el ajuste al dato de cantidad de grupos sugerido por Servicios Escolares. La relación entre las tablas va generando las reglas de producción

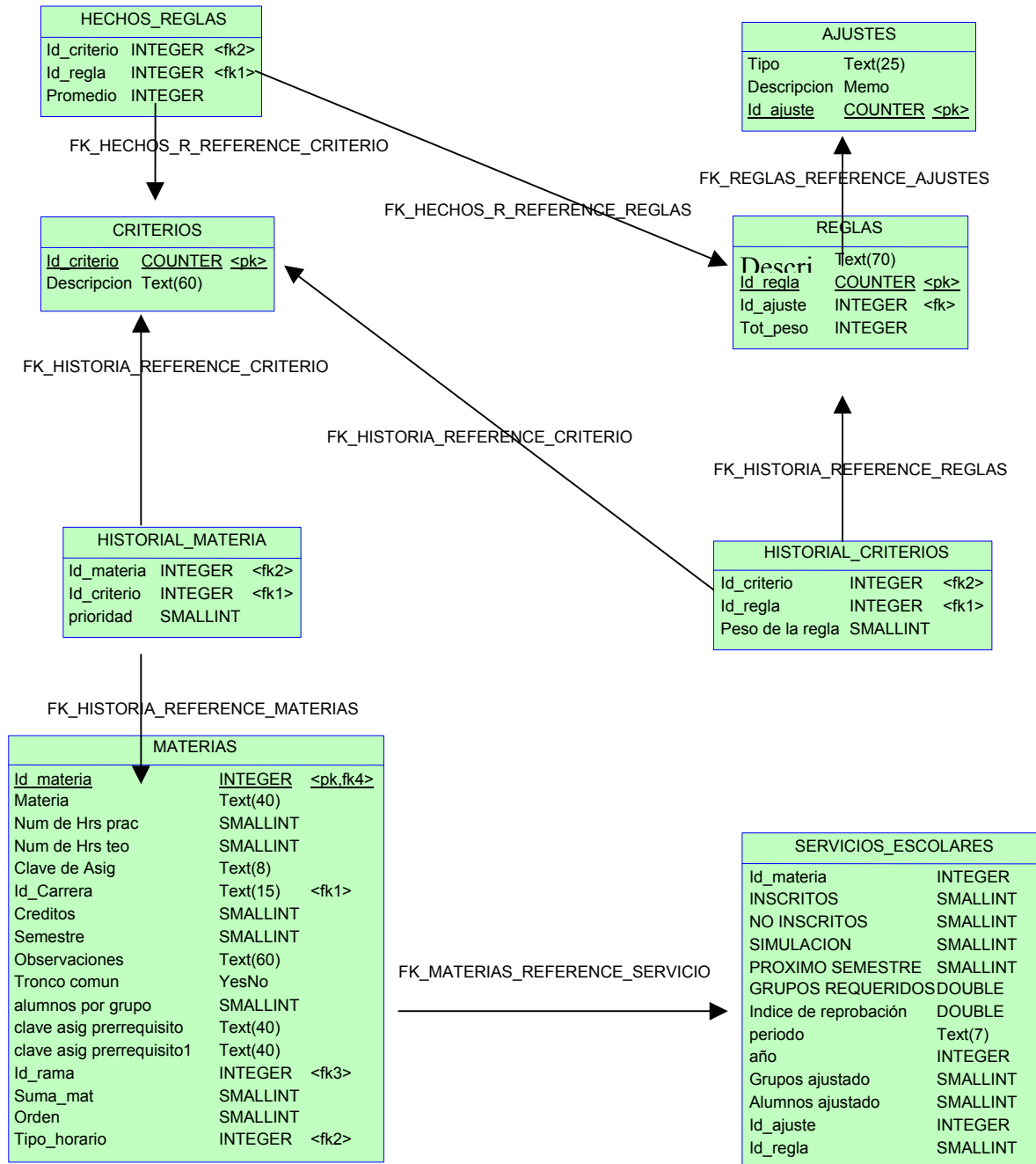


Fig. 4.12 Diagrama E-R del módulo para determinar y ajustar número de grupos

En este diagrama se muestra la base de datos que almacena la información que manipula el módulo Algoritmo Genético, las tablas más representativas son POBLACION donde se guardan todas las versiones que forman las generaciones de horarios y CROMOSOMAS que almacena el registro de evaluación de cada versión de horarios

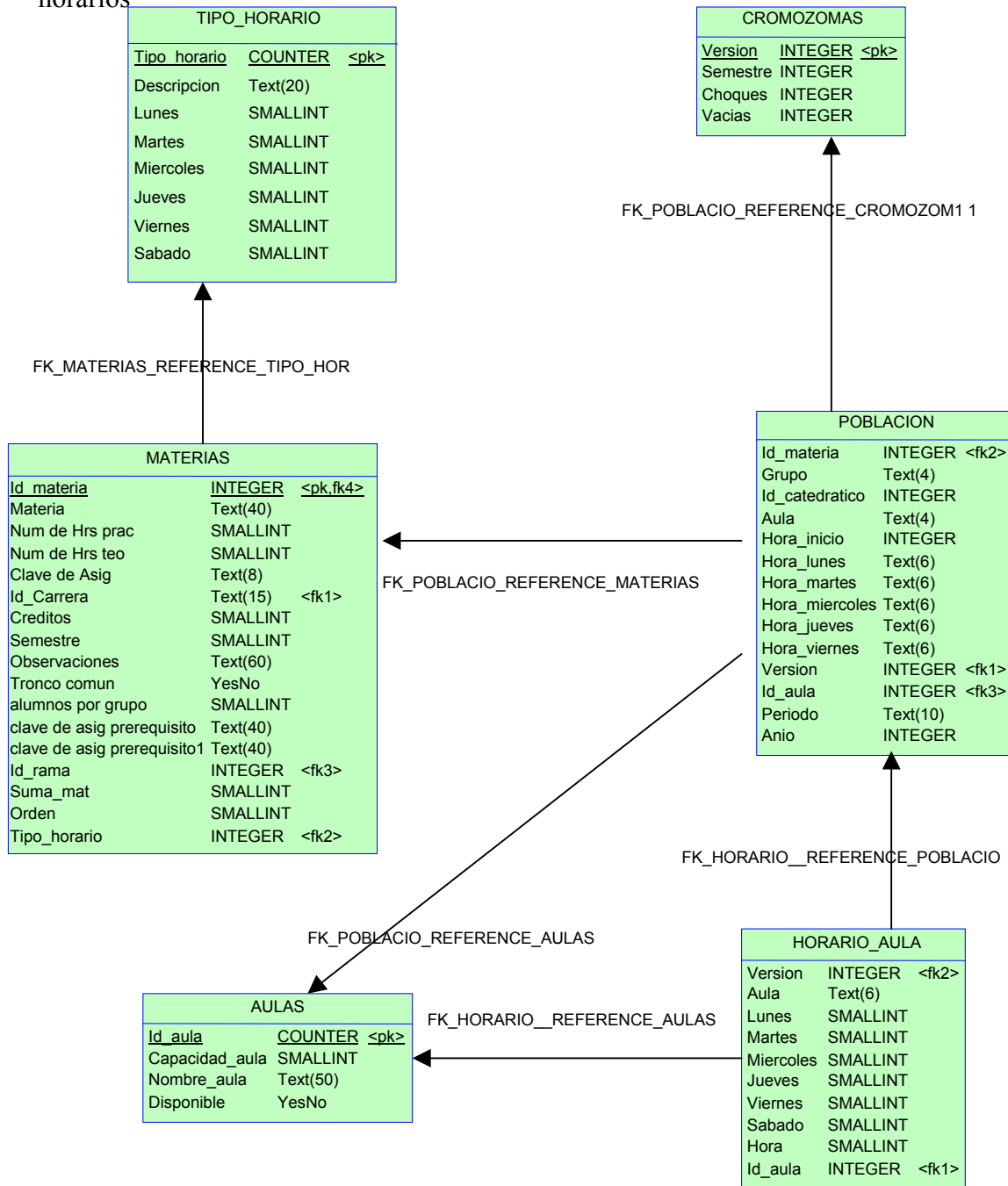


Fig. 4.13 Diagrama E-R del módulo para asignar aulas (Algoritmo Genético)

En esta Figura 4.14 se pueden observar las tablas involucradas en el proceso para generar los paquetes de materias por semestre y la asignación de horario a cada grupo del paquete utilizando la técnica Programación Lógica de Restricciones, nótese que no existe una tabla de restricciones, éstas se aplican por software a cada grupo generado

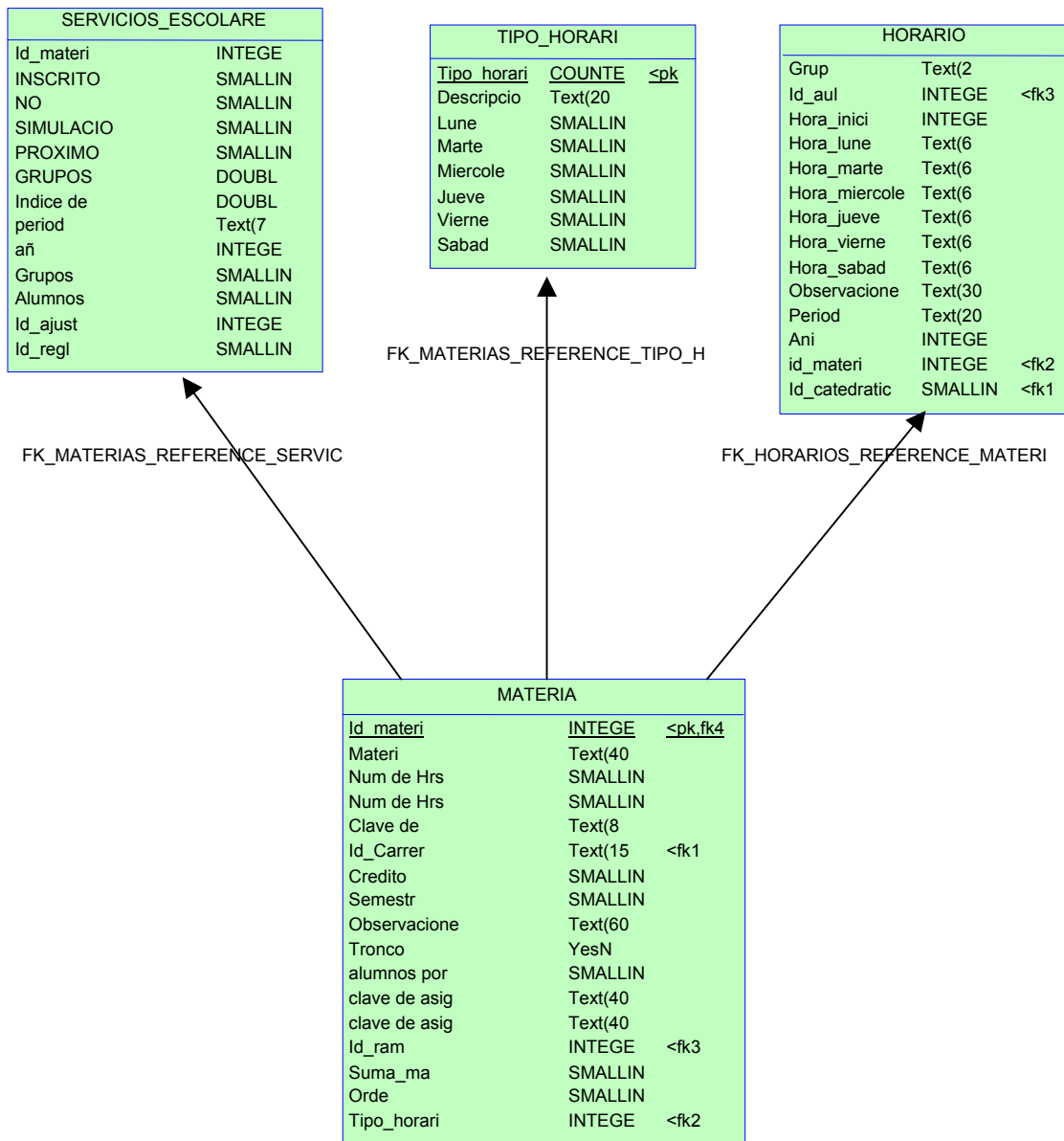


Fig.4.14 Diagrama E-R del módulo para asignar horario (PLR)

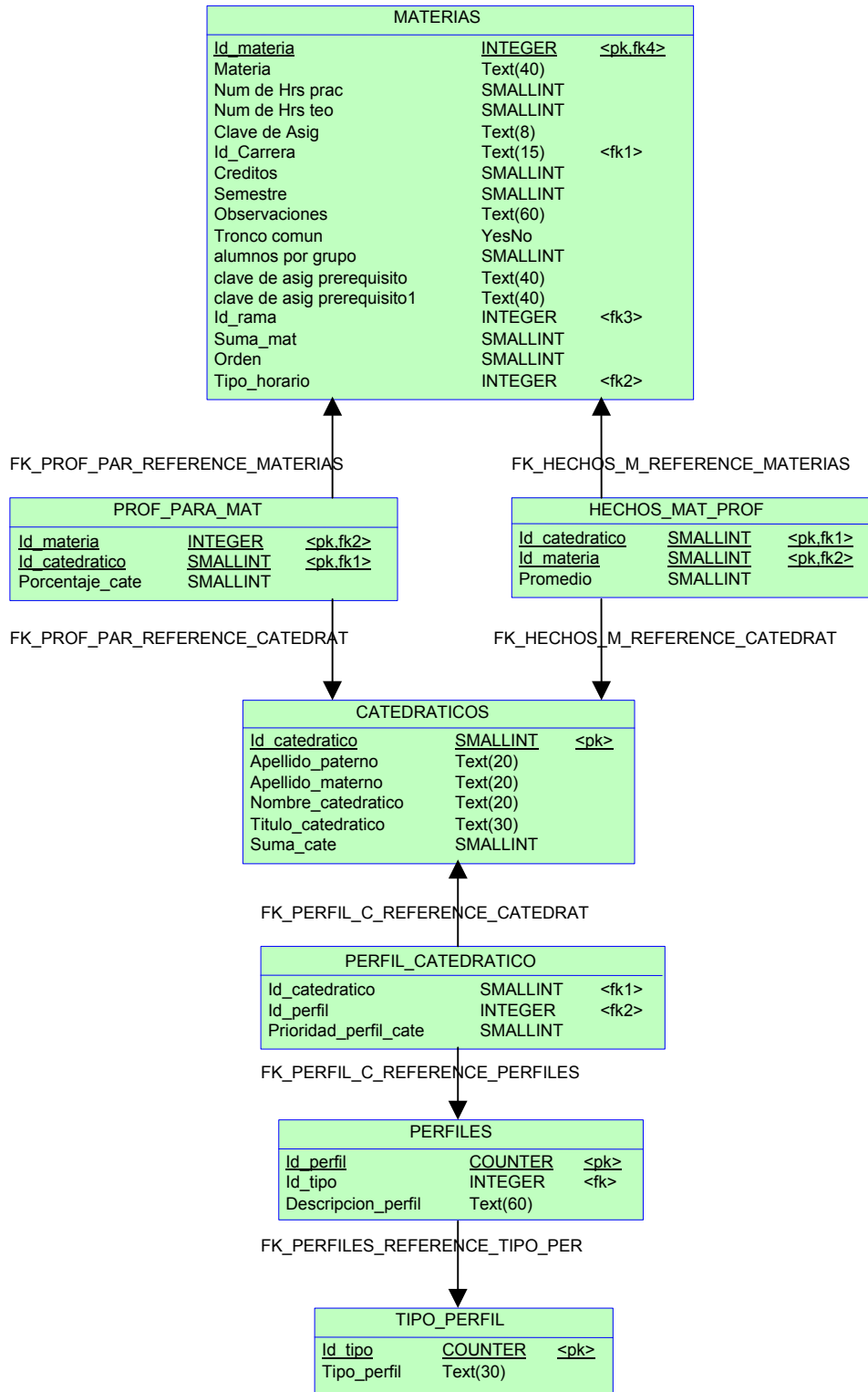


Fig. 4.15 Diagrama E-R del módulo designa mejor catedrático (Sistema Experto)

En las figuras de la página anterior y ésta (Fig. 4.15 y 4.16) se muestran las tablas utilizadas por los Sistemas Expertos para sugerir que materias puede impartir un profesor y también sugiere que profesores pueden impartir una asignatura.

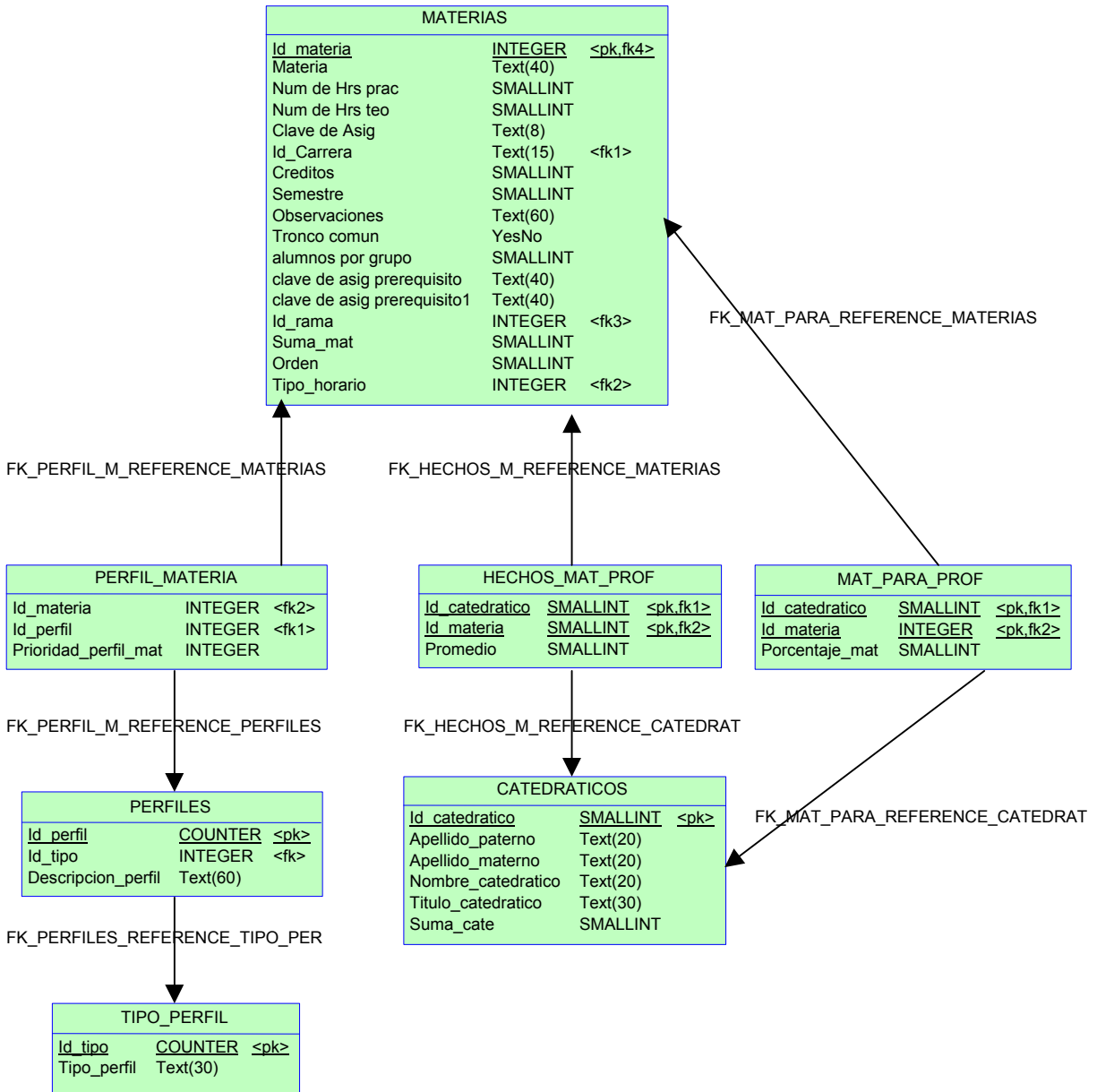


Fig. 4.16 Diagrama E-R del módulo selecciona mejor materia (sistema experto)

### 4.5.5 Diccionario de la Base de Datos

<b>Nombre</b>	HORARIOS_CON_CONOCIMIENTO
<i>Comentario</i>	Base de datos del sistema “Generación de horarios mediante Sistemas Basados en Conocimiento”
<i>DBMS</i>	Microsoft Access 97

#### Lista de Tablas

<i>Nombre</i>	<i>Tipo de Objeto</i>
AJUSTES	Table
AULAS	Table
CARRERAS	Table
CATEDRATICOS	Table
CRITERIOS	Table
CROMOSOMAS	Table
HECHOS	Table
HECHOS_MAT_PROF	Table
HECHOS_PROF_MAT	Table
HECHOS_REGLAS	Table
HISTORIAL_CRITERIOS	Table
HISTORIAL_MATERIA	Table
HORARIO_AULA	Table
HORARIOS	Table
MAT_PARA_PROF	Table
MATERIAS	Table
PERFIL_CATEDRATICO	Table
PERFIL_MATERIA	Table
PERFILES	Table
POBLACION	Table
PROF_PARA_MAT	Table
RAMA_MATERIA	Table
REGLAS	Table
SERVICIOS_ESCOLARES	Table
TIPO_HORARIO	Table
TIPO_PERFIL	Table



**Tabla AJUSTES**

<b>Función</b>	Almacena los datos de los tipos de ajuste que se aplican por el Sistema Experto al dato estadístico que define cuantos grupos se deben abrir de cada materia
----------------	--

**Lista de columnas de la tabla AJUSTES**

<i>Nombre</i>	<i>Tipo de Dato</i>	<i>Long</i>	<i>Llave</i> <i>Primaria Foranea</i>		<i>Comentario</i>
Tipo	Text(25)	25			Nombre del tipo de ajuste
Descripción	Memo				Explicación del ajuste a realizar al número de grupos calculados segun estadísticos
Id_ajuste	COUNTER	4	X		Identificador del ajuste

**Tabla AULAS**

<b>Función</b>	Almacena los datos de las aulas que existen en cada edificio del instituto
----------------	--

**Lista de columnas de la tabla AULAS**

<i>Nombre</i>	<i>Tipo de Dato</i>	<i>Long</i>	<i>Llave</i> <i>Primaria Foranea</i>		<i>Comentario</i>
Id_aula	COUNTER	4	X		Identificador del aula
Capacidad_aula	SMALLINT	4			Número máximo de alumnos en el aula
Nombre_aula	Text(50)	50			Nombre del aula (edificio+numero)
Disponible	YesNo	1			Disponibilidad del aula

**Tabla CARRERAS**

<b>Función</b>	Almacena los datos de las carreras que se imparten en el ITZ
----------------	--

**Lista de columnas de la tabla CARRERAS**

<i>Nombre</i>	<i>Tipo de Dato</i>	<i>Long</i>	<i>Llave</i> <i>Primaria Foranea</i>		<i>Comentario</i>
Id_Carrera	Text(15)	15	X		Clave de la carrera
Nombre_carrera	Text(50)	50			Nombre de la carrera

**Tabla CATEDRATICOS**

<b>Función</b>	Almacena los datos de los CATEDRÁTICOS que imparten clases en las diferentes carreras del Instituto. Se utilizan los datos en los Sistemas Expertos que definen cual es el mejor catedrático para impartir una materia
----------------	--

**Lista de columnas de la tabla CATEDRATICOS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_catedratico	SMALLINT	4	X		Número de tarjeta del catedrático
Apellido_paterno	Text(20)	20			Apellido paterno del catedrático
Apellido_materno	Text(20)	20			Apellido materno del catedrático
Nombre_catedratico	Text(20)	20			Nombre del catedrático
Titulo_catedratico	Text(30)	30			Grado de estudios y título del catedrático
Suma_cate	SMALLINT	4			Suma total de las pesos de los perfiles pertenecientes al catedrático

**Tabla CRITERIOS**

Función	Tabla de los criterios utilizados por el experto para crear las reglas que definen el tipo de ajuste inferido por el Sistema Experto para definición de grupos por abrir
---------	--

**Lista de columnas de la tabla CRITERIOS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_criterio	COUNTER	4	X		Identificador del criterio aplicado a una materia para crear grupos
Descripcion	Text(60)	60			Descripción del criterio

**Tabla CROMOSOMAS**

Función	Almacena los datos de la evaluación de cada versión de horarios generada ya sea por la población inicial o por los operadores genéticos, se genera un cromosoma por cada semestre en cada versión
---------	---

**Lista de columnas de la tabla CROMOSOMAS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Version	INTEGER	4	X		Número de individuo o versión de horarios
Semestre	INTEGER	4			Semestre al que pertenece el dato
Choques	INTEGER	4			Número de choques de versión y semestre
Vacias	INTEGER	4			Número de aulas vacías del individuo

**Tabla HECHOS\_MAT\_PROF**

Función	Esta tabla almacena el resultado de la inferencia del Sistema Experto materias para profesor, indica para cada catedrático las materias y el porcentaje de certeza con que puede impartirlas
---------	--

**Lista de columnas de la tabla HECHOS\_MAT\_PROF**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_catedratico	SMALLINT	4		X	Identificador del catedrático
Id_materia	SMALLINT	4		X	Identificador de la asignatura
Promedio	SMALLINT	4			Porcentaje de certeza con que un profesor puede impartir una materia (resultado de la inferencia)

**Tabla HECHOS\_PROF\_MAT**

Función	Almacena los datos de la inferencia del Sistema Experto que calcula para una materia determinada cuales son los mejores maestros para impartirla
---------	--

**Lista de columnas de la tabla HECHOS\_PROF\_MAT**

Nombre	Tipo de Dato	Longitud	Llave		Comentario
			Primaria	Foránea	
Id_catedratico	SMALLINT	4		X	Identificador del catedrático
Id_materia	SMALLINT	4		X	Identificador de la asignatura
Promedio	SMALLINT	4			Porcentaje de certeza con que una materia puede ser impartida por un profesor (resultado de la inferencia)

**Tabla HECHOS\_REGLAS**

Función	Almacena los porcentajes de certeza con que se dispara una regla del Sistema Experto que define los grupos por abrir
---------	--

**Lista de columnas de la tabla HECHOS\_REGLAS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_criterio	SMALLINT	4		X	Identificador del criterio
Id_regla	SMALLINT	4		X	Identificador de la regla
Promedio	INTEGER	4			Porcentaje de certeza con que una regla dispara un criterio

**Tabla HISTORIAL\_CRITERIOS**

Función	Almacena las reglas formadas por los criterios utilizados por el Sistema Experto para definir los grupos por abrir
---------	--

**Lista de columnas de la tabla HISTORIAL\_CRITERIOS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_criterio	INTEGER	4		X	Identificador del criterio perteneciente a la regla
Id_regla	INTEGER	4		X	Identificador de la regla
Peso de la regla	SMALLINT	4			Peso del criterio en la regla

**Tabla HISTORIAL\_MATERIA**

Función	Tabla que almacena la definición de cada materia con base en los criterios utilizados por las reglas para definir por materia el número de grupos por abrir
---------	---

**Lista de columnas de la tabla HISTORIAL\_MATERIA**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4		X	Identificador de la materia para su asignación de criterios
Id_criterio	INTEGER	4		X	Identificador del criterio que se aplica a la materia para describirla
Prioridad	SMALLINT	4			Peso del criterio en la materia

**Tabla HORARIOS**

Función	Almacena los datos del condensado final de horarios, aquí se almacena el producto final del sistema para su posterior publicación y distribución entre la comunidad tecnológica
---------	---

**Lista de columnas de la tabla HORARIOS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4		X	Identificador de la materia
Grupo	Text(2)	2			Identificador del grupo A, B, C
Id_catedratico	SMALLINT	4		X	Catedrático asignado al grupo
Id_aula	INTEGER	4		X	Aula asignada al grupo
Hora_inicio	INTEGER	4			Hora de inicio del grupo
Hora_lunes	Text(6)	6			Horas que se imparte el grupo el día lunes

Hora_martes	Text(6)	6			
Hora_miercoles	Text(6)	6			
Hora_jueves	Text(6)	6			
Hora_viernes	Text(6)	6			
Hora_sabado	Text(6)	6			
Observaciones	Text(30)	30			Observaciones al grupo
Periodo	Text(20)	20			Periodo Ene-jun o Ago-dic
Anio	INTEGER	4			Año en que se utiliza el horario

**Tabla HORARIO\_AULA**

Función	Tabla auxiliar utilizada como almacenamiento temporal durante el proceso de evaluación en el Algoritmo Genético, guarda los datos de ocupación, choques y no utilización de cada aula, por hora, día y versión
---------	--

**Lista de columnas de la tabla HORARIO\_AULA**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Version	INTEGER	4	X		Número de individuo o versión
Aula	Text(6)	6			Nombre del aula
Lunes	SMALLINT	4			Número de horas que se utiliza el aula el día lunes
Martes	SMALLINT	4			
Miercoles	SMALLINT	4			
Jueves	SMALLINT	4			
Viernes	SMALLINT	4			
Sabado	SMALLINT	4			
Hora	SMALLINT	4			Hora de utilización del aula
Id_aula	INTEGER	4		X	Identificador del aula en la versión

**Tabla MATERIAS**

Función	Esta es una de las principales tablas del proyecto, se utiliza en todos los módulos del sistema, contiene información base para cálculos diversos además contiene también las llaves de diferentes tablas que describen de manera de manera especial a las materias.
---------	--

**Lista de columnas de la tabla MATERIAS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	COUNTER	4	X		Identificación de la materia llave principal
Materia	Text(40)	40			Nombre de la materia
Num de Hrs prac	SMALLINT	4			Número de horas prácticas
Num de Hrs teo	SMALLINT	4			Número de horas teóricas
Clave de Asig	Text(8)	8			Clave oficial de la materia
Id_Carrera	Text(15)	15		X	Carrera a la que pertenece la materia
Creditos	SMALLINT	4			Créditos de la materia

Semestre	SMALLINT	4			Semestre en que se imparte la materia
Observaciones	Text(60)	60			
Tronco comun	YesNo	1			Definición de especialidad o tronco común
alumnos por grupo	SMALLINT	4			Capacidad sugerida de los grupos de esta materia
clave de asig prerequisite	Text(40)	40			Clave de asignatura del prerequisite
clave de asig prerequisite1	Text(40)	40			Clave de asignatura del prerequisite
Id_rama	INTEGER	4		X	Identificador de la rama a la que pertenece la materia
Suma_mat	SMALLINT	4			Suma total de los perfiles pertenecientes a la materia
Orden	SMALLINT	4			Es el número de materia en un paquete
Tipo_horario	INTEGER	4		X	El tipo de horario asignado a materia

**Tabla MAT\_PARA\_PROF**

Función	La tabla almacena para cada catedrático las asignaturas que puede impartir y el grado de certeza con que puede hacerlo, se almacena en esta tabla el resultado de la inferencia del Sistema Experto
---------	---

**Lista de columnas de la tabla MAT\_PARA\_PROF**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_catedratico	SMALLINT	4	X		Número de tarjeta del profesor
Id_materia	INTEGER	4	X		Identificador de la materia
Porcentaje_mat	SMALLINT	4			Porcentaje de certeza con que un profesor puede impartir una materia

**Tabla PERFILES**

Función	Tabla que contiene una relación de todas las características de las materias, características necesarias para poder impartirlas, es al mismo tiempo la tabla que contiene la relación general de los conocimientos que poseen los profesores
---------	--

**Lista de columnas de la tabla PERFILES**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_perfil	COUNTER	4	X		Identificador del perfil para asignaciones de materias a catedráticos
Id_tipo	INTEGER	4		X	Identificador del tipo del perfil
Descripcion_perfil	Text(60)	60			Descripción de una característica del perfil de una materia o un catedrático

**Tabla PERFIL\_CATEDRATICO**

Función	Tabla que contiene el perfil de todos los catedráticos (pueden llamarse las reglas) del Sistema Experto que recomienda profesores para las materias el perfil de un catedrático esta definido por los conceptos que domina de las materias
---------	--

**Lista de columnas de la tabla PERFIL\_CATEDRATICO**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_catedratico	SMALLINT	4		X	Identificador del catedrático para definición de su perfil académico
Id_perfil	INTEGER	4		X	Identificador del perfil que define a un catedrático
Prioridad_perfil_cate	SMALLINT	4			Porcentaje en que contribuye el perfil especificado en la definición del catedrático

**Tabla PERFIL\_MATERIA**

Función	Esta tabla almacena las reglas que definen el perfil de una materia, contiene las características y temas que forman el temario de la materia
---------	---

**Lista de columnas de la tabla PERFIL\_MATERIA**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4		X	Identificador de la materia en la regla de la materia
Id_perfil	INTEGER	4		X	Identificador del perfil perteneciente a una materia
Prioridad_perfil_mat	INTEGER	4			Porcentaje en que contribuye este perfil particular a definir una materia

**Tabla PROF\_PARA\_MAT**

Función	Esta tabla almacena la inferencia del Sistema Experto para determinar cuales son los mejores profesores para impartir una materia, contiene para cada asignatura una lista de los profesores y el porcentaje de certeza con que pueden desempeñarse si la imparten
---------	--

**Lista de columnas de la tabla PROF\_PARA\_MAT**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4	X		Identificador de materia
Id_catedratico	SMALLINT	4	X		Identificador de catedrático
Porcentaje_cate	SMALLINT	4			% de certeza con que el catedrático X puede impartir la materia Y

**Tabla POBLACION**

Función	La tabla es la más importante para el módulo de Algoritmo Genético porque almacena los datos de cada individuo, hipótesis o versión que son generados ya sea por la población inicial aleatoria o por los diversos operadores genéticos, contiene copias (versiones) del condensado de horarios con aulas aleatorias
---------	--

**Lista de columnas de la tabla POBLACION**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4		X	Identificador de la materia
Grupo	Text(4)	4			Grupo de la materia
Id_catedratico	INTEGER	4		X	Identificador del catedratico
Aula	Text(4)	4		X	Nombre del aula (edificio + numero)
Hora_inicio	INTEGER	4			Hora de inicio del grupo
Hora_lunes	Text(6)	6			Hora inicio del lunes
Hora_martes	Text(6)	6			
Hora_miercoles	Text(6)	6			
Hora_jueves	Text(6)	6			
Hora_viernes	Text(6)	6			
Version	INTEGER	4	X		Número de la versión o hipótesis
Periodo	Text(10)	10			Periodo o semestre del grupo
Anio	INTEGER	4			Año del grupo

**Tabla RAMA\_MATERIA**

Función	La tabla almacena la clasificación de las materias de acuerdo a las ciencias a la que pertenecen
---------	--

**Lista de columnas de la tabla RAMA\_MATERIA**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_rama	COUNTER	4	X		Identificador de la rama de una materia (clasificación)
Nombre_rama	Text(30)	30			Descripción de la rama de una materia



**Tabla REGLAS**

Función	Esta tabla almacena las reglas utilizadas para generar los grupos que se deben ofertar el próximo semestre indicando para cada regla el tipo de ajuste que debe aplicarse
---------	---

**Lista de Columnas de la Tabla REGLAS**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Descripcion	Text(70)	70			Descripción de la regla para ajustar el número de grupos
Id_regla	COUNTER	4	X		Identificador de la regla
Id_ajuste	INTEGER	4		X	Identificador del tipo de ajuste que se aplica cuando esta regla se dispara para alguna materia
Tot_peso	INTEGER	4			Total de los pesos de las características de esta regla

**Tabla SERVICIOS\_ESCOLARES**

Función	En esta tabla se almacenan los datos estadísticos de cada materia que provienen de Servicios Escolares para el cálculo del número de grupos que deben abrirse para el próximo semestre, contiene también los parámetros proporcionados por el Sistema Experto para realizar el ajuste al número de grupos calculado inicialmente
---------	--

**Lista de columnas de la tabla SERVICIOS\_ESCOLARES**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_materia	INTEGER	4	X		Identificador de la materia
INSCRITOS	SMALLINT	4			Número de alumnos inscritos actualmente en el grupo
NO INSCRITOS	SMALLINT	4			Número de alumnos que pueden cursar la materia pero no se inscribieron
SIMULACION	SMALLINT	4			Simulación de Servicios Escolares de alumnos que podrán cursar la materia
PROXIMO SEMESTRE	SMALLINT	4			Resultado del calculo de Serv. Escolares de alumnos que cursarán la materia
GRUPOS REQUERIDOS	DOUBLE	4			Grupos requeridos para abrir próximo semestre según cálculos iniciales
Indice de reprobacion periodo	DOUBLE	4			Indice de reprobación de la materia
	Text(7)	7			Periodo de cálculo de horarios actual (semestre ene-jun o ago-dic)

año	INTEGER	4			Año de cálculo de horarios
Grupos ajustado	SMALLINT	4			Número de grupos a solicitar para abrir próximo semestre
Alumnos ajustado	SMALLINT	4			Cálculo final del numero de alumnos del próximo semestre en esta materia
Id_ajuste	INTEGER	4	X		Tipo del ajuste aplicado por el Sistema Experto al campo grupos_requeridos
Id_regla	SMALLINT	4	X		Regla que se disparo para realizar el ajuste

**Tabla TIPO\_HORARIO**

Función	Esta tabla almacena los formatos de los distintos tipos de horario que existen y se ofrecen para cada materia, horarios de 2,3,4,5 días a la semana, y de 1,2,3 o 4 horas por día
---------	---

**Lista de columnas de la tabla TIPO\_HORARIO**

Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Tipo_horario	COUNTER	4	X		Identificador del tipo de horario
Descripcion	Text(20)	20			Descripción del tipo de horario
Lunes	SMALLINT	4			Número de horas del lunes
Martes	SMALLINT	4			Número de horas del martes
Miercoles	SMALLINT	4			Número de horas del miércoles
Jueves	SMALLINT	4			Número de horas del jueves
Viernes	SMALLINT	4			Número de horas del viernes
Sabado	SMALLINT	4			Número de horas del sábado

**Tabla TIPO\_PERFIL**

Función	Aquí se almacena una clasificación de los perfiles de acuerdo a la rama de conocimiento a la que pertenecen
---------	---

**Lista de columnas de la tabla TIPO\_PERFIL**

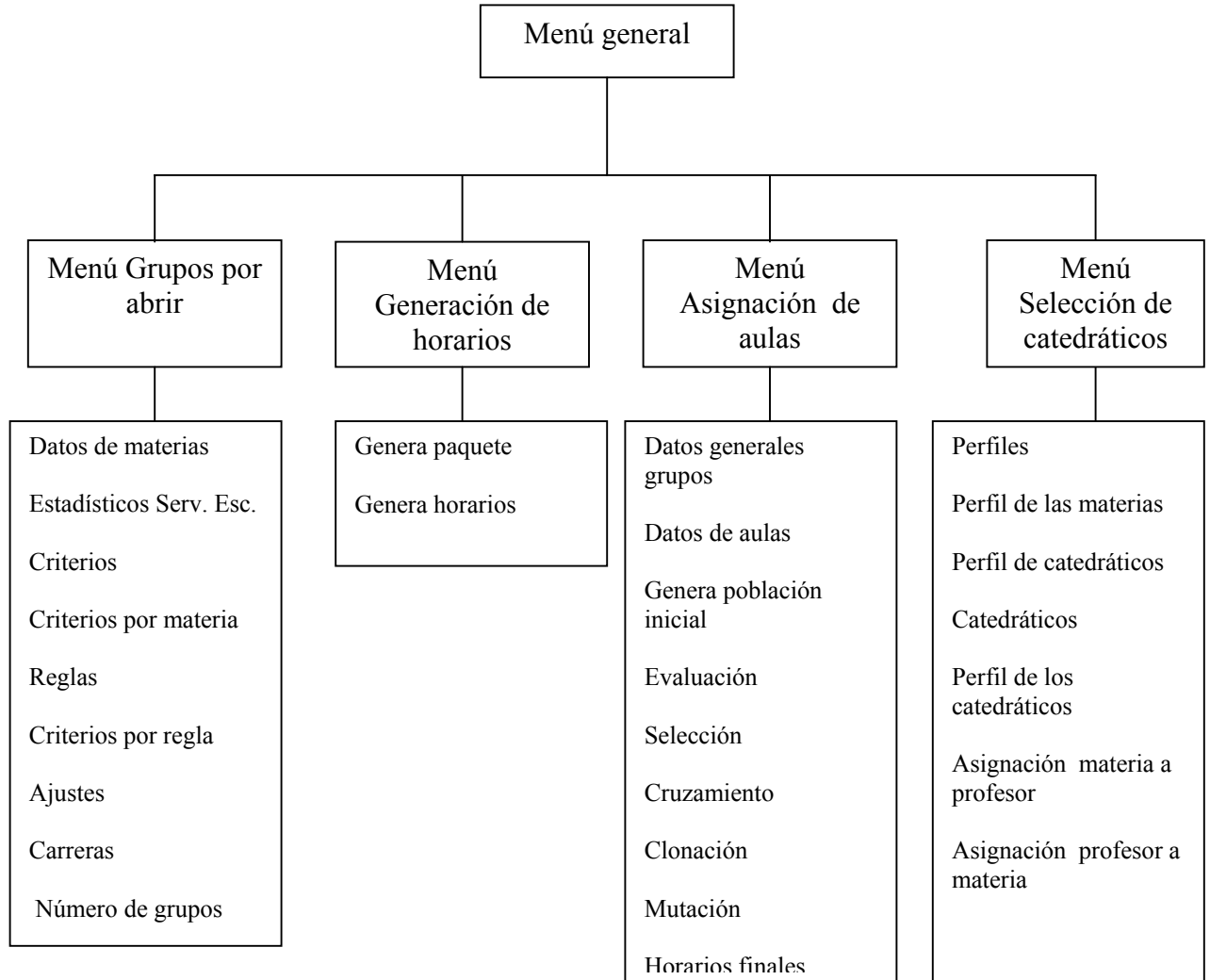
Nombre	Tipo de Dato	Long	Llave		Comentario
			Primaria	Foránea	
Id_tipo	COUNTER	4	X		Identificador del tipo de perfil
Tipo_perfil	Text(30)	30			Descripción del tipo de perfil de los catedráticos o las materias

### 4.5.6 Lista de funciones o programas

Nombre Programa	Descripción del programa
Ajustes	Datos generales de los tipos de ajustes que se aplican al número de grupos por abrir al dato de Servicios Escolares
Aulas	Captura los datos de las aulas disponibles para clase
Captura_horarios	Este programa realiza los cambios posteriores a la generación
Carreras	Captura los datos de las carreras del tecnológico
Catedraticos	Datos generales de los catedráticos
Clonacion	Obtiene una copia (clon) del mejor individuo de una generación
Cons ajuste	Muestra y/o imprime los datos generales de los ajustes
Cons aulas	Emite una pantalla de consulta general de aulas
Cons captura_horarios	Muestra una pantalla de consulta de los horarios generados
Cons carrera	Despliega y/o imprime las carreras
Cons catedratico	Muestra/imprime los datos generales de los catedráticos
Cons criterio	Formulario que muestra y/o despliega los criterios generales
Cons critXmat	Muestra y/o imprime los criterios de una materia
Cons critXregla	Despliega/imprime los criterios que forman una regla
Cons gpos abrir	Explicación de la inferencia realizada (reglas disparadas)
Cons hist_crit	Muestra las reglas de producción de los criterios
Cons materia	Pantalla que despliega y/o imprime datos de las materias
Cons materia_gral	Pantalla de consulta general de los datos de materias
Cons_perf_cate	Consulta/imprime los datos del perfil académico de los catedráticos
Cons_perf_mate	Consulta/imprime los datos del perfil académico de las materias
Cons perfil	Muestra/imprime datos de los perfiles
Cons perfiles	Consulta a la lista de perfiles o características
Cons perfilxmat	Consulta a las reglas de producción del perfil de una materia
Cons rama	Consulta a la lista de ramas (clasificación de las materias)
Cons reglas	Consulta a la base de reglas para ajustar número de grupos
Cons_serv_esc	Consulta a las tablas que tienen información de Servicios Escolares
Cons_serv_escol	Pantalla que muestra y/o imprime los datos estadísticos de Serv. Esc.
Cons tipo_perfil	Consulta a los tipos de perfil (clasificación de perfiles)
Crit X mat	Formulario que captura los criterios pertenecientes a una materia
Crit X regla	Captura los criterios que forman una regla de producción
Criterios	Pantalla que captura el conocimiento de criterios para el S.E.
Criteriosxmat	Pantalla de captura para los criterios por materia
Cruzamiento	Programa que realiza la combinación (cruza) de individuos de la población en el Algoritmo Genético

Evaluacion	Programa que realiza la creación del cromosoma de evaluación al evaluar las versiones de horarios de una generación
Explica_ajustes	Módulo de explicación del porque se aplicó determinado ajuste a una materia
Explica_mat_p_prof	Explicación del porque una materia puede ser impartida por un profesor con cierto grado de certidumbre
Explica_prof_p_mat	Explicación del porque un profesor puede impartir una materia con cierto grado de certidumbre
Genera_horarios	Sistema que utiliza Programación Lógica de Restricciones para generar los paquetes de horarios de un semestre
Genera_paquete	Muestra el paquete generado por el proceso
Grupo_nuevo	Programa que va generando los paquetes
Grupos_por_abrir	Inferencia realizada para definir el ajuste aplicado a los grupos definidos por Servicios Escolares
Historial_criterios	Captura los criterios necesarios para hacer una regla
Historial_materia	Captura los criterios por materia
Mat_para_prof	Inferencia realizada para determinar que profesores pueden impartir una materia
Materias	Actualización al archivo datos generales de las materias
Menu_algoritmos_geneticos	Menú del módulo para asignar aula a los grupos
Menu_generacion_grupos	Menú del Sistema Experto para definir cuantos grupos abrir
Menu_asigna_horario	Menú del módulo asignación de horario
Menu_principal	Menú principal del sistema
Menu_sel_catedra	Menú del Sistema Experto para selección de catedráticos a materia
Mutacion	Programa que hace la modificación de información de aulas de alguna versión (versión)
Perfil_catedratico	Captura el perfil académico de los catedráticos
Perfil_materia	Captura el perfil académico de las materias
Perfiles	Datos generales del perfil académico de materias y catedráticos
Poblacion_inicial	Programa que genera versiones de horarios y que de manera aleatoria le va asignando aula a los grupos
Portada_principal	Es la ventana de entrada al sistema
Prof_para_mat	Inferencia realizada para determinar que materias pueden ser impartidas por un profesor
Rama	Programa que captura las posibles ramas de las materias
Reglas	Datos generales de las reglas de producción para definir el No. de grupos por abrir
Selección	Proceso que selecciona las mejores versiones de horarios (mejores individuos) y elimina las menos aptas
Serv_escolares	Pantalla de captura de los datos estadísticos de Servicios Escolares
Tipo_perfil	Programa que captura la clasificación de los perfiles de materias y profesores

### 4.5.7 Diagrama de Arquitectura (Menús)





# CAPÍTULO 5

## IMPLEMENTACIÓN

### 5.1 Pantallas del sistema

La Figura 5.1 muestra la portada inicial del proyecto. Contiene los logotipos de las tres instituciones participantes y el nombre del proyecto.



Fig 5.1 Portada inicial del proyecto

La Figura 5.2 es la pantalla que proporciona la información de los autores del proyecto.



Fig 5.2 Acerca del sistema

La Figura 5.3 muestra el menú principal del proyecto contiene y 4 módulos. La primera opción incluye los módulos administrativo y generación de grupos, la segunda opción incluye selección de maestro y selección de materia.

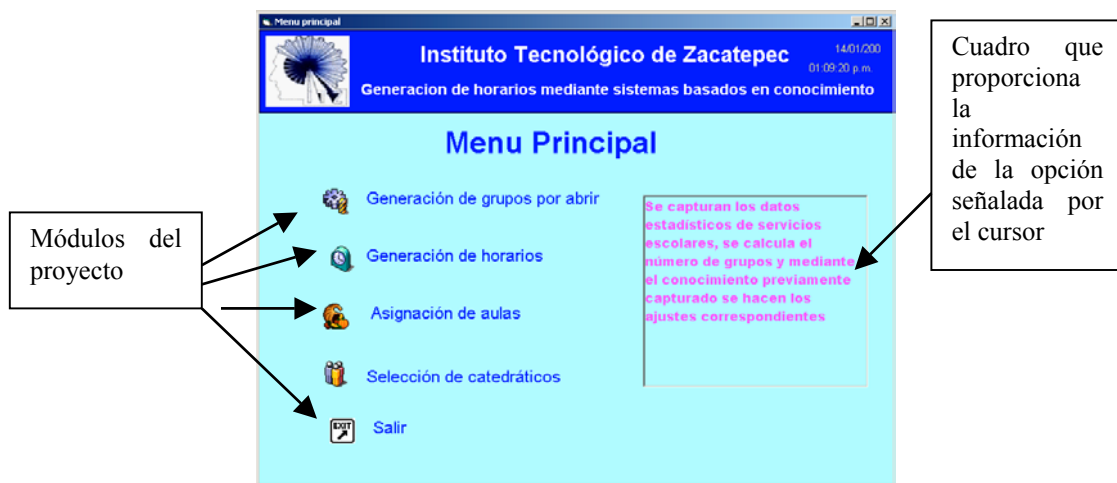


Fig 5.3 Menú principal del proyecto



En la Figura 5.4 se tiene la pantalla del menú que agrupa los programas del módulo administrativo y Sistema Experto para determinar el número de grupos por abrir según los cálculos realizados con los datos de Servicios Escolares y el ajuste a ese dato.



Fig 5.4 Menú grupos por abrir

Este programa del módulo administrativo (Figura 5.5) sirve para capturar la información general de las asignaturas que componen una carrera.

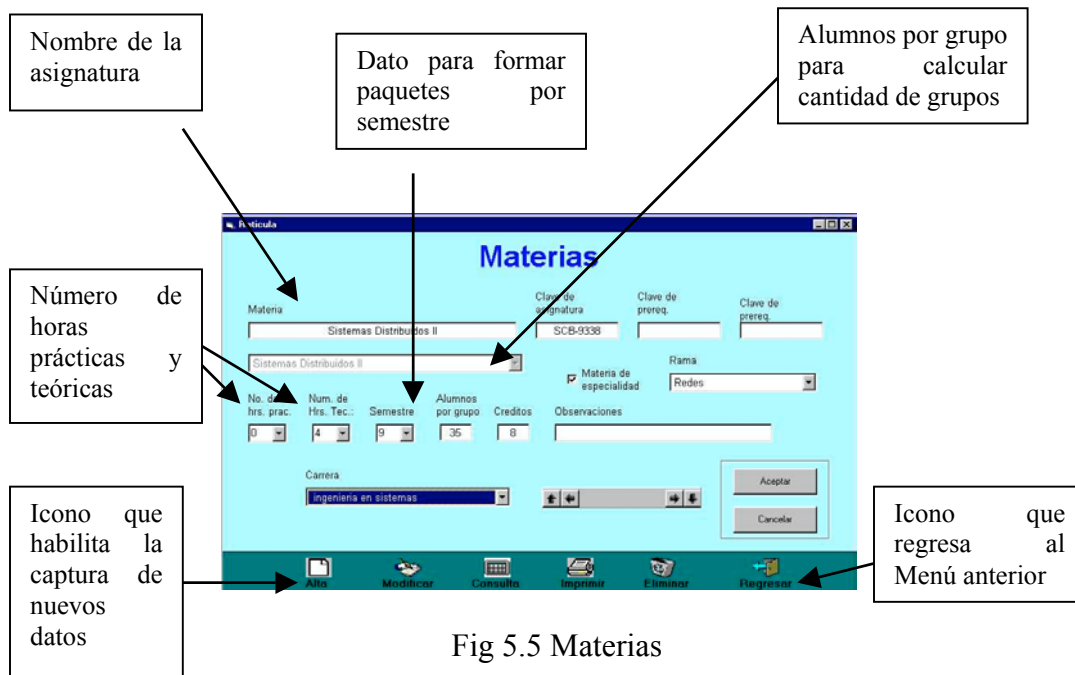


Fig 5.5 Materias

La Figura 5.6 es un ejemplo de la consulta de datos en pantalla.

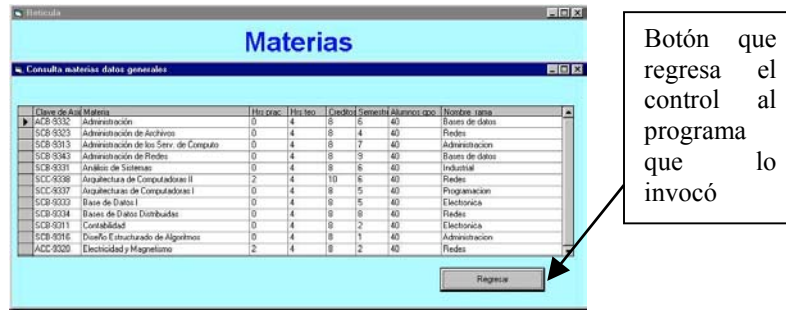


Fig 5.6 Consulta Materias

La Figura 5.7 es la pantalla donde se capturan los datos estadísticos provenientes de Servicios Escolares, contiene la información necesaria para realizar el cálculo original del número de grupos de cada asignatura que se debe abrir para ofertarse a los alumnos el siguiente semestre, la fórmula mostrada proporciona tal dato.

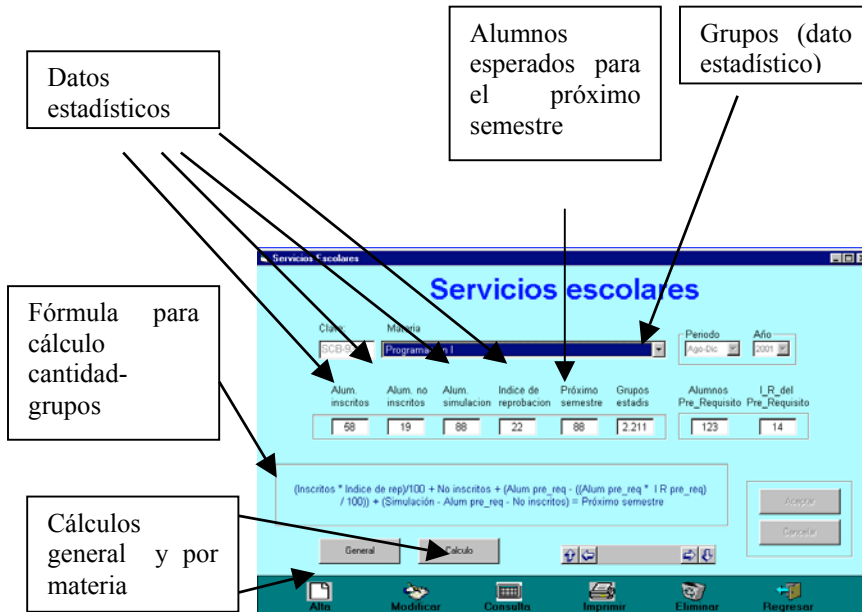


Fig 5.7 Pantalla Servicios Escolares

En la Figura 5.8 se muestra la pantalla de consulta de los datos estadísticos proporcionados por Servicios Escolares.

Clave	Materia	Inscritos	No inscritos	Simulación	Ind. de rep.	Prox. sem.	Gpos. req.
SCB-9316	Diseño Estructurado de Algoritmos	123	300	300	14	300	7.5
SCB-9324	Lenguajes y Autómatas	30	19	152	5	104	2.6
SCB-9317	Programación I	58	19	88	22	88	2.2
SCB-9328	Ingeniería de Software	121	63	123	21	136	3.9
SCB-9325	Programación de Sistemas I	40	28	94	20	100	2.5
SCB-9334	Bases de Datos Distribuidas	120	44	156	52	217	5.4
ACB-9336	Ingeniería Económica	92	27	107	6	111	2.8
ACC-9320	Electricidad y Magnetismo	91	4	87	17	101	2.5
SCB-9318	Programación II	117	30	132	18	115	2.9
SCC-9335	Sistemas Digitales	56	66	171	22	162	4.1
SCB-9315	Introducción a la Ing. en Sist. Comp.	171	48	154	39	221	5.5

Fig 5.8 Pantalla consulta Servicios Escolares

Esta pantalla captura las características por materia que dan origen al conocimiento necesario para definir el tipo de ajuste que se aplica al dato del número de grupos proporcionado por los cálculos estadísticos de la pantalla anterior.

**Criterios por materia**

Materias: Ingeniería Económica

Criterios por materia	Prioridad
Asignatura pertenece a semestres menores o iguales a 7	31
Asignatura no tiene prerrequisitos	46
Asignatura tiene requisito de creditos minimo	66
Asignatura tiene nivel de complejidad medio	71

Seleccione el criterio: Asignatura tiene bajo indice de reprobación

Elija la prioridad: 40

Buttons: Aceptar, Cancelar

Callouts:
 

- Criterios de una materia (points to the list of criteria)
- Grado de importancia de esa característica para la (points to the priority column)

Fig 5.9 Pantalla criterios por materia

En la Figura 5.10 se muestra un ejemplo de la consulta a la base de datos del sistema en la parte de criterios por materia.

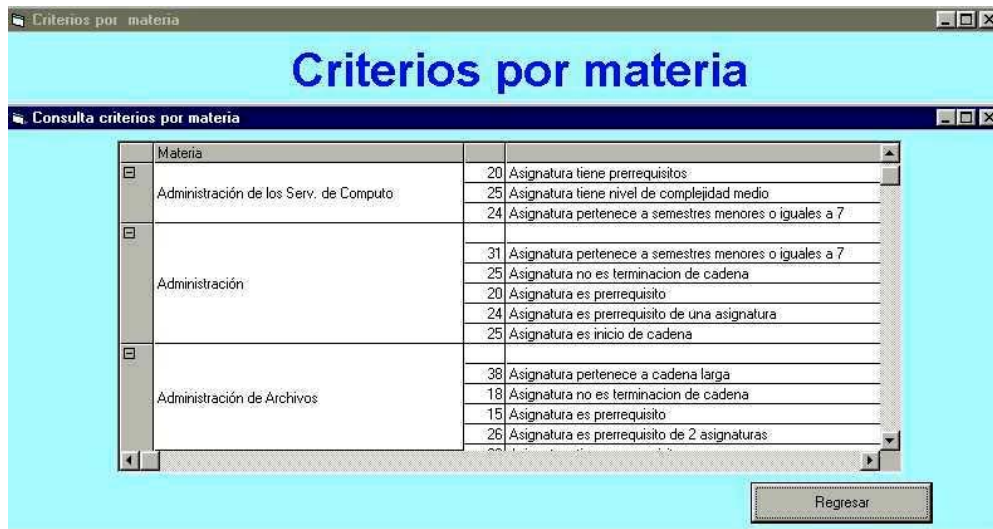


Fig 5.10 Pantalla consulta criterios por materia

Esta pantalla (fig. 5.11) proporciona al sistema la capacidad de capturar, modificar, borrar e imprimir las carreras del Instituto Tecnológico de Zacatepec

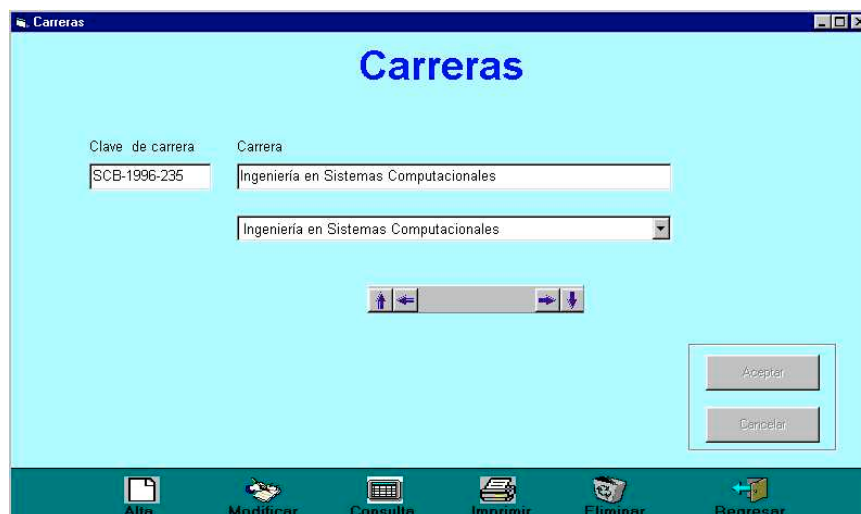


Fig 5.11 Pantalla carreras

La Figura 5.12 proporciona al sistema la capacidad para capturar los criterios (conocimientos) del experto para definir a cada materia.

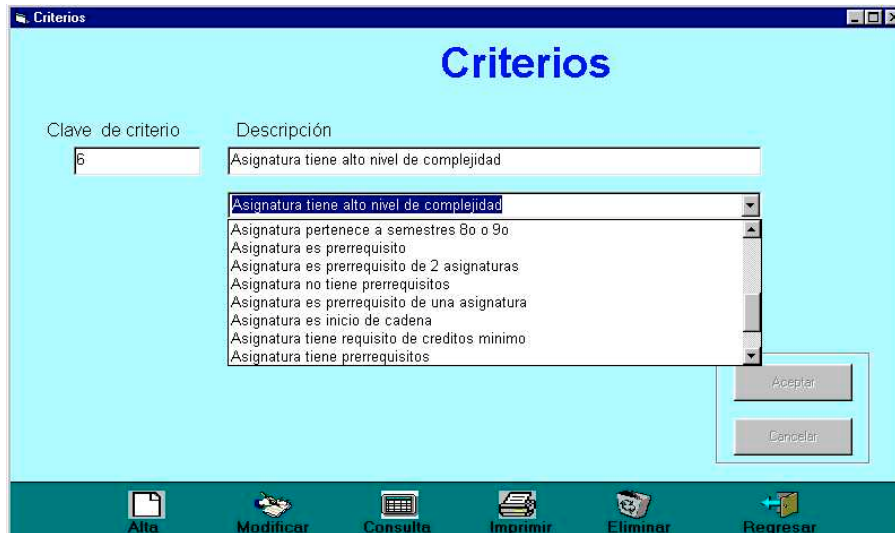


Fig 5.12 Pantalla criterios

La Figura 5.13 muestra un ejemplo de consulta al archivo de los criterios capturados en el sistema.

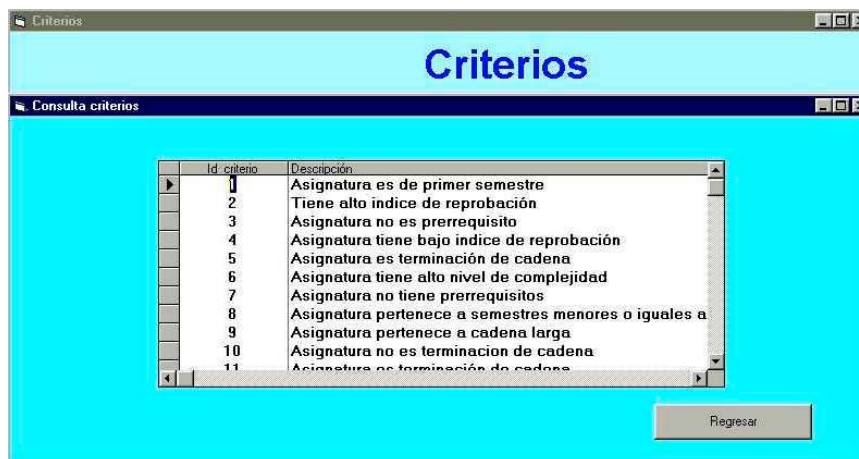


Fig 5.13 Pantalla consulta criterios

La Figura 5.14 muestra la pantalla de captura de las reglas que componen el Sistema Experto para determinar el tipo de ajuste que se aplicará al dato estadístico del número de grupos que se van a abrir el próximo semestre, cada regla tiene asignado un tipo de ajuste.



Fig 5.14 Pantalla reglas

La Figura 5.15 muestra un ejemplo de la consulta al archivo de las reglas que utiliza el Sistema Experto para determinar número de grupos por ofertar el próximo semestre.

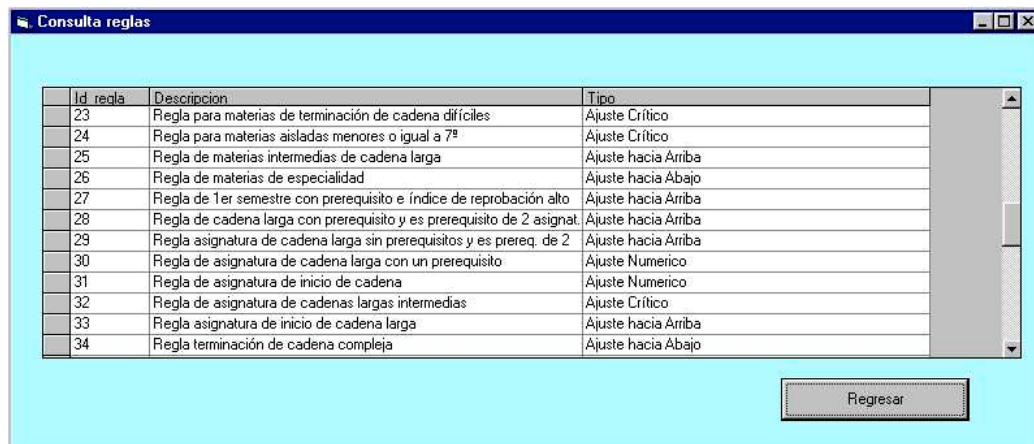


Fig 5.15 Pantalla consulta reglas

La Figura 5.16 muestra la pantalla de captura al archivo de ajustes, nótese que la descripción explica en extenso la acción que se realiza al aplicar este ajuste proveniente de una regla disparada por los criterios pertenecientes a la definición de una materia.



Fig 5.16 Pantalla ajustes

La Figura 5.17 muestra la pantalla de la consulta a los ajustes que contiene el sistema.

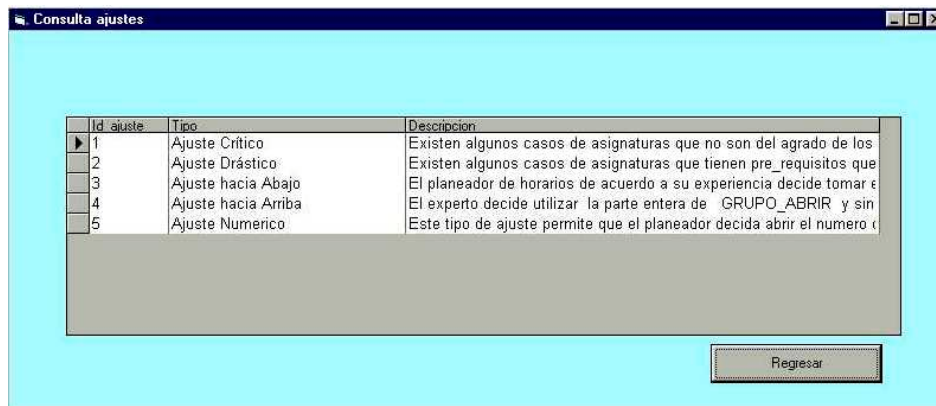


Fig 5.17 Pantalla consulta ajustes



La Figura 5.18 muestra la pantalla que captura los criterios para cada regla. Una regla está formada por el mismo tipo de criterios que definen a las materias.

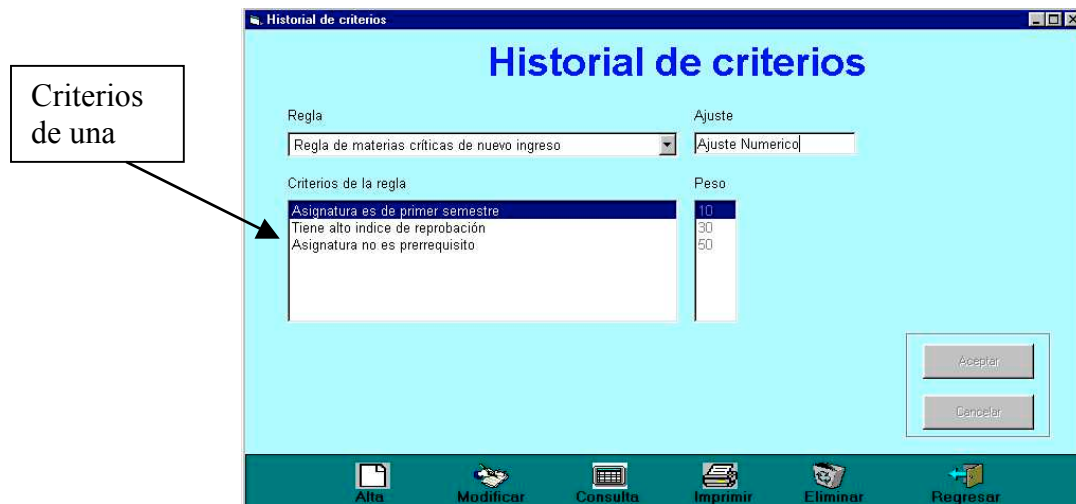


Fig 5.18 Pantalla criterios por regla

La Figura 5.19 muestra el ejemplo de una consulta de los criterios que definen a una regla.



Fig 5.19 Pantalla consulta criterios por regla



La Figura 5.20 es la pantalla más importante de módulo de generación de grupos porque es donde se realiza la inferencia para calcular el tipo de ajuste que se va a aplicar a cada materia, con base en las reglas que se disparan y que hacen *match* los criterios de una regla y los criterios de una materia

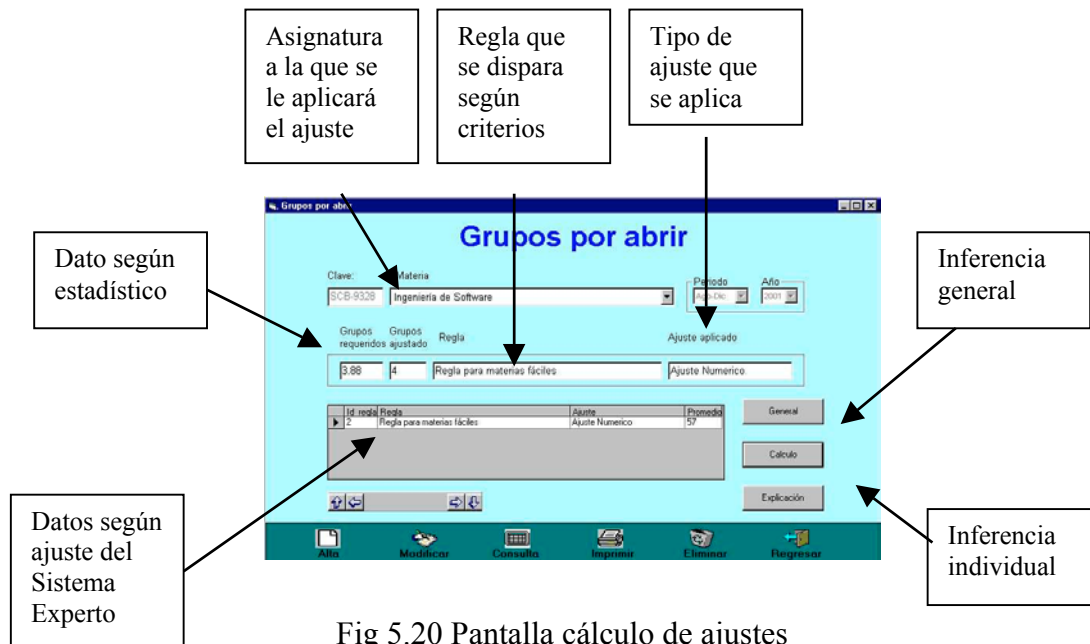


Fig 5.20 Pantalla cálculo de ajustes

Esta pantalla (Figura 5.21) ofrece una explicación mostrando los criterios que definen una materia y su correspondiente regla disparada, mostrando los criterios de la regla que hacen *match* con los criterios de la materia.

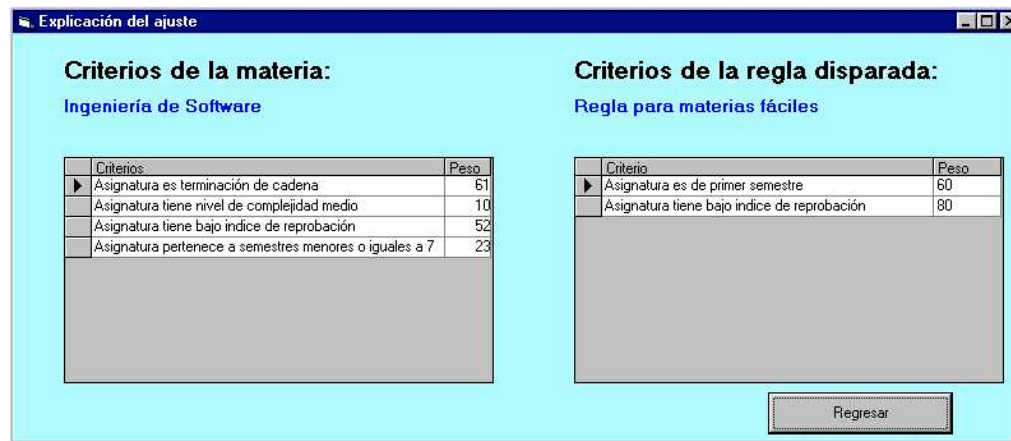


Fig 5.21 Pantalla explicación de la inferencia

La Figura 5.22 muestra el formulario que forma los paquetes de materias según la retícula de una carrera, muestra el número de grupos por cada materia que se van a formar y hace los paquetes asignando los grupos AA, AB, AC, AD.

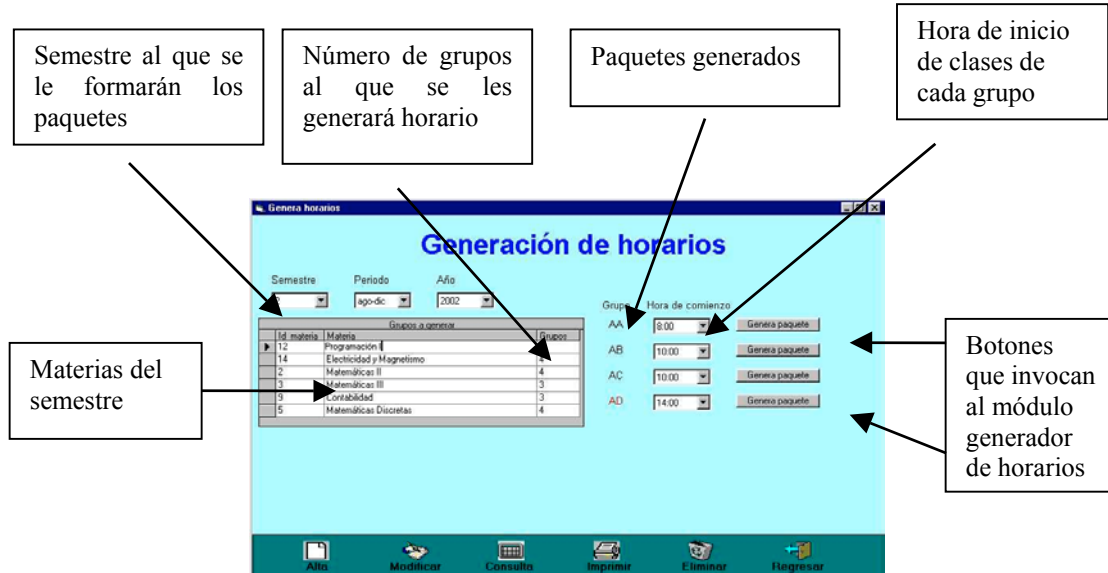


Fig 5.22 Pantalla inicial del módulo generación de horarios

La Figura 5.23 es el formulario del módulo Programación Lógica de Restricciones que genera los horarios por paquete satisfaciendo las restricciones para formar un horario libre de choques, continuo (sin horas muertas) cumpliendo las horas por materia, etc.

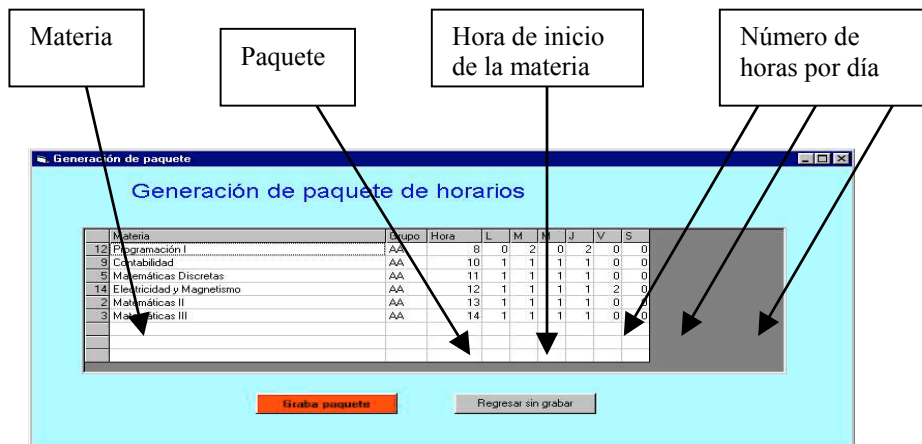


Fig 5.23 Pantalla principal del módulo generación de horarios

La Figura 5.24 muestra el módulo para asignar aula a los horarios que ya tienen materia, grupo y hora.

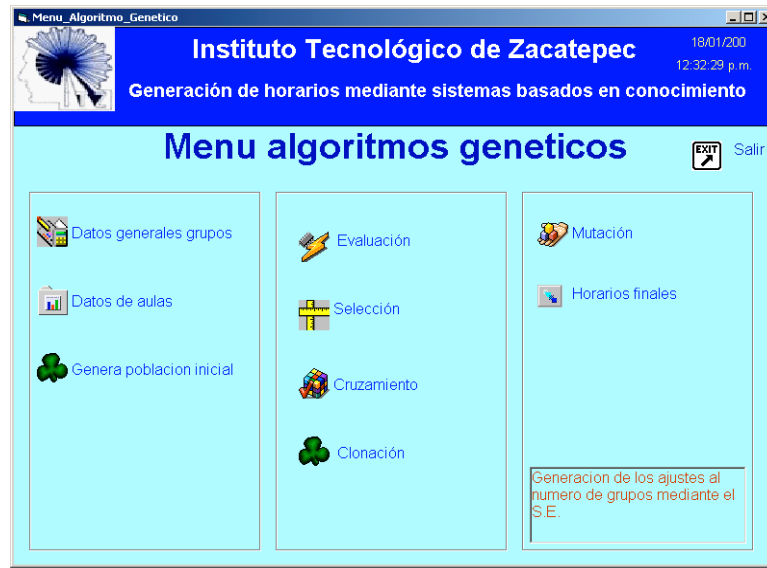


Fig 5.24 Pantalla principal del menú Algoritmos Genéticos

Esta pantalla (Figura 5.25) se utiliza para hacer altas, bajas, consulta y modificaciones a los grupos que se les va a dar aula.

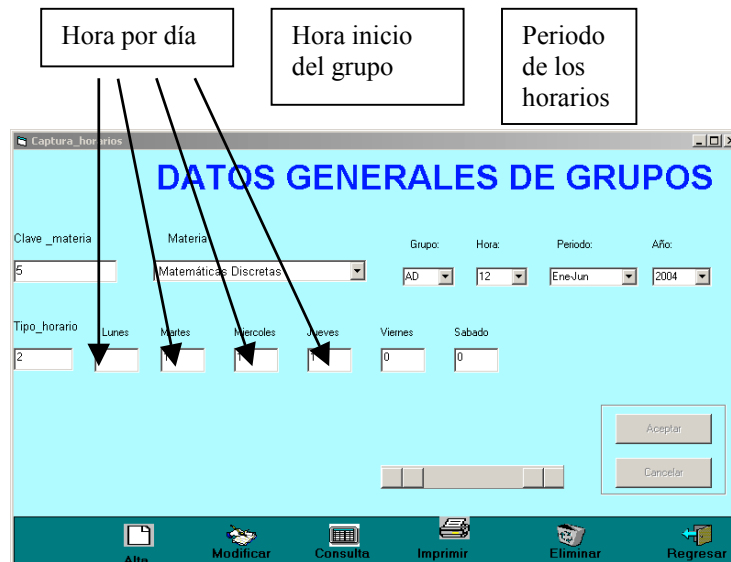


Fig 5.25 Pantalla de administración de grupos

La Figura 5.26 muestra una consulta de la pantalla administración de grupos.

Id materia	Materia	Tipo horario	Grupo	Periodo	Año	Id aula	Hora in
27	Lenguajes y Automata	2	AA	Ene-Jun	2003	16	18
56	Sistemas Operativos	2	AA	Ene-Jun	2003	16	18
5	Matemáticas Discretas	2	AA	Ene-Jun	2003	16	12
71	Teoría de Señales	2	AB	Ene-Jun	2003	16	10
71	Teoría de Señales	2	AC	Ene-Jun	2003	16	12
71	Teoría de Señales	2	AA	Ene-Jun	2003	16	14
74	Nuevas Tecnologías	2	AC	Ene-Jun	2003	16	16
74	Nuevas Tecnologías	2	AA	Ene-Jun	2003	16	7
58	Arquitectura de Computadores	3	AA	Ene-Jun	2003	16	8
58	Arquitectura de Computadores	3	AB	Ene-Jun	2003	16	12
21	Administración de Empresas	2	AA	Ene-Jun	2003	16	14

Regresar

Fig 5.26 Pantalla de consulta de administración de grupos

La Figura 5.27 muestra la pantalla para actualizar las aulas disponibles por las personas que generan los horarios, estas aulas están disponibles para que el Algoritmo Genético las utilice para asignarles grupos durante el proceso.

**Aulas**

Clave de aula:     Edificio y número de aula:     Capacidad:     Disponible:

↑ ←    → ↓

Fig 5.27 Pantalla del control de aulas

La Figura 5.28 muestra la consulta de las aulas que utiliza el módulo generación de aulas.

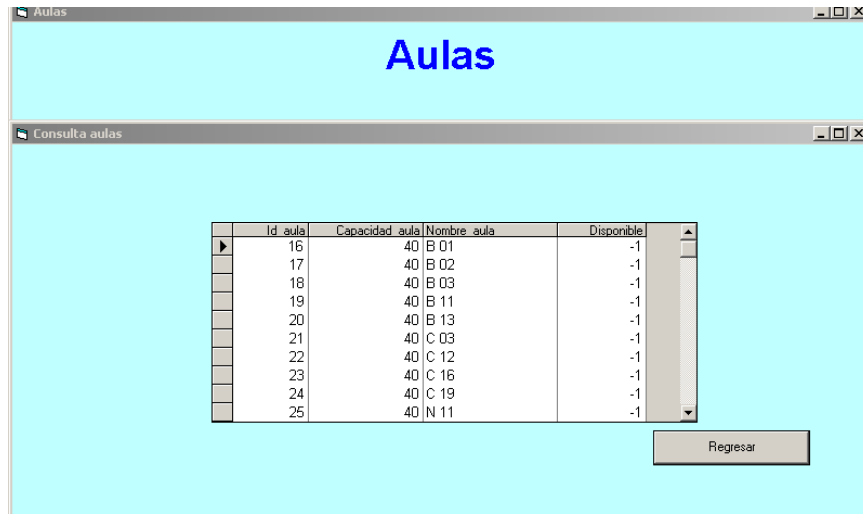


Fig 5.28 Pantalla de consultas del control de aulas

La Figura 5.29 muestra la pantalla que genera la población inicial a partir de los datos generales proveniente de los módulos anteriores.

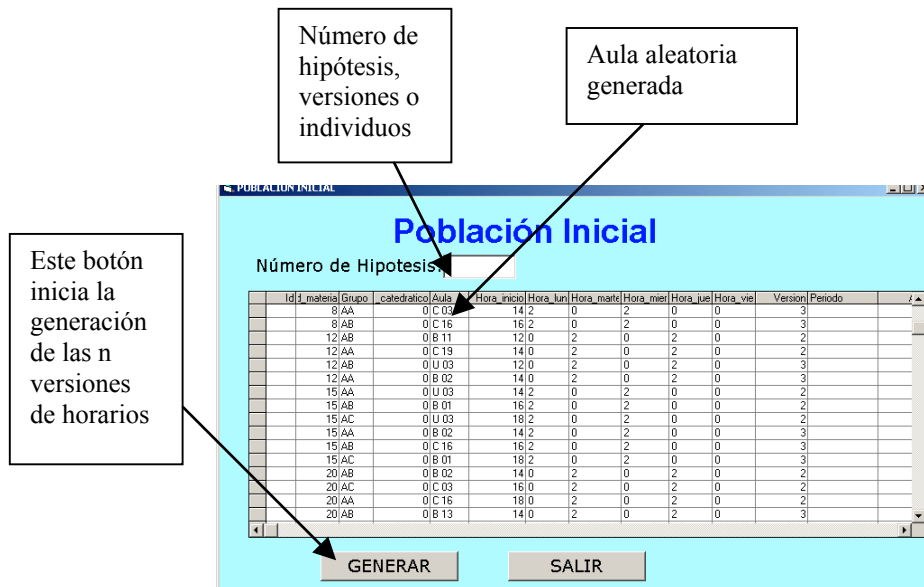


Fig 5.29 Pantalla de genera población inicial

La Figura 5.30 muestra la pantalla que evalúa las versiones originadas y genera para cada una de ellas un cromosoma cuyos genes contienen la información obtenida

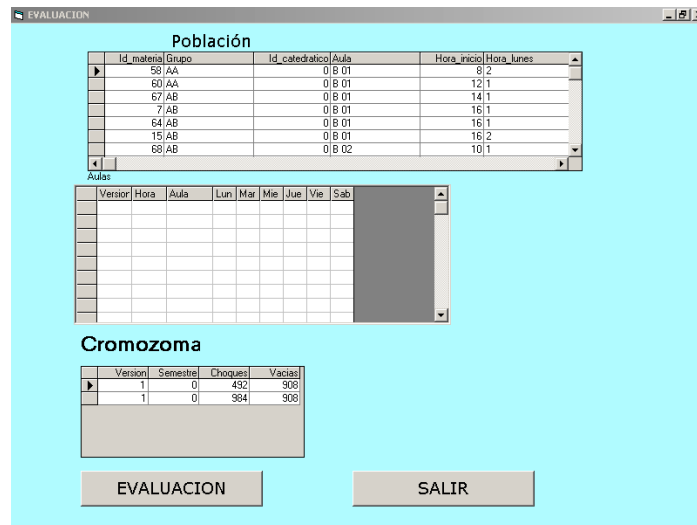


Fig 5.30 Pantalla de evaluación de versiones

La Figura 5.31 muestra el formulario del proceso de selección de las mejores versiones de horarios, toma aquellas que tienen menor numero de choques y mayor espacio de ocupación de aulas

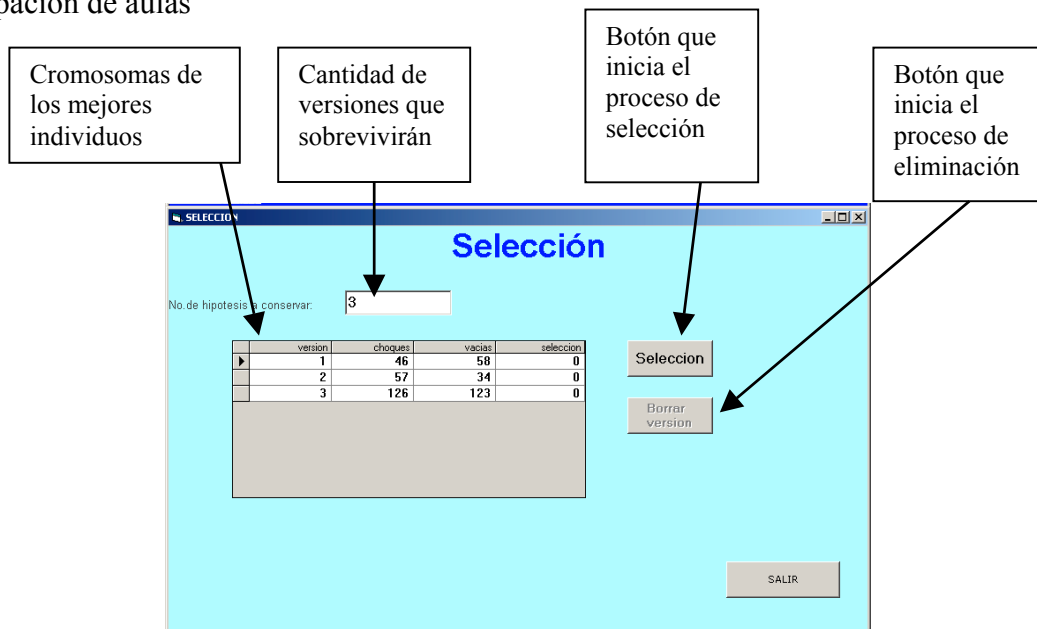


Fig. 5.31 Pantalla del proceso de selección

La Figura 5.32 muestra la pantalla del proceso de clonación donde se selecciona la mejor versión y se obtiene un clon idéntico con un nuevo número de versión.

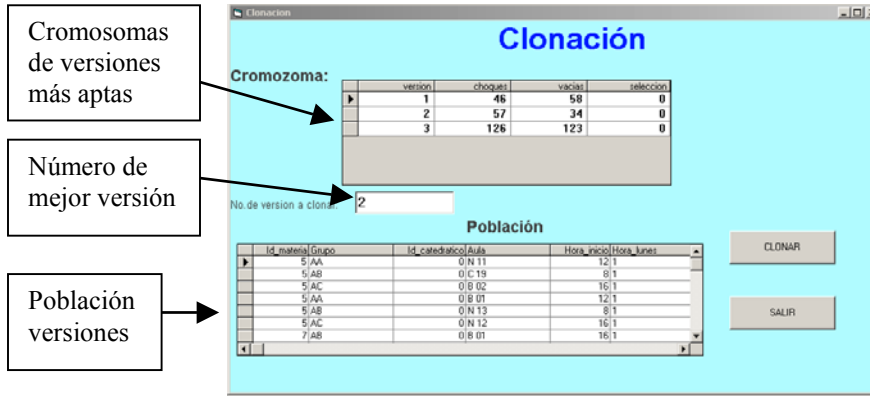


Fig. 5.32 Pantalla del proceso de clonación

La Figura 5.33 muestra la pantalla del proceso de mutación utilizado para hacer el cambio “mutar” los datos de una versión para romper algún ciclado de la tendencia hacia el resultado o simplemente para acelerar el proceso de optimización

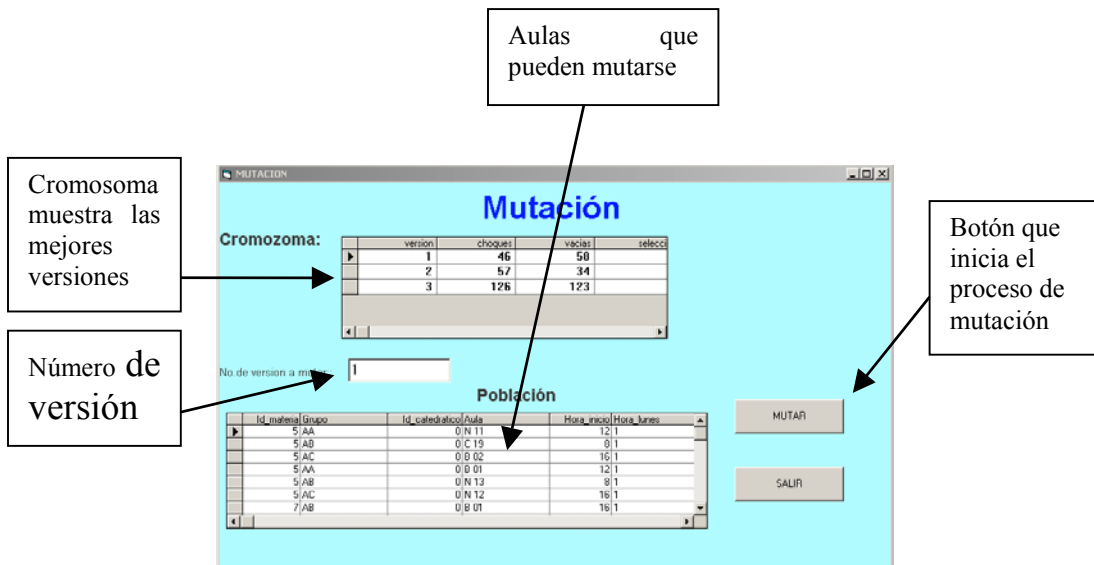


Fig. 5.33 Pantalla de mutación

La Figura 5.34 muestra la pantalla donde se realiza el cruzamiento utilizando plantillas predeterminadas dirigiendo la cruce de versiones por paquetes de materias agrupadas por semestre y cada plantilla genera una nueva versión con la combinación de fragmentos de varias versiones.

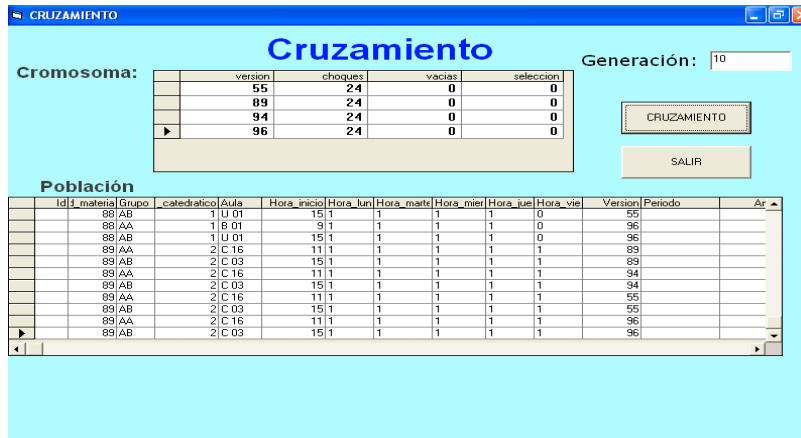


Fig. 5.34 Pantalla del proceso de cruzamiento

La Figura 5.35 permite la selección y envío de la mejor versión de horarios y adaptarlo al módulo administrativo para su distribución y publicación.

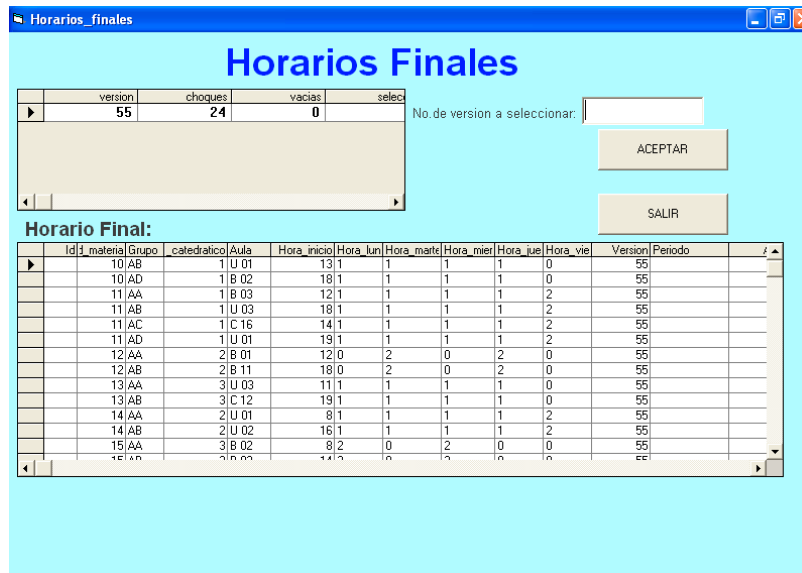


Fig. 5.35 Horarios finales



La Figura 5.36 muestra el menú de los módulos que sugiere maestros para impartir una asignatura y sugerencia de asignatura para algún maestro.



Fig 5.36 Pantalla menú selección de catedráticos

La Figura 5.37 muestra la pantalla para la captura de características de los conocimientos requeridos para dar una materia, que son las mismas características que definen los conocimientos de un profesor para impartir las materias.

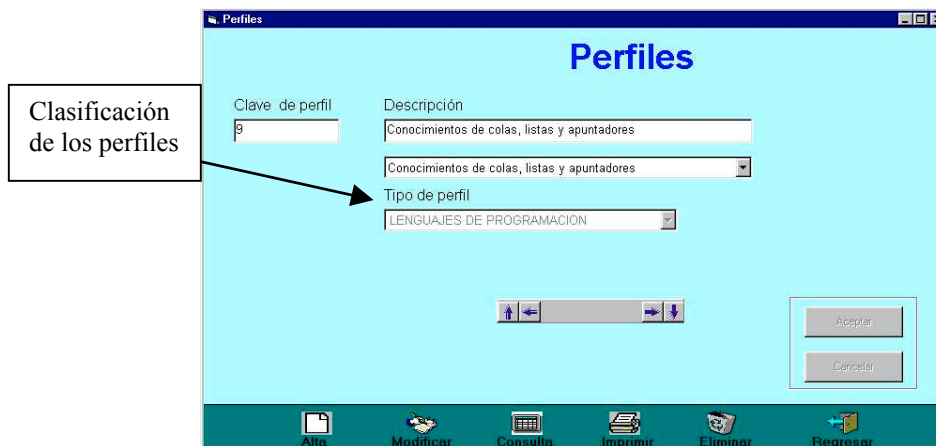


Fig 5.37 Pantalla de perfiles

La Figura 5.38 muestra una consulta de los perfiles agrupados por tipo.

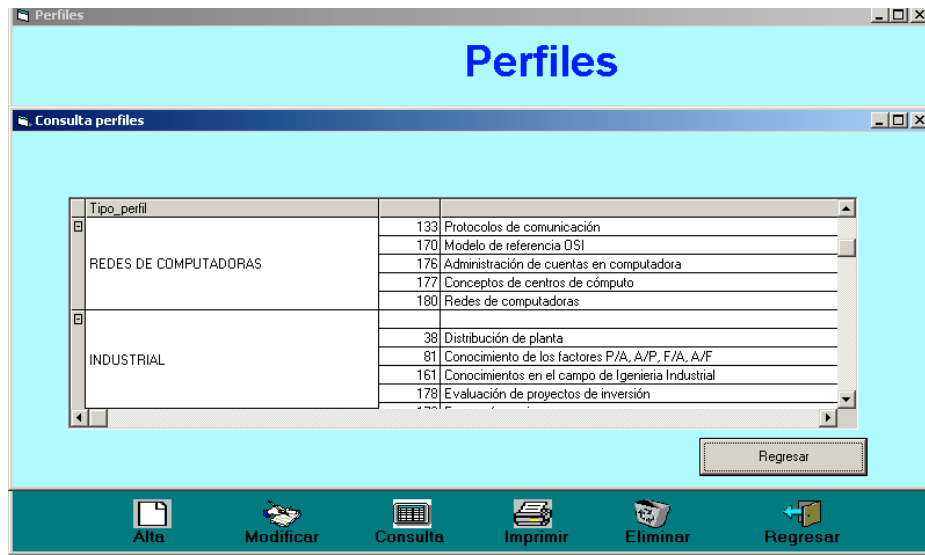


Fig 5.38 Pantalla de consulta perfiles

La Figura 5.39 muestra la pantalla en la que se definen los conocimientos requeridos para poder impartir una asignatura de acuerdo al perfil y contenido temático.

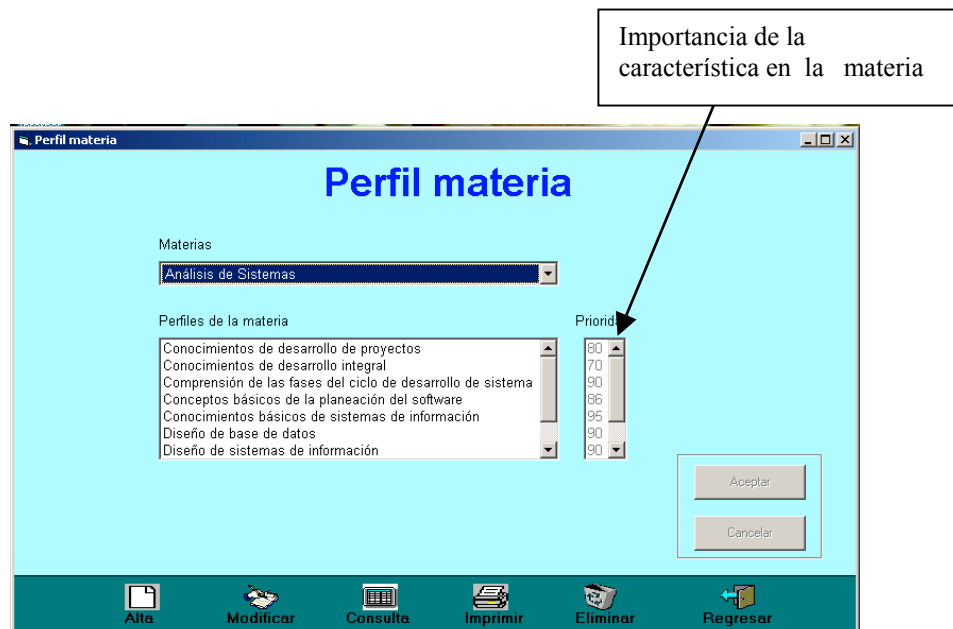


Fig 5.39 Pantalla perfiles de las materias

La Figura 5.40 muestra un ejemplo de la consulta al perfil de una materia.

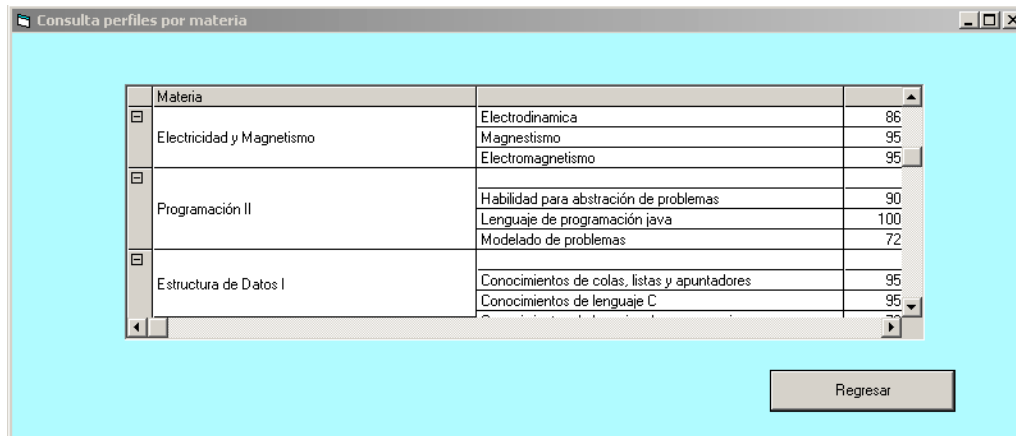


Fig 5.40 Pantalla consulta perfiles de las materias

La Figura 5.41 es utilizada para capturar los datos de los catedráticos que imparten las asignaturas de una carrera



Fig 5.41 Pantalla captura datos de catedráticos

La Figura 5.42 es una muestra de la consulta al archivo de los catedráticos

# Cred.	Apellido paterno	Apellido materno	Nombre catedratico	Titulo catedratico
148	Botello	Uribe	Guadalupe	Lic. en Administracion
151	Lagunas	Ocampo	Oscar	Ing. Quimico
168	Lara	Acono	Noemí	Lic. en Informatica
329	Velazquez	Velazco	MARco Aurelio	Lic. en Informatica
361	Rodriguez	Zavaleta	Estela	Lic. en Informatica
362	Aranda	Benitez	Borisito	Lic. en Computación
363	Sanchez	Cadena	Olivia	Ing. Sistemas Computacionales
366	Rojas	Vilchis	Guadalupe	Lic. en Informatica
370	Leyva	Adan	Sabina	Ing. Industrial

Fig 5.42 Pantalla consulta datos de catedráticos

La Figura 5.43 muestra la pantalla en que se captura el perfil de los catedráticos, el perfil son los conocimientos que tiene el profesor que lo capacitan para impartir alguna asignatura

Conocimientos del profesor

Grado de dominio del tema

Barra de selección de perfiles

Perfiles del catedrático	Prioridad
Conocimientos de lenguaje C	50
Materia requiere conocimientos de algebra lineal	75
Conocimientos de ensamblador	75
Dominio del Algebra	100
Manejo de la geometria	100
Conocimientos de circuitos lógicos	100
Circuitos digitales	100

Selección de perfiles:

Fig 5.43 Pantalla captura perfil de catedráticos

La Figura 5.44 muestra una consulta típica al archivo de perfiles de los catedráticos



Fig 5.44 Pantalla consulta perfil de catedráticos

La Figura 5.45 muestra la pantalla de inferencia final que sugiere quienes son los catedráticos más capacitados para impartir una materia, haciendo la inferencia mediante la búsqueda de *matches* entre el perfil de la asignatura y el perfil del catedrático.

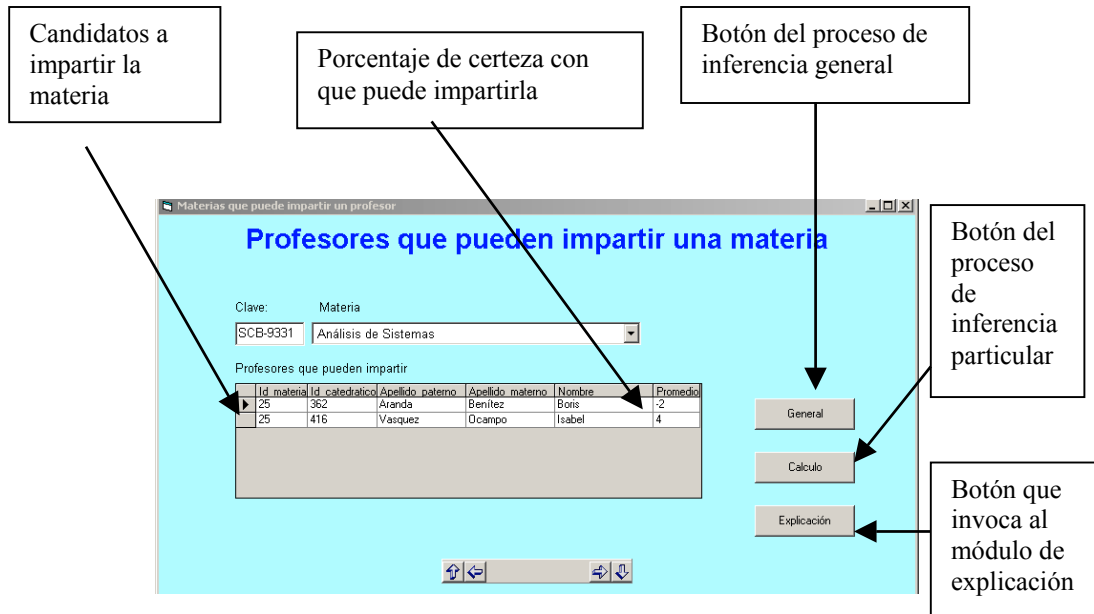


Fig 5.45 Pantalla muestra inferencia catedráticos para materia

La Figura 5.46 muestra la pantalla de explicación donde se indican los perfiles de la materia y el perfil del profesor.

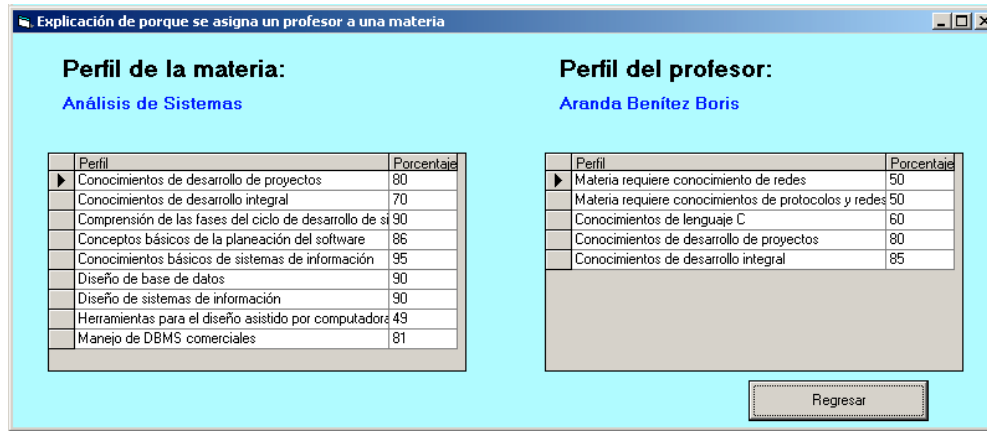


Fig 5.46 Pantalla explicación porque un maestro puede dar una materia

La Figura 5.47 muestra el resultado de la inferencia del proceso que sugiere las materias que puede impartir un profesor con base en el perfil de la materia y el perfil del profesor.

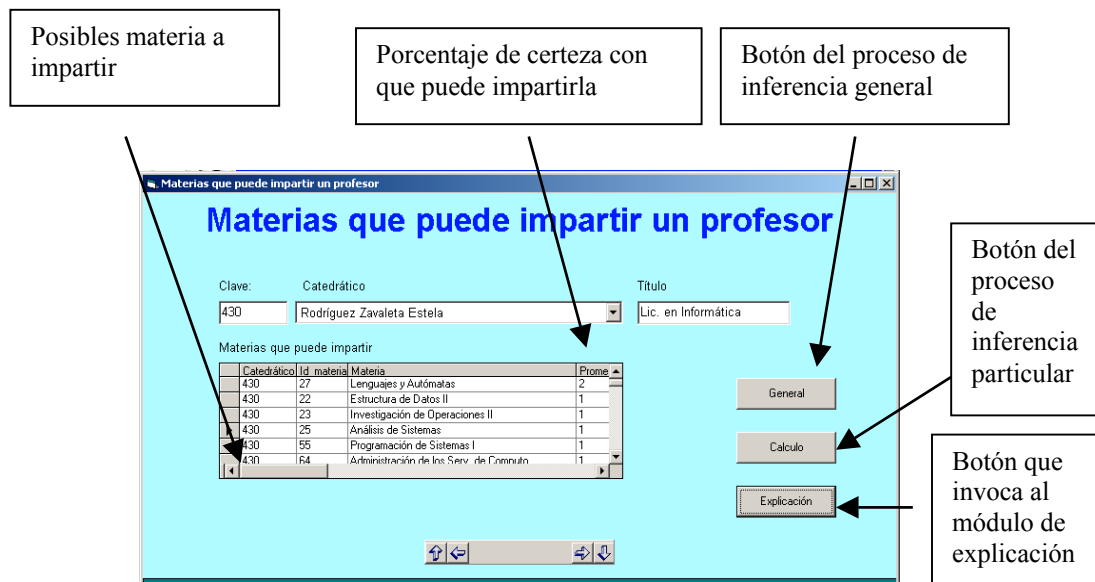


Fig 5.47 Pantalla materias que puede impartir un profesor

La Figura 5.48 muestra la pantalla de explicación donde se indican el perfil de la materia y el perfil del profesor.

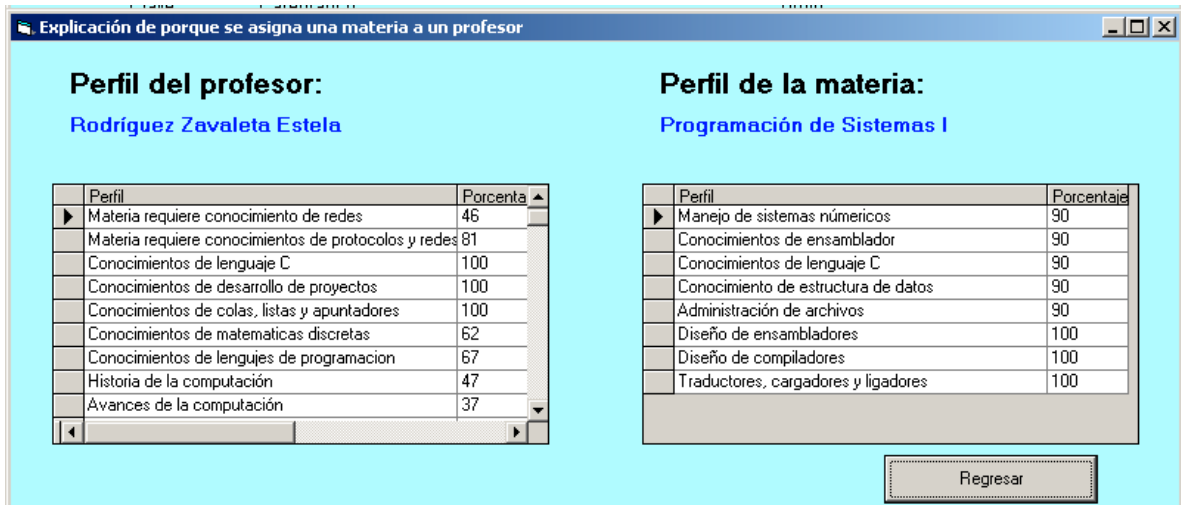


Fig 5.48 Pantalla de explicación porque un profesor puede dar una materia





# CAPÍTULO 6

## PRUEBAS Y RESULTADOS

### 6.1 Pruebas del sistema

Se pueden detectar, dentro del desarrollo de sistemas, las actividades para realizar pruebas: las producidas durante la etapa de requerimientos, de diseño preliminar y las de diseño detallado. Clasificadas de otra manera éstas pueden ser: funcionales, de integración y de implementación.

La finalidad de realizar *Pruebas funcionales* es detectar errores que pudieran haberse dado durante las fases de análisis y de especificación. Por otro lado, ellas pueden ayudar a probar si el diseño es correcto, ya que esta fase en sus dos niveles, preliminar y detallado, está basada en el análisis y especificación de requerimientos del sistema. Esta actividad es guiada por los objetivos y requerimientos plasmados. Se busca en esta prueba la posibilidad de que se haya incurrido en errores de conceptualización y/o de especificación. Las pruebas se organizan tomando como guía las diferentes opciones del sistema final.

Las *Pruebas de integración* se realizan cuando se procede a interconectar los diferentes subsistemas que constituyen el futuro sistema. Como es conocido, un sistema puede ser desarrollado por diferentes personas. Por esta razón, es de suma importancia probar la interconexión de los diferentes subsistemas y así verificar que ha sido realizada de manera correcta.

Las *Pruebas de implementación o codificación* son las que se realizan directamente sobre los programas que se están desarrollando durante la fase de implementación. Estos módulos o programas a su vez constituyen subsistemas. Por esta razón, la integración de los diferentes módulos que forman un subsistema deben ser probados. Estas pruebas deben estar basadas en la especificación de diseño preliminar y detallado [Gerez, 1985].

Los siguientes rubros, describen las pruebas más importantes que se consideraron para validar el sistema, las pruebas funcionales. Las pruebas de implementación se realizaron durante el desarrollo de la codificación de los diferentes módulos.

## 6.2 PRUEBAS DEL SISTEMA

### 6.2.1 PRUEBA 1: ÁREAS DE MENÚS

**Objetivo:** Guiar al usuario en el desarrollo del proceso. Esta prueba verifica que el sistema tenga como opciones principales y submenús los señalados en la definición de requerimientos.

**Criterios de aceptación:** Cada prueba verifica que el sistema produzca los resultados señalados en la definición de requerimientos.

**Procedimiento de la prueba:** A continuación se despliegan las pantallas correspondientes a los menús principales y submenús. Las opciones principales del proyecto son dos: Menu Principal y Datos de los autores.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todas las opciones activadas, el sistema presentó los submenús definidos en la etapa de requerimientos.

La Figura 6.1 muestra la portada inicial del proyecto. Contiene los logotipos de las tres instituciones participantes y el nombre del proyecto.



Fig 6.1 Portada inicial del proyecto

La Figura 6.2 muestra el menú principal del proyecto y contiene 4 módulos. La primera opción incluye los módulos administrativo y generación de grupos, la segunda opción incluye selección de maestro y selección de materia.



Fig 6.2 Menú principal del proyecto

En la Figura 6.3 se tiene la pantalla del menú que agrupa los programas del módulo administrativo y Sistema Experto para determinar el número de grupos por abrir según los cálculos realizados con los datos de Servicios Escolares y el ajuste a ese dato.



Fig 6.3 Menú grupos por abrir

La Figura 6.4 muestra el menú de los módulos que sugiere maestros para impartir una asignatura y sugerencia de asignatura para algún maestro.



Fig 6.4 Pantalla menú selección de catedráticos

La Figura 6.5 muestra el módulo para asignar aula a los horarios que ya tienen materia, grupo y hora.



Fig 6.5 Pantalla principal del menú Algoritmos Genéticos

### 6.2.2 PRUEBA 2: VALIDACIÓN DE ENTRADAS

**Objetivo:** Evitar pérdidas de tiempo, con entradas erróneas de datos y resultados sin validez.

**Criterios de aceptación:** Esta prueba deberá mostrar de manera exacta la información al seleccionar la opción deseada.

**Procedimiento de la prueba:** Se verifica que todas las cajas de texto tengan el dato que se pide y una vez dados todos los datos se guardan para que estos datos posteriormente sean procesados por el sistema y almacenen los datos correctos.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todas las opciones activadas, el sistema presentó los resultados correctos a cada opción. Además los desplegados de mensajes de error se presentaron de manera correcta en todas las pruebas.

La Figura 6.6 muestra la captura de datos de una asignatura y envía el correspondiente mensaje de error si una caja de texto se encuentra vacía



Fig 6.6 Pantalla de captura de datos de las Materias

La Figura 6.7 muestra una pantalla de captura de datos asignando criterios a una materia y valida la no duplicidad de criterios en esa materia, se envía el correspondiente mensaje de error si una entrada ya fue capturada.

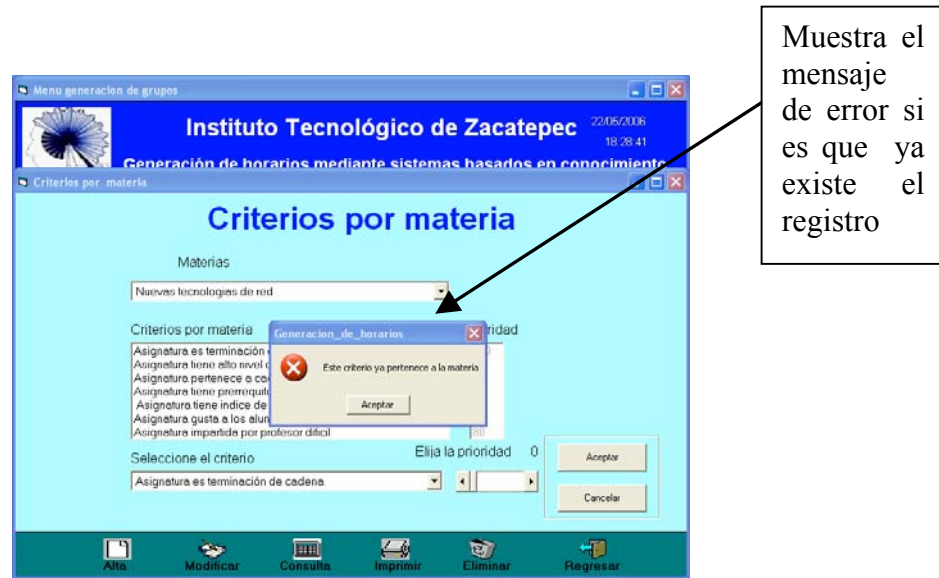


Fig 6.7 Pantalla de captura de datos de Criterios por Materia

La Figura 6.8 muestra una pantalla de captura de datos de las carreras, al encontrar un registro ya capturado muestra el mensaje de error si una carrera ya fue capturada.



Fig 6.8 Pantalla de captura de datos de Carreras

### 6.2.3 PRUEBA 3: DESPLEGADO DE RESULTADOS

#### 6.2.3.1 MÓDULO DE GENERACIÓN DE GRUPOS POR ABRIR

**Objetivo:** Proporcionar los datos que el módulo pide, para que posteriormente el sistema genere el número de grupos sugerido por abrir para cada materia.

**Criterios de aceptación:** Esta prueba muestra la lista de grupos que se deben abrir después de aplicar los criterios del Sistema Experto.

**Procedimiento de la prueba:** Una vez que se proporcionan los datos que el Sistema Experto requiere, el sistema generará una lista de los grupos que se deben abrir el próximo semestre.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todos los casos presentados, el sistema generó la lista en forma satisfactoria y con la información correcta.

La Figura 6.9 es la pantalla más importante de módulo de generación de grupos porque es donde se realiza la inferencia para calcular el tipo de ajuste que se va a aplicar a cada materia, con base en las reglas que se disparan y que hacen *match* los criterios de una regla y los criterios de una materia

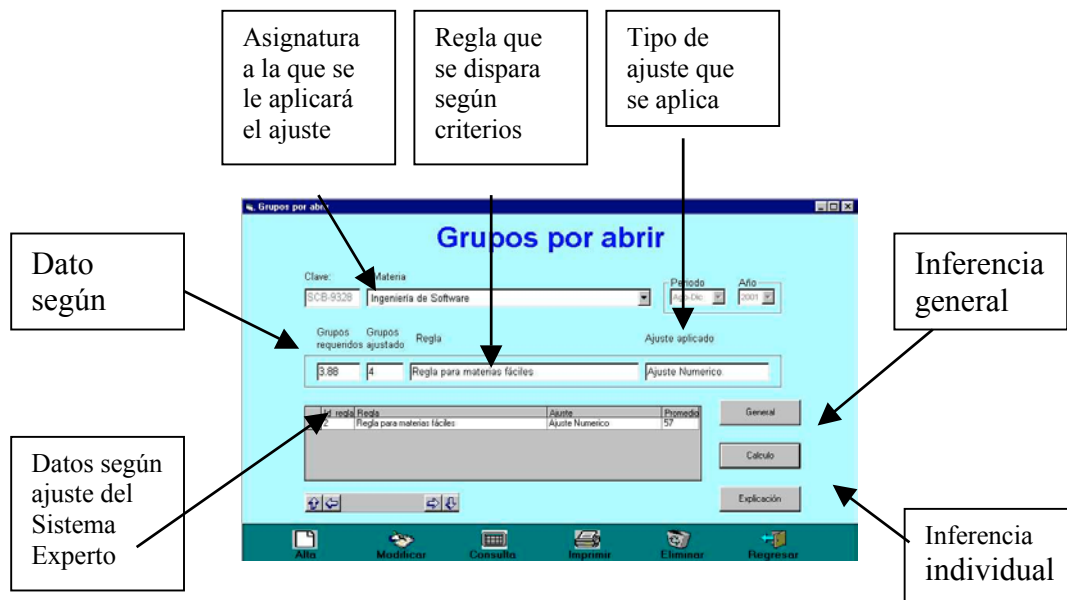


Fig 6.9 Pantalla cálculo de ajustes

### 6.2.3.2 MÓDULO DE GENERACIÓN DE HORARIOS

**Objetivo:** Proporcionar los datos que el módulo pide, para que posteriormente el sistema genere el horario por semestre de las materias de una carrera.

**Criterios de aceptación:** Esta prueba debe mostrar los horarios para las materias sin errores, sin choques de hora en un paquete.

**Procedimiento de la prueba:** Recibe como entrada cuantos grupos abrir de cada materia en cada semestre y genera primero los paquetes, posteriormente a cada paquete de materias le asigna la hora por materia.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todos los casos presentados, el sistema generó los horarios correspondientes.

La Figura 6.10 muestra el formulario que forma los paquetes de materias según la retícula de una carrera, muestra el número de grupos por cada materia que se van a formar y hace los paquetes asignando los grupos AA, AB, AC, AD.

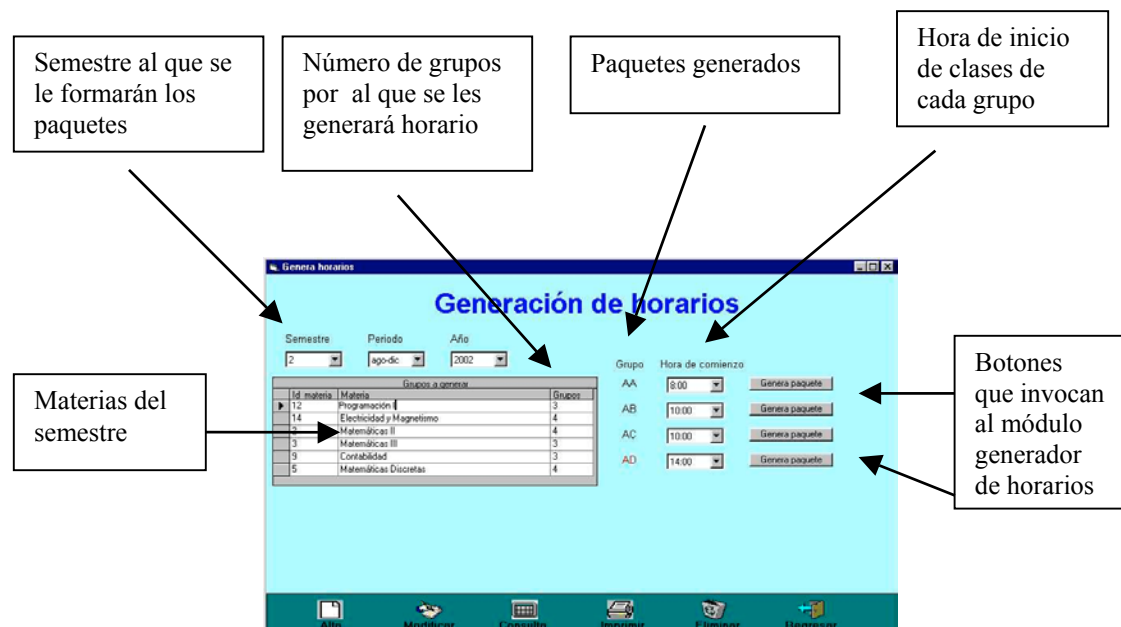


Fig 6.10 Pantalla inicial del módulo generación de horarios



La Figura 6.11 es el formulario del módulo Programación Lógica de Restricciones que genera los horarios por paquete satisfaciendo las restricciones para formar un horario libre de choques, continuo (sin horas muertas) cumpliendo las horas por materia, etc.

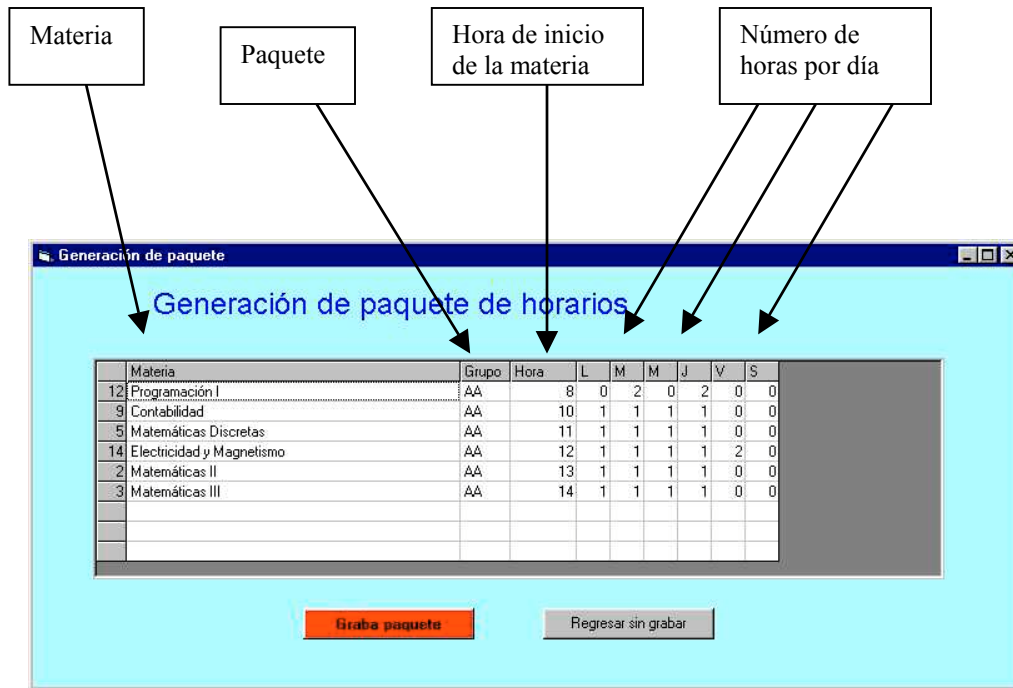


Fig 6.11 Pantalla principal del módulo generación de horarios

### 6.2.3.3 MÓDULO ASIGNACIÓN DE AULA

**Objetivo:** Utilizar los datos generados en el módulo anterior para hacer la asignación de aulas a los paquetes de materias.

**Criterios de aceptación:** Esta prueba debe mostrar los horarios de las materias con una asignación de aula sin choques.

**Procedimiento de la prueba:** Se generan las aulas utilizando los procedimientos de los Algoritmos Genéticos durante varias generaciones hasta lograr una generación sin choques en el aula.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todos los casos presentados, el sistema generó las aulas correspondientes.

La Figura 6.12 muestra la pantalla para actualizar las aulas disponibles por las personas que generan los horarios, estas aulas están disponibles para que el Algoritmo Genético las utilice para asignarles grupos durante el proceso.

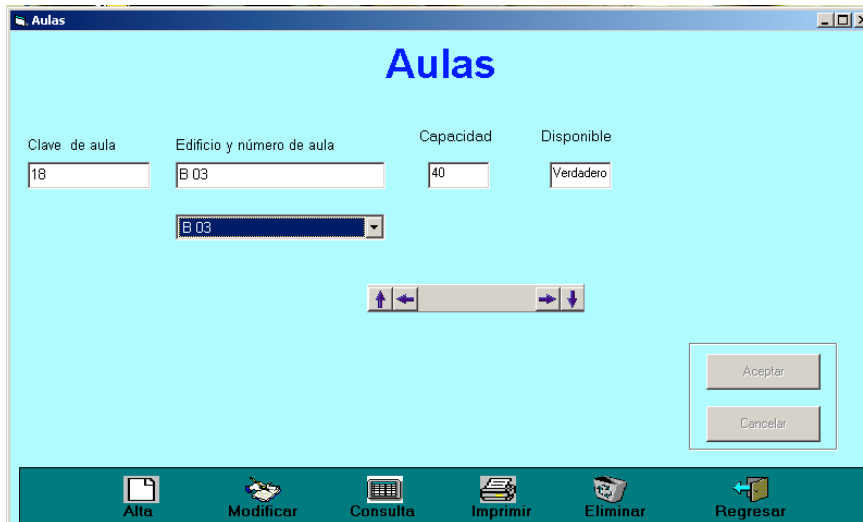


Fig 6.12 Pantalla del control de aulas

La Figura 6.13 muestra la consulta de las aulas que utiliza el módulo generación de aulas.

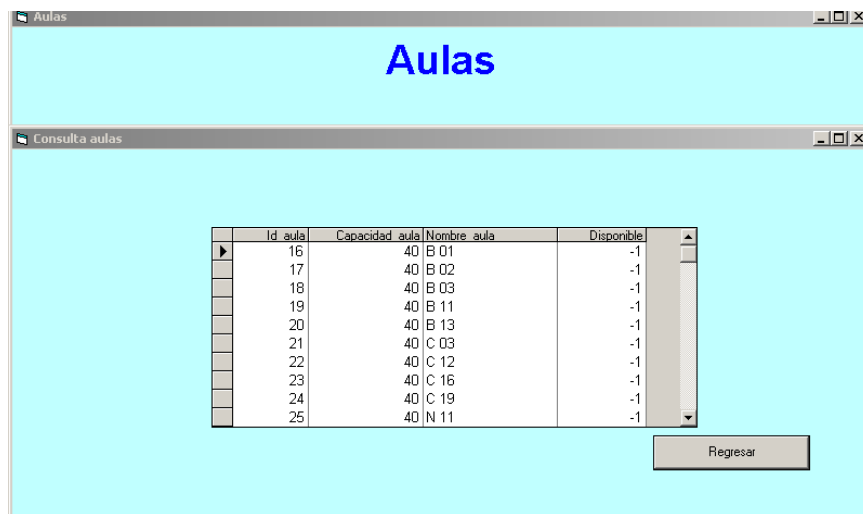


Fig 6.13 Pantalla de consultas del control de aulas

La Figura 6.14 muestra la pantalla que genera la población inicial a partir de los datos generales proveniente de los módulos anteriores.

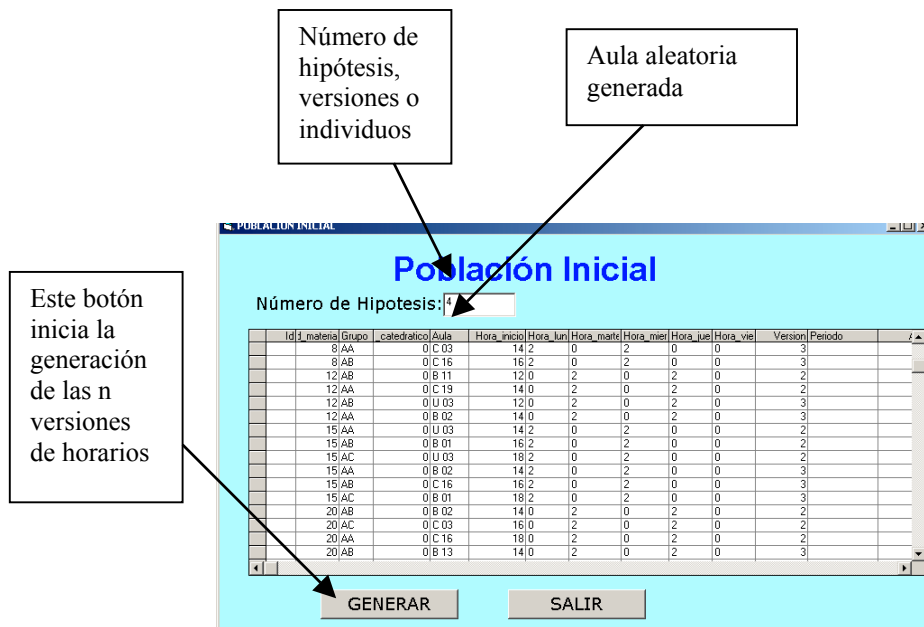


Fig 6.14 Pantalla de genera población inicial

La Figura 6.15 muestra la pantalla que evalúa las versiones originadas y genera para cada una de ellas un cromosoma cuyos genes contienen la información obtenida

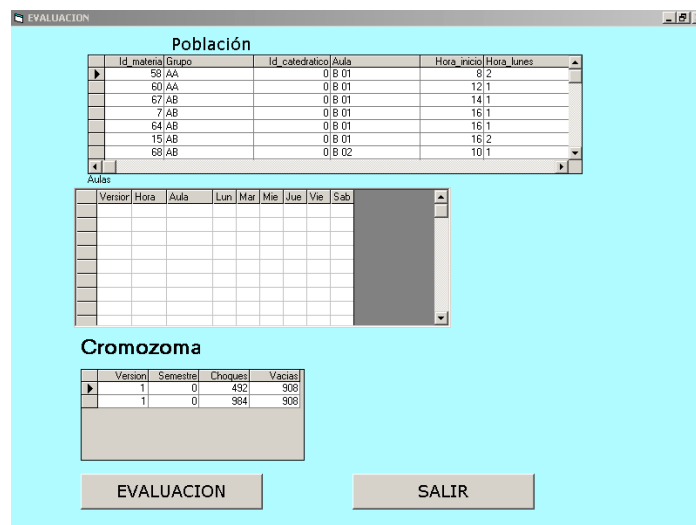


Fig 6.15 Pantalla de evaluación de versiones

La Figura 6.16 muestra el formulario del proceso de selección de las mejores versiones de horarios, toma aquellas que tienen menor número de choques y mayor espacio de ocupación de aulas

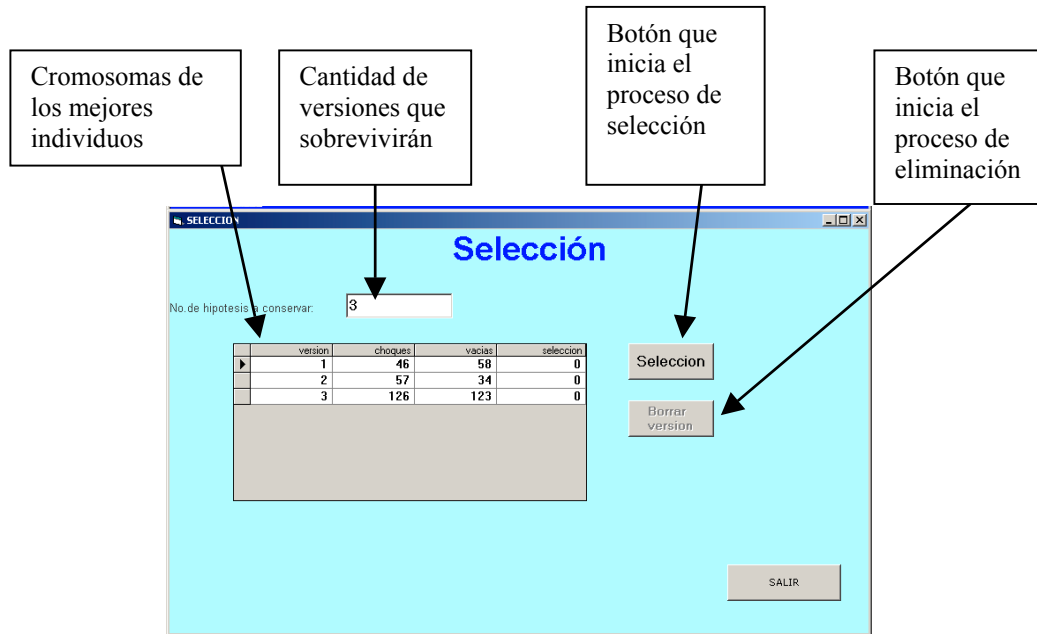


Fig. 6.16 Pantalla del proceso de selección

La Figura 6.17 muestra la pantalla del proceso de clonación donde se selecciona la mejor versión y se obtiene un clon idéntico con un nuevo número de versión.

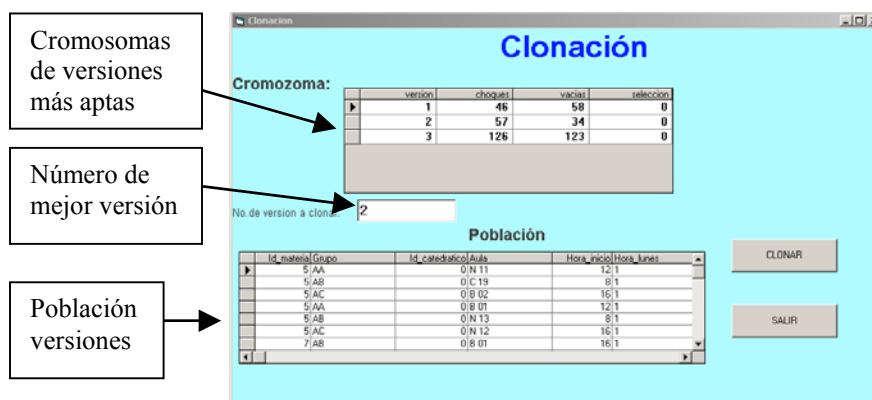


Fig. 6.17 Pantalla del proceso de clonación

La Figura 6.18 muestra la pantalla del proceso de mutación utilizado para hacer el cambio “mutar” los datos de una versión para romper algún ciclado de la tendencia hacia el resultado o simplemente para acelerar el proceso de optimización

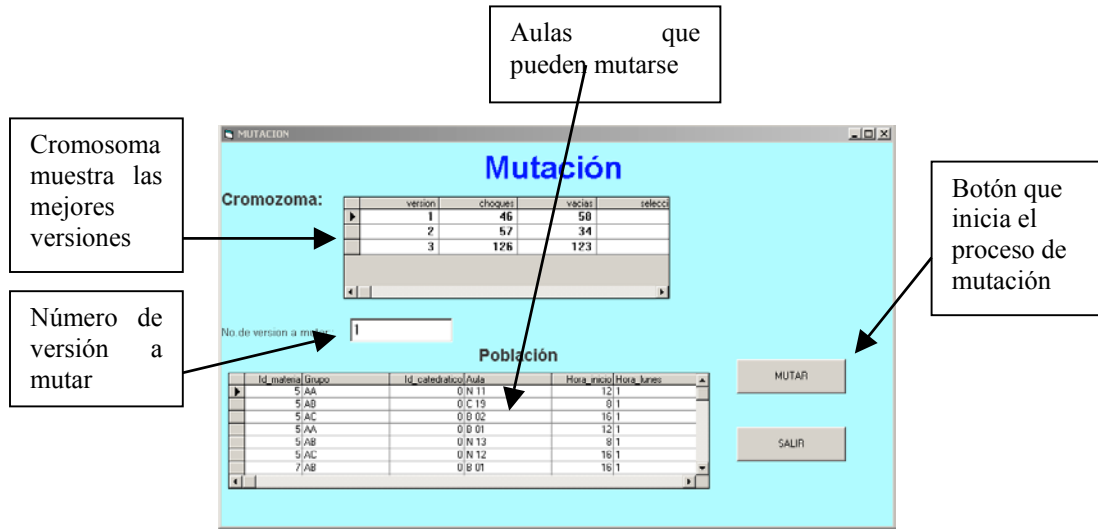


Fig. 6.18 Pantalla de mutación

La Figura 6.19 muestra la pantalla donde se realiza el cruzamiento utilizando plantillas predeterminadas dirigiendo la cruce de versiones por paquetes de materias agrupadas por semestre y cada plantilla genera una nueva versión con la combinación de fragmentos de varias versiones.



Fig. 6.19 Pantalla del proceso de cruzamiento

La Figura 6.20 permite la selección y envío de la mejor versión de horarios y adaptarlo al módulo administrativo para su distribución y publicación.

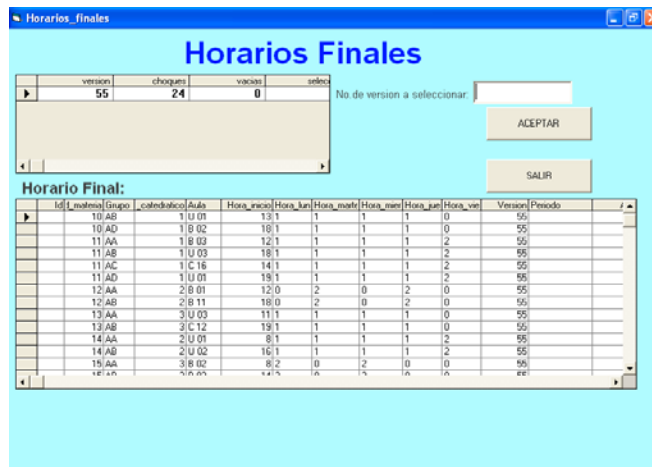


Fig. 6.20 Horarios finales

### 6.2.3.4 MÓDULO SELECCIÓN DE CATEDRATICOS

**Objetivo:** Capturar los datos de los profesores y su perfil profesiográfico, además capturar los conocimientos necesarios para impartir una asignatura para poder sugerir cual es el mejor catedrático que puede impartir una asignatura.

**Criterios de aceptación:** Esta prueba debe mostrar la sugerencia de los mejores catedráticos para una materia y las mejores materias para un catedrático.

**Procedimiento de la prueba:** Se darán los datos iniciales de los maestros y el sistema deberá generar una lista a partir de la inferencia del Sistema Experto respectivo y viceversa.

**Resultado de la prueba:** Los resultados fueron satisfactorios. En todos los casos presentados, el sistema infirió las materias y los catedráticos correspondientes

La Figura 6.21 muestra la pantalla en que se captura el perfil de los catedráticos, el perfil son los conocimientos que tiene el profesor que lo capacitan para impartir alguna asignatura.

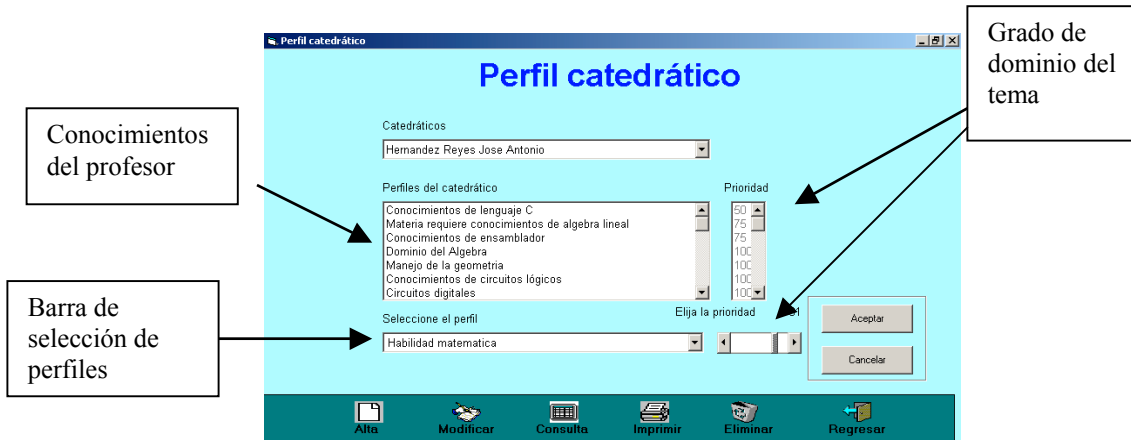


Fig 6.21 Pantalla captura perfil de catedráticos

La Figura 6.22 muestra la pantalla de inferencia final que sugiere quienes son los catedráticos más capacitados para impartir una materia, haciendo la inferencia mediante la búsqueda de *matches* entre el perfil de la asignatura y el perfil del catedrático.

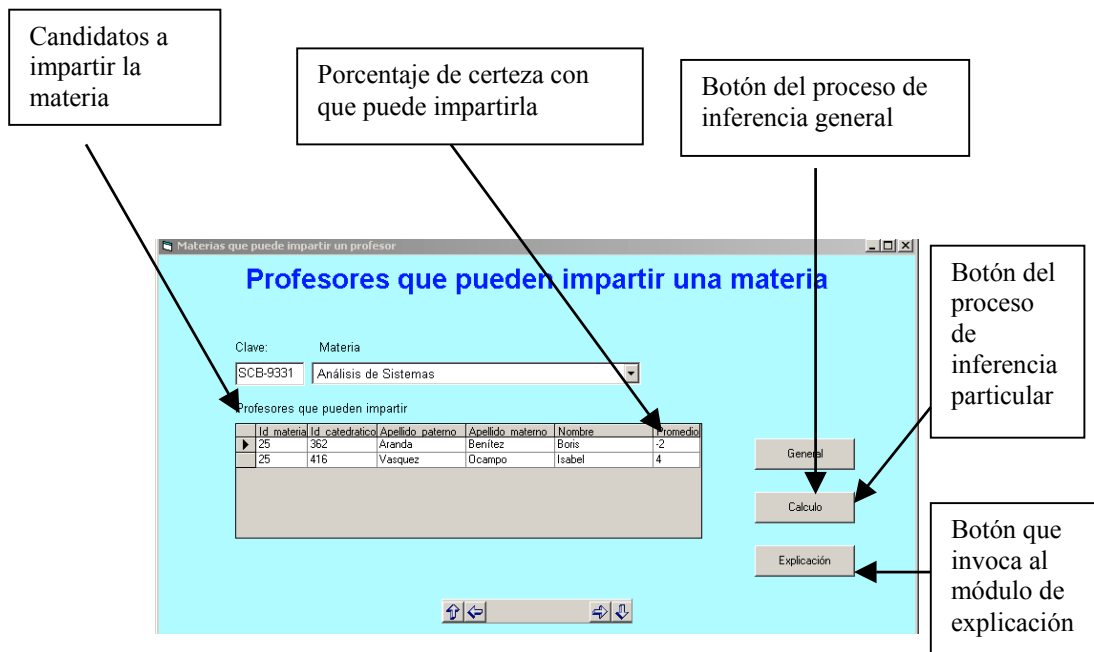


Fig 6.22 Pantalla muestra inferencia catedráticos para materia

La Figura 6.23 muestra la pantalla de explicación donde se indican los perfiles de la materia y el perfil del profesor.

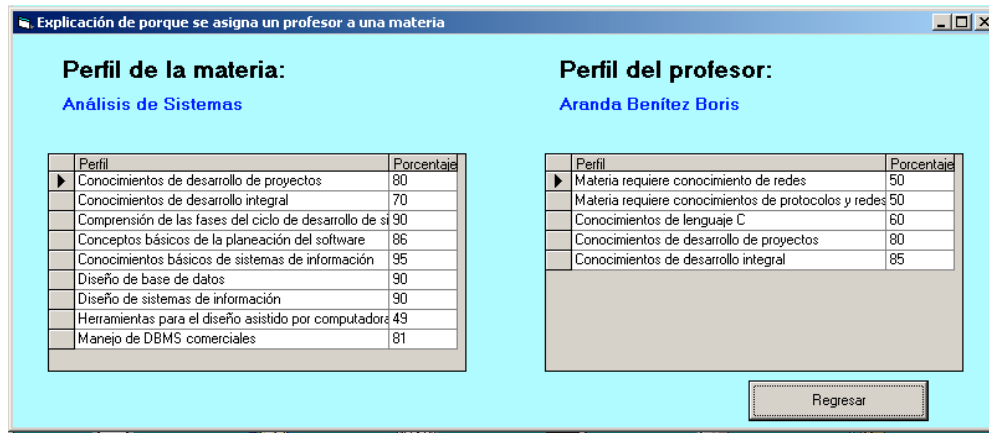


Fig 6.23 Pantalla explicación porque un maestro puede dar una materia

La Figura 6.24 muestra el resultado de la inferencia del proceso que sugiere las materias que puede impartir un profesor con base en el perfil de la materia y el perfil del profesor.

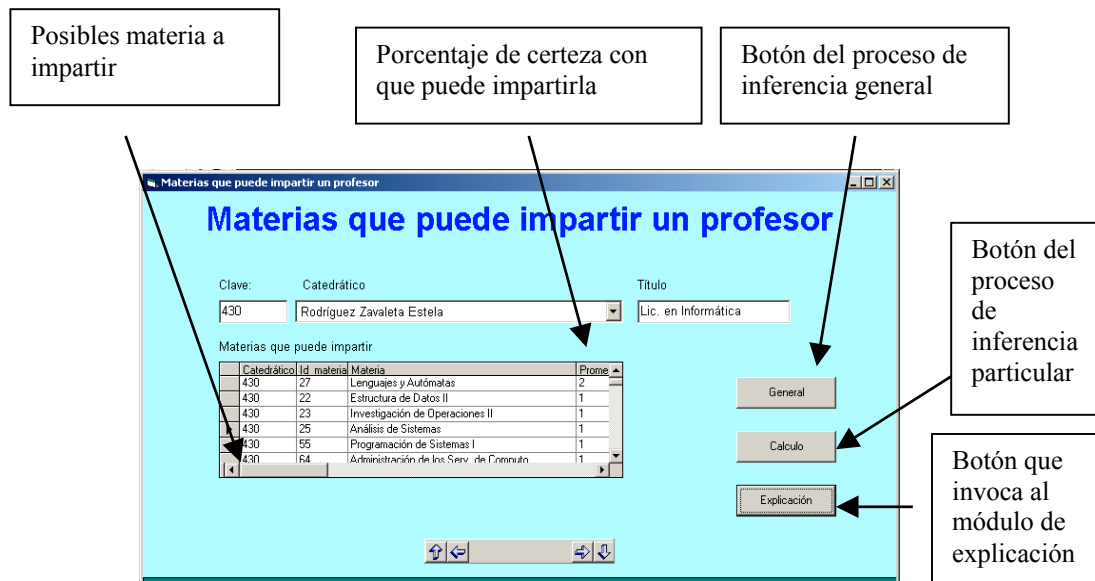


Fig 6.24 Pantalla materias que puede impartir un profesor



La Figura 6.25 muestra la pantalla de explicación donde se indican el perfil de la materia y el perfil del profesor.

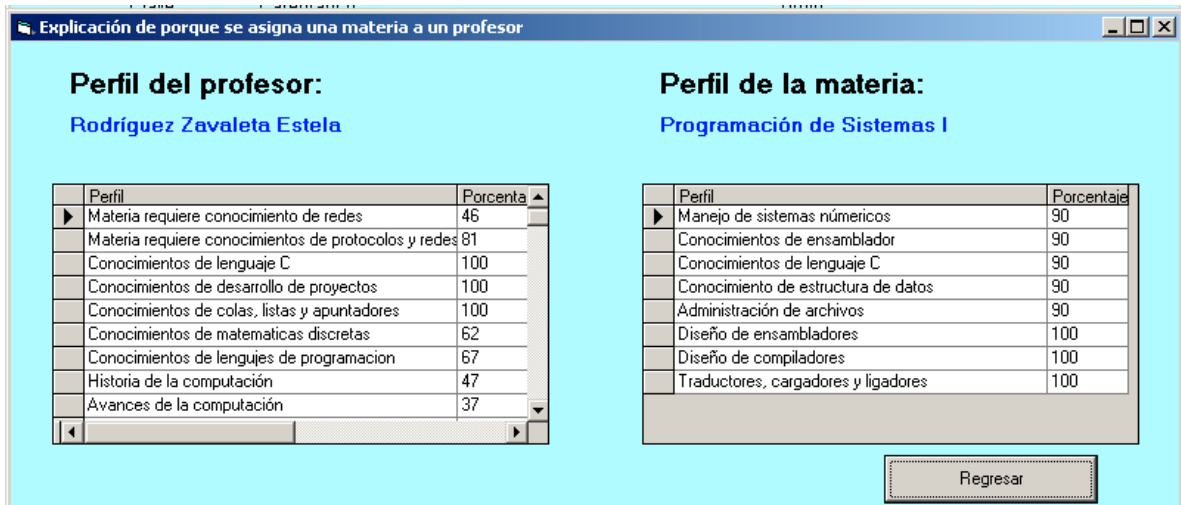


Fig 6.25 Pantalla de explicación porque un profesor puede dar una materia



## CAPÍTULO 7

### CONCLUSIONES Y TRABAJOS FUTUROS

#### 7.1 CONCLUSIONES

Los métodos de programación lineal proporcionan una solución parcial al problema de la generación de horarios, utilizan una “receta” predefinida para lograr la optimización de la asignación de recursos como lo hace el método Húngaro. La Inteligencia Artificial soluciona el problema utilizando la heurística, el conocimiento y la experiencia de un experto humano para solucionar el problema, se decidió utilizar esta metodología para explorar una solución aproximada al problema.

En el ITZ la generación de horarios se hacía de forma manual, no se utilizaba ningún método automatizado. Al aplicar algunas técnicas de la Inteligencia Artificial se ha logrado generar de manera automatizada los horarios de las materias que se ofertarán el próximo semestre.

La aplicación de varias ramas de la Inteligencia Artificial para la resolución del problema de la generación de horarios viene a demostrar que si es posible generar un condensado de horarios para el Instituto Tecnológico de Zacatepec que es una escuela de nivel superior y que estos horarios satisfacen adecuadamente las restricciones fuertes, medias y suaves. Para ello fue necesaria la combinación de varias disciplinas de la computación para generar cada una de las partes que componen un horario y también recomendar los catedráticos mas apropiados para impartir dichos grupos, nuevamente queda demostrado que la Inteligencia Artificial sirve para resolver problemas que las técnicas convencionales de computación consideran problema NP-completo.

La utilización de las técnicas de los Sistemas Expertos para ajustar el número de grupos por abrir el próximo semestre, permite aplicar el conocimiento y experiencia de la persona encargada de elaborar los horarios y así disminuir el número de grupos cancelados, el conocimiento e intuición del experto humano realizan un ajuste fino al número de grupos y como puede notarse, el sistema lo hace de manera similar.

Para el módulo de generación de los grupos y los horarios, la utilización de la técnica llamada Programación Lógica de Restricciones permitió la satisfacción de las restricciones que hacen que los horarios sean considerados factibles para su aplicación a

los alumnos el próximo semestre, ya que los paquetes de materias por semestre tienen horarios corridos y sin choques en horas, número de materias completo y buena distribución de horas pares e impares.

Se utilizó también la técnica de optimización llamada Algoritmos Genéticos para realizar la asignación de aulas a cada grupo generado, logrando con ello minimizar automáticamente el número de choques de materias en aulas. En las pruebas realizadas se logró que en la sexta generación de individuos ya sólo hubiera hasta un máximo de 24 choques lo que indicaba que sólo 4 ó 5 grupos compartían aula a la misma hora, esto era fácilmente corregible realizando una inspección visual a la versión de horarios final y ajustando los últimos errores.

Por último, nuevamente se aplicaron los Sistemas Expertos en el módulo de sugerencia de materias que puede impartir un maestro y en el módulo maestros que pueden impartir una determinada materia, vale la pena comentar que durante las pruebas realizadas, los catedráticos del departamento de Sistemas y Computación capturaban su perfil profesiográfico y al pedir al sistema la sugerencia de cuáles materias podrían impartir, los Sistemas Expertos del proyecto inferían exactamente las materias que los catedráticos están impartiendo en este semestre y las que habían impartido en algún semestre anterior, demostrando que en efecto para profesores nuevos la recomendación del sistema sería válida.

## **7.2 TRABAJOS FUTUROS**

Se sugiere que se amplíe el proyecto para que a cada grupo generado con aula y hora se le asigne de manera automática un catedrático, utilizando ya sea Algoritmos Genéticos o Programación Lógica de Restricciones.

Otra posible mejora puede ser incluir la Lógica Difusa en los Sistemas Expertos de los módulos que sugieren las materias que puede impartir un catedrático y los catedráticos que pueden impartir una asignatura.

Como ya se comprobó, la factibilidad de generar los horarios de manera automática se propone también la exploración de otras ramas y técnicas de la computación (Redes Neuronales, Búsqueda Tabú, Grafos Coloreados, etc.) para comparar el rendimiento, tiempo, esfuerzo y así poder determinar cuáles son mejores o peores técnicas para resolver el problema de la generación de horarios.

## REFERENCIAS BIBLIOGRÁFICAS

[Abramson, 1991] Abramson, D. “Constructing School Timetables Using Simulated Annealing: Secuencial and Parallel Algorithms”, Management Science, PWS Publishing, Boston, 1998.

[Beasley, 1993] Beasley, D. & Bull, D.R., & Martin, R.R., “An Overview of Genetic Algorithms: Part 1, Fundamentals”, University Computing.

[Bull, 1993] Bull, D.R. & Beasley, D. & Martin, R.R., “An Overview of Genetic Algorithms: Part 2”, University Computing Magazine, 1993.

[Burke,1994]Burke, E.K. “A Genetic Algorithm for University Timetabling”, AISB workshop on evolutionary computing, Leeds, 1994.

[Burke, 1997] Burke Edmund K., “Automated Scheduling Optimization and Planning Group (ASAP)” ,PATAT’97 Proceedings of the 2<sup>nd</sup> Intenational Conference on the Practice And Theory of Automated Timetabling, 1997.

[Burke,2000] Burke, Edmund & Eliman, David & Weare, Rupert, “A Genetic Algorithm Based University Timetabling System”, PATAT’2000 Proceedings of the 5<sup>th</sup> Intenational Conference on the Practice And Theory of Automated Timetabling, 2000.

[Burke,2001] Burke E.K. & Eliman D.G.& Weare R., “Automated Scheduling Of University Exams”, Leeds, 2001.

[Burke,2002] Burke, Edmund & Eliman, David & Weare, Rupert, “Examination Timetabling in British Universities – A Survey”, Leeds, 2002.

[Carrasco, 2004] Carrasco, Marco & Pato, Margarida, “Solving Real Class/Teacher Timetabling Problems using Neural Networks”, Facultade de Ciencias, University of Lisboa, Portugal.

[Chorafas, 1990] Chorafas, Dimitris N., “Knowledge Engineering”. Van Nostrand Reinhold, New York, USA.

- [Corne, 1993] Corne, D., "Solving the module exam scheduling problem with genetic algorithms" , Proceedings of the 6<sup>th</sup> International conference in Industrial and Engineering applications of Artificial Intelligence and Expert Systems, 1993.
- [Daellenbach, 1987] Daellenbach, Hans & McNickle, Donald, "Introducción a técnicas de Investigación de Operaciones", CECSA, México, 1987
- [Davis, 1991] Davis, L. "Handbook of Genetic Algorithms", Van Nostrand. Reinhold, New York, NY.
- [Demist, 2000] Demist, "Departmental Timetabling Project, Proposal, DeMIST interface to Timetabling Package Project", Demist University, 2000.
- [Detcher, 1990] Detcher, R. "Enhancement Schemes for Constraint Processing: Back Jumping, Learning and Cutset Decomposition", Artificial Intelligence 41 (1990).
- [Eliman ,2000] Eliman David, "Specialised Recombinative Operators for Timetabling Problems", Cambridge University Press, New York, 2000.
- [Elmohamed, 2000] Elmohamed, Saleh, "The timetabling problem" PATAT'2000 Proceedings of the 5<sup>th</sup> International Conference on the Practice And Theory of Automated Timetabling, 2000.
- [Elmohamed, 2001] Elmohamed, Saleh, "Constraint based university course timetabling", 4<sup>th</sup> World Congress on Timetabling. Application of advanced information Technologies, London, 2001.
- [Fahrion y Dollanski, 1992] Fahrion, R. & G Dollanski, "Construction of University Faculty Timetables using Logic Programming", ACM, Sigsoft Notes, vol 9, no. 3, May 1992.
- [Frenzel, 1986] Frenzel, Louis E., "Crash Course in Artificial Intelligence and Expert Systems". Howard W. Sams & Co., McMillan Inc., Indianapolis, USA.
- [Gerez, 1985] Gerez, Victor & Mier, Mauricio, "Desarrollo y administración de programas de computadora", McGraw Hill, México, 1985.
- [Gervás, 1999] Gervás, P. & San Miguel, B. "Sequential Building of Constrained Timetables Using Rule-Based Heuristics: An Expert System for Automated Timetabling", Congreso Internacional de Investigación en Ciencias Computacionales.

[Giarratamo, 1998] Giarratamo, Joseph y Gary Riley, "Expert Systems. Principles and Programming", PWS Kent Publishing Company , Boston, USA.

[Gines, 1986] Gines, B. & Boose, J., "Knowledge Acquisition for Knowledge-Based Systems". Vol 1&2. Academic Press, London , U. K.

[Goldberg, 1989] Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley.

[Kang y White, 1994 ] Kang, L. & White, G.M., "A Logic Approach to the Resolution of Constrains in Timetabling", European Journal of Operational Research, vol 61, 1994.

[Kandel, 1991] Kandel, Abraham, "Fuzzy Expert Systems", CRC Press Inc., London, U.K., 1991.

[Kendall,1999] Kendall y Kendall, "Análisis y diseño de sistemas", 3ª edición, Pearson Educación.

[Koza, 1992] Koza, J.R., "Genetic Programming: On the Programming of Computers by means of Natural Selection" Cambridge, MA: MIT Press

[Martínez, 1999] Martínez Alfaro Horacio, "Solución al problema de asignación de salones de clases, horario y grupo mediante templado simulado", Memorias 6º Congreso Internacional de Mecatrónica e Inteligencia Artificial, México, 1999.

[Medina, 2004] Medina Pasaflores Horacio, "Fundamentos de Algoritmos Genéticos", Tesis de licenciatura, Zacatepec, Mexico, 2004.

[Michalewicz, 1992] Michalewicz, Z., "Genetic algorithms + Data Structures = Evolution Programs", Springer-Verlag, New York, NY.

[Negoita,1985] Negoita, ConstantinV., Expert Systems and Fuzzy Systems. The Benjamin Cummings Publishing Co., Menlo Park, CA., USA.

[Nielsoft, 2000] Nielsoft informatica, "Manual de usuario Mimosa for windows".

[Paechter, 1994] Paechter, B., "Two solutions to the general timetabling problem using evolutionary methods", IEEE WCCI ,1994.

[Pluss,1999] Pluss J.J., "Sistemas expertos un enfoque pragmático", UNR.

[Rich, 1996] Rich, E. y Knight, K., “Inteligencia Artificial”, McGraw Hill Interamericana, Madrid, España.

[Russell, 1996] Russell, S. y Norvig, P., “Inteligencia Artificial, un enfoque moderno”, Prentice Hall, Mexico, 1996.

[Sanchez,1999] Sanchez y Beltran, “Sistemas expertos una metodología de programación”, Macrobit.

[Schedul,2000] SchedulExpert, “Course scheduling in post educational institutions”, Ed. Addison Wesley, 2000.

[Trick, 1999 ] Trick Michael, “Network resources for coloring a graph”, Oracle Press, Ed. McGraw Hill, 1999.

[Turban, 1995] Turban Efraim, “Decision Support and Expert Systems. Management Support Systems.” Fourth Edition. Orentice Hall, NJ, USA.

[Waterman,1986] Waterman, Donald A., “A Guide to Expert Systems.” Addison-Wesley Publishing Co. USA.

[Weare,2000] Weare, Rupert & Burke, Edmundo & Eliman , David, “A Questionnaire on Exam Timetabling in Universities”, Annual review of computer Science, Volume 2, Palo Alto, 2000.

[Wilke, 2002] Wilke, P. & Grobner, M. & Oster, N. “A hibrid Genetic Algorithm for School Timetabling”, Advances in Artificial Intelligence. Springer Lecture Notes in Computer Science, New York.

[Yoshikawa, 1994] Yoshikawa, M. & Kaneko, K.“A Constraint-based Approach to High School Timetabling Problems: A Case Study”, AAAI-94 12° National Conference on Artificial Intelligence.



## **Referencias electrónicas**

<http://www.monografias.com/trabajos12/inteartf/inteartf.shtml>

Obregón Arturo, Inteligencia artificial y Sistemas expertos, IIMAS-UNAM (1992)

[http://www.lafacu.com/apuntes/informatica/sist\\_cono/default.htm](http://www.lafacu.com/apuntes/informatica/sist_cono/default.htm)

Cruz Ochoa Wilson, Sistemas de conocimiento

[http://www.ingenieroseninformatica.org/recursos/tutoriales/sist\\_exp.html](http://www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp.html)

Zambrano Molina Darwin, Sistemas expertos

<http://redcientifica.com/doc/doc199008210001.html>

Samper Marquez Juan José, Introducción a los sistemas expertos, Madrid, 2000

<http://mismonografias.com.ar/Sistemas%20experto%20fuzzy.html>

Arede Dario, Sistema experto fuzzy

<http://cruzrojaguaya.org/inteligencia/inter.html>

Ayuquina Martha & Arellano Gregorio, Inteligencia artificial, Universidad de Guayaquil

<http://dmi.uib.es/abasolo/intart/1-introduccion.html>

Abasolo María José Dra., Inteligencia Artificial