

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO
PARA EL APOYO AL PROCESO DE ASIGNACIÓN DE LA CARGA
ACADÉMICA USANDO PROGRAMACIÓN CON RESTRICCIONES**

**BIBIANA MARÍA GONZÁLEZ ALVAREZ
JORGE ANDRÉS QUINTERO CÁRDENAS**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2010**

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO
PARA EL APOYO AL PROCESO DE ASIGNACIÓN DE LA CARGA
ACADÉMICA USANDO PROGRAMACIÓN CON RESTRICCIONES**

**BIBIANA MARÍA GONZÁLEZ ALVAREZ
JORGE ANDRÉS QUINTERO CÁRDENAS**

**Proyecto presentado como requisito para optar por el título de Ingeniero(a)
de Sistemas y Computación**

**Asesor del Proyecto
Ing. Guillermo Roberto Solarte Martínez**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2010**

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Pereira, Junio de 2010

DEDICATORIA

A mi mamá por el apoyo y motivación brindados a lo largo de mi carrera universitaria y durante la realización de este proyecto.

Bibiana María González Alvarez

A mi madre, por no desfallecer, por su apoyo total e incondicional, por no perder la ilusión de ver este proceso culminado.

Jorge Andrés Quintero Cárdenas

AGRADECIMIENTOS

Queremos dar un reconocimiento a todos los profesores que hicieron parte de este proceso de formación como ingenieros de sistemas y computación, queremos darles las gracias por intentar transmitirnos parte de sus conocimientos, por compartir con nosotros sus experiencias y por realizar la labor de educadores, la cual tiene un impacto social tan importante, a todos y cada uno los recordaremos como excelentes profesionales, esperamos demostrar en nuestra vida laboral la calidad de nuestra formación y dejar en alto el nombre de nuestra universidad.

Igualmente y en especial, a nuestro guía y tutor, el ingeniero Guillermo Roberto Solarte, quien nos acompañó durante el desarrollo de todas las etapas de este proyecto.

A la ingeniera Luz Stella Valencia, quien fue la persona que nos sugirió la idea del proyecto. Cuando el estudiante se enfrenta a la realización de la tesis de grado la primera dificultad que encuentra es no saber que hacer.

Al ingeniero Gilberto Vargas, queremos también hacerle un reconocimiento; en su labor de docente es impecable y muy profesional, además de poseer un conocimiento extenso en ingeniería de sistemas, pero no solo se destaca en su labor académica, también lo hace en su labor directiva, mostrando interés por las dificultades del estudiante.

A la ingeniera Ana Maria López Echeverri, quien nos llevó de la mano en la última etapa de desarrollo, ayudándonos a optimizar y aterrizar la aplicación en cuanto a la realidad del proceso.

A todos y cada uno, muchas gracias.

CONTENIDO

	Página
INTRODUCCIÓN	27
1. TÍTULO DEL PROYECTO	28
2. PLANTEAMIENTO DEL PROBLEMA	29
3. IMPORTANCIA DE LA INVESTIGACIÓN	30
4. JUSTIFICACIÓN	31
5. OBJETIVOS	32
5.1. Objetivo general	32
5.2. Objetivos específicos	32
6. MARCO REFERENCIAL	33
6.1. ANTECEDENTES EN LA SOLUCIÓN DEL PROBLEMA	33
6.2. ACTUALIDAD DE LA PROGRAMACIÓN CON RESTRICCIONES	37
6.3. REFERENTES DEL MARCO TEÓRICO	39
6.3.1. Programación lógica	39
6.3.2. Programación con restricciones	40
6.3.2.1. Ejemplo de un problema que se puede formular como un CSP (Constraint Satisfaction Problem)	41
6.3.2.2. Resolución de un CSP	42

6.3.2.3. Modelado de un CSP	42
6.3.2.4. Conceptos CSP	44
6.3.2.5. Notación CSP	44
6.3.2.6. Restricciones	45
6.3.2.7. Consistencia de un CSP	45
6.3.2.8. Algoritmos de búsqueda	46
6.3.2.8.1. El árbol de búsqueda	46
6.3.2.8.2. Backtracking cronológico (BT)	47
6.3.2.8.3. Algoritmos Look-back	48
6.3.2.8.3.1. Backjumping (BJ)	48
6.3.2.8.3.2. Conflict-directed backjumping (CBJ)	48
6.3.2.8.3.3. Learning	49
6.3.2.8.4. Algoritmos Look-ahead	49
6.3.2.8.4.1. Forward Checking (FC)	49
6.4. INTRODUCCION A UML	50
6.4.1. Diagrama de Casos de Uso	50
6.4.2. Diagramas de Interacción	52
6.4.2.1. Diagrama de Secuencia	52
6.4.2.2. Diagrama de Colaboración	53
6.4.3. Diagrama de Actividades	53
6.4.4. Diagrama de Estados	54
6.4.5. Diccionario de Datos	56
6.4.6. Requerimientos del Usuario	56

6.4.6.1. Requerimientos Funcionales	56
6.4.6.2. Requerimientos No Funcionales	56
6.4.7. Diagrama de Interacción con el Usuario	56
6.4.8. Diagrama de Componentes	57
6.4.9. Diagrama de Despliegue	57
6.4.10. Diagrama de Clases	58
6.4.11. Modelo Entidad-Relación	61
7. ANÁLISIS DEL SISTEMA	64
7.1. TRABAJO DE CAMPO	64
7.1.1. Entrevista	64
7.1.2. Requerimientos del sistema	66
7.1.3. Modelado del dominio	67
7.2. MODELO MATEMÁTICO	71
7.2.1. Restricciones	74
7.2.1.1. Profesores catedráticos	74
7.2.1.2. Profesores de contrato	74
7.2.1.2.1. Horas reglamentarias	74
7.2.1.2.2. Sobrecarga	74
7.3. METODOLOGÍA DE DESARROLLO	76
7.4. MODELO DE CASOS DE USO	77
7.4.1. Diagramas y Descripción de Casos de Uso	77
7.5. MODELO DINÁMICO	104

7.5.1. Diagramas de Secuencia	104
7.5.2. Diagramas de Colaboración	127
7.6. MODELO DE ESTADOS	150
7.6.1. Diagrama de Estados	150
7.7. MODELO FUNCIONAL	154
7.7.1. Diagrama de Actividades	154
7.8. MODELO FINAL DE OBJETOS	158
7.8.1. Diagrama de Clases	158
7.9. DICCIONARIO DE DATOS	159
7.9.1. Diccionario de clases	159
7.9.2. Diccionario de atributos clase	160
7.9.3. Diccionario de métodos clase	163
7.10. REQUERIMIENTOS NO FUNCIONALES	175
8. DISEÑO DEL SISTEMA	176
8.1. ARQUITECTURA DEL SISTEMA	176
8.2. IDENTIFICACIÓN DE SUBSISTEMAS	178
8.2.1. Diagrama de Subsistemas	178
8.2.2. Descripción de cada Subsistema	179
8.3. DESCRIPCIÓN DE SUBSISTEMAS DE APLICACIÓN	180
8.3.1. Modelo general de las aplicaciones	180
8.3.1.1. Diagramas de Secuencia de Ventanas	180
8.3.2. Modelo Dinámico de la aplicación	184

8.3.2.1. Diagramas de Interacción con el usuario	184
8.3.3. Modelo de componentes	188
8.3.3.1. Diagrama de Componentes	188
8.3.4. Modelo de Distribución	189
8.3.4.1. Diagrama de Despliegue	189
8.4. DISEÑO DE INTERFACES	190
8.4.1. Diagrama de Interacción con el usuario de los componentes a nivel de objetos	190
8.4.2. Diseño estético	194
8.5. DISEÑO DE LA BASE DE DATOS	233
8.5.1. Modelo Entidad-Relación	233
8.5.2. Scripts	234
8.5.3. Descripción de las tablas	236
9. CONCLUSIONES	241
10. RECOMENDACIONES	242
11. FUTURAS LÍNEAS DE INVESTIGACIÓN	243
BIBLIOGRAFÍA	244

ÍNDICE DE FIGURAS

	Pág.
Figura 1. Mapa dividido en cuatro regiones (Ejemplo de problema CSP)	41
Figura 2. Grafo de restricciones	42
Figura 3. Esquema de sumado	43
Figura 4. Árbol de búsqueda	47
Figura 5. Representación Actor	50
Figura 6. Representación Caso de Uso	50
Figura 7. Inclusión	51
Figura 8. Extensión	51
Figura 9. Generalización	51
Figura 10. Diagrama de Secuencia	52
Figura 11. Ejemplo Diagrama de Colaboración	53
Figura 12. Ejemplo Diagrama de Actividad	54
Figura 13. Ejemplo Diagrama de Estados	55
Figura 14. Diagrama de Interacción con el Usuario	56
Figura 15. Ejemplo Diagrama de Componentes	57
Figura 16. Ejemplo Diagrama de Despliegue	57
Figura 17. Representación Clase	58
Figura 18. Herencia	60
Figura 19. Agregación	60

Figura 20. Asociación	60
Figura 21. Dependencia o instanciación	61
Figura 22. Ejemplo Dependencia o instanciación	61
Figura 23. Ejemplo Modelo Entidad-Relación	63
Figura 24. Esquema propuesto de secuencia de horas	72
Figura 25. Diagrama de Caso de Uso Administrar Profesores	77
Figura 26. Diagrama de Caso de Uso Administrar Materias	85
Figura 27. Diagrama de Caso de Uso Administrar Usuarios	90
Figura 28. Diagrama de Caso de Uso Administrar Carga	96
Figura 29. Diagrama de Secuencia Administrar Profesores Escenario Adicionar	104
Figura 30. Diagrama de Secuencia Administrar Profesores Escenario Modificar	105
Figura 31. Diagrama de Secuencia Administrar Profesores Escenario Consultar	106
Figura 32. Diagrama de Secuencia Administrar Profesores Escenario Eliminar	107
Figura 33. Diagrama de Secuencia Administrar Profesores Escenario Establecer Horario	108
Figura 34. Diagrama de Secuencia Administrar Profesores Escenario Adicionar Materia	109
Figura 35. Diagrama de Secuencia Administrar Profesores Escenario Eliminar Materia	110
Figura 36. Diagrama de Secuencia Administrar Materias Escenario Adicionar	111
Figura 37. Diagrama de Secuencia Administrar Materias Escenario Modificar	112

Figura 38.	Diagrama de Secuencia Administrar Materias Escenario Consultar	113
Figura 39.	Diagrama de Secuencia Administrar Materias Escenario Eliminar	114
Figura 40.	Diagrama de Secuencia Administrar Usuarios Escenario Adicionar	115
Figura 41.	Diagrama de Secuencia Administrar Usuarios Escenario Modificar	116
Figura 42.	Diagrama de Secuencia Administrar Usuarios Escenario Consultar	117
Figura 43.	Diagrama de Secuencia Administrar Usuarios Escenario Eliminar	118
Figura 44.	Diagrama de Secuencia Administrar Usuarios Escenario Restablecer Clave	119
Figura 45.	Diagrama de Secuencia Administrar Carga Escenario Adicionar Grupo	120
Figura 46.	Diagrama de Secuencia Administrar Carga Escenario Consultar	121
Figura 47.	Diagrama de Secuencia Administrar Carga Escenario Eliminar Grupo	122
Figura 48.	Diagrama de Secuencia Administrar Carga Escenario Asignar Carga	123
Figura 49.	Diagrama de Secuencia Administrar Carga Escenario Reiniciar Carga	124
Figura 50.	Diagrama de Secuencia Administrar Carga Escenario Asignar Grupo	125
Figura 51.	Diagrama de Secuencia Administrar Carga Escenario Liberar Grupo	126
Figura 52.	Diagrama de Colaboración Administrar Profesores Escenario Adicionar	127

Figura 53.	Diagrama de Colaboración Administrar Profesores Escenario Modificar	128
Figura 54.	Diagrama de Colaboración Administrar Profesores Escenario Consultar	129
Figura 55.	Diagrama de Colaboración Administrar Profesores Escenario Eliminar	130
Figura 56.	Diagrama de Colaboración Administrar Profesores Escenario Establecer Horario	131
Figura 57.	Diagrama de Colaboración Administrar Profesores Escenario Adicionar Materia	132
Figura 58.	Diagrama de Colaboración Administrar Profesores Escenario Eliminar Materia	133
Figura 59.	Diagrama de Colaboración Administrar Materias Escenario Adicionar	134
Figura 60.	Diagrama de Colaboración Administrar Materias Escenario Modificar	135
Figura 61.	Diagrama de Colaboración Administrar Materias Escenario Consultar	136
Figura 62.	Diagrama de Colaboración Administrar Materias Escenario Eliminar	137
Figura 63.	Diagrama de Colaboración Administrar Usuarios Escenario Adicionar	138
Figura 64.	Diagrama de Colaboración Administrar Usuarios Escenario Modificar	139
Figura 65.	Diagrama de Colaboración Administrar Usuarios Escenario Consultar	140
Figura 66.	Diagrama de Colaboración Administrar Usuarios Escenario Eliminar	141
Figura 67.	Diagrama de Colaboración Administrar Usuarios Escenario Restablecer Clave	142

Figura 68.	Diagrama de Colaboración Administrar Carga Escenario Adicionar Grupo	143
Figura 69.	Diagrama de Colaboración Administrar Carga Escenario Consultar	144
Figura 70.	Diagrama de Colaboración Administrar Carga Escenario Eliminar Grupo	145
Figura 71.	Diagrama de Colaboración Administrar Carga Escenario Asignar Carga	146
Figura 72.	Diagrama de Colaboración Administrar Carga Escenario Reiniciar Carga	147
Figura 73.	Diagrama de Colaboración Administrar Carga Escenario Asignar Grupo	148
Figura 74.	Diagrama de Colaboración Administrar Carga Escenario Liberar Grupo	149
Figura 75.	Diagrama de Estados Caso de Uso Administrar Profesores	150
Figura 76.	Diagrama de Estados Caso de Uso Administrar Materias	151
Figura 77.	Diagrama de Estados Caso de Uso Administrar Usuarios	152
Figura 78.	Diagrama de Estados Caso de Uso Administrar Carga	153
Figura 79.	Diagrama de Actividades Administrar Profesores	154
Figura 80.	Diagrama de Actividades Administrar Materias	155
Figura 81.	Diagrama de Actividades Administrar Usuarios	156
Figura 82.	Diagrama de Actividades Administrar Carga	157
Figura 83.	Diagrama de Clases	158
Figura 84.	Arquitectura Web-enable	177
Figura 85.	Diagrama de Subsistemas	178
Figura 86.	Diagrama de Secuencia de Ventanas – Administrar Profesores	180

Figura 87.	Diagrama de Secuencia de Ventanas – Administrar Materias	181
Figura 88.	Diagrama de Secuencia de Ventanas – Administrar Usuarios	182
Figura 89.	Diagrama de Secuencia de Ventanas – Administrar Carga	183
Figura 90.	Diagrama de Interacción con el Usuario – Administrar Profesores	184
Figura 91.	Diagrama de Interacción con el Usuario – Administrar Materias	185
Figura 92.	Diagrama de Interacción con el Usuario – Administrar Usuarios	186
Figura 93.	Diagrama de Interacción con el Usuario – Administrar Carga	187
Figura 94.	Diagrama de Componentes	188
Figura 95.	Diagrama de Despliegue	189
Figura 96.	Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Profesores	190
Figura 97.	Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Materias	191
Figura 98.	Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Usuarios	192
Figura 99.	Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Carga	193
Figura 100.	Ventana Inicio de Sesión	194
Figura 101.	Ventana Menú Principal	195
Figura 102.	Ventana Menú Principal – Administrar Profesores	196
Figura 103.	Ventana Menú Principal – Administrar Materias	197
Figura 104.	Ventana Menú Principal – Administrar Usuario	198
Figura 105.	Ventana Menú Principal – Administrar Carga	199
Figura 106.	Ventana Adicionar Profesor	200
Figura 107.	Ventana Modificar Profesor - Seleccionar Profesor	201

Figura 108.	Ventana Modificar Profesor	202
Figura 109.	Ventana Consultas Profesor - Seleccionar Consulta y Vista	203
Figura 110.	Ventana Eliminar Profesor	204
Figura 111.	Ventana Horario Profesor - Seleccionar Profesor	205
Figura 112.	Ventana Horario Profesor - Horario	206
Figura 113.	Ventana Adicionar Materia a Perfil - Seleccionar Profesor	207
Figura 114.	Ventana Adicionar Materia a Perfil - Seleccionar Materia	208
Figura 115.	Ventana Eliminar Materia a Perfil - Seleccionar Profesor	209
Figura 116.	Ventana Eliminar Materia a Perfil - Seleccionar Materia	210
Figura 117.	Ventana Adicionar Materia	211
Figura 118.	Ventana Modificar Materia - Seleccionar Materia	212
Figura 119.	Ventana Modificar Materia	213
Figura 120.	Ventana Consultas Materia - Seleccionar Consulta y Vista	214
Figura 121.	Ventana Eliminar Materia	215
Figura 122.	Ventana Adicionar Administrador	216
Figura 123.	Ventana Modificar Usuario - Seleccionar Usuario	217
Figura 124.	Ventana Modificar Usuario	218
Figura 125.	Ventana Consultas Usuario - Seleccionar Consulta y Vista	219
Figura 126.	Ventana Eliminar Usuario	220
Figura 127.	Ventana Restablecer Clave - Seleccionar Usuario	221
Figura 128.	Ventana Adicionar Grupo	222
Figura 129.	Ventana Consultas Carga - Seleccionar Consulta y Vista	223
Figura 130.	Ventana Eliminar Grupo – Seleccionar Materia	224

Figura 131.	Ventana Eliminar Grupo – Seleccionar Grupo	225
Figura 132.	Ventana Asignación de Carga – Seleccionar jornada y categoría	226
Figura 133.	Ventana Asignación de Carga – Lista de Profesores	227
Figura 134.	Ventana Asignación de Carga – Resultados	228
Figura 135.	Ventana Reiniciar Carga	229
Figura 136.	Ventana Reiniciar Carga – Seleccionar tipo	230
Figura 137.	Ventana Asignar Grupo	231
Figura 138.	Ventana Liberar Grupo	232
Figura 139.	Modelo Entidad-Relación	233

ÍNDICE DE TABLAS

	Pág.
Tabla 1. Componentes del modelo “send+more=money”	43
Tabla 2. Requerimientos del sistema	66
Tabla 3. Espacio de tiempo de clases establecido por la universidad	72
Tabla 4. Eventos Caso de Uso Administrar Profesores Escenario Adicionar	78
Tabla 5. Eventos Caso de Uso Administrar Profesores Escenario Modificar	79
Tabla 6. Eventos Caso de Uso Administrar Profesores Escenario Consultar	80
Tabla 7. Eventos Caso de Uso Administrar Profesores Escenario Eliminar	81
Tabla 8. Eventos Caso de Uso Administrar Profesores Escenario Establecer Horario	82
Tabla 9. Eventos Caso de Uso Administrar Profesores Escenario Adicionar Materia	83
Tabla 10. Eventos Caso de Uso Administrar Profesores Escenario Adicionar Materia	84
Tabla 11. Eventos Caso de Uso Administrar Materias Escenario Adicionar	86
Tabla 12. Eventos Caso de Uso Administrar Materias Escenario Modificar	87
Tabla 13. Eventos Caso de Uso Administrar Materias Escenario Consultar	88
Tabla 14. Eventos Caso de Uso Administrar Materias Escenario Eliminar	89
Tabla 15. Eventos Caso de Uso Administrar Usuarios Escenario Adicionar	91
Tabla 16. Eventos Caso de Uso Administrar Usuarios Escenario Modificar	92
Tabla 17. Eventos Caso de Uso Administrar Usuarios Escenario Consultar	93
Tabla 18. Eventos Caso de Uso Administrar Usuarios Escenario Eliminar	94

Tabla 19. Eventos Caso de Uso Administrar Usuarios Escenario Restablecer Clave	95
Tabla 20. Eventos Caso de Uso Administrar Carga Escenario Adicionar Grupo	97
Tabla 21. Eventos Caso de Uso Administrar Carga Escenario Consultar	98
Tabla 22. Eventos Caso de Uso Administrar Carga Escenario Eliminar Grupo	99
Tabla 23. Eventos Caso de Uso Administrar Carga Escenario Asignar Carga	100
Tabla 24. Eventos Caso de Uso Administrar Carga Escenario Reiniciar Carga	101
Tabla 25. Eventos Caso de Uso Administrar Carga Escenario Asignar Grupo	102
Tabla 26. Eventos Caso de Uso Administrar Carga Escenario Liberar Grupo	103
Tabla 27. Diccionario de Clases	159
Tabla 28. Diccionario de atributos de la clase DB	160
Tabla 29. Diccionario de atributos de la clase Profesor	161
Tabla 30. Diccionario de atributos de la clase Materia	161
Tabla 31. Diccionario de atributos de la clase Usuario	162
Tabla 32. Diccionario de atributos de la clase Perfil	162
Tabla 33. Diccionario de métodos de clase BD	163
Tabla 34. Diccionario de métodos de clase profesor	164
Tabla 35. Diccionario de métodos de clase materia	164
Tabla 36. Diccionario de métodos de clase usuario	165
Tabla 37. Diccionario de métodos de clase perfil	166
Tabla 38. Diccionario de métodos de clase carga	166
Tabla 39. Diccionario de métodos de clase controlador	167
Tabla 40. Diccionario de métodos de clase interfaz	171

Tabla 41. Descripción de la tabla usuario	236
Tabla 42. Descripción de la tabla materia	236
Tabla 43. Descripción de la tabla perfil	237
Tabla 44. Descripción de la tabla profesor	238
Tabla 45. Descripción de la tabla horario_profesor	238
Tabla 46. Descripción de la tabla horario_materia	239

GLOSARIO

- **Algoritmo**¹

Conjunto de instrucciones o de pasos a seguir para la resolución de un problema específico.

- **Análisis**²

Fase del ciclo de desarrollo de sistemas que consta de dos tareas: llevar a cabo una investigación preliminar y realizar un análisis detallado. En esta etapa los analistas se encargan de analizar los requerimientos del sistema (a veces se incluye en esta etapa la recolección de requerimientos también). Esta etapa centra su atención en la interacción de los usuarios con el sistema.

- **Aplicación**³

Software que utiliza un computador para llevar a cabo tareas; también se le conoce como programa.

- **Base de datos**⁴

Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc. Las bases de datos son uno de los grupos de aplicaciones de productividad personal más extendidos.

- **Diseño**⁵

El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del

¹ Thomson Course Technology. 2005. Computer and Internet Technology Definitions in spanish. Cánada. Shana Rosental. [Libro en línea]. Disponible desde Internet en: <<http://books.google.com/books?id=DoF9lJnKtwC&pg=PA86&dq=diccionarios+de+tecnolog%C3%ADa+e+inform%C3%A1tica&lr=&hl=es&cd=9#v=onepage&q&f=false>>

² Ibid.

³ Ibid.

⁴ Mataix Lorda M. Mataix Hidalgo M. 1999. Diccionario de Electrónica, Informática y Energía Nuclear. España. Ediciones Díaz de Santos S.A. [Libro en línea]. Disponible desde Internet en: <<http://books.google.com/books?id=BVDd8xRNQ3IC&pg=PA293&dq=diccionarios+de+tecnologia+informatica+internet+istemas&hl=es&cd=10#v=onepage&q&f=false>>

⁵ Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill. [Libro en línea]. Disponible desde Internet en: <<http://omarenm.wordpress.com/2008/03/03/ingenieria-del-software-roger-s-pressman/>>

diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas tales como implementar todos los requisitos explícitos contenidos en el modelo de análisis y acumular todos los requisitos implícitos que desea el cliente.

El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

- **Dominio⁶**

Es el conjunto de valores que puede tomar la variable independiente x.

- **Explosión combinatoria⁷**

La explosión combinatoria se encuentra en situaciones donde las elecciones están compuestas secuencialmente, es decir, dado un conjunto de elementos se pueden obtener diferentes arreglos ordenados de estos, permitiendo una vasta cantidad de posibilidades. Situaciones de este tipo ocurren en problemas de inversión financiera, manejo de inventarios, diseño de circuitos integrados, manejo de recursos hidráulicos, mantenimiento de sistemas, etc.

- **Instanciación⁸**

Proceso de creación de un objeto a partir de una clase.

- **Inteligencia artificial⁹**

Es una de las áreas de las ciencias computacionales encargadas de la creación de hardware y software que tenga comportamientos inteligentes

⁶ Ibid.

⁷ Gutiérrez M. De los Cobos S. Pérez B. 1994. Búsqueda Tabú: Un Procedimiento Heurístico para Solucionar Problemas de Optimización Combinatoria. México. [Revista en línea]. Disponible desde Internet en: <<http://www.azc.uam.mx/publicaciones/enlinea2/num1/1-3.htm>>

⁸ Larman C. 1999. UML y patrones: introducción al análisis y diseño orientado a objetos. Prentice Hall. [Libro en línea]. Disponible desde Internet en: < <http://www.scribd.com/doc/457980/UML-y-Patrones-by-Craig-Larman>>

⁹ Ralph m. Stair y george e. Reynolds, 2002. Principios de Sistemas de Información. Inteligencia Artificial y Sistemas expertos, Editorial Thomson

- **Interfaz**¹⁰

Engloba la forma en la que el operador interactúa con el ordenador, los mensajes que éste recibe en pantalla, las respuestas del ordenador a la utilización de periféricos de entrada de datos, etc.

- **Modelo matemático**¹¹

Se define como una descripción desde el punto de vista de las matemáticas de un hecho o fenómeno del mundo real, desde el tamaño de la población, hasta fenómenos físicos como la velocidad, aceleración o densidad. El objetivo del modelo matemático es entender ampliamente el fenómeno y tal vez predecir su comportamiento en el futuro.

- **Optimización combinatorial**¹²

Buscar la mejor opción entre un conjunto de un número finito de elementos que pueden ser generados mediante reglas que definen el problema a tratar.

- **Paradigma**¹³

Un paradigma es una técnica, un modelo o un conjunto de herramientas para representar la solución de problemas específicos

- **Programación lógica**¹⁴

Los programas se ejecutan verificando la presencia de una cierta condición habilitadora y, cuando se satisface, ejecutan una acción apropiada. También se le conoce como lenguaje con base a reglas.

¹⁰ <http://www.glosarium.com/term/>

¹¹ STEWART, James. 2002. Cálculo, Trascendentes Tempranas". 4 ed. Tr. de Andrés Sestier. México, Ed. Thomson

¹² Gutiérrez M. De los Cobos S. Pérez B. 1994. Búsqueda Tabú: Un Procedimiento Heurístico para Solucionar Problemas de Optimización Combinatoria. México. [Revista en línea]. Disponible desde Internet en: <<http://www.azc.uam.mx/publicaciones/enlinea2/num1/1-3.htm>>

¹³ <http://www.glosarium.com/term/>

¹⁴ Louden K. 2004. Programación Lógica. Teoría y Práctica, P. Julián, M. Alpuente, Pearson Prentice may. [Libro en línea]. Disponible desde Internet en: <http://books.google.com/books?id=MnrVc_GVKbMC&pg=PA493&dq=Programaci%C3%B3n+L%C3%B3gica.+Teor%C3%ADa+y+Pr%C3%A1ctica,&hl=es&cd=2#v=onepage&q&f=false>

- **Prolog**¹⁵

(Programming In Logic). Es un lenguaje de programación que tiene su aplicación práctica en el desarrollo de software destinado a Sistemas Expertos.

- **Subsistema**¹⁶

Un subsistema no es ni una función ni un objeto, sino un paquete de clases, asociaciones, operaciones, sucesos y restricciones interrelacionados, y que tienen una interfaz razonablemente bien definida y pequeña con los demás subsistemas. Normalmente, un subsistema se identifica por los servicios que proporciona. Un servicio es un grupo de funciones relacionadas que comparten algún propósito común, tal como el procesamiento de entrada-salida, dibujar imágenes o efectuar cálculos aritméticos. Un subsistema define una forma coherente de examinar un aspecto del problema.

- **Tupla**¹⁷

Una tupla, en matemáticas, es una secuencia ordenada de objetos, esto es, una lista con un número limitado de objetos (una secuencia *infinita* se denomina en matemática como una familia). Las tuplas se emplean para describir objetos matemáticos que tienen estructura, es decir que son capaces de ser descompuestos en un cierto número de componentes.

- **Variable**¹⁸

Estructura de programación que contiene datos. Puede contener números o caracteres alfanuméricos y el programador le asigna un nombre único. Mantiene los datos hasta que un nuevo valor se le asigna o hasta que el programa termine. Una variable es un identificador que se utiliza para representar un dato individual; es decir, una cantidad numérica o carácter de forma parecida a la constante pero en este caso, su valor es variable, asignado en alguna parte del programa.

¹⁵ <http://www.glosarium.com/term/>

¹⁶ Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill. [Libro en línea]. Disponible desde Internet en: <<http://omarenm.wordpress.com/2008/03/03/ingenieria-del-software-roger-s-pressman/>>

¹⁷ STEWART, James. 2002. Cálculo, Trascendentes Tempranas". 4 ed. Tr. de Andrés Sestier. México, Ed. Thomson

¹⁸ <http://www.glosarium.com/term/>

RESUMEN

Este proyecto nació de identificar una necesidad en el proceso de asignación de la carga académica en el Programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira.

El propósito era diseñar e implementar una aplicación que permitiera optimizar dicho proceso logrando así el mejor aprovechamiento de los recursos con los que se cuenta actualmente. Para darle solución al problema se planteó un modelo matemático para ser aplicado usando la técnica de programación con restricciones la cual se ajusta muy bien a la resolución de este tipo de problemas.

INTRODUCCIÓN

Actualmente toda empresa, compañía, o entidad de cualquier tipo que maneje cierta cantidad de información o que requiera la gestión de determinados procesos para su desenvolvimiento en la respectiva área, debe contar con herramientas tecnológicas para estar a la vanguardia.

Es por eso que la mayoría de estas organizaciones ha ido incursionando en el mundo de la tecnología, que si bien puede llegar a ser complejo, el hecho de estar inmersos en él, se convierte casi que en una necesidad, si no se quiere llegar a estar obsoletos en el mercado correspondiente; además que facilita muchas de las tareas inherentes a los diferentes procesos que se realizan.

La constante evolución de la tecnología, hace que cada día se desarrollen nuevas aplicaciones y se diseñen innovadores dispositivos dependiendo de las necesidades de quienes los requieran.

El programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira gestiona de forma manual la distribución de la carga académica de los profesores adscritos al programa, es decir, todo lo relacionado con la creación de los grupos de todas las asignaturas y la respectiva asignación de los docentes a dichos grupos.

Debido a esto, se plantea una solución mediante el análisis, diseño e implementación de un software que le permita a los directivos del programa mejorar la gestión del proceso, ahorrando tiempo y sacándole mejor provecho a las herramientas con que se cuenta y a los recursos que se tienen destinados para tal fin.

Para ello, se formularon una serie de preguntas a la Ingeniera Ana María López Echeverri Coordinadora del Programa de Ingeniería de Sistemas y Computación quien es la persona encargada de todo el proceso de asignación de la carga académica, con el fin de recolectar las inquietudes y sugerencias al respecto y poder implementar así un software acorde a sus necesidades, ya que ella será el usuario principal de éste.

1. TÍTULO DEL PROYECTO

Análisis, diseño e implementación de un sistema informático para el apoyo al proceso de asignación de la carga académica usando programación con restricciones.

2. PLANTEAMIENTO DEL PROBLEMA

Días previos al inicio de cada semestre académico, las directivas del programa involucradas directamente en el proceso de creación de cursos (grupos) y asignación de horarios a profesores y estudiantes, se ven enfrentadas a diversos inconvenientes que retrasan o hacen más complicado dicho proceso. Por ejemplo, deben tener en cuenta el número mínimo de estudiantes necesario para que un grupo pueda ser creado y no generar cruces de horarios en ningún momento. No solo las directivas se ven afectadas a la hora de la matrícula, también los alumnos debido a los retrasos en la generación de la nueva configuración de horarios. La creación de grupos nuevos implica la asignación de un profesor, lo que conllevaría tal vez una reacomodación total o parcial de los horarios ya establecidos.

Entonces se podría decir que comienza de nuevo el trabajo de las directivas encargadas de la asignación de la carga académica. Se deben empezar a considerar las solicitudes de los estudiantes con el fin de facilitarles y/o proporcionales su normal proceso académico, es decir, darles las garantías para que puedan desarrollar su plan de estudio.

Las variables que se deben considerar a la hora de generar los respectivos horarios son: con respecto a los docentes, se debe tener en cuenta el estado en el que éstos se encuentran con la universidad dependiendo del tipo de contratación, de los horarios de disponibilidad (Horario de Contrato menos Horario de Clases para profesores que no estén contratados por horas) y de la capacidad que tengan para dictar una determinada asignatura, es decir, su perfil; con respecto a las asignaturas, se debe guardar relación para fijar las de un mismo semestre, ya que para programar un grupo se debe cumplir no solo que un profesor la pueda dictar sino que a esa hora no haya otro grupo programado de igual semestre. Este sería un caso crítico en el que cada materia cuenta con un solo grupo, para los casos en los que se cuente con más grupos hay mayor libertad para la asignación.

3. IMPORTANCIA DE LA INVESTIGACIÓN

La implementación de este software la consideramos pertinente por los aportes que brinda a la solución práctica de tipo social, económico y educativo de la Universidad. Esta herramienta es muy útil para dicha institución porque le permite optimizar el proceso de la distribución de la carga y de esta manera mejorar el aprovechamiento de los recursos relacionados como lo son docentes, asignaturas, entre otros.

En el aspecto social es importante porque permite realizar una mejor asignación de cursos (horas de clase, asignaturas,...) teniendo en cuenta las necesidades de los alumnos y de los docentes, lo que implicaría mejores condiciones laborales y académicas; en el ámbito económico porque permite realizar una mejor distribución de los recursos humanos y físicos de la Universidad y en el aspecto educativo porque se pueden ofrecer mejores condiciones de estudio, es decir, poder aceptar un mayor número de estudiantes sin afectar la calidad de la educación, ya que habría una mejor distribución de las horas de clase y de los salones permitiendo a los alumnos organizar su respectivo horario de acuerdo a sus necesidades académicas.

4. JUSTIFICACIÓN

Desde hace muchos años, las distintas organizaciones y/o empresas, en búsqueda de ser competitivas y debido a la continua evolución de la Informática y todo lo relacionado con la tecnología, han ido incursionando en este mundo. La manera como se maneja la información ha ido cambiando y casi ha obligado a los Directivos de toda institución a automatizar los procesos y a optimizar la forma en que se manipula la información. Tal es el caso de la Universidad que hace unos pocos años modificó el sistema de matrículas de los estudiantes cada semestre, facilitando ese proceso.

Pero como se mencionaba anteriormente, es casi una necesidad el hecho de actualizarse constantemente para suprimir técnicas y procesos obsoletos que pueden llegar a retrasar determinadas actividades en una organización particular impidiendo así que puedan ser competitivas frente a otras.

Para las Directivas de la Universidad y específicamente para las del Programa de Ingeniería de Sistemas y Computación, el proceso de asignación de la carga académica se ha convertido en una tarea complicada debido a los diversos cambios que se ven obligados a realizar para poder llevar a cabo con éxito dicho proceso y teniendo en cuenta que se realiza manualmente. De ahí surge entonces la necesidad de implementar un sistema informático que a través de reportes sirva como herramienta de apoyo a los Directivos del programa para que puedan desempeñar con éxito esta ardua tarea.

El problema de la asignación de la carga académica no solo le compete al Programa de Ingeniería de Sistemas y Computación sino que abarca todos los programas de la Universidad Tecnológica de Pereira, y más aún a los programas nuevos que sufren más cambios en su recurso profesoral, viéndose esto directamente reflejado en un mayor trabajo por parte de los responsables de este proceso en cada facultad. El alcance de este proyecto está limitado al programa de Ingeniería de Sistemas y Computación, pero puede extenderse en cualquier momento a otras facultades ajustándolo a sus requerimientos.

Muchos de los proyectos que se han realizado en tesis anteriores no han tenido mucha aplicación o ni siquiera han llegado a usarse alguna vez, este en cambio será aplicado dentro de la universidad en una tarea que es vital al inicio del semestre.

5. OBJETIVOS

5.1. General

- Analizar, diseñar e implementar un sistema informático para el apoyo al proceso de asignación de la carga académica usando programación con restricciones.

5.2. Específicos

- Construir un software para la gestión del proceso de asignación de la carga académica del programa de Ingeniería de Sistemas y Computación, que disminuya su complejidad y proporcione información clara a los profesores acerca de la programación de sus clases.
- Recopilar y consignar una base teórica coherente y suficiente que soporte la técnica empleada, teniendo en cuenta el pasado y el presente de la misma.
- Realizar una inspección del proceso manual que permita formar una idea clara de los requerimientos del software y con base en ello, realizar los diagramas respectivos que ilustren la aplicación desde el punto de vista funcional.
- Definir la arquitectura del software y diseñar las interfaces que permitan la interacción del usuario con la aplicación de una manera lo más intuitiva posible.
- Combinar los datos mediante técnicas de programación con restricciones y generar una o varias soluciones o alternativas que ayuden al directivo a escoger el horario que mejor se acomode a los requerimientos. Puede darse el caso que no sea posible generar una solución por la calidad que tienen los recursos de ser limitados, es decir, en algún momento el sistema puede ver agotada la disponibilidad profesoral.

6. MARCO REFERENCIAL

6.1. ANTECEDENTES EN LA SOLUCIÓN DEL PROBLEMA

En este capítulo se mencionarán soluciones o aplicaciones previas, enfocadas a la resolución de problemas similares al abordado en este proyecto, las cuales nos sirvieron como base para implementar y mejorar las funcionalidades del software adaptándolo a las necesidades del programa de Ingeniería de Sistemas y Computación de la Universidad.

Para eso se consultó información acerca de otros programas relacionados con la creación de horarios de los cuales mencionaremos algunos. Para ampliar la información sobre alguno de ellos, se puede visitar la página asociada.

- **aSc Time Tables**¹⁹

Este programa fue implementado por la empresa desarrolladora aSc Applied Software Consultants para crear horarios de primaria y secundaria. Es uno de los generadores de horarios más completos y eficientes que existen, razón por la cual ha sido comprado por escuelas de 114 países y además ha ganado premios como medallas de oro en exposiciones internacionales. Está disponible en varios idiomas.

Comentarios. Este programa tiene funcionalidades muy útiles para la generación óptima de los diferentes horarios de clase dependiendo de las exigencias dadas por el usuario. Se tienen en cuenta factores importantes para evitar asignaciones erróneas o incompatibilidades en lo que a profesores, horas y salones se refiere. Uno de estos errores puede ser el cruce de horarios, situación que precisamente también se pretende evitar con la implementación del programa. Aunque es un software de buena calidad enfocado para colegios su aplicación para otro tipo de instituciones educativas como universidades es limitada o no aplicaría dependiendo de las condiciones.

¹⁹ aSc Applied Software Consultants. 2009. aSc Time Tables. Disponible desde Internet en: http://www.asctimetables.com/timetables_es.html.

- **GHC Generador de Horarios**²⁰

Este programa fue implementado por la empresa informática Peñalara Software; es la principal herramienta de España de generación de horarios académicos para centro de enseñanza. Posee 1500 clientes activos en España, Portugal, Marruecos y Latinoamérica.

Comentarios. La verificación del horario antes de ser generado para evitar errores o incompatibilidades es una buena herramienta para la optimización de la tarea de creación del horario. En nuestro caso, antes de generar los reportes respectivos se podrán hacer verificaciones sobre las condiciones y restricciones dadas inicialmente. Por otro lado, la versión gratuita de este programa está limitada a solo 10 profesores lo que casi obliga al pago para poder acceder al programa sin limitación alguna.

- **gp-Untis - Horario Software para colegios**²¹

Es un programa implementado por el grupo Gruber & Petters de Austria, desarrolladores de software para gestión de información horaria. Posee más de 14.000 usuarios en más de 80 países y todo tipo de centros de enseñanza; ha sido traducido a más de 20 idiomas.

Comentarios. Aunque tiene funcionalidades útiles para la creación de los horarios como la impresión del horario en forma de lista que es más adecuado que la vista en cuadrícula horaria porque se destaca claramente la sucesión de horas y además la adaptación de módulos al paquete estándar según las necesidades adicionales, el acceso o inserción de dichos módulos tiene un costo y la versión demo con la que se pueden utilizar todas las funciones impide guardar los datos introducidos y el número de grupos es limitado. Sin embargo, se puede decir que en general es un programa que brinda soluciones óptimas a los usuarios dependiendo de sus necesidades.

²⁰ Peñalara Software. 2009. GHC Generador de Horarios. Disponible en Internet desde: URL: <http://www.penalara.com/ghc.asp>.

²¹ grupo Gruber & Petters de Austria. 2009. gp-Untis - Horario Software para colegios. Disponible en Internet desde: http://www.grupet.at/home_es.php

- **DocCF – Software de Gestión Escolar²²**

Este es un programa implementado por el Grupo CF Developer, empresa de servicios informáticos para España y Latinoamérica, que ayuda a optimizar algunas tareas realizadas por los directivos de una institución educativa. El programa tiene varias funcionalidades (todo en una sola aplicación) agrupadas en Módulos que cabe mencionar, aunque solo se comentará sobre la Asignación y control de horarios. Disponible en español.

Comentarios. Este programa, además de la gestión de horarios para instituciones educativas realiza otro tipo de gestiones como económica, de biblioteca, entre otras. Tiene un costo de 250 dólares para la versión básica y 500 dólares para la versión completa.

Este tipo de software con interfaces elaboradas y herramientas de programación nuevas, requiere que la máquina en donde vaya a ser instalado, cumpla con una serie de requerimientos como los son las librerías de ejecución de java o .NET si es el caso, nuestra aplicación solo necesita que la maquina tenga un navegador estándar como lo es el Mozilla Firefox o Internet Explorer y presenta total compatibilidad con ambos, no se necesita ningún plugin adicional.

- **OPTIHPER Asignación Optimizada de Horarios al Personal²³**

Es un programa desarrollado en la Universidad Politécnica de Valencia España que gestiona la asignación objetiva y equilibrada de recursos humanos como horarios y tareas al personal de una empresa, basándose en condiciones y/o restricciones dadas. Está dirigido a empresas de distribución comercial, grandes superficies, de transporte (aéreo, terrestre y marítimo), hospitales y centros docentes.

Comentarios. Este programa se asemeja más a un software de propósito general, en la materia claro está, por esta razón se tendría que invertir un esfuerzo adicional acondicionándolo al propósito que compete, la distribución de la carga académica en una universidad, en este caso la UTP.

Esta aplicación es un producto del grupo de investigación IA-GPS de la universidad politécnica de valencia, no tienen representación en Colombia.

²² Grupo CF Developer. 2009. DocCF – Software de Gestión Escolar. Disponible en Internet desde: <http://www.grupocfdeveloper.com/index.htm>

²³ Barber Federico – Universidad Politécnica de Valencia. 2009. OPTIHPER Asignación Optimizada de Horarios al Personal. Disponible en Internet desde: <http://users.dsic.upv.es/grupos/gps/optihper/>

Cabe aclarar que todos estos programas mencionados tienen funcionalidades que no se adaptan o no están enfocados a suplir completamente las necesidades que tiene el programa de Ingeniería de Sistemas y Computación de la Universidad en términos de la distribución de la carga académica. Es por eso, que se quiso implementar este sistema informático utilizando metodologías y conceptos que permitieran brindar una solución óptima al problema a abordar. En el siguiente capítulo se podrá encontrar información referente a la tecnología empleada, los conceptos fundamentales y la terminología pertinente para la implementación concreta de la solución.

6.2. ACTUALIDAD DE LA PROGRAMACIÓN CON RESTRICCIONES

En esta sección se podrá obtener de forma clara el concepto, las aplicaciones e información en general relacionada con el paradigma de programación con restricciones, el cual permitió resolver el problema planteado inicialmente.

Aunque la programación con restricciones no es muy reciente, es una tecnología que no se ha dejado de usar ni se considera obsoleta, por el contrario, está suscitando un interés creciente en la comunidad investigadora por varios motivos:

- La teoría que sustenta la programación con restricciones es una teoría muy sólida, lo que hace este paradigma mucho más robusto.
- Su campo de investigación es muy heterogéneo, abarca desde tópicos puramente teóricos en la lógica matemática hasta aplicaciones totalmente prácticas en la industria. Como consecuencia de ello, el paradigma está atrayendo un amplio interés comercial pues es muy apropiado para el modelado de una extensa gama de problemas de optimización y, en particular, esos problemas que involucran restricciones heterogéneas y búsqueda combinatoria.
- Hoy en día es protagonista en la resolución de problemas bastante complejos.

La compañía NICTA, Centro de Excelencia en Tecnologías de Información y Comunicación (TIC) de Australia, es la organización más grande de este país dedicada a la investigación de las TIC's; de entre sus proyectos se destacan la "Plataforma de programación con Restricciones" llamada G12, un software para resolver problemas industriales de optimización combinatorial de gran escala. En el proyecto se sustenta que la revolución en el procesamiento de la información, ha permitido que empresas de todos los tamaños puedan tener a la mano la información actualizada de todas sus actividades y recursos, la Plataforma de Programación con Restricciones permitirá convertir esta información en beneficios inmediatos mediante el acompañamiento en la toma de decisiones y la asignación de recursos de una forma optimizada. El proyecto de investigación se divide en cuatro líneas encaminadas a construir versiones mejoradas, las líneas son: lenguajes de modelado, capacidades de solución (tecnología de búsqueda y solución), control del lenguaje de modelado y ambiente de solución de problemas.²⁴

Existen algunas organizaciones como la ACP (Association for Constraint Programming) que coordinan y promocionan la aplicación de la programación con

²⁴ NICTA (Australia's ICT Research Centre of Excellence). 2008; 8 de marzo de 2010. Constraint Programming Platform. [web en línea] Disponible desde Internet en: <http://www.nicta.com.au/research/projects/constraint_programming_platform>

restricciones en diferentes áreas del mundo científico, impulsan la enseñanza en instituciones académicas y su adopción en el mundo industrial.²⁵

En particular, la ACP organiza la conferencia anual internacional de principios y práctica de la programación con restricciones.

En las últimas dos versiones de la conferencia una realizada en Sydney, Australia²⁶ y la otra en Lisboa, Portugal²⁷, se trataron temas relacionados con la optimización del método mismo, es decir, mejoras en el modelado, conteo de soluciones y algoritmos, cuantificación de variables con dominios inciertos, preferencia de restricciones para evitar sistemas sobre-restringidos y detección de simetría en espacios de búsqueda evitando también búsquedas redundantes. La otra parte de la conferencia estuvo dedicada a aplicaciones como la planeación y la secuenciación de eventos en muchas áreas, oleoductos de gran tamaño, sistemas ferroviarios, sistemas altamente configurables, etc.

En la Universidad Cornell de Estados Unidos existe un instituto llamado “Instituto para la sostenibilidad computacional” (Institute for Computational Sustainability) el cual tuvo participación en la última conferencia realizada por la ACP anteriormente mencionada. En dicha conferencia, el instituto realizó el primer taller sobre razonamiento con restricciones y optimización para la sostenibilidad computacional, en donde argumentaron la posibilidad de aplicar la ciencia computacional, la ciencia de la información, investigación de operaciones, la matemática aplicada y la estadística para el balanceo de las necesidades ambientales, económicas y sociales para un desarrollo sostenible. De esta manera podemos ver la heterogeneidad de los campos de investigación de la programación con restricciones.

Siguiendo con los conceptos fundamentales, a continuación se podrá encontrar de forma concreta acerca de la Programación Lógica y la Programación con Restricciones.

²⁵ ACP (Association for Constraint Programming). 2005. [web en línea] Disponible desde Internet en <<http://4c.ucc.ie/a4cp/>>

²⁶ The 14th International Conference on Principles and Practice of Constraint Programming. 2008. Sydney, Australia. [web en línea] Disponible desde Internet en <<http://ww2.cs.mu.oz.au/cp2008/>>

²⁷ The 15th International Conference on Principles and Practice of Constraint Programming. 2009. Lisbon, Portugal. [web en línea] Disponible desde Internet en <<http://centria.di.fct.unl.pt/conferences/cp2009/>>

6.3. REFERENTES DEL MARCO TEÓRICO

6.3.1. Programación lógica. La programación lógica tiene sus raíces en la automatización de la demostración de teoremas basado en el método de resolución del matemático, científico y profesor de la universidad de Siracusa de Estados Unidos, Alan Robinson, quien en su obra principal introduce el principio de resolución, la noción y el algoritmo de unificación. Usando su resolución, se pueden probar teoremas dados como fórmulas de lógica de primer orden, con el fin de obtener un 'sí' o 'no' como respuesta a una pregunta, aunque bajo este modelo, no era posible realizar cálculos con una respuesta de este tipo.

La solución a este inconveniente fue sugerida por el estadounidense, lógico y científico de la computación, Robert Kowalski quien propuso una versión modificada de la resolución, que consistía en un subconjunto de la lógica de primer orden que permitía generar sustituciones que en últimas satisfacían la fórmula original. Esta sustitución puede ser interpretada como el resultado del cálculo. Este acercamiento después se llamó Programación Lógica.

Al mismo tiempo, el francés, científico de la computación y profesor de la universidad de Montreal, Alain Colmerauer y sus compañeros, trabajaban en un lenguaje de programación para el procesamiento del lenguaje natural basados en la demostración del teorema automatizado, lo que los llevó a la creación del Prolog.

Prolog puede ser visto como la realización práctica de la idea de programación lógica. Este comenzó como un lenguaje de programación para crear aplicaciones de procesamiento del lenguaje natural, pero gracias a las contribuciones de varios investigadores fue transformado en un lenguaje de propósito general.

Colmerauer, en su experimentación con Prolog, solucionó algunas de sus más importantes limitantes, por ejemplo, podía resolver ecuaciones entre términos, pero no entre cadenas. Usando la terminología actual, Prolog solo soporta un solucionador de restricciones. Esto lo llevo a diseñar los sucesores de Prolog, Prolog II, Prolog III y Prolog IV. Cada uno implementa nuevas características con respecto a su antecesor y representa un avance en la programación lógica hacia la programación con restricciones.

En particular, en Prolog III, se pueden plantear restricciones sobre listas, valores booleanos y reales. Prolog III puede verse como la primera realización de la programación lógica con restricciones.

Prolog, al ser una lenguaje de Programación con restricciones, es una muy buena opción para adoptarlo en la aplicación, ya que el proceso de adición de grupos cuando se distribuye la carga académica, consiste básicamente en encontrar una

combinación que satisfaga los límites que imponen los recursos finitos de la universidad, estos límites se traducen en restricciones y finalmente se llevan a Prolog.²⁸

6.3.2. Programación con restricciones. El concepto formal de restricción fue usado originalmente en física y optimización combinatorial. La primera vez que fue adoptado en la ciencia computacional fue hecho por el estadounidense, científico de la computación y pionero del Internet, Ivan Sutherland, quien lo usó para describir su sistema de dibujo interactivo 'Sketchpad'. En los setentas, fueron propuestos varios lenguajes experimentales que usaban la noción de restricción y confiaban en el concepto de solución de restricciones. También en los setentas, se formuló el concepto de problema de satisfacción de restricciones en el campo de la inteligencia artificial y se usó para describir problemas de visión por computador. Después, se identificó que el concepto de propagación de restricciones era una vía crucial para copiar en la explosión combinatoria cuando se resolvían problemas de satisfacción de restricciones usando búsqueda de arriba hacia abajo.

Búsqueda de arriba hacia abajo (Top-down search) es un nombre genérico para una serie de búsquedas en las cuales se pretende construir una solución tratando de extender una solución parcial mediante la adición de restricciones. En el caso más simple, cada restricción asigna un nuevo valor a otra variable. El ejemplo más común de este tipo de búsqueda es el backtracking o vuelta atrás, el cual se remonta al siglo diecinueve. Después llegó una mejora al backtracking, que se llamó búsqueda de ramificación y poda (Branch and bound search), la cual fue usada por primera vez en optimización combinatorial. En los ochentas salieron los primeros lenguajes de programación con restricciones, los más significativos fueron los basados en el paradigma de programación lógica. La razón principal del éxito del acercamiento a la programación con restricciones es que los predicados de las restricciones y de la programación lógica son ambas relaciones matemáticas y las variables son vistas como desconocidas en términos de álgebra, lo que contrasta con los lenguajes imperativos, en los cuales estas son entes cambiantes, pero en cada momento son entidades conocidas como en el cálculo.

El paradigma resultante es llamado programación lógica con restricciones, y Prolog III como se dijo anteriormente, es un ejemplo de lenguaje de programación que aplica este paradigma.²⁹

²⁸ Apt, Krzysztof R. Wallace, M. Diciembre 2006; Constraint Logic Programming Using Eclipse. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780511345852>

²⁹ Apt, Krzysztof R. Wallace, M. Diciembre 2006; Constraint Logic Programming Using Eclipse. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780511345852>

La programación con restricciones hoy en día se utiliza para resolver problemas de muchas áreas tales como inteligencia artificial, investigación operativa, bases de datos, sistemas expertos, entre otros. Un ejemplo de aplicación muy típico en la programación con restricciones es el scheduling o planificación de tiempos, materia que enmarca este proyecto.

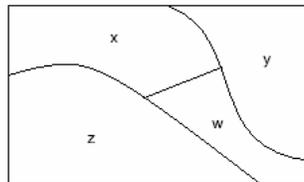
En muchas situaciones encontradas a diario y en las cuales se tenga que tomar una decisión, se estará sujeto a restricciones que limitan a la hora de hacer una elección.

En los últimos años, la programación con restricciones ha ganado importancia gracias a su potencial para resolver grandes problemas reales, por tal motivo, esta tecnología fue reconocida por la ACM (Association for Computer Machinery) como una de las direcciones estratégicas en la investigación en informática, sin embargo, este tipo de programación, no es muy conocida y comprendida.

La programación con restricciones es básicamente un sistema en donde se intenta resolver un problema basados en sus restricciones, esta se divide en dos ramas: la satisfacción de restricciones y la resolución de restricciones, la diferencia principal es que la primera trabaja con dominios finitos mientras que la segunda con dominios infinitos o más complejos.

6.3.2.1. Ejemplo de un problema que se puede formular como un CSP (Constraint Satisfaction Problem)

Figura 1. Mapa dividido en cuatro regiones

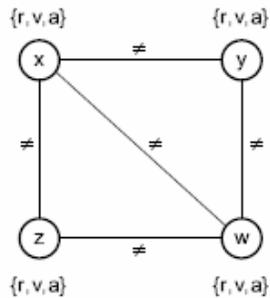


Fuente: Revista Iberoamericana de Inteligencia Artificial N° 20³⁰

Un CSP clásico es el problema de la coloración de un mapa. Se tiene un mapa dividido en varias regiones y se tiene un conjunto de colores, el problema consiste en colorearlas de tal forma que las regiones adyacentes tengan diferente color.

³⁰ Barber, F. Salido, M; Julio de 2003. Revista Iberoamericana de Inteligencia Artificial. 20 ed. Introduction to constraint programming. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en: <http://erevista2.aepia.org/index.php/ia/article/view/827/819> [con acceso el 15-4-2009].

Figura 2. Grafo de restricciones



Fuente: Revista Iberoamericana de Inteligencia Artificial N° 20.³¹

Las regiones, que han sido llamadas w, x, y, z, son las variables del problema, las cuales pueden tomar los valores r, v, a, y se ha definido una restricción por cada par de regiones adyacentes, en la figura se puede ver el grafo que ilustra la situación.

6.3.2.2. Resolución de un CSP. La resolución de un CSP se divide en dos etapas:

- **Modelado.** Consiste en llevar el problema a términos de los componentes de un CSP, variables, valores posibles o dominios y restricciones.
- **Procesamiento.** Para procesar los elementos formulados en el modelado existen dos técnicas:
 - ✓ **Técnica de consistencia.** Basada en eliminar valores inconsistentes de los dominios de las variables.
 - ✓ **Algoritmos de búsqueda.** Se trata de buscar en el espacio de soluciones de una forma sistemática hasta encontrar una solución o probar que no existe.

6.3.2.3. Modelado de un CSP. De la manera como se modele el problema va a depender mucho la complejidad para resolverlo; como se dijo antes, el modelado consiste en expresar el problema en términos de variables, dominios y restricciones.

³¹ Ibid.

Ejemplo. El conocido problema ‘send+more=money’ que consiste en encontrar valores para cada letra de tal forma que se cumpla la operación indicada, los elementos serían:

- Variables = {s, e, n, d, m, o, r, y}
- Dominio = {0,...,9}
- Restricciones: $10^3(s+m)+10^2(e+o)+10(n+r)+d+e = 10^4m+10^3o+10^2n+10e+y$
Todas diferentes(s, e, n, d, m, o, r, y)

Figura 3. Esquema de sumado

```

  s e n d
+ m o r e
-----
m o n e y

```

Fuente: Constraint Logic Programming using Eclipse.

Este modelo no es muy eficiente ya que todas las variables hacen parte de la primera restricción, lo que hace obligatorio instanciar o dar valor a todas las variables antes de cada comprobación, por lo tanto no se puede recortar el espacio de búsqueda, retardando la solución.

Una forma de optimizar el modelo es generar ecuaciones más pequeñas a partir de las operaciones intermedias que se realizan antes de llegar al resultado final, para esto se requieren las variables c1, c2, c3, que son las variables de acarreo. Analizando el problema, m debe tomar el valor 1 y el dominio de s es {1,...,9}, el modelo queda planteado de la siguiente manera:³²

Tabla 1. Componentes del modelo “send+more=money”

Variables	Dominio	Restricciones
c1, c2, c3	{0,1}	$e + d = y + 10c1$
M	{1}	$c1 + n + r = e + 10c2$
S	{1,...,9}	$c2 + e + o = n + 10c3$
e, n, d, o, r, y	{0,...,9}	$c3 + s + m = 10m + o$ Todas diferentes (s, e, n, d, m, o, r, y)

Fuente: los autores.

³² Apt, Krzysztof R. Wallace, M. Diciembre 2006; Constraint Logic Programming Using Eclipse. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780511345852>.

6.3.2.4. Conceptos CSP. Un CSP (Constraint Satisfaction Problem) es un problema de satisfacción de restricciones definido por la terna (X, D, C) .

- X : conjunto de n variables $\{x_1, \dots, x_n\}$
- $D = \langle D_1, \dots, D_n \rangle$ vector de dominios, donde D_i corresponde a X_i .
- C : conjunto finito de restricciones, cada restricción C_n está definida sobre un conjunto de variables $\{x_1, \dots, x_n\}$, el cual limita los valores que las variables pueden tomar.

Asignación (instanciación). Es un par variable-valor (x, a) en donde el valor a es asignado a la variable x . La asignación de un conjunto de variables se puede denotar por una tupla de la forma $((x_1, a_1), \dots, (x_i, a_i))$, la cual es localmente consistente si satisface todas las restricciones que intervienen.

Por simplicidad $((x_1, a_1), \dots, (x_i, a_i))$ se define como (a_1, \dots, a_i) .

Solución. Se tendrá cuando se encuentre una tupla consistente que contenga todas las variables, es decir, de la forma (a_1, a_2, \dots, a_n) .

En la búsqueda se puede encontrar una solución cualquiera, todas las soluciones o una óptima partiendo de alguna función objetivo.³³

6.3.2.5. Notación CSP. La notación que se usará en este capítulo para la descripción de los problemas de satisfacción de restricciones será la siguiente:

General:

- n : número de variables del CSP.
- La longitud del dominio de una variables x_i se denota por $d_i = |D_i|$.
- c : número de restricciones.
- Aridad: la aridad de una restricción es la cantidad de variables que intervienen en ella, el máximo valor es k .

Variables. Para las variables usaremos las últimas letras del alfabeto, x, y, z ; denotaremos el conjunto de todas las variables así: $X_{i, \dots, j}$.

Dominios/Valores. El dominio de una variable x_i se denota por D_i . Usaremos las primeras letras del alfabeto para indicar valores individuales de algún dominio, a, b, c .³⁴

³³ Rossi F. Van Beek P. Walsh T. Agosto 2006; Handbook of Constraint Programming. Holanda. Elsevier. [libro en línea] Disponible desde Internet en: http://www.elsevier.com/wps/find/bookaudience.cws_home/708863/description

6.3.2.6. Restricciones. Las restricciones son por lo general formulas matemáticas, en ellas intervienen un número k de variables, $\{x_1, \dots, x_k\}$, se denotan por $C_{1..k}$. Una restricción binaria es una restricción en donde intervienen dos variables, x_i y x_j , sería entonces la restricción C_{ij} . En ocasiones no es necesario hacer referencia a las variables de la restricción, en este caso simplemente sería C . Por último, para hacer referencia específicamente a las variables que intervienen en una restricción $C_{i..k}$ se usa la notación $X_{C_{i..k}}$

La restricción $x \leq 1$ es una restricción unaria, es decir, solo posee una variable, en este caso x , la restricción $5x_1 + 23x_2 - 12x_3 \geq 0$ es una restricción ternaria.

Supóngase que tenemos una restricción $C_{i..k}$, una tupla p de $C_{i..k}$ es un elemento del producto cartesiano $D_i \times \dots \times D_k$, si p satisface a $C_{i..k}$, p es una tupla permitida o válida y se dice que es soporte para cada uno de sus valores. Al proceso de verificar si una tupla es válida se le llama comprobación de la consistencia.

Considérese la restricción $x_1 + x_2 \leq x_3 + x_4$ con dominios $\{1,2\}$, el conjunto de tuplas válidas sería $\{(1,1,1,1), (1,1,1,2), (1,1,2,1), (1,1,2,2), (2,1,2,2), (1,2,2,2), (1,2,1,2), (1,2,2,1), (2,1,1,2), (2,1,2,1), (2,2,2,2)\}$, y el conjunto de tuplas no válidas o no permitidas sería $\{(1,2,1,1), (2,1,1,1), (2,2,1,1), (2,2,1,2), (2,2,2,1)\}$; al nombrar todas las tuplas válidas y no válidas de una restricción se está haciendo una representación por extensión de ella.³⁵

6.3.2.7. Consistencia de un CSP. Para encontrar la solución de un CSP, se usan algoritmos de búsqueda basados en backtracking, sin embargo, suele presentarse lo que se conoce con el nombre de explosión combinatoria, es decir, un espacio de búsqueda muy grande, por lo tanto se considera que el backtracking por sí solo no es un método muy eficiente. Cuando en la búsqueda se encuentra un valor o un conjunto de valores que no satisfacen alguna propiedad de consistencia, es decir, valores inválidos, se presenta una inconsistencia local, éstas inconsistencias son relativas porque pueden ser inconsistentes con respecto a unas condiciones pero consistentes respecto a otras, en general, se usa el término consistencia global para referirnos a una solución en la que se eliminan los valores que no cumplen todas las condiciones.

Aparte de las restricciones explícitas de un CSP, existen restricciones implícitas que ayudan a acotar los espacios de búsqueda, si el algoritmo es capaz de encontrar éstas restricciones, ahorrará esfuerzo.

³⁴ Apt, Krzysztof. 2003; Principles of Constraint Programming. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780521825832>.

³⁵ Apt, Krzysztof. 2003; Principles of Constraint Programming. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780521825832>.

Ejemplo:

Variables: x, y, z.

Dominios: {0,1}, {2,3}, {1,2}, respectivamente.

Restricciones: $y < z$, $x \neq y$.

Si se hace la búsqueda por medio de backtracking, las variables se instanciarán en el orden x,y,z, y se probarán las 2^3 combinaciones posibles, todo esto para finalmente descubrir que no hay solución. Si se observa la primera restricción, no hay valores que la satisfagan, si el algoritmo tuviera la capacidad de identificar esta inconsistencia local, ahorraría esfuerzo de búsqueda.³⁶

6.3.2.8. Algoritmos de búsqueda. La base de los algoritmos de búsqueda para resolución de CSP's es el backtracking el cual se comporta similar al recorrido de un árbol con el objetivo de encontrar una o varias soluciones que se construyen por etapas hasta llegar a una solución completa; si en alguna etapa se detecta que no se puede llegar a una solución completa, se devuelve, elimina los elementos añadidos en cada etapa y continua el recorrido por los nodos no explorados.

Existen algoritmos completos, que garantizan encontrar una solución si esta existe, y algoritmos incompletos, que aunque son menos costosos y más eficientes, no garantizan encontrar una solución, éstos últimos incluyen algoritmos genéticos, búsqueda tabú, etc.

Algunos de los algoritmos de búsqueda completa que se han desarrollado para CSP's binarios son: backtracking cronológico, backjumping, conflict-directed backtracking, backtracking dinámico, forward checking, minimal forward checking, algoritmos híbridos como forward checking con conflict-directed backtracking y manteniendo arco-consistencia, algunos se han extendido a CSP's no binarios.³⁷

6.3.2.8.1. El árbol de búsqueda. Para asignar a las variables las múltiples combinaciones que genera el espacio de búsqueda, los algoritmos se apoyan en un árbol llamado árbol de búsqueda en donde se hace una *exploración primero en profundidad*. Suponiendo que el orden en el que se instancian las variables no cambia durante el proceso, un nodo en un nivel k del árbol, representa un estado

³⁶ Barber, F. Salido, M; Julio de 2003. Revista Iberoamericana de Inteligencia Artificial. 20 ed. Introduction to constraint programming. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

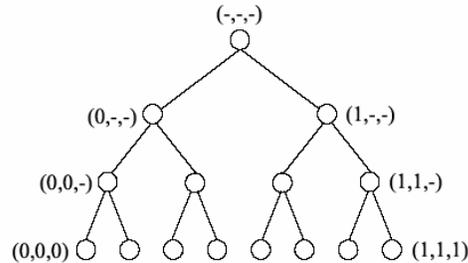
<<http://erevista2.aepia.org/index.php/ia/article/view/827/819>> [con acceso el 15-4-2009].

³⁷ Manyá, F. Gomes, C; 2003. Revista Iberoamericana de Inteligencia Artificial. 19 ed. Solution Techniques for Constraint Satisfaction Problems. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista.aepia.org/index.php/ia/article/view/362/348>>

en donde se han asignado valores a las variables x_1, \dots, x_k , por consiguiente las variables x_{k+1}, \dots, x_n aun no tienen valor. La raíz del árbol se toma como el nivel 0, allí no hay ningún valor asignado.

Figura 4. Árbol de búsqueda



Fuente: Revista Iberoamericana de Inteligencia Artificial N° 20³⁸

Cada nodo del árbol tiene una tupla asociada formada por las variables que hasta ese punto se han instanciado, en el nivel 1 se le ha asignado valor a x_1 , en el nivel 2 se le ha asignado valor a x_1 y x_2 , así sucesivamente hasta el nivel n , en donde los nodos tienen asociadas tuplas que contienen todas las n variables del problema instanciadas. Si una n -tupla es consistente se afirma que es solución del problema, si la asignación parcial en un nodo intermedio es consistente, se dice que el nodo es consistente. Los nodos cercanos a la raíz son nodos superficiales y los nodos cercanos a las hojas son nodos profundos.

En la figura 4 se muestra como se van asignando valores en cada nodo a medida que se recorre el árbol en profundidad, vemos que la raíz representa la tupla $(-, -, -)$ y que en cada nodo siguiente, se agrega un valor a la tupla que proviene de la variable que determinada el nivel. Al recorrer el árbol en profundidad, se llama variable actual a la que se instancia en el nodo en donde se encuentre, las variables pasadas son las que ya se instanciaron, es decir, las que corresponden a niveles menos profundos y variables futuras son las variables restantes que se encuentran en niveles mas profundos.

6.3.2.8.2. Backtracking cronológico (BT). Asumiendo que el orden de las variables y sus valores no cambia, este algoritmo trabaja de la siguiente manera. El algoritmo toma la siguiente variable, le asigna un valor dentro de su dominio y prueba las restricciones de la variable actual y las restricciones de las variables

³⁸ Barber, F. Salido, M; Julio de 2003. Revista Iberoamericana de Inteligencia Artificial. 20 ed. Introduction to constraint programming. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en: <http://erevista2.aepia.org/index.php/ia/article/view/827/819> [con acceso el 15-4-2009].

pasadas, si se han satisfecho las restricciones, toma la siguiente variable y repite el proceso; si por el contrario la asignación actual presenta una inconsistencia, se deshace la asignación y se toma otro valor, si no se logra encontrar un valor consistente entre todos los valores posibles, se llega a un estado sin salida (dead-end), en este caso el algoritmo se devuelve a la variable inmediatamente anterior y le asigna un nuevo valor. Cuando todas las variables tienen un valor, se dice que se ha llegado a una solución, o cuando se han probado todas las combinaciones sin éxito, se dice que no existe solución. Cabe anotar que solo pueden ser probadas las restricciones en donde no intervengan variables futuras, éstas tendrán que esperar a que el algoritmo llegue a esas variables.

Este algoritmo es muy simple, ineficiente y poco inteligente, por lo que ignora las restricciones con variables futuras y por ser incapaz de recordar procesos previos, puede repetir la misma acción varias veces³⁹. Véanse ahora los algoritmos look-back y look-ahead que ofrecen mejoras al BT.

6.3.2.8.3. Algoritmos Look-back. Realizan un análisis de la información del problema para mejorar su eficiencia en las situaciones sin salida, al igual que el BT, guardan la característica de comprobación hacia atrás. Algunos algoritmos de este tipo son los siguientes⁴⁰:

6.3.2.8.3.1. Backjumping (BJ). Este algoritmo tiene la capacidad de retroceder a cualquier variable pasada que esté causando conflicto con algún valor de la variable actual debido a una restricción que relacione ambas variables, de esta manera se haría un cambio en el valor de dicha variable pasada buscando habilitar valores para la variable actual⁴¹.

6.3.2.8.3.2. Conflict-directed backjumping (CBJ). Presenta una mejora al mecanismo usado en BJ; llamemos x_i a la variable actual y x_j a una variable pasada cualquiera, la clave del método consiste en guardar información de los conflictos que tiene una variable x_i con alguna variable x_j , cada vez que se detecta

³⁹ Manyá, F. Gomes, C; 2003. Revista Iberoamericana de Inteligencia Artificial. 19 ed. Solution Techniques for Constraint Satisfaction Problems. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista.aepia.org/index.php/ia/article/view/362/348>>

⁴⁰ Barber, F. Salido, M; Julio de 2003. Revista Iberoamericana de Inteligencia Artificial. 20 ed. Introduction to constraint programming. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista2.aepia.org/index.php/ia/article/view/827/819>> [con acceso el 15-4-2009].

⁴¹ Manyá, F. Gomes, C; 2003. Revista Iberoamericana de Inteligencia Artificial. 19 ed. Solution Techniques for Constraint Satisfaction Problems. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista.aepia.org/index.php/ia/article/view/362/348>>

una inconsistencia entre una variable x_i y una variable pasada x_j , x_j se añade al conjunto conflicto de x_i . Cuando se presenta una situación sin salida, el algoritmo busca en el conjunto, el conflicto de la variable actual y salta al elemento más cercano a ella. Este método requiere tener una estructura de datos más elaborada para guardar los datos de los conflictos⁴².

6.3.2.8.3.3. Learning. Cuando se alcanza una situación sin salida en x_i , se sabe que la combinación variable-valor representada por la tupla $((x_1, a_1), \dots, (x_{i-1}, a_{i-1}))$ causa una inconsistencia y por lo tanto no está permitida, este método almacena este hecho en forma de restricciones implícitas y las usa para podar el espacio de búsqueda. De esta manera el algoritmo aprende qué combinaciones no están permitidas para x_1, \dots, x_{i-1} .⁴³

6.3.2.8.4. Algoritmos Look-ahead. Como ya vimos, los algoritmos de look-back solo hacen comprobación de consistencia hacia atrás, ignorando las variables futuras. Los algoritmos look-ahead, comprueban hacia atrás y también hacia delante⁴⁴.

6.3.2.8.4.1. Forward Checking (FC). Este es el caso más común de look-ahead; FC prueba hacia adelante el valor de x_i y el valor de las variables futuras que están relacionadas con esta, si se encuentra un valor inconsistente en ellas, es eliminado temporalmente de su dominio, si en alguna etapa un dominio se queda vacío, se prueba con otro valor de la variable actual x_i , en el momento que se agoten los valores de la variable actual, se realizará backtracking cronológico.⁴⁵

Todos estos conceptos y algoritmos forman un conjunto de herramientas que facilitaron la implementación del sistema informático, brindando así la solución esperada para el problema inicial. Ahora para hablar de otros conceptos y/o terminología pertinente, se encontrará en el siguiente capítulo la información general referente a la metodología UML, mediante la cual se pudo modelar la aplicación a través de diagramas que muestran su funcionalidad.

⁴² Manyá, F. Gomes, C; 2003. Revista Iberoamericana de Inteligencia Artificial. 19 ed. Solution Techniques for Constraint Satisfaction Problems. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista.aepia.org/index.php/ia/article/view/362/348>>

⁴³ Barber, F. Salido, M; Julio de 2003. Revista Iberoamericana de Inteligencia Artificial. 20 ed. Introduction to constraint programming. Malaga, España. Lawrence Mandow. [Revista en línea] Disponible desde Internet en:

<<http://erevista2.aepia.org/index.php/ia/article/view/827/819>> [con acceso el 15-4-2009].

⁴⁴ Apt, Krzysztof. 2003; Principles of Constraint Programming. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780521825832>.

⁴⁵ Ibid.

6.4. INTRODUCCIÓN A UML

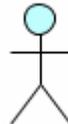
El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

6.4.1. Diagrama de casos de uso. El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

Actor. Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

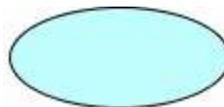
Figura 5. Representación de Actor



Fuente: los autores.

Caso de Uso. Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Figura 6. Representación Caso de Uso



Fuente: los autores.

Relaciones. Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML, el cual describe notación gráfica para esas relaciones.

- **Inclusión (include o use).** Es una forma de interacción, un caso de uso dado puede "incluir" otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido.

Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual, desde el caso de uso que lo incluye hasta el caso de uso incluido, con la etiqueta "«include»".

Figura 7. Inclusión



Fuente: los autores.

- **Extensión (extend).** Es otra forma de interacción, un caso de uso dado, (la extensión) puede *extender* a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación, es una flecha de punta abierta con línea discontinua, desde el caso de uso extensión al caso de uso extendido, con la etiqueta «extend».

Figura 8. Extensión



Fuente: los autores.

- **Generalización:** En la tercera forma de relaciones entre casos de uso, existe una relación generalización/especialización. Un caso de uso dado puede estar en una forma especializada de un caso de uso existente. La notación es una línea sólida terminada en un triángulo dibujado desde el caso de uso especializado al caso de uso general.

Figura 9. Generalización



Fuente: los autores.

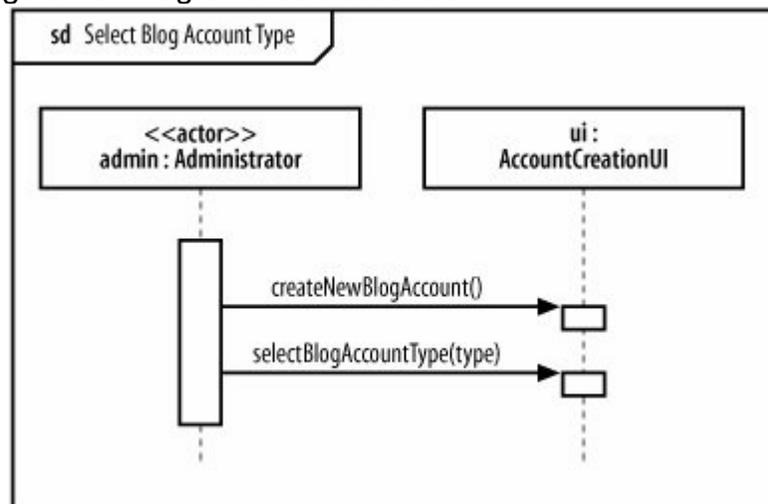
6.4.2. Diagramas de Interacción. Los diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento.

Habitualmente, un diagrama de interacción capta el comportamiento de un solo caso de uso. El diagrama muestra cierto número de objetos y los mensajes que se pasan entre estos objetos dentro del caso de uso.

Hay dos tipos de diagramas de interacción: diagramas de secuencia y diagramas de colaboración.

6.4.2.1. Diagramas de Secuencia. Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

Figura 10. Diagrama de secuencia

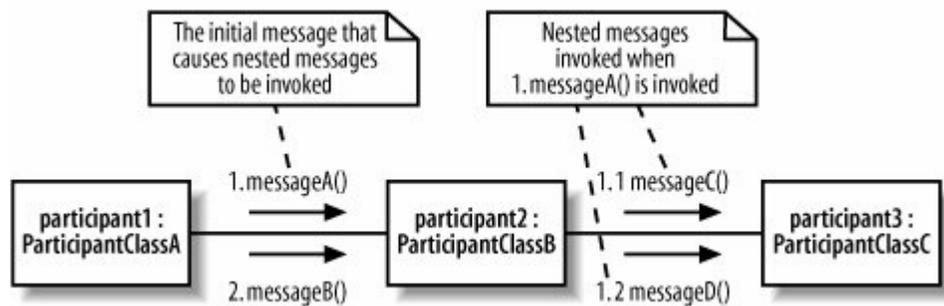


Fuente: Libro Learning UML 2.0⁴⁶

⁴⁶ Hamilton Kim, Miles Rusell. Abril 2006. Learning UML 2.0. Ed. O'Reilly. Sebastopol. [Libro en línea] Disponible desde Internet en: <<http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>>

6.4.2.2. Diagrama de Colaboración. Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

Figura 11. Ejemplo Diagrama de Colaboración



Fuente: Libro Learning UML 2.0⁴⁷

6.4.3. Diagrama de Actividad. Un diagrama de Actividad demuestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en él.

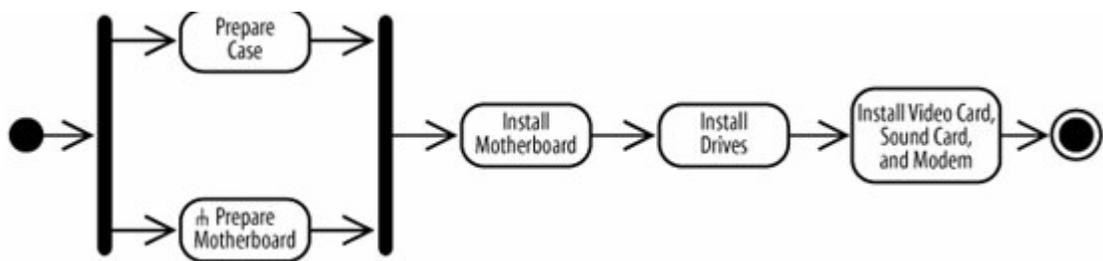
Composición

- **Inicio.** El inicio de un diagrama de actividad es representado por un círculo de color negro sólido.
- **Actividad.** Una actividad representa la acción que será realizada por el sistema la cual es representada dentro de un ovalo.
- **Transición.** Una transición ocurre cuando se lleva acabo el cambio de una actividad a otra, la transición es representada simplemente por una línea con una flecha en su terminación para indicar dirección.

⁴⁷ Ibid.

- **Ramificación (*Branch*).** Una ramificación ocurre cuando existe la posibilidad que ocurra más de una transición (resultado) al terminar determinada actividad. Este elemento es representado a través de un rombo.
- **Unión (*Merge*).** Una unión ocurre al fusionar dos o más transiciones en una sola transición o actividad. Este elemento también es representado a través de un rombo.
- **Fin.** El fin de un diagrama de actividad es representado por un círculo, con otro círculo concéntrico de color negro sólido.
- **Canales (*Swimlanes*).** En determinadas ocasiones ocurre que un diagrama de actividad se expanda a lo largo de más de un entidad o actor, cuando esto ocurre el diagrama de actividad es particionada en canales (*swimlanes*), donde cada canal representa la entidad o actor que esta llevando acabo la actividad.

Figura 12. Ejemplo Diagrama de Actividad



Fuente: Libro Learning UML 2.0⁴⁸

6.4.4. Diagrama de Estados. Un Diagrama de Estados muestra la secuencia de estados por los que pasa bien un caso de uso, bien un objeto a lo largo de su vida, o bien todo el sistema. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

Representación:

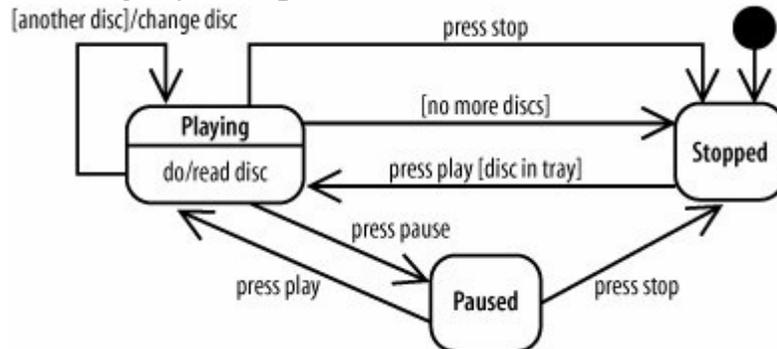
- **Estado.** Se representa como una caja redondeada con el nombre del estado en su interior. La caja de un estado puede tener 1 o 2 compartimentos. En el

⁴⁸ Ibid.

primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

- **Transición.** Se representa como una flecha desde el estado origen al estado destino.
- **Acción de entrada.** Aparece en la forma entrada/acción_asociada donde acción_asociada es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta.
- **Acción de salida.** Aparece en la forma salida/acción_asociada. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.
- **Acción interna.** Es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma nombre_de_evento/acción_asociada.

Figura 13. Ejemplo Diagrama de Estados



Fuente: Libro Learning UML 2.0⁴⁹

Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (finalización del caso de uso o destrucción del objeto). El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad, los estados inicial y final son pseudoestados, pues un objeto no puede “estar” en esos estados, pero nos sirven para saber cuáles son las transiciones iniciales y finales.

⁴⁹ Ibid.

6.4.5. Diccionario de Datos. El diccionario de datos es un listado organizado de todos elementos que forman parte del flujo de datos de todo el sistema en el que se guardan los detalles y descripción de todos estos elementos. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema; su contenido también se emplea durante el diseño.

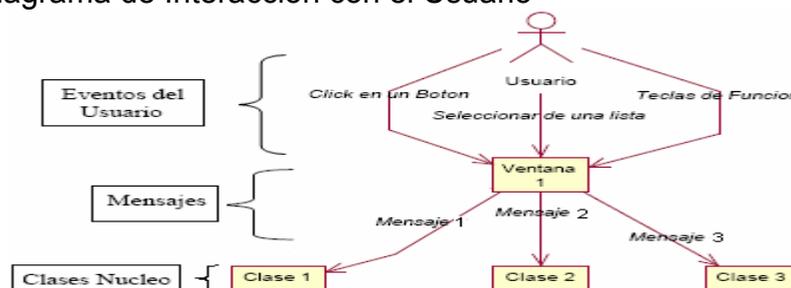
6.4.6. Requerimientos del usuario. Deben describir los requerimientos funcionales y no funcionales, que sean comprensibles para los usuarios sin conocimiento técnico detallado.

6.4.6.1. Requerimientos funcionales. Describen las funciones que el sistema va a hacer. Estos requerimientos dependen del tipo de software que se desarrolla, de los posibles usuarios del software y del enfoque en la organización al redactar los requerimientos; los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas y salidas, excepciones, etc.

6.4.6.2 Requerimientos no funcionales. Son aquellos requerimientos que no se refieren directamente a las propiedades emergentes, como son la fiabilidad, el tiempo de respuesta del sistema y la capacidad de almacenamiento: de forma alternativa define las restricciones del sistema de los dispositivos de entrada y salida.

6.4.7. Diagrama de interacción con el usuario. Su objetivo es modelar los eventos (parte cambiante con el tiempo) que el usuario puede generarle a la ventana o módulo y los mensajes que éste módulo le envía a las clases núcleo.

Figura 14. Diagrama de Interacción con el Usuario



Fuente: Libro Learning UML 2.0⁵⁰.

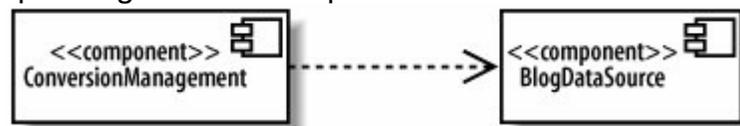
⁵⁰ Ibid.

6.4.8. Diagrama de Componentes. Los diagramas de componentes describen los elementos físicos del Sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable.

Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

Los elementos que lo componen son: componentes, interfaces y relaciones de dependencia, generalización, asociación, realización.

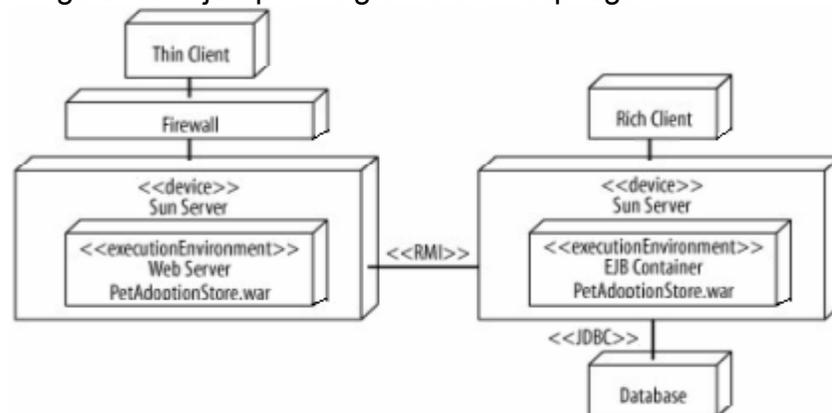
Figura 15. Ejemplo Diagrama de Componentes



Fuente: Libro Learning UML 2.0⁵¹

6.4.9. Diagrama de Despliegue. Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de ejecución.

Figura 16. Ejemplo Diagrama de Despliegue



Fuente: Libro Learning UML 2.0⁵²

⁵¹ Hamilton Kim, Miles Rusell. Abril 2006. Learning UML 2.0. Ed. O'Reilly. Sebastopol. [Libro en línea] Disponible desde Internet en: <<http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>>

6.4.10. Diagrama de Clases. Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases esta compuesto por los siguientes elementos:

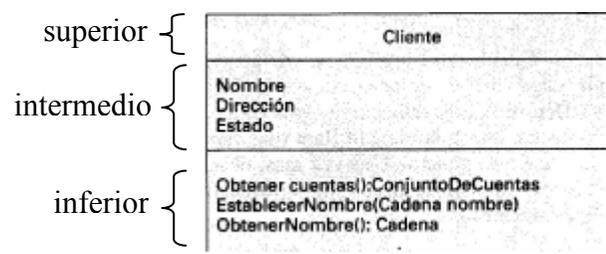
- **Clase.** Atributos, métodos y visibilidad.
- **Relaciones.** Herencia, Composición, Agregación, Asociación y Uso.

Elementos

- **Clase.** Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones:

Figura 17. Representación Clase



Fuente: Libro Ingeniería del Software II.⁵³

En donde:

- ✓ **Superior.** Contiene el nombre de la Clase.
- ✓ **Intermedio.** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

⁵² Hamilton Kim, Miles Rusell. Abril 2006. Learning UML 2.0. Ed. O'Reilly. Sebastopol. [Libro en línea] Disponible desde Internet en: <<http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>>

⁵³ Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill.

- ✓ **Inferior.** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).
- **Atributos.** Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:
 - ✓ **Public.** Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
 - ✓ **Private.** Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).
 - ✓ **Protected.** Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven (ver herencia).
- **Métodos.** Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:
 - ✓ **Public.** Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
 - ✓ **Private.** Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
 - ✓ **Protected.** Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (ver herencia).

Relaciones entre Clases. Ahora ya definido el concepto de Clase, es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- uno o muchos: 1..* (1..n)
- 0 o muchos: 0..* (0..n)
- número fijo: m (m denota el número).

- ✓ **Herencia (Especialización/Generalización).** Indica que una subclase hereda los métodos y atributos especificados por una Súper Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Súper Clase (public y protected).

Figura 18. Herencia



Fuente: los autores.

- ✓ **Agregación.** Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:
 - **Por Valor.** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada *Composición*.
 - **Por Referencia:** Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada *Agregación*.

Figura 19. Agregación



Fuente: los autores.

- ✓ **Asociación.** La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

Figura 20. Asociación



Fuente: los autores.

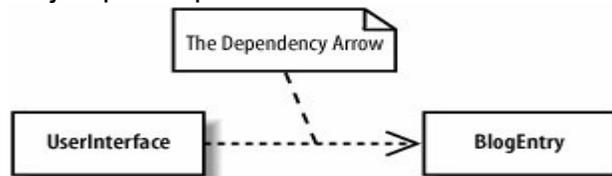
- ✓ **Dependencia o Instanciación (uso).** Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase).

Figura 21. Dependencia o Instanciación



Fuente: los autores.

Figura 22. Ejemplo Dependencia o Instanciación



Fuente: Libro Learning UML 2.0.⁵⁴

Cabe destacar que el objeto creado (en este caso la Ventana gráfica) no se almacena dentro del objeto que lo crea (en este caso la Aplicación).

6.4.11. Modelo Entidad Relación

Este modelo se obtiene en tiempo de diseño de la base de datos. Fue propuesto por el Doctor en Ciencias de la Computación y Matemáticas Aplicadas Peter Chen en 1976 y desde entonces se viene utilizando de una forma muy global.

Se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones. Con este modelo conseguimos representar de manera grafica la estructura lógica de una base de datos.

Los principales elementos del modelo entidad-relación son las entidades con sus atributos y las relaciones entre entidades.

⁵⁴ Hamilton Kim, Miles Rusell. Abril 2006. Learning UML 2.0. Ed. O'Reilly. Sebastopol. [Libro en línea] Dsponible desde Internet en: <<http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>>

Elementos

- **Entidad.** Se trata de un objeto del que se recoge información de interés de cara a la base de datos. Gráficamente se representan mediante un rectángulo. Un ejemplo sería la entidad banco, donde se recogerían los datos relativos a ese banco, como puede ser el nombre, el número de sucursal, la dirección, etc.

Dentro de las entidades pueden ser fuertes o débiles. Las fuertes son las que no dependen de otras entidades para existir, mientras que las entidades débiles siempre dependen de otra entidad sino no tienen sentido por ellas mismas.

- **Relación.** Podemos definir la relación como una asociación de dos o más entidades. A cada relación se le asigna un nombre para poder distinguirla de las demás y saber su función dentro del modelo entidad-relación. Otra característica es el grado de relación, siendo las de grado 1 relaciones que solo relacionan una entidad consigo misma. Las de grado 2 son relaciones que asocian dos entidades distintas, y las de grado n que se tratan de relaciones que unen más de dos entidades. Las relaciones se representan gráficamente con rombos, dentro de ellas se coloca el nombre de la relación.

Correspondencia entre dos relaciones

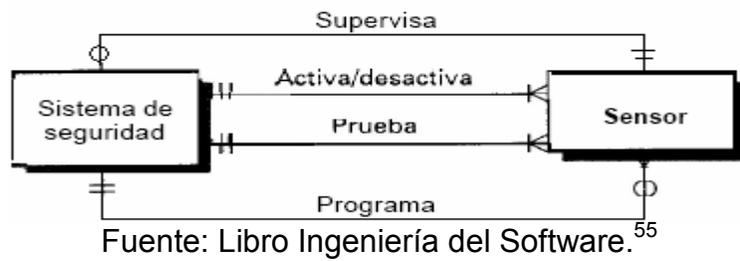
- ✓ **1:1.** Uno a uno, a cada ocurrencia de una entidad le corresponde como máximo una ocurrencia de la otra entidad relacionada.
- ✓ **1:N.** Uno a Mucho, a cada ocurrencia de la entidad A le pueden corresponder varias de la entidad B.
- ✓ **N:M.** Muchos a muchos, cada ocurrencia de una entidad puede contener varias de la otra entidad relacionada y viceversa.

Para finalizar las características de la relación tenemos la cardinalidad que define el número máximo y mínimo de ocurrencias de cada tipo de entidad.

Se representa con los valores máximo como mínimo encerrados entre paréntesis encima de la relación. (máximo, mínimo)

- **Atributo.** Se define como cada una de las propiedades de una entidad o relación. Cada atributo tiene un nombre y todos los posibles valores que puede tener. Dentro de una entidad tiene que haber un atributo principal que identifica a la entidad y su valor tiene que ser único. Un ejemplo de atributo principal sería el DNI dentro de la entidad persona.

Figura 23. Ejemplo Modelo Entidad-Relación



Previa a la realización de los diagramas descritos anteriormente, es necesario hacer un debido análisis del sistema y de los requisitos o requerimientos del usuario. A continuación se podrá ver detalladamente lo que se hizo en este proceso.

⁵⁵ Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill.

7. ANÁLISIS DEL SISTEMA

En este capítulo están plasmadas las necesidades que fueron identificadas durante cada etapa de recolección de información como la entrevista, los requerimientos del sistema y con los cuales se pudo modelar el dominio de acuerdo al problema planteado inicialmente.

7.1. TRABAJO DE CAMPO

7.1.1. Entrevista. La siguiente es la entrevista realizada a la Ingeniera Ana María López Coordinadora del Programa de Ingeniería de Sistemas y Computación en la que se plantearon los inconvenientes que se generan durante el proceso de distribución de la carga académica del cual ella se encarga manualmente.

- Mencione los aspectos más importantes y/o limitantes que debe considerar a la hora de generar los horarios.
 - ✓ Evitar cruce de horarios.
 - ✓ Registrar la información de los docentes referente a su disponibilidad horaria y a los perfiles de asignaturas que poseen de acuerdo a sus conocimientos y/o capacidades.
 - ✓ De acuerdo al tipo de contratación de los profesores, tener cuenta el número máximo de horas de clase que se les puede asignar, incluyendo las de sobrecarga.
 - ✓ Realizar la respectiva asignación de los grupos dependiendo de la disponibilidad que tengan los docentes de dictar una asignatura, ya sea en jornada diurna, especial o ambas.
 - ✓ Tener en cuenta que la asignación debe poder realizarse en paralelo a todos los docentes en cuatro fases diferentes: Planta, Transitorio Tiempo Completo, Transitorio Medio Tiempo y Catedrático.
- De todo el proceso ¿cuál es la parte que le genera mayor dificultad o esfuerzo?

Completar la respectiva carga a cada profesor dependiendo del tipo de contratación que tengan con la Universidad.

- ¿Considera que por la complejidad del proceso, sería conveniente implementarlo en un sistema computacional?

Si, porque las herramientas informáticas son de gran utilidad en la gestión de procesos de casi cualquier índole.

- A grandes rasgos, ¿cómo se imagina el software? (Puede recurrir a la especificación de la tarea que considere como principal).

Una aplicación web con la que pueda simular la carga académica de los profesores, además que permita la gestión de toda la información involucrada en este proceso y que genere informes de la misma de una manera más rápida y eficiente.

7.1.2. Requerimientos del sistema

La siguiente tabla muestra la especificación de los diferentes requerimientos o necesidades que se plantearon y definieron luego de la recolección de la información proporcionada por el usuario y discutida con los desarrolladores de la aplicación.

Tabla 2. Requerimientos del sistema

Referencia	Descripción	Categoría
R1	Captura el login y la contraseña del usuario para que sea validado en el sistema.	EVIDENTE
R2	Captura información sobre el nuevo profesor.	EVIDENTE
R3	Captura información sobre nuevo materia.	EVIDENTE
R4	Captura información sobre nueva usuario.	EVIDENTE
R5	Captura información sobre nuevo grupo.	EVIDENTE
R6	Valida formulario para Adición de nuevo profesor.	EVIDENTE
R7	Valida formulario para Adición de materia a profesor.	EVIDENTE
R8	Valida formulario para Adición de nueva materia.	EVIDENTE
R9	Valida formulario para Adición de nuevo usuario.	EVIDENTE
R10	Valida formulario para Adición de grupo de una materia.	EVIDENTE
R11	Valida formulario para Modificación de profesores existentes.	EVIDENTE
R12	Valida formulario para Modificación de materias registradas.	EVIDENTE
R13	Valida formulario para Modificación de usuarios existentes.	EVIDENTE
R14	Valida formulario para Modificación de grupos existentes.	EVIDENTE
R15	Permite adicionar un profesor al sistema.	EVIDENTE
R16	Permite adicionar una materia.	EVIDENTE
R17	Permite adicionar un usuario.	EVIDENTE
R18	Permite adicionar un grupo de una materia.	EVIDENTE
R19	Permite adicionar una materia a profesor.	EVIDENTE
R20	Permite modificar un profesor.	EVIDENTE
R21	Permite modificar una materia.	EVIDENTE
R22	Permite modificar un usuario.	EVIDENTE
R23	Permite modificar un grupo.	EVIDENTE
R24	Permite consultar información de profesores.	EVIDENTE
R25	Permite consultar información de materias.	EVIDENTE
R26	Permite consultar información de usuarios.	EVIDENTE
R27	Permite consultar información de grupos.	EVIDENTE
R28	Permite eliminar profesor.	EVIDENTE
R29	Permite eliminar materia a profesor.	EVIDENTE
R30	Permite eliminar materia.	EVIDENTE
R31	Permite eliminar usuario.	EVIDENTE
R32	Permite eliminar grupo.	EVIDENTE
R33	Permite asignar la carga	EVIDENTE
R34	Permite reiniciar la carga	EVIDENTE
R35	Permite asignar grupo	EVIDENTE
R36	Permite liberar grupo	EVIDENTE
R37	Permite establecer horas disponibles de profesor.	EVIDENTE
R38	Permite restablecer contraseña	EVIDENTE
R39	Gestiona la información en la base de datos.	OCULTO

Fuente: los autores.

7.1.3. Modelado del Dominio. De acuerdo al planteamiento del problema y a los requerimientos del sistema la aplicación se dividió en 4 secciones: Administrar Profesores, Administrar Materias, Administradores (si el usuario es Administrador se llama Administradores y si el usuario es Profesor, se llama Administrar Usuarios) y Administrar Carga. Además se hizo necesario implementar 8 clases: Profesor, Materia, Usuario, Perfil, Carga, Controlador, Interfaz y db.

Cuando el usuario accede a la sección Administrar Profesores se le presenta un submenú con las siguientes opciones: Adicionar, Modificar, Consultar, Eliminar, Establecer Horario, Adicionar Materia y Eliminar Materia.

- La opción Adicionar le permite al usuario Administrador ingresar información de un nuevo profesor a través de un formulario entregado por el sistema; para esto se hace uso de las clases Profesor, Usuario, Controlador, Interfaz y db.
- La opción Modificar permite realizar cambios en determinada información de un profesor. Si el usuario es Administrador puede escoger de una lista el profesor al cual le desea modificar los datos, pero si el usuario es Profesor solo puede modificar su(s) apellido(s) o su(s) nombre(s); para esto se hace uso de las clases Profesor, Controlador, Interfaz y db.
- La opción Consultar permite obtener información general de todos los profesores registrados en el sistema y de las materias que tengan asociadas dependiendo de los conocimientos y capacidades que tengan para dictarlas; para esto se hace uso de las clases Profesor, Controlador, Interfaz y db.
- La opción Eliminar le permite al usuario Administrador elegir de una lista de profesores registrados al que desea eliminar de la base de datos; para esto se hace uso de las clases Profesor, Usuario, Controlador, Interfaz y db.
- La opción Establecer Horario permite asignar horas disponibles en las que se le pueden asignar horas de clase a un profesor. Si el usuario es Administrador puede escoger de una lista el profesor al cual le desea asignar horas disponibles, pero si el usuario es Profesor solo puede establecer sus propias horas; para esto se hace uso de las clases Profesor, Materia, Controlador, Interfaz y db.
- La opción Adicionar Materia le permite al usuario Administrador adicionarle una materia a los profesores dependiendo de las competencias que éstos tengan; para esto se hace uso de las clases Perfil, Controlador, Interfaz y db.
- La opción Eliminar Materia le permite al usuario Administrador elegir de una lista de profesores al que se le desea eliminar una asignatura, seleccionada

previamente; para esto se hace uso de las clases Perfil, Controlador, Interfaz y db.

Cuando el usuario accede a la sección Administrar Materias se le presenta un submenú con las siguientes opciones: Adicionar, Modificar, Consultar y Eliminar.

- La opción Adicionar le permite al usuario Administrador ingresar información de una asignatura del programa a través de un formulario entregado por el sistema; para esto se hace uso de las clases Materia, Controlador, Interfaz y db.
- La opción Modificar le permite al usuario Administrador realizar cambios en determinada información de una materia; para esto se hace uso de las clases Materia, Controlador, Interfaz y db.
- La opción Consultar le permite a los usuarios obtener información general de todas las materias registradas en el sistema y de las materias por semestre; para esto se hace uso de las clases Materia, Controlador, Interfaz y db.
- La opción Eliminar le permite al usuario Administrador elegir de una lista de las materias registradas la que desea eliminar de la base de datos; para esto se hace uso de las clases Materia, Controlador, Interfaz y db.

Cuando el usuario accede a la sección Administradores (Administrar Usuarios) se le presenta un submenú con las siguientes opciones: Adicionar, Modificar, Consultar, Eliminar, Restablecer Clave.

- La opción Adicionar le permite al usuario Administrador ingresar información de un nuevo usuario a través de un formulario entregado por el sistema; para esto se hace uso de las clases Usuario, Controlador, Interfaz y db.
- La opción Modificar permite a los usuarios realizar cambios en determinada información. Si el usuario es Administrador puede escoger de una lista el profesor al cual le desea modificar los datos, pero si el usuario es Profesor solo puede modificar su propia información; para esto se hace uso de las clases Usuario, Controlador, Interfaz y db.
- La opción Consultar permite obtener información general de todos los usuarios registrados en el sistema, ya sean administradores o profesores; para esto se hace uso de las clases Usuario, Controlador, Interfaz y db.

- La opción Eliminar le permite al usuario Administrador elegir de una lista de usuarios registrados al que desea eliminar de la base de datos; para esto se hace uso de las clases Usuario, Controlador, Interfaz y db.
- La opción Restablecer Clave le permite al usuario Administrador reasignar la clave que se le dio por defecto a un nuevo usuario cuando creó la cuenta, esto debido a un olvido de la clave de acceso al sistema; para esto se hace uso de las clases Usuario, Controlador, Interfaz y db.

Cuando el usuario accede a la sección Administrar Carga se le presenta un submenú con las siguientes opciones: Adicionar Grupo, Consultar, Eliminar Grupo, Asignar Carga, Reiniciar Carga, Asignar Grupo y Liberar Grupo.

- La opción Adicionar Grupo le permite al usuario Administrador crear un grupo de una materia registrada en la base de datos, a través de un formulario entregado por el sistema; para esto se hace uso de las clases Carga, Controlador, Interfaz y db.
- La opción Consultar le permite a los usuarios obtener información de todos los grupos existentes de las asignaturas registradas en la base de datos y de los grupos que tienen asignados los profesores; para esto se hace uso de las clases Carga, Controlador, Interfaz y db.
- La opción Eliminar Grupo le permite al usuario Administrador borrar un grupo de una materia registrada en la base de datos; para esto se hace uso de las clases Profesor, Carga, Controlador, Interfaz y db.
- La opción Asignar Carga le permite al usuario Administrador realizar de manera automática la respectiva asignación de los grupos a los profesores, escogiendo la jornada (diurna o especial), el tipo de asignación (Planta, Transitorio Tiempo Completo, Transitorio Medio Tiempo y/o Catedrático) y por último estableciendo el número máximo de horas de clase que se le pueden fijar a un docente; para esto se hace uso de las clases Carga, Controlador, Interfaz y db.
- La opción Reiniciar Carga le permite al usuario Administrador eliminar la asignación de profesores a los grupos creados de las asignaturas; para esto se hace uso de las clases Carga, Controlador, Interfaz y db.
- La opción Asignar Grupo le permite al usuario Administrador realizar una elección manual del profesor para dictar un grupo de una determinada asignatura, sin tener consideraciones de perfil o de disponibilidad del docente, solo garantizando que no haya cruces con otros grupos que ya tenga

asignados; para esto se hace uso de las clases Profesor, Controlador, Interfaz y db.

- La opción Liberar Grupo le permite al usuario Administrador eliminar de manera manual el docente que tiene asignado determinado grupo; para esto se hace uso de las clases Profesor, Controlador, Interfaz y db.

Luego de especificar la manera como se manejan las secciones en que se dividió el menú de la aplicación y las clases que están involucradas en la implementación de cada una, se podrá ver en el siguiente capítulo de qué forma se abordó el problema a solucionar dándole así una solución óptima, llegando a una funcionalidad como la descrita en este modelado del dominio del sistema.

7.2 MODELO MATEMÁTICO

En este capítulo se describe claramente la manera cómo se va a abordar el problema planteado para poder darle una solución óptima que satisfaga las necesidades del usuario final.

Para tal fin y debido a que el método de solución escogido para resolver el problema de asignación de horarios o distribución de carga académica es programación con restricciones, el modelo matemático debe estar dado en términos de la terna (X, D, C):

X: Variables del sistema.

D: Dominios de las variables.

C: Restricciones definidas sobre las variables del sistema.

Los dominios de las variables son similares a conjuntos, donde el nombre del conjunto es la variable y sus elementos representan los posibles valores que puede tomar, por esta razón tendremos operaciones de dos tipos, unas en donde las variables se toman como los dominios que representan, es decir operaciones sobre conjuntos, y otras en donde las variables se toman como tal.

La universidad tiene establecidos unos horarios en los cuales pueden dictarse las clases dependiendo si es jornada diurna o especial.

Jornada diurna: lunes a viernes 6 AM – 10 PM.

Jornada especial: lunes a viernes 6 PM - 10 PM y sábados 7 AM - 8 PM.

El espacio de tiempo se muestra en la Tabla 3.

Cada unidad de tiempo es numerada, de tal manera que la primera hora del lunes corresponda a la hora 1 y la última hora del mismo día corresponda a la 16, para continuar la secuencia, tomamos la primera hora del martes como la 17, así sucesivamente hasta llegar a la hora 80 para jornada diurna y 94 para jornada especial, como se muestra en la figura 24.

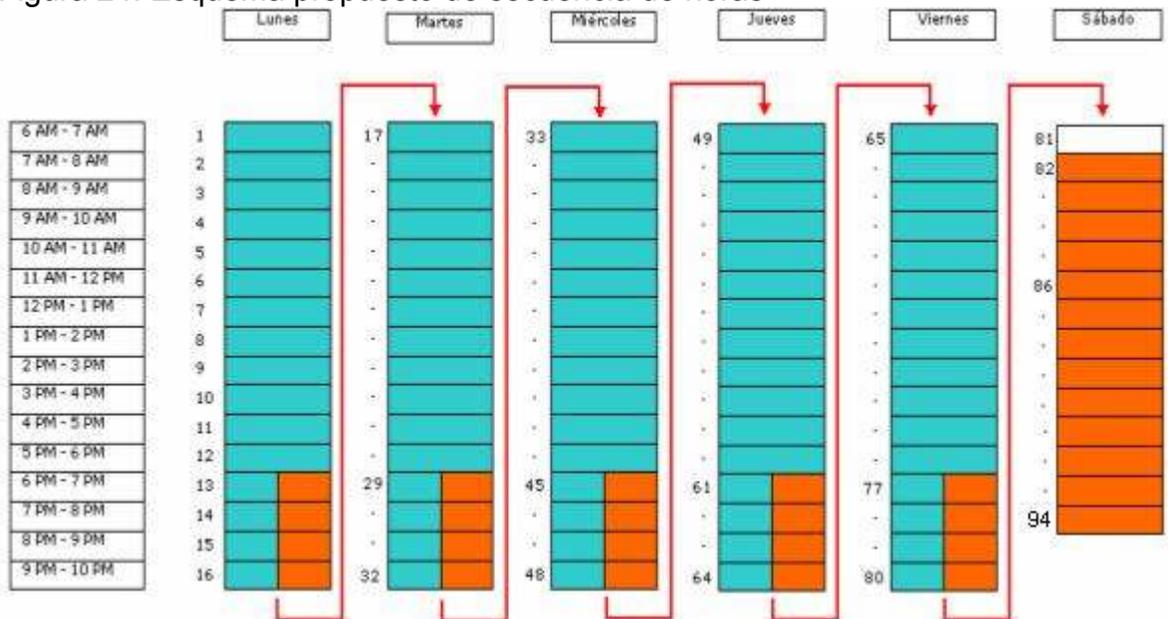
Tabla 3. Espacio de tiempo de clases establecido por la universidad

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
6 am. - 7 am.						
7 am. - 8 am.						
8 am. - 9 am.						
9 am. - 10 am.						
10 am. - 11 am.						
11 am. - 12 pm.						
12 pm. - 1 pm.						
1 pm. - 2 pm.						
2 pm. - 3 pm.						
3 pm. - 4 pm.						
4 pm. - 5 pm.						
5 pm. - 6 pm.						
6 pm. - 7 pm.						
7 pm. - 8 pm.						
8 pm. - 9 pm.						
9 pm. - 10 pm.						



Fuente: los autores.

Figura 24. Esquema propuesto de secuencia de horas



Fuente: los autores.

El modelo de solución aborda el problema considerando que la carga se encuentra fija, lo que representa un dominio acotado para cada grupo.

El sistema puede encontrarse en múltiples estados de asignación dependiendo de el comportamiento de la fase anterior, en ese sentido puede definirse un estado S_0 en donde todos los profesores tienen asignación nula o cero, y un estado S_r , cualquiera posterior a S_0 .

Procedimiento

Sea el conjunto Categorías definido por los tipos de contrato establecidos por la universidad, este conjunto permite priorizar la asignación según el tipo de contrato.

$$Categorías = \{planta, transitorio_tc, transitorio_mt, catedratico\} \quad 1 \leq i \leq |Categorías|$$

Sea i una variable de iteración, esta variará de 1 hasta el número de elementos del conjunto Categorías y c_i un elemento de dicho conjunto.

Sea el conjunto Grupos definido por todos los grupos registrados que pertenecen a una jornada j y que no han sido asignados:

$$Grupos = \{g : g \text{ es grupo no asignado con jornada } j\} \quad 1 \leq k \leq |Grupos|$$

Sea k una variable de iteración, esta variará de 1 hasta el número de elementos del conjunto Grupos y g_k un elemento de dicho conjunto.

Sea el conjunto Profesores definido por los docentes de categoría c_i que están en capacidad de dictar g_k y están habilitados en jornada j :

$$Profesores = \left\{ p : p \text{ es docente de categoría } c_i \wedge p \text{ incluye } g_k \text{ en su perfil} \wedge \right. \\ \left. p \text{ esta habilitado en jornada } j \right\}$$

$$1 \leq q \leq |Profesores|$$

Sea q una variable de iteración, esta variará de 1 hasta el número de elementos del conjunto Profesores y p_q un elemento de dicho conjunto.

Se desea asignar un profesor p_q a un grupo g_k , considerando que p_q tenga asociado en su perfil la materia representada por $g_{materia,k}$ y tanto p_q como g_k corresponden a una jornada j .

7.2.1. Restricciones

Parte de las restricciones se han definido en la especificación de los conjuntos mencionados, defínanse ahora las restricciones que delimiten la asignación con respecto a los topes de cada profesor, intensidad horaria, sobrecarga y jornada.

7.2.1.1. Profesores catedráticos

Los profesores catedráticos tienen un trato menos restrictivo en materia de topes. Entiéndase por *tope_contrato* el número máximo de horas que se le pueden asignar, en este caso la palabra contrato no hace referencia a su tipo de vinculación.

$$c_i = \text{catedratico}$$
$$\mathcal{G}_{\text{intensidad_horaria},k} + P_{\text{carga_diurna},q} + P_{\text{carga_especial},q} \leq P_{\text{tope_contrato},q}$$
$$\mathcal{G}_{\text{horario},k} \subseteq P_{\text{horario},q}$$

7.2.1.2. Profesores de contrato

Los profesores de contrato manejan dos topes, el tope de contrato y el tope de sobrecarga.

$$c_i = \text{planta} \vee c_i = \text{transitorio_tc} \vee c_i = \text{transitorio_mt}$$

7.2.1.2.1 Horas reglamentarias

Jornada diurna

$$\mathcal{G}_{\text{intensidad_horaria},k} + P_{\text{carga_diurna},q} \leq P_{\text{tope_contrato},q}$$
$$\mathcal{G}_{\text{horario},k} \subseteq P_{\text{horario},q}$$

7.2.1.2.2. Sobrecarga

Jornada diurna

$$(P_{\text{carga_diurna},q} \geq P_{\text{tope_contrato},q},$$
$$\mathcal{G}_{\text{intensidad_horaria},k} + P_{\text{carga_diurna},q} + P_{\text{carga_especial},q} \leq P_{\text{tope_contrato},q} + P_{\text{tope_sobrecarga},q},$$

$$\begin{aligned}
& \mathcal{G}_{\text{horario},k} \subseteq \mathcal{P}_{\text{horario},q} \\
& \quad \vee \\
& (P_{\text{carga_diurna},q} < P_{\text{tope_contrato},q} , \\
& \mathcal{G}_{\text{intensidad_horaria},k} + P_{\text{carga_diurna},q} \leq P_{\text{tope_contrato},q} , \\
& \mathcal{G}_{\text{horario},k} \subseteq \mathcal{P}_{\text{horario},q})
\end{aligned}$$

Jornada especial

$$\begin{aligned}
& P_{\text{carga_diurna},q} \geq P_{\text{tope_contrato},q} \\
& \mathcal{G}_{\text{intensidad_horaria},k} + P_{\text{carga_diurna},q} + P_{\text{carga_especial},q} \leq P_{\text{tope_contrato},q} + P_{\text{tope_sobrecarga},q} \\
& \mathcal{G}_{\text{horario},k} \subseteq \mathcal{P}_{\text{horario},q}
\end{aligned}$$

Después de plantear las restricciones se definió una función final que se encarga de hacer todo el proceso de iteración.

$$\text{asignacion}_{ikq}(c_i, \mathcal{G}_k, p_q)$$

En cada iteración, la función tratará de encontrar una asignación para la tupla $(c_i, \mathcal{G}_k, p_q)$ teniendo en cuenta las restricciones definidas; es necesario actualizar los dominios en determinados puntos de la iteración, de esta manera se excluyen de ellos los valores encontrados en cada asignación.

7.2 METODOLOGÍA DE DESARROLLO

En esta sección se define la metodología de desarrollo de software que se utilizó en este proyecto y se sustentan las razones por las cuales fue escogida.

Teniendo en cuenta las características y/o la naturaleza del proyecto, se consideró que la metodología más adecuada es la XP (Extreme Programming, en español Programación Extrema), la cual es una de las más utilizadas para proyectos de corto plazo y corto equipo.

Aunque esta metodología consiste en la programación rápida o extrema, teniendo como parte del equipo al usuario final, se tomaron como base las respuestas obtenidas de la entrevista realizada a la Coordinadora del Programa antes de comenzar el proyecto, pudiendo recolectar y analizar los requerimientos en cuanto a la necesidad de implementar el software con el que se desea dar solución a los inconvenientes generados en el proceso de asignación de la carga académica. Con esta información se pudo desarrollar todo el proyecto adicionándole mejoras y otras funcionalidades.

También se tuvieron muy presentes algunas de las características principales de esta metodología, las cuales permiten adelantarse al futuro o a los posibles errores, es decir, ir realizando pruebas y a medida que surjan fallas éstas se puedan ir corrigiendo. Además se reutilizó el código para lo que se crearon patrones o modelos estándares, permitiendo que los cambios sean más flexibles.

Esta metodología también plantea unos derechos que tiene el (los) desarrollador(es), los cuales son: decidir cómo se implementan los procesos, crear el sistema con la mejor calidad posible, pedir al cliente en cualquier momento aclaraciones de los requerimientos y cambiar los requerimientos en base a nuevos descubrimientos.

En cuanto a los modelos, paradigmas o filosofías de desarrollo en los cuales se pueden enfocar las diferentes metodologías, para este caso la XP, se consideró como el más óptimo para realizar este software el Modelo en Cascada, ya que está dividido en fases que por lo general son secuenciales, pero que en algún momento pueden llegar a ser concurrentes o incluso retomar alguna ya implementada. Este modelo usualmente maneja las etapas de análisis de requisitos, diseño del sistema, codificación, pruebas, implantación y mantenimiento.

Una de las etapas más importantes es la de análisis en la que se definen los requerimientos del sistema y se crea una definición del sistema para fundamentar todo el trabajo de ingeniería. En el siguiente capítulo se podrán ver los diagramas que modelan toda la funcionalidad del software implementado.

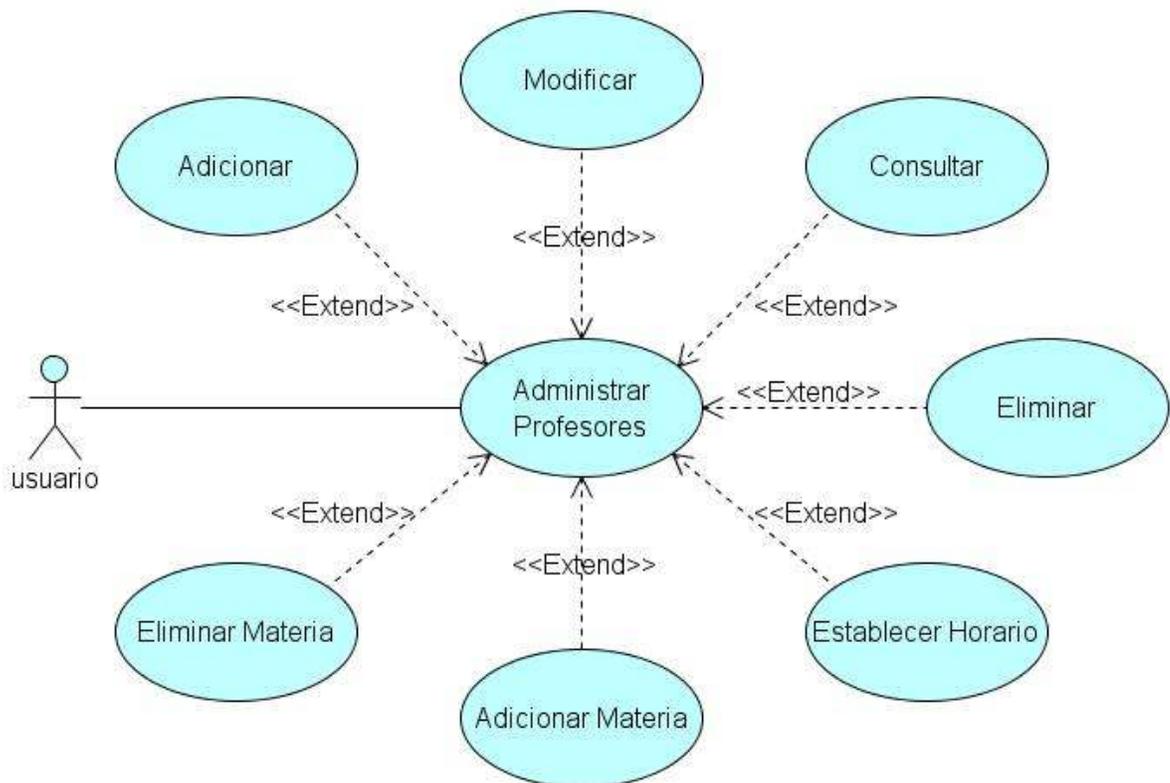
7.3 MODELO DE CASOS DE USO

En esta parte se describe la funcionalidad del sistema utilizando casos de uso los cuales son casi que los principales medios para modelar la interacción entre usuario y sistema.

7.3.1 Diagramas y descripción de Casos de Uso

- **Administrar Profesores**

Figura 25. Caso de Uso Administrar Profesores



Fuente: los autores.

Escenario Adicionar

Tabla 4. Eventos Caso de Uso Administrar Profesores Escenario Adicionar

Caso de uso	Administrar Profesores.
Escenario	Adicionar.
Actor	Administrador.
Propósito	Ingresar un profesor a la base de datos.
Resumen	El actor ingresa la información correspondiente al profesor para ser registrado en el sistema.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R2, R6, R15, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores. 2. Selecciona la opción Adicionar. 5. Diligencia el formulario con la información del profesor. 6. Selecciona la opción Adicionar. 7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	3. Muestra un formulario para obtener los datos del profesor. 6. Se registra la operación en la base de datos y se envía un mensaje de adición exitosa. 8. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 4. 5. Se muestra mensaje de error: El registro ya existe. Regresa a la línea 3.	

Fuente: los autores.

Escenario Modificar

Tabla 5. Eventos Caso de Uso Administrar Profesores Escenario Modificar

Caso de uso	Administrar Profesores.
Escenario	Modificar.
Actor	Administrador, profesor.
Propósito	Modificar la información de un profesor existente en la base de datos.
Resumen	El actor modifica la información referente al nombre y/o el tipo de contrato del profesor.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R11, R20, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Modificar.	3. Muestra una lista de los profesores registrados en el sistema.
4. Elige el profesor a modificar.	
5. Selecciona la opción Aceptar.	6. Muestra un formulario con los datos del profesor.
7. Modifica los datos del profesor.	
8. Selecciona la opción Actualizar.	9. Se registra la operación en la base de datos y se envía un mensaje de modificación exitosa.
10. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	11. Fin de la transacción.
CURSOS ALTERNOS	
2. Si el usuario es Profesor, saltar a la línea 6.	
8. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 7.	

Fuente: los autores.

Escenario Consultar

Tabla 6. Eventos Caso de Uso Administrar Profesores Escenario Consultar

Caso de uso	Administrar Profesores.
Escenario	Consultar.
Actor	Administrador, profesor.
Propósito	Consultar información de los profesores existentes en la base de datos.
Resumen	El actor elige esta opción para realizar diversas consultas con información básica de los profesores registrados.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R24, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Consultar.	3. Muestra una lista de las consultas disponibles.
4. Selecciona la consulta a realizar.	
5. Selecciona el tipo de vista	
6. Selecciona la opción Aceptar.	7. Muestra la información solicitada.
8. Recibe la información.	9. Fin de la transacción.
CURSOS ALTERNOS	
5. Selecciona Tipo de vista Pantalla: muestra la información en pantalla.	
5. Selecciona Tipo de vista Impresión: muestra la información en un documento para imprimir.	
6. Se muestra mensaje de error: No hay información registrada al respecto. Regresa a la línea 4.	

Fuente: los autores.

Escenario Eliminar

Tabla 7. Eventos Caso de Uso Administrar Profesores Escenario Eliminar

Caso de uso	Administrar Profesores.
Escenario	Eliminar.
Actor	Administrador.
Propósito	Eliminar un profesor existente en la base de datos.
Resumen	El actor elige un profesor existente en la base de datos para eliminarlo de ésta.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R28, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Eliminar.	3. Muestra una lista de los profesores registrados en el sistema.
4. Elige el profesor a eliminar.	
5. Selecciona la opción Eliminar.	6. Muestra un mensaje para recibir confirmación.
7. Selecciona la opción Aceptar.	8. Se registra la operación en la base de datos y se envía un mensaje de eliminación exitosa.
9. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	10. Fin de la transacción.
CURSOS ALTERNOS	
7. Se muestra mensaje de error: existen registros que dependen de éste, imposible realizar la operación solicitada. Regresa a la línea 3.	

Fuente: los autores.

Escenario Establecer Horario

Tabla 8. Eventos Caso de Uso Administrar Profesores Escenario Establecer Horario

Caso de uso	Administrar Profesores.
Escenario	Establecer Horario.
Actor	Administrador, Profesor.
Propósito	Establecer las horas disponibles del profesor.
Resumen	El Administrador puede realizar modificaciones al horario disponible de un docente en particular.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R37, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Establecer Horario.	3. Muestra una lista de los profesores registrados en el sistema.
4. Selecciona el profesor.	
5. Selecciona la opción Aceptar.	6. Muestra una representación del horario de clases establecido por la universidad.
7. Revisa y/o modifica el horario deseado.	
8. Selecciona la opción Actualizar.	9. Se registra la operación en la base de datos y se muestra una representación actualizada del horario escogido por el profesor.
	10. Fin de la transacción.
CURSOS ALTERNOS	
2. Si el usuario es Profesor, la opción cambia de nombre a Consultar Horario; el profesor no puede modificar su horario.	
2. Si el usuario es Profesor, saltar a la línea 6 y luego a la línea 10.	

Fuente: los autores.

Escenario Adicionar Materia

Tabla 9. Eventos Caso de Uso Administrar Profesores Escenario Adicionar Materia

Caso de uso	Administrar Profesores.
Escenario	Adicionar Materia.
Actor	Administrador.
Propósito	Adicionar una o varias asignaturas a un profesor para dependiendo de las competencias que tenga.
Resumen	El Administrador elige la(s) asignatura(s) de acuerdo a los conocimientos y/o capacidades de un profesor registrado en la base de datos, esto con el fin de establecer un perfil.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R7, R19, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Adicionar Materia.	3. Muestra una lista de los profesores registrados en el sistema.
4. Selecciona el profesor al cual se le asignará la materia. Clic en Enviar.	5. Muestra el listado de asignaturas a seleccionar.
6. Selecciona la(s) materia(s).	
7. Selecciona la opción Adicionar.	8. Se registra la operación en la base de datos y se envía un mensaje de asignación exitosa.
9. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	10. Fin de la transacción.

Fuente: los autores.

Escenario Eliminar Materia

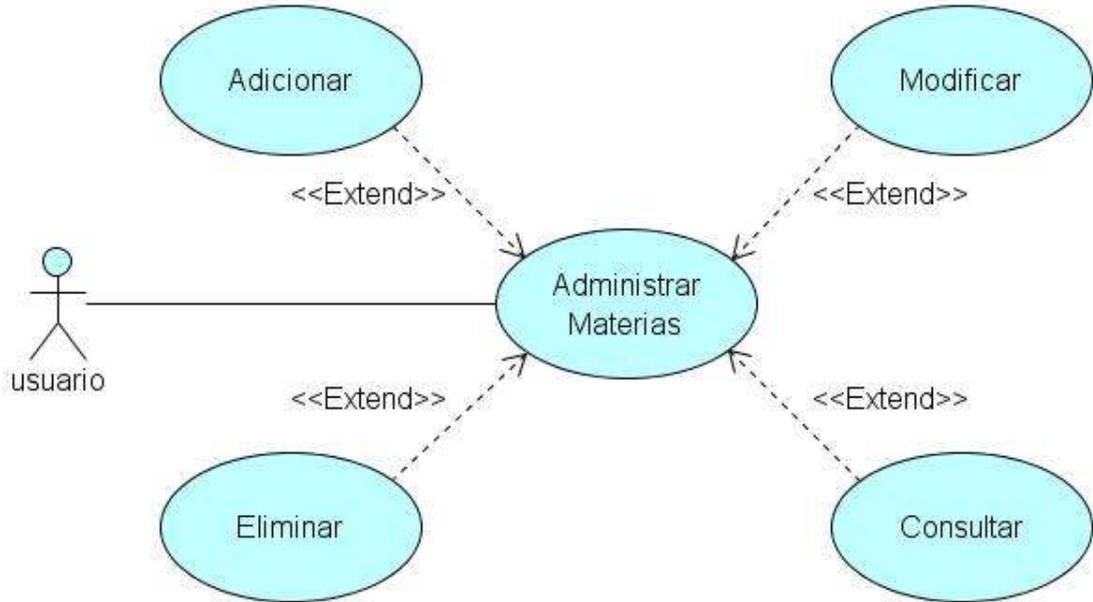
Tabla 10. Eventos Caso de Uso Administrar Profesores Escenario Eliminar Materia

Caso de uso	Administrar Profesores.
Escenario	Eliminar Materia.
Actor	Administrador.
Propósito	Eliminar una asignatura del perfil de un profesor.
Resumen	El Administrador elige la(s) asignatura(s) que desea eliminar del perfil de un profesor.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R29, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Profesores.	
2. Selecciona la opción Eliminar Materia.	3. Muestra una lista de los profesores registrados en el sistema.
4. Selecciona el profesor al cual se le quitará un perfil. Clic en	
5. Selecciona la(s) materia(s) a eliminar.	
6. Selecciona la opción Eliminar.	7. Se registra la operación en la base de datos y se envía un mensaje de eliminación exitosa.
8. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	9. Fin de la transacción.
CURSOS ALTERNOS	
6. Se muestra mensaje de error: existen registros que dependen de éste, imposible realizar la operación solicitada. Regresa a la línea 3.	

Fuente: los autores.

- **Administrar Materias**

Figura 26. Caso de Uso Administrar Materias



Fuente: los autores.

Escenario Adicionar

Tabla 11. Eventos Caso de Uso Administrar Materias Escenario Adicionar

Caso de uso	Administrar Materias.
Escenario	Adicionar.
Actor	Administrador.
Propósito	Ingresar una materia a la base de datos.
Resumen	El Administrador ingresa la información correspondiente a la materia para ser registrada en el sistema.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R3, R8, R16, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Materias.	
2. Selecciona la opción Adicionar.	3. Muestra un formulario para obtener los datos de la materia.
4. Diligencia el formulario con la información de la materia.	
5. Selecciona la opción Adicionar.	6. Se registra la operación en la base de datos y se envía un mensaje de adición exitosa.
7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	8. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 4.	
5. Se muestra mensaje de error: El registro ya existe. Regresa a la línea 3.	

Fuente: los autores.

Escenario Modificar

Tabla 12. Eventos Caso de Uso Administrar Materias Escenario Modificar

Caso de uso	Administrar Materias.
Escenario	Modificar.
Actor	Administrador.
Propósito	Modificar la información de una materia en la base de datos.
Resumen	El Administrador modifica la información referente al nombre, tipo e intensidad horaria de la materia.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R13, R22, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Materias.	
2. Selecciona la opción Modificar.	3. Muestra una lista de las materias registradas en el sistema.
4. Elige la materia a modificar.	
5. Selecciona la opción Enviar.	6. Muestra un formulario con los datos de la materia.
7. Modifica los datos de la materia.	
8. Selecciona la opción Actualizar.	9. Se registra la operación en la base de datos y se envía un mensaje de modificación exitosa.
10. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	11. Fin de la transacción.
CURSOS ALTERNOS	
8. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 7.	

Fuente: los autores.

Escenario Consultar

Tabla 13. Eventos Caso de Uso Administrar Materias Escenario Consultar

Caso de uso	Administrar Materias.
Escenario	Consultar.
Actor	Administrador, profesor.
Propósito	Consultar las materias existentes en la base de datos.
Resumen	El actor elige esta opción para consultar información referente a las materias o asignaturas registradas en el sistema.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R25, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Materias.	
2. Selecciona la opción Consultar.	3. Muestra una lista de las consultas disponibles.
4. Selecciona la consulta a realizar.	
5. Selecciona el tipo de vista.	
6. Selecciona la opción Aceptar.	7. Muestra la información solicitada.
8. Recibe la información.	9. Fin de la transacción.
CURSOS ALTERNOS	
5. Selecciona Tipo de vista Pantalla: muestra la información en pantalla.	
5. Selecciona Tipo de vista Impresión: muestra la información en un documento para imprimir.	
6. Se muestra mensaje de error: La consulta no recuperó ningún registro. Regresa a la línea 4.	

Fuente: los autores.

Escenario Eliminar

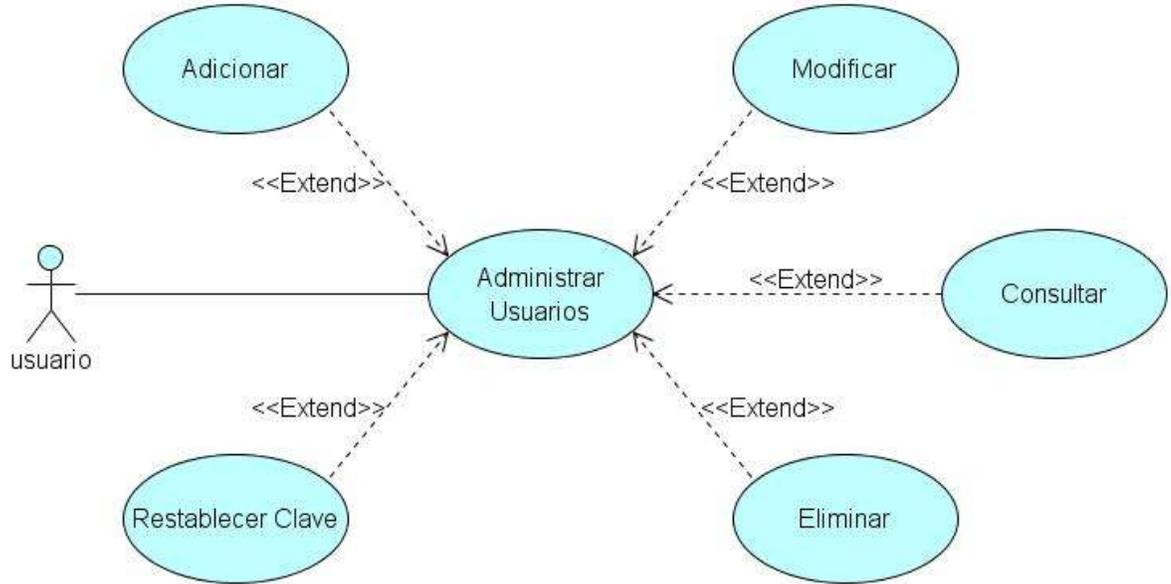
Tabla 14. Eventos Caso de Uso Administrar Materias Escenario Eliminar

Caso de uso	Administrar Materias.
Escenario	Eliminar.
Actor	Administrador.
Propósito	Eliminar una materia existente en la base de datos.
Resumen	El Administrador elige una materia existente en la base de datos para eliminarla de ésta.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R30, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Materias.	
2. Selecciona la opción Eliminar.	3. Muestra una lista de las materias registradas en el sistema.
4. Elige la materia a eliminar.	
5. Selecciona la opción Eliminar.	6. Muestra un mensaje para recibir confirmación.
7. Selecciona la opción Aceptar.	8. Se registra la operación en la base de datos y se envía un mensaje de eliminación exitosa.
9. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	10. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error: existen registros que dependen de éste, imposible realizar la operación solicitada. Regresa a la línea 3.	

Fuente: los autores.

- **Administrar Usuarios**

Figura 27. Caso de Uso Administrar Usuarios



Fuente: los autores.

Escenario Adicionar

Tabla 15. Eventos Caso de Uso Administrar Usuarios Escenario Adicionar

Caso de uso	Administrar Usuarios.
Escenario	Adicionar.
Actor	Administrador.
Propósito	Ingresar un usuario a la base de datos.
Resumen	El administrador ingresa la información correspondiente al usuario para ser registrado en el sistema.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R4, R9, R17, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Usuarios.	
2. Selecciona la opción Adicionar.	3. Muestra un formulario para obtener los datos del usuario.
4. Diligencia el formulario con la información del usuario.	
5. Selecciona la opción Adicionar.	6. Se registra la operación en la base de datos y se envía un mensaje de adición exitosa.
7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	8. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 4.	
5. Se muestra mensaje de error: El registro ya existe. Regresa a la línea 3.	

Fuente: los autores.

Escenario Modificar

Tabla 16. Eventos Caso de Uso Administrar Usuarios Escenario Modificar

Caso de uso	Administrar Usuarios.
Escenario	Modificar.
Actor	Administrador, profesor.
Propósito	Modificar la información de un usuario existente en la base de datos.
Resumen	El actor modifica la información referente al nombre, tipo de usuario y clave.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R13, R22, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Usuarios.	
2. Selecciona la opción Modificar.	3. Muestra una lista de los usuarios registrados en el sistema.
4. Elige el usuario a modificar.	
5. Selecciona la opción Aceptar.	6. Muestra un formulario con los datos del usuario.
7. Modifica los datos del usuario.	
8. Selecciona la opción Actualizar.	9. Se registra la operación en la base de datos y se envía un mensaje de modificación exitosa.
10. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	11. Fin de la transacción.
CURSOS ALTERNOS	
2. Si el usuario es Profesor, saltar a la línea 6.	
8. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 7.	

Fuente: los autores.

Escenario Consultar

Tabla 17. Eventos Caso de Uso Administrar Usuarios Escenario Consultar

Caso de uso	Administrar Usuarios.
Escenario	Consultar.
Actor	Administrador, profesor.
Propósito	Consultar los usuarios existentes en la base de datos.
Resumen	El actor elige esta opción para consultar información básica de los usuarios registrados.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R26, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Usuarios.	
2. Selecciona la opción Consultar.	3. Muestra una lista de las consultas disponibles.
4. Selecciona la consulta a realizar.	
5. Selecciona el tipo de vista	
6. Selecciona la opción Aceptar.	7. Muestra la información solicitada.
8. Recibe la información.	9. Fin de la transacción.
CURSOS ALTERNOS	
5. Selecciona Tipo de vista Pantalla: muestra la información en pantalla.	
5. Selecciona Tipo de vista Impresión: muestra la información en un documento para imprimir.	
6. Se muestra mensaje de error: La consulta no recuperó ningún registro. Regresa a la línea 4.	

Fuente: los autores.

Escenario Eliminar

Tabla 18. Eventos Caso de Uso Administrar Usuarios Escenario Eliminar

Caso de uso	Administrar Usuarios.
Escenario	Eliminar.
Actor	Administrador.
Propósito	Eliminar un usuario existente en la base de datos.
Resumen	El Administrador elige un usuario existente en la base de datos para eliminarlo de ésta.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R31, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Usuarios.	
2. Selecciona la opción Eliminar.	3. Muestra una lista de los usuarios registrados en el sistema.
4. Elige el usuario a eliminar.	
5. Selecciona la opción Eliminar.	6. Muestra un mensaje para recibir confirmación.
7. Selecciona la opción Aceptar.	8. Se registra la operación en la base de datos y se envía un mensaje de eliminación exitosa.
9. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	10. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error: existen registros que dependen de éste, imposible realizar la operación solicitada. Regresa a la línea 3.	

Fuente: los autores.

Escenario Restablecer Clave

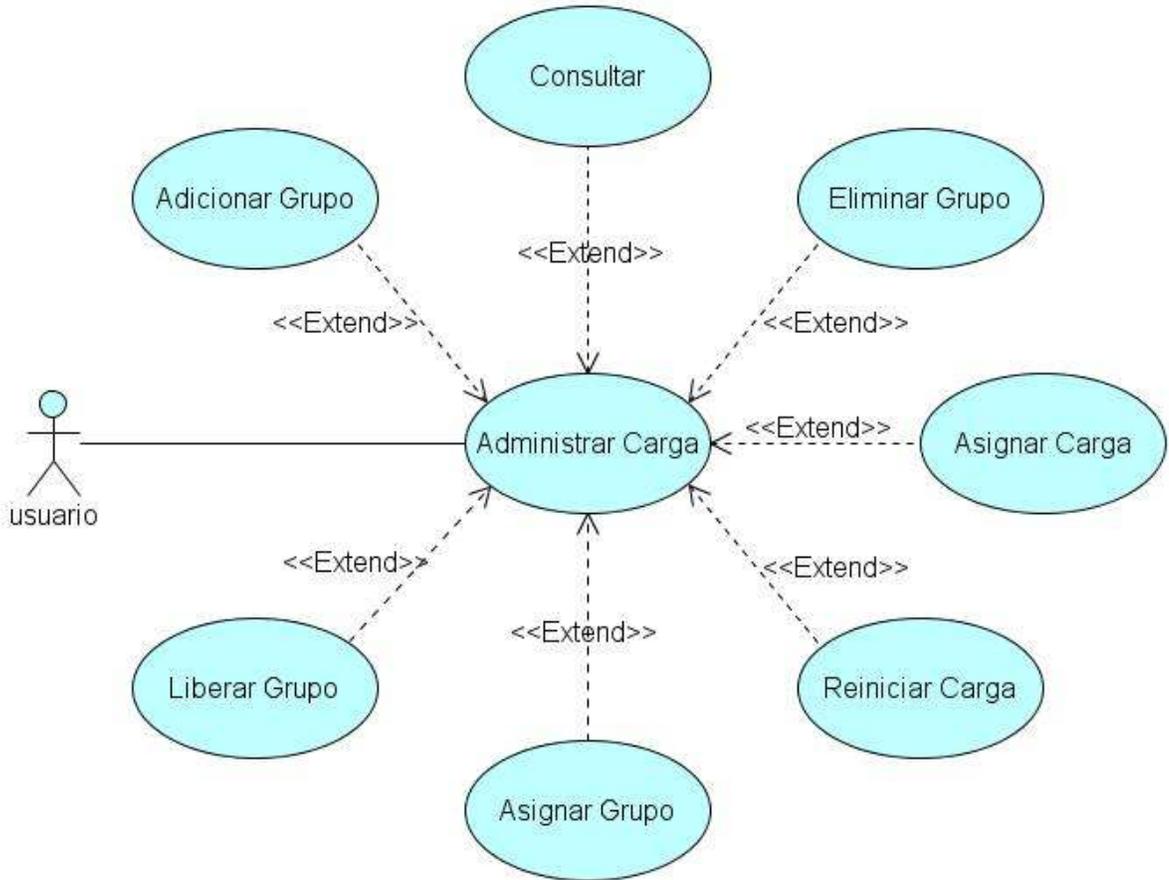
Tabla 19. Eventos Caso de Uso Administrar Usuarios Escenario Restablecer Clave

Caso de uso	Administrar Usuarios.
Escenario	Restablecer Clave.
Actor	Administrador.
Propósito	Retornar a la clave asignada por defecto a un usuario.
Resumen	Cuando a un usuario se le olvida la clave de acceso, el Administrador puede volver a asignar la clave (por defecto) que se le dio al momento de adicionar el nuevo usuario.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R38, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Usuarios.	
2. Selecciona la opción Restablecer Clave.	3. Muestra una lista de los usuarios registrados en el sistema.
4. Selecciona el usuario.	
5. Selecciona la opción Restablecer.	6. Se registra la operación en la base de datos y se envía un mensaje de éxito.
7. Al terminar la operación el actor recibe el mensaje confirmando que se realizó con éxito.	8. Fin de la transacción.

Fuente: los autores.

- **Administrar Carga**

Figura 28. Caso de Uso Administrar Carga



Fuente: los autores.

Escenario Adicionar Grupo

Tabla 20. Eventos Caso de Uso Administrar Carga Escenario Adicionar Grupo

Caso de uso	Administrar Carga.
Escenario	Adicionar Grupo
Actor	Administrador.
Propósito	Crear un grupo de una determinada asignatura.
Resumen	El Administrador elige la materia y la jornada y establece el número de grupo y el respectivo horario que desea para ese grupo.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R10, R18, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Carga.	
2. Selecciona la opción Adicionar Grupo.	5. Muestra un formulario para diligenciar los datos del grupo.
4. Diligencia el formulario.	
5. Selecciona la opción Adicionar.	6. Se registra la operación en la base de datos y se envía un mensaje de adición exitosa.
7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	8. Fin de la transacción.
CURSOS ALTERNOS	
5. Se muestra mensaje de error en formato al digitar información inválida. Regresa a la línea 4.	
5. Se muestra mensaje de error: El registro ya existe. Regresa a la línea 3.	

Fuente: los autores.

Escenario Consultar

Tabla 21. Eventos Caso de Uso Administrar Carga Escenario Consultar

Caso de uso	Administrar Carga.
Escenario	Consultar.
Actor	Administrador, profesor.
Propósito	Consultar los grupos creados de las asignaturas.
Resumen	El actor elige esta opción para consultar información referente a los grupos registrados en el sistema.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R27, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Carga.	
2. Selecciona la opción Consultar.	3. Muestra una lista de las consultas disponibles.
4. Selecciona la consulta a realizar.	
5. Selecciona el tipo de vista.	
6. Selecciona la opción Aceptar.	7. Muestra la información solicitada.
8. Recibe la información.	9. Fin de la transacción.
CURSOS ALTERNOS	
5. Selecciona Tipo de vista Pantalla: muestra la información en pantalla.	
5. Selecciona Tipo de vista Impresión: muestra la información en un documento para imprimir.	
6. Se muestra mensaje de error: La consulta no recuperó ningún registro. Regresa a la línea 4.	

Fuente: los autores.

Escenario Eliminar Grupo

Tabla 22. Eventos Caso de Uso Administrar Carga Escenario Eliminar Grupo

Caso de uso	Administrar Carga.
Escenario	Eliminar Grupo
Actor	Administrador.
Propósito	Eliminar un grupo de una determinada asignatura.
Resumen	El Administrador elige la materia y el grupo respectivo que desea eliminar de la base de datos.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R32, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Carga.	
2. Selecciona la opción Eliminar Grupo.	3. Muestra la lista de asignaturas registradas.
4. Selecciona la materia de la cual se desea eliminar el grupo.	
5. Selecciona la opción Enviar.	6. Muestra una lista de los grupos creados de la materia seleccionada.
7. Selecciona la opción Eliminar correspondiente al grupo que desea borrar.	8. Muestra un mensaje para recibir confirmación.
9. Selecciona la opción Aceptar.	10. Se registra la operación en la base de datos y se envía un mensaje de eliminación exitosa.
11. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	12. Fin de la transacción.
CURSOS ALTERNOS	
7. Se muestra mensaje de error: existen registros que dependen de éste, imposible realizar la operación solicitada. Regresa a la línea 6.	

Fuente: los autores.

Escenario Asignar Carga

Tabla 23. Eventos Caso de Uso Administrar Carga Escenario Asignar Carga

Caso de uso	Administrar Carga.
Escenario	Asignar Carga.
Actor	Administrador.
Propósito	Realizar la asignación de los docentes a los grupos.
Resumen	El Administrador selecciona las opciones de jornada, sobrecarga y categoría para realizar la asignación de la carga académica.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R33, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción Administrar Carga. 2. Selecciona la opción Asignar Carga. 4. Diligencia formulario. 5. Selecciona la opción Enviar. 7. Selecciona profesor(es). 8. Verifica y/o modifica el número máximo (Tope Contrato) de horas a asignar a cada profesor. 9. Selecciona la opción Iniciar. 11. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito. 	<ol style="list-style-type: none"> 3. Muestra un formulario para seleccionar las opciones de asignación. 6. Muestra un listado de los profesores dependiendo de lo seleccionado en la línea 3, con la información del contrato y la carga de cada uno. 10. Se registra la operación en la base de datos y se envía un mensaje de éxito. 12. Fin de la transacción.
CURSOS ALTERNOS	
<ol style="list-style-type: none"> 9. Si ya se hizo esa misma asignación previamente, el sistema muestra mensaje: No se realizaron nuevas asignaciones. 	

Fuente: los autores.

Escenario Reiniciar Carga

Tabla 24. Eventos Caso de Uso Administrar Carga Escenario Reiniciar Carga

Caso de uso	Administrar Carga.
Escenario	Reiniciar Carga.
Actor	Administrador.
Propósito	Borra la asignación de los profesores a los grupos de las asignaturas.
Resumen	El Administrador selecciona reiniciar la carga para eliminar la asignación de docentes a los respectivos grupos; esta operación puede ser parcial o completa dependiendo de las opciones que le brinda el sistema tales como Liberación Completa, Conservar Última Selección o Realizar Nueva Selección.
Tipo	Primario y esencial.
Referencias cruzadas	R1, R34, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Carga.	
2. Selecciona la opción Reiniciar Carga.	3. Muestra una lista de los tipos de reinicio posibles.
4. Selecciona tipo de reinicio.	
1. Selecciona Procesar.	6. Muestra mensaje solicitando confirmación.
7. Confirma seleccionando Aceptar.	8. Se registra la operación en la base de datos y se envía un mensaje de éxito.
9. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	10. Fin de la transacción.

Fuente: los autores.

Escenario Asignar Grupo

Tabla 25. Eventos Caso de Uso Administrar Carga Escenario Asignar Grupo

Caso de uso	Administrar Carga.
Escenario	Asignar Grupo.
Actor	Administrador.
Propósito	Asignar manualmente un docente a un determinado grupo.
Resumen	El Administrador elige un docente y un grupo para realizar la respectiva asignación sin tener consideraciones de perfil ni disponibilidad del profesor. Solo se garantiza que no se presente conflicto con otro(s) grupo(s) que ya tenga asignado(s).
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R35, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción Administrar Carga. 2. Selecciona la opción Asignar Grupo. 4. Elige el profesor y el grupo para realizar la asignación. 5. Elige la opción Asignar. 7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito. 	<ol style="list-style-type: none"> 3. Muestra una lista de los profesores y grupos registrados en el sistema. 6. Se registra la operación en la base de datos y se envía un mensaje de asignación exitosa. 8. Fin de la transacción.
CURSOS ALTERNOS	
<ol style="list-style-type: none"> 5. Se muestra mensaje de error: Imposible realizar la asignación, una o más horas presentan cruce con otro grupo. Regresa a la línea 3. 	

Fuente: los autores.

Escenario Liberar Grupo

Tabla 26. Eventos Caso de Uso Administrar Carga Escenario Liberar Grupo

Caso de uso	Administrar Carga.
Escenario	Liberar Grupo.
Actor	Administrador.
Propósito	Eliminar manualmente una asignación de profesor a grupo creado con anterioridad.
Resumen	El Administrador elige un grupo de una asignatura para quitarle el docente asignado. Esto es una eliminación manual.
Tipo	Opcional y esencial.
Referencias cruzadas	R1, R36, R39.
CURSO NORMAL DE LOS EVENTOS	
Acción del autor	Respuesta del sistema
1. Selecciona la opción Administrar Carga.	
2. Selecciona la opción Liberar Grupo.	3. Muestra una lista de grupos asignados previamente.
4. Elige el grupo.	
5. Selecciona la opción Liberar.	6. Se registra la operación en la base de datos y se envía un mensaje de liberación exitosa.
7. Al terminar la operación el usuario recibe el mensaje confirmando que se realizó con éxito.	8. Fin de la transacción.

Fuente: los autores.

7.5 MODELO DINÁMICO

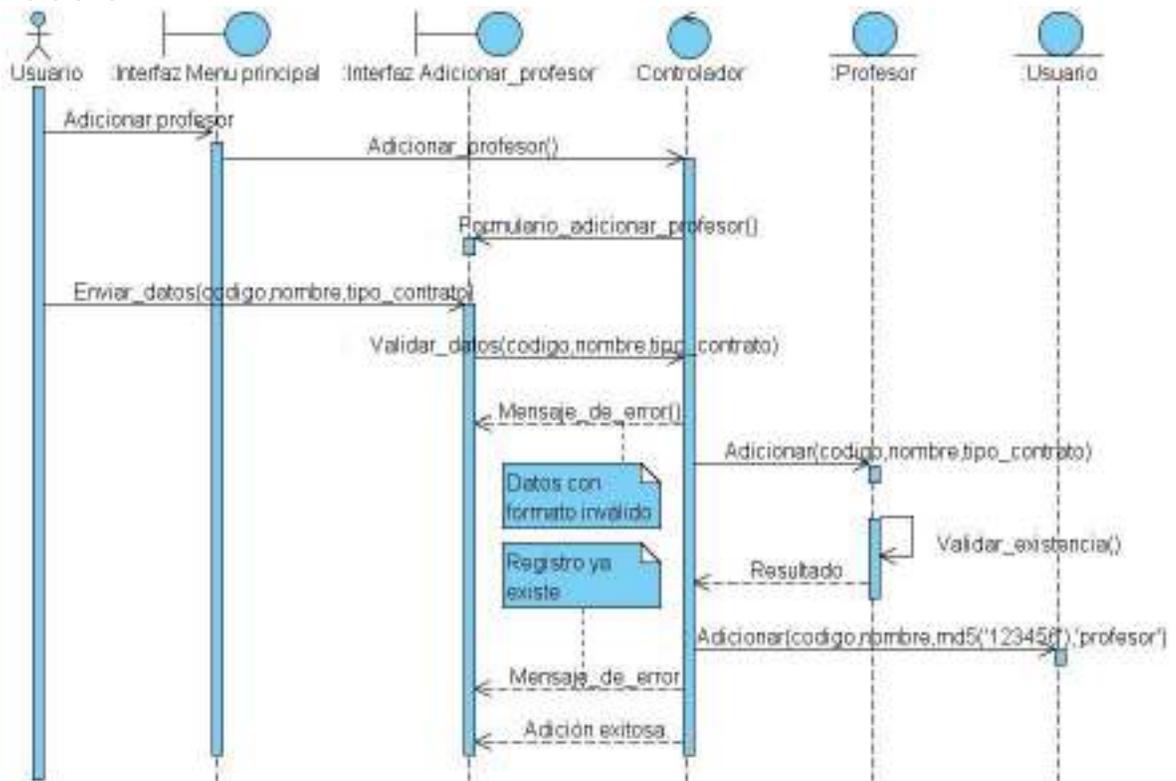
En este apartado se ilustrará el comportamiento del sistema a lo largo del tiempo utilizando los diagramas de colaboración y los diagramas de secuencia

7.5.1. Diagramas de Secuencia

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Adicionar

La figura 29 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Adicionar. El administrador elige la opción Adicionar del submenú Administrar Profesores y el sistema le permite ingresar la información correspondiente a un profesor para ser registrado en la base de datos.

Figura 29. Diagrama de Secuencia Caso de Uso Administrar Profesor Escenario Adicionar

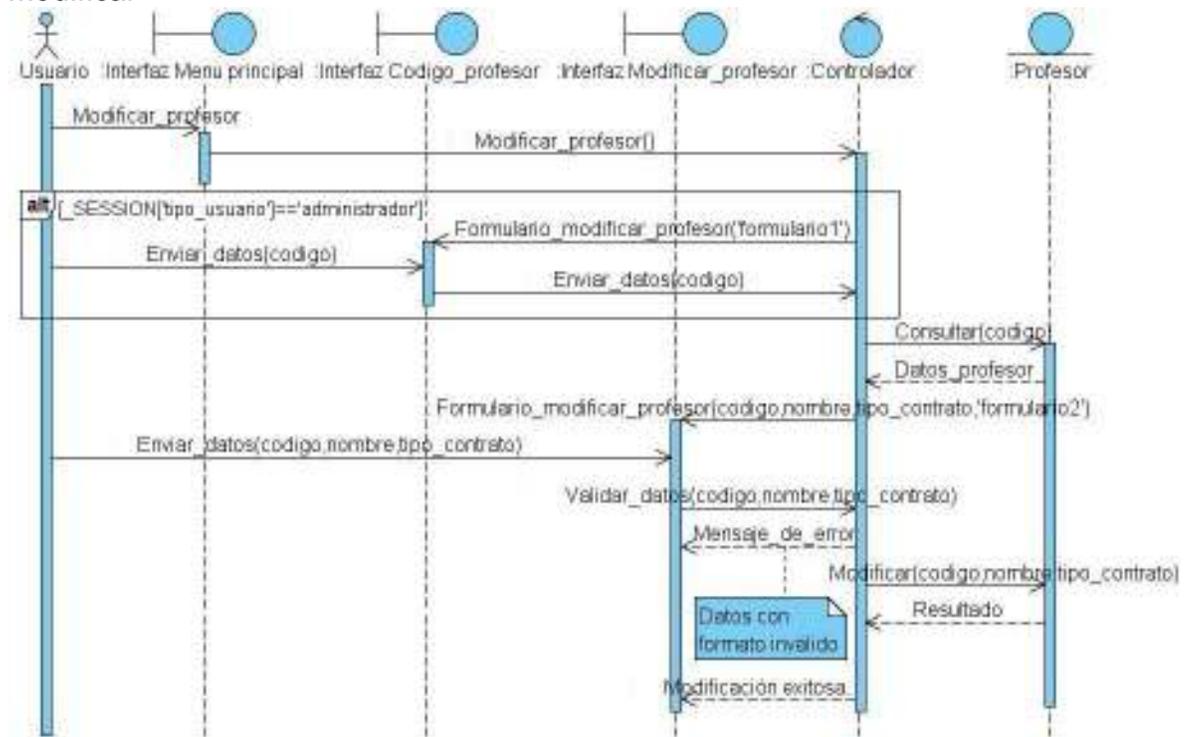


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Modificar

La figura 30 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Modificar. El usuario elige la opción Modificar del submenú Administrar Profesores y el sistema le permite modificar la información de un profesor existente en la base de datos.

Figura 30. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Modificar

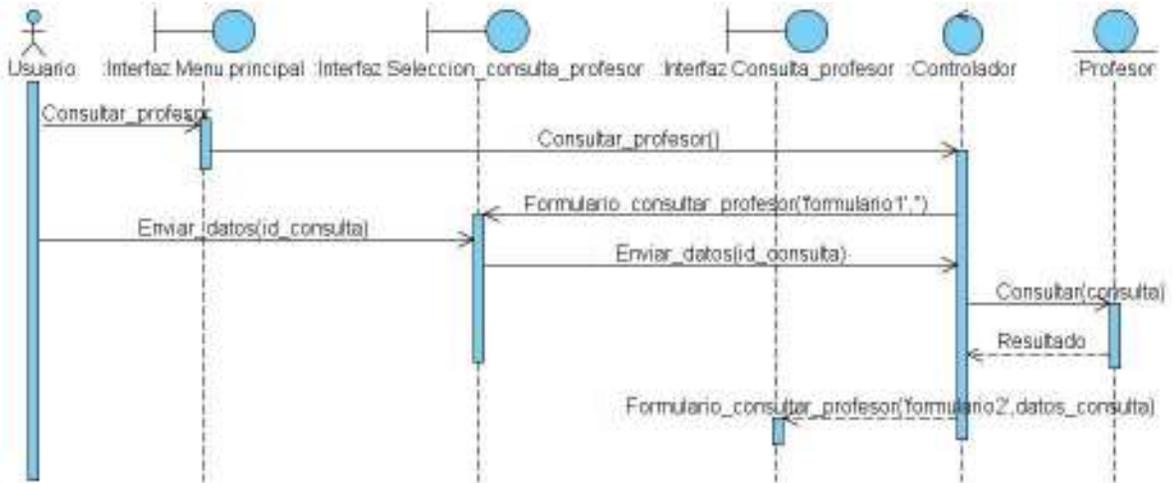


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Consultar

La figura 31 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Consultar. El usuario elige la opción Consultar del submenú Administrar Profesores y el sistema le permite realizar diversas consultas con información básica de los profesores registrados y de las asignaturas que tienen asignadas como perfil.

Figura 31. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Consultar

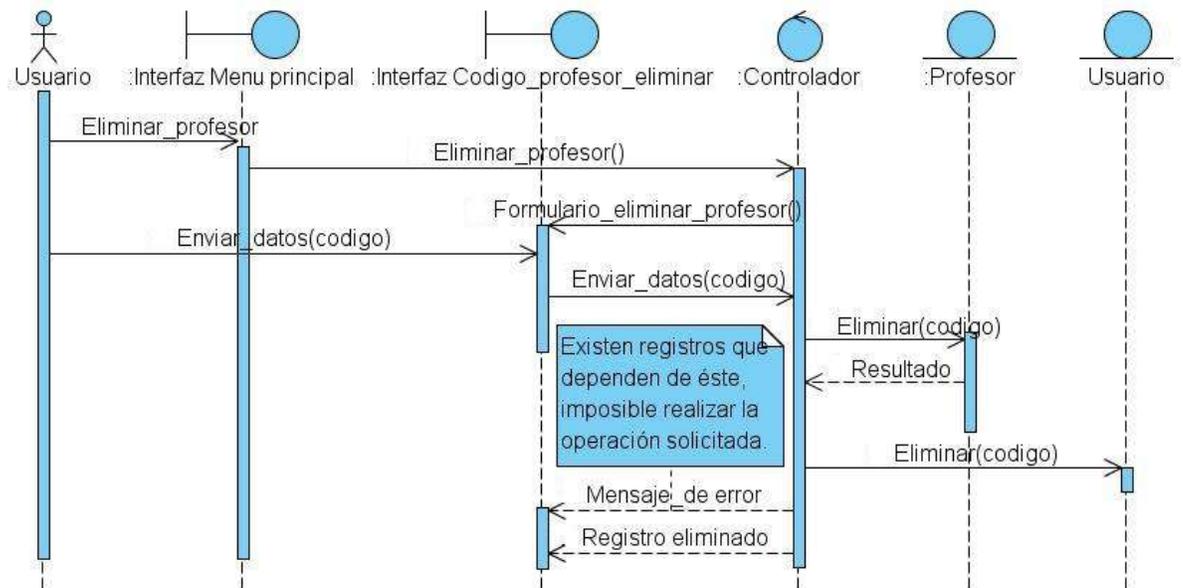


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Eliminar

La figura 32 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Eliminar. El administrador elige la opción Eliminar del submenú Administrar Profesores y el sistema le permite elegir un profesor existente en la base de datos para eliminarlo de ésta.

Figura 32. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Eliminar

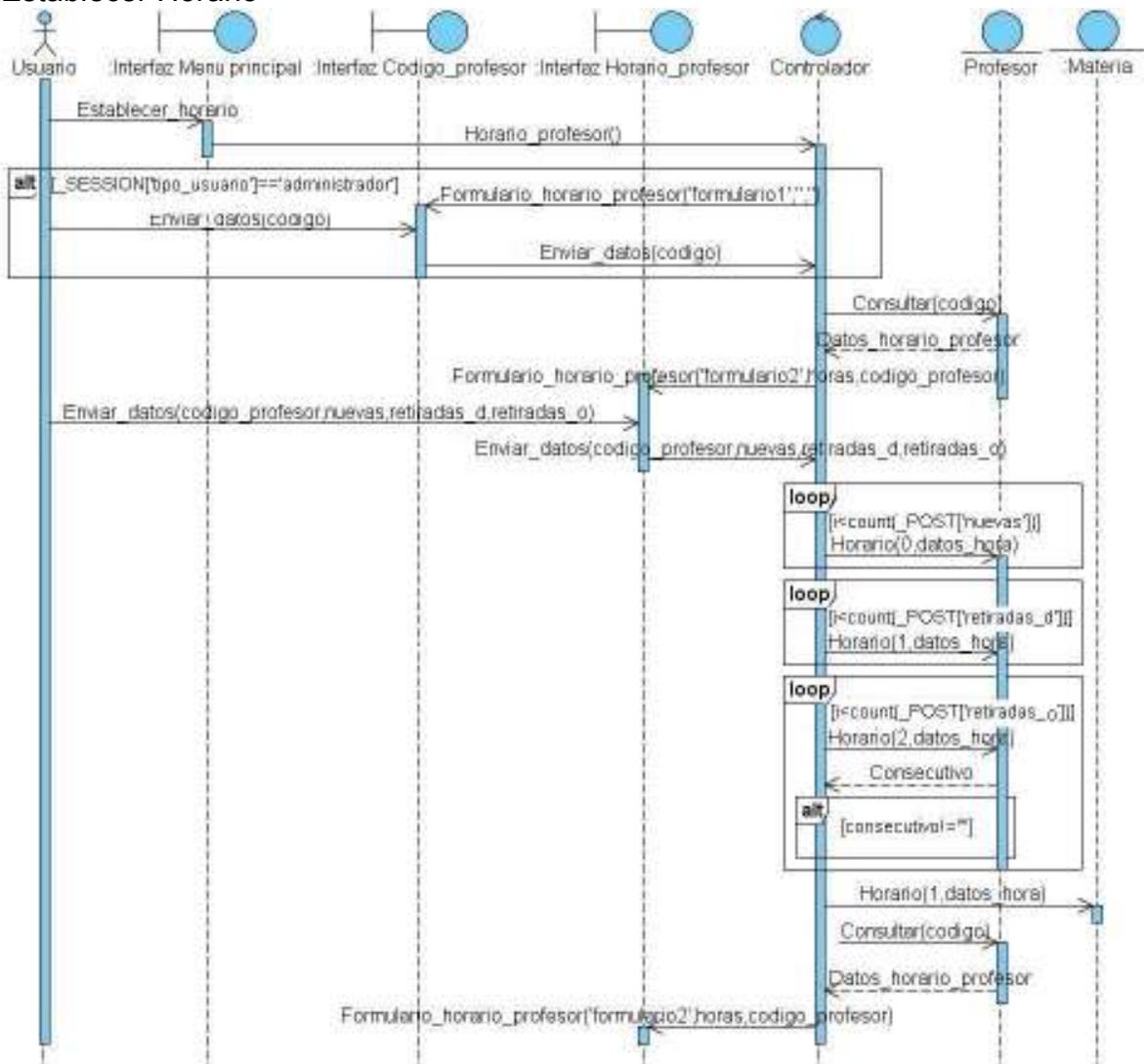


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Establecer Horario

La figura 33 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Establecer Horario. El usuario elige la opción Establecer Horario del submenú Administrar Profesores y el sistema le permite ingresar horas disponibles para que posteriormente sean asignadas las horas de clase a los profesores.

Figura 33. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Establecer Horario

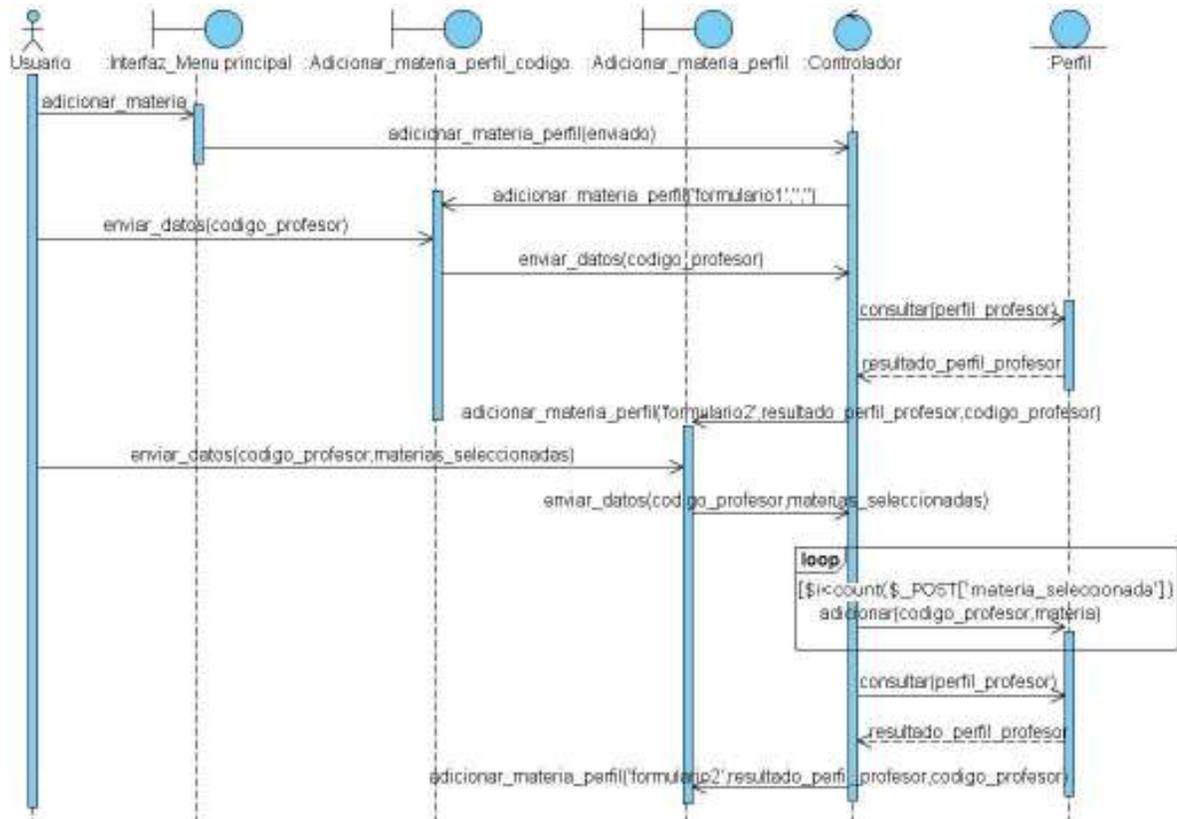


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Adicionar Materia

La figura 34 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Adicionar Materia. El usuario elige la opción Adicionar Materia del submenú Administrar Profesores y el sistema le permite al Administrador asignarle las asignaturas que el docente esté en capacidad de dictar para que posteriormente le sean asignados los grupos respectivos.

Figura 34. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Adicionar Materia

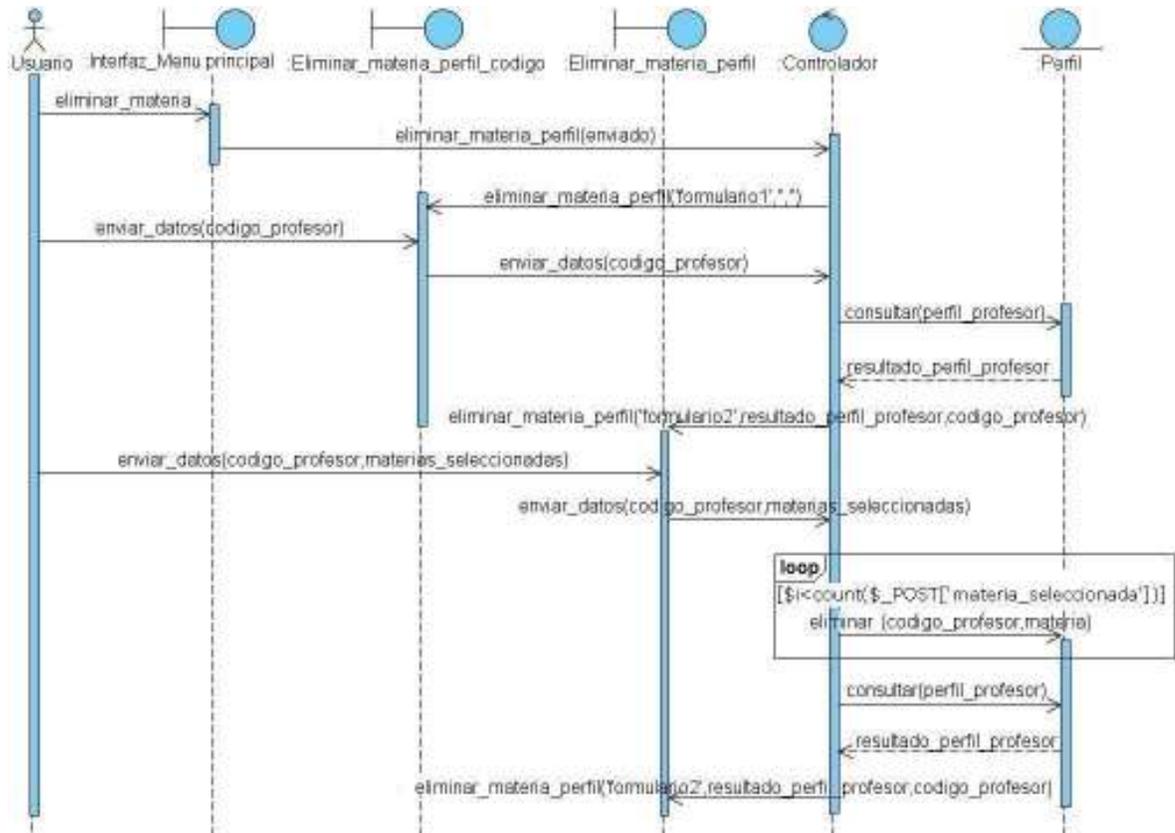


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Eliminar Materia

La figura 35 muestra la secuencia de eventos para el caso de uso Administrar Profesores en el escenario Eliminar Materia. El usuario elige la opción Eliminar Materia del submenú Administrar Profesores y el sistema le permite al Administrador eliminarle la(s) asignatura(s) al docente para que luego no sea considerado para dictar grupos de determinadas asignaturas.

Figura 35. Diagrama de Secuencia Caso de Uso Administrar Profesores Escenario Eliminar Materia

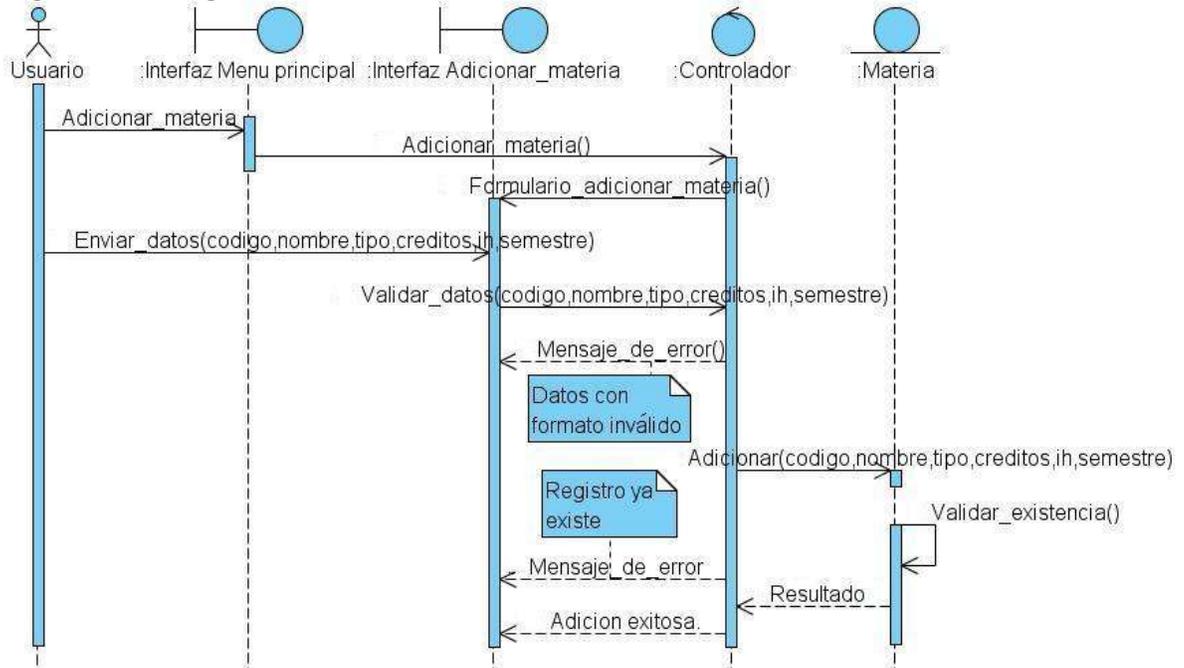


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Materias Escenario Adicionar

La figura 36 muestra la secuencia de eventos para el caso de uso Administrar Materias en el escenario Adicionar. El administrador elige la opción Adicionar del submenú Administrar Materias y el sistema le permite ingresar la información correspondiente a una materia para ser registrada en la base de datos.

Figura 36. Diagrama de Secuencia Administrar Materias Escenario Adicionar

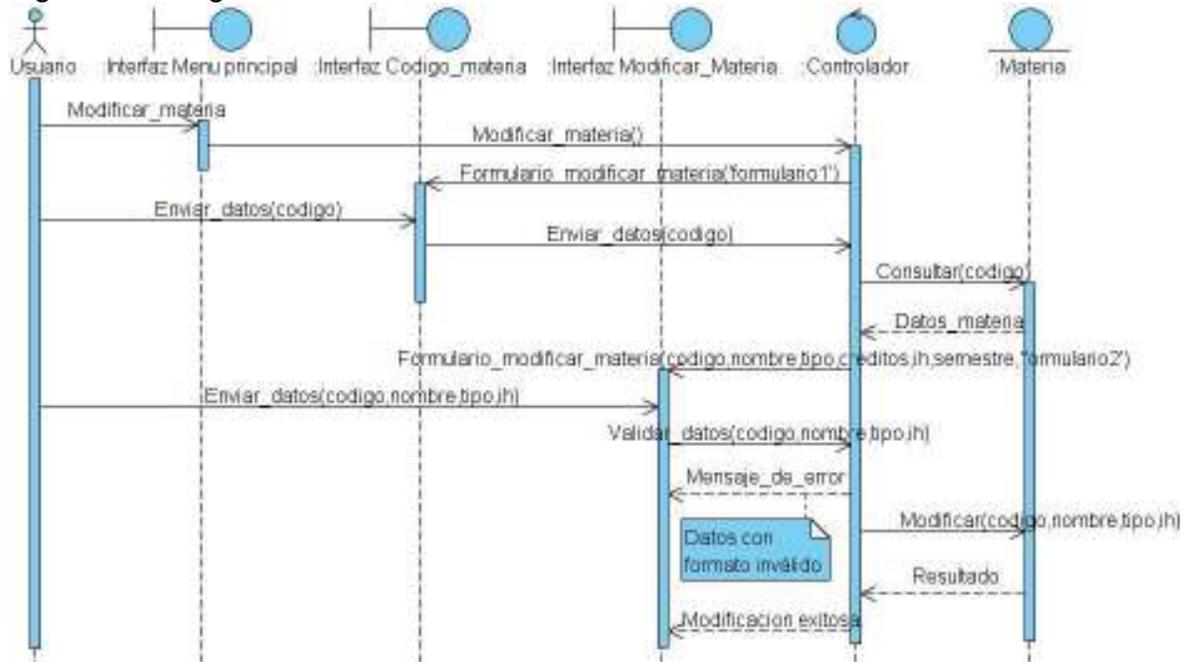


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Materias Escenario Modificar

La figura 37 muestra la secuencia de eventos para el caso de uso Administrar Materias en el escenario Modificar. El administrador elige la opción Modificar del submenú Administrar Materias y el sistema le permite modificar la información de una materia existente en la base de datos.

Figura 37. Diagrama de Secuencia Administrar Materias Escenario Modificar

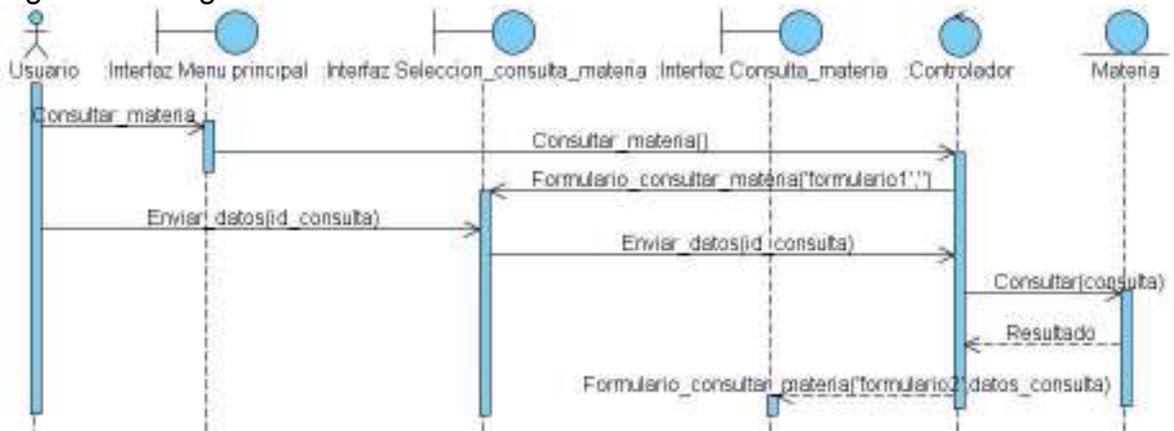


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Materias Escenario Consultar

La figura 38 muestra la secuencia de eventos para el caso de uso Administrar Materias en el escenario Consultar. El usuario elige la opción Consultar del submenú Administrar Materias y el sistema le permite consultar información referente a las materias o asignaturas registradas en el sistema.

Figura 38. Diagrama de Secuencia Administrar Materias Escenario Consultar

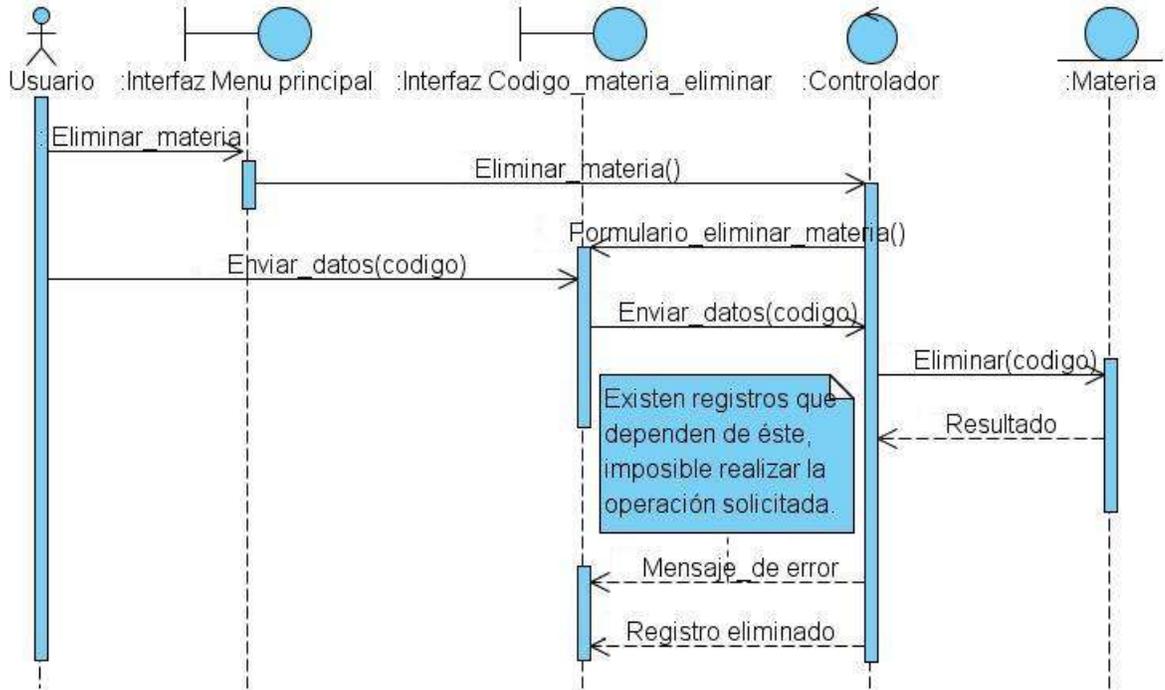


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Materias Escenario Eliminar

La figura 39 muestra la secuencia de eventos para el caso de uso Administrar Materias en el escenario Eliminar. El administrador elige la opción Eliminar del submenú Administrar Materias y el sistema le permite elegir una materia existente en la base de datos para eliminarla de ésta.

Figura 39. Diagrama de Secuencia Administrar Materias Escenario Eliminar

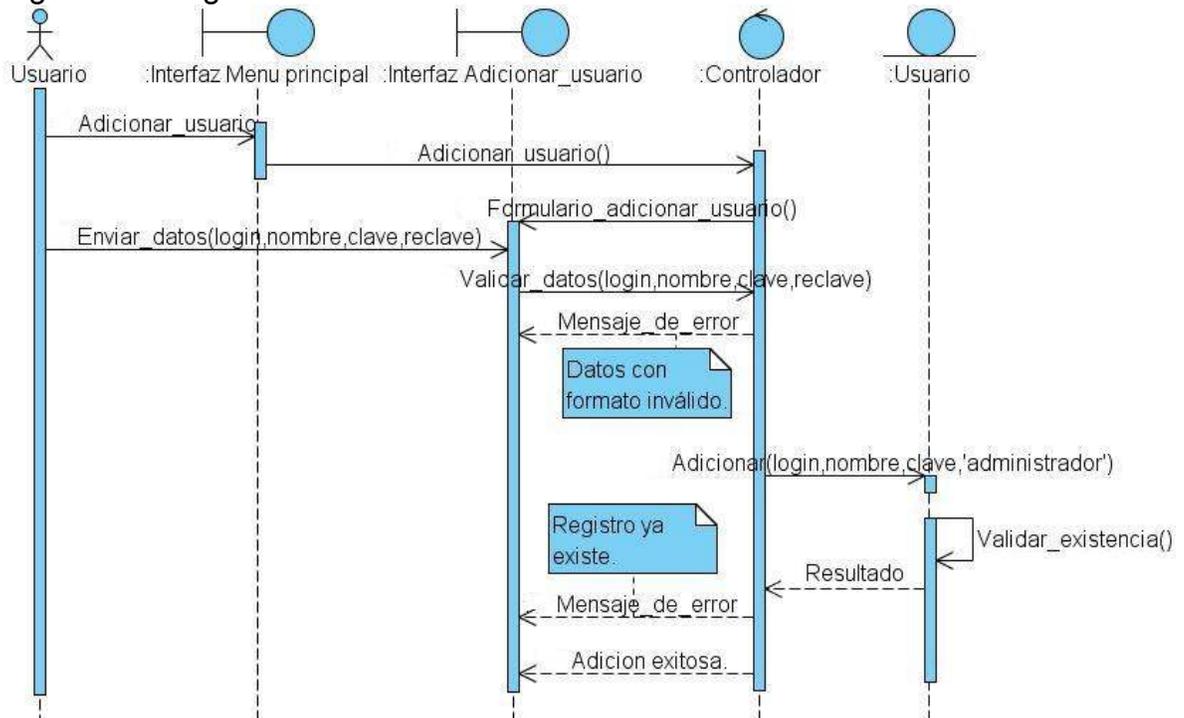


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Usuarios Escenario Adicionar

La figura 40 muestra la secuencia de eventos para el caso de uso Administrar Usuarios en el escenario Adicionar. El administrador elige la opción Adicionar del submenú Administrar Usuarios y el sistema le permite ingresar la información correspondiente al usuario para ser registrada en la base de datos.

Figura 40. Diagrama de Secuencia Administrar Usuarios Escenario Adicionar

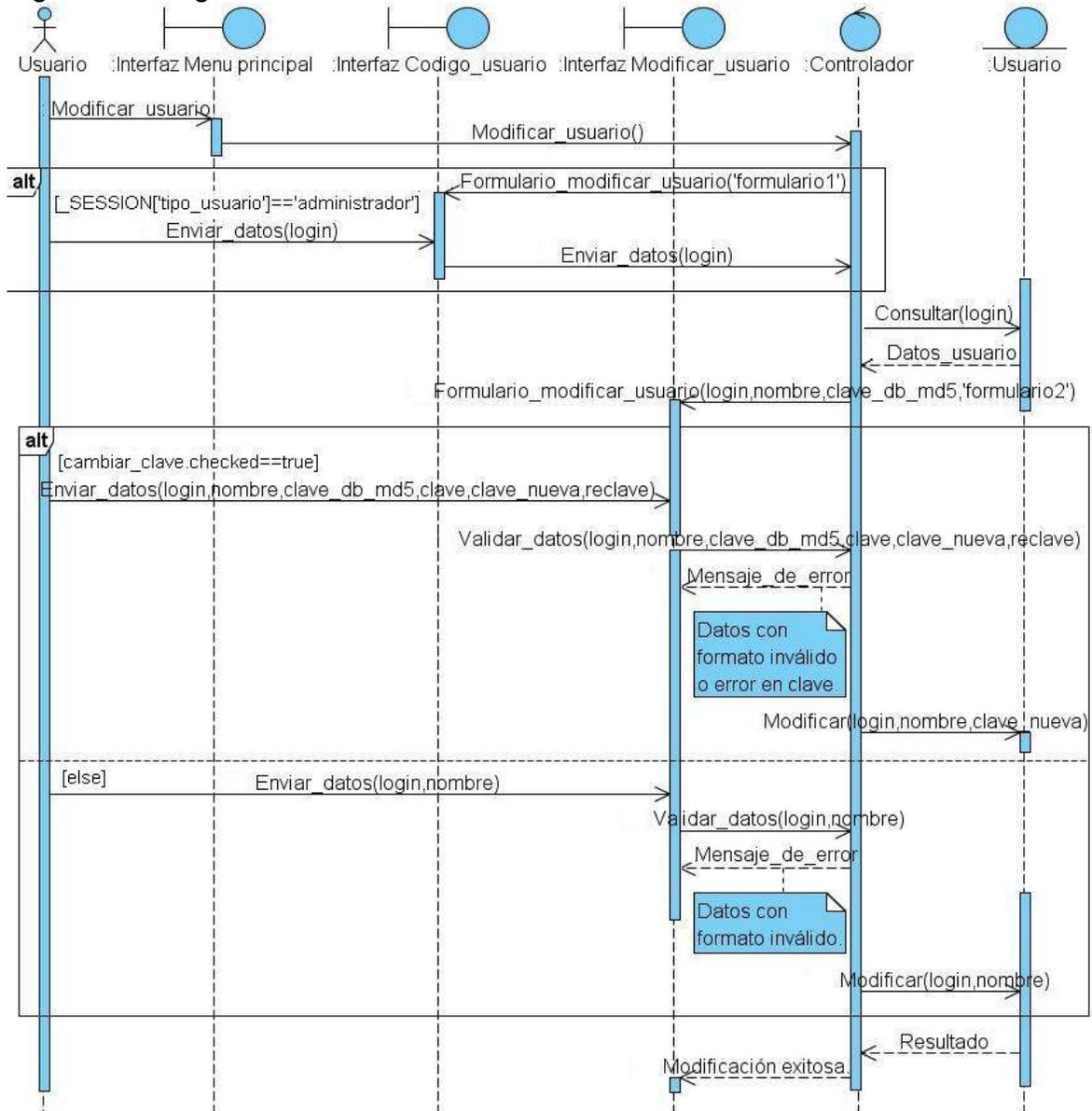


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Usuarios Escenario Modificar

La figura 41 muestra la secuencia de eventos para el caso de uso Administrar Usuarios en el escenario Modificar. El usuario elige la opción Modificar del submenú Administrar Usuarios y el sistema le permite realizar cambios en la información de un usuario existente en la base de datos.

Figura 41. Diagrama de Secuencia Administrar Usuarios Escenario Modificar

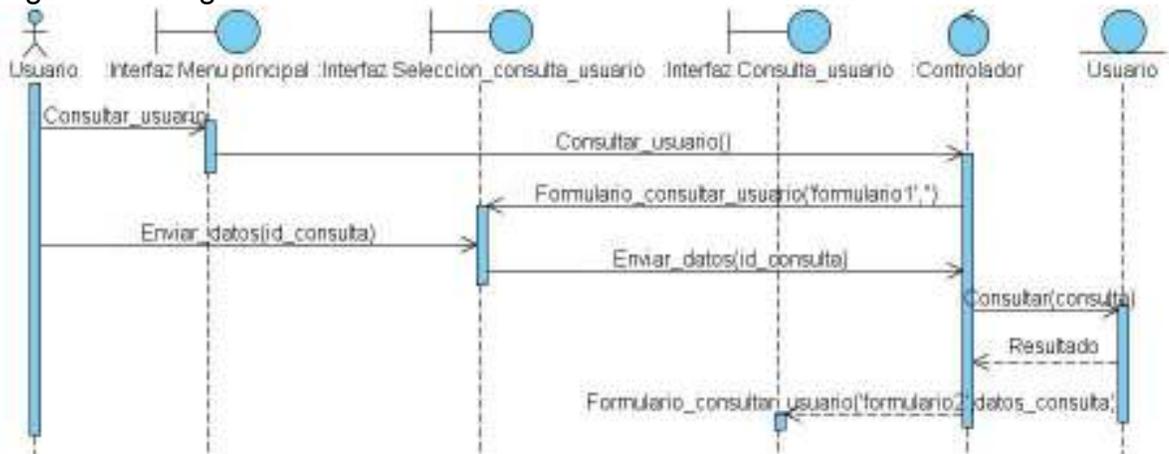


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Usuarios Escenario Consultar

La figura 42 muestra la secuencia de eventos para el caso de uso Administrar Usuarios en el escenario Consultar. El usuario elige la opción Consultar del submenú Administrar Usuarios y el sistema le permite consultar información básica de los usuarios registrados en la base de datos.

Figura 42. Diagrama de Secuencia Administrar Usuarios Escenario Consultar

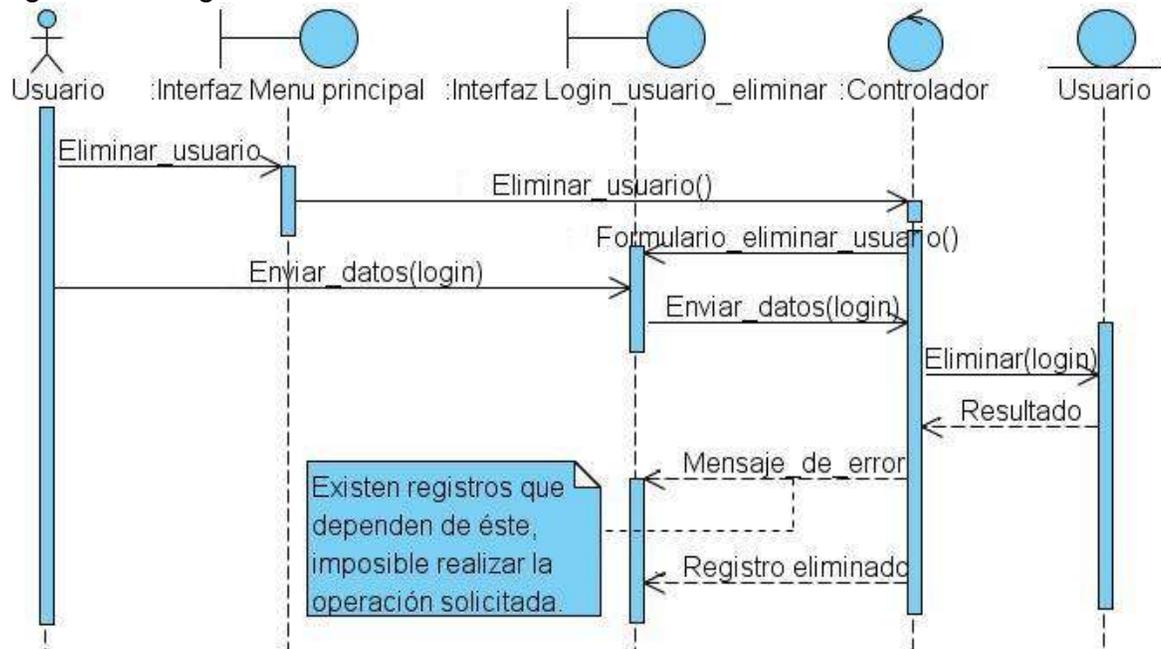


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Usuarios Escenario Eliminar

La figura 43 muestra la secuencia de eventos para el caso de uso Administrar Usuarios en el escenario Eliminar. El administrador elige la opción Eliminar del submenú Administrar Usuarios y el sistema le permite elegir un usuario existente en la base de datos para eliminarlo de ésta.

Figura 43. Diagrama de Secuencia Administrar Usuarios Escenario Eliminar

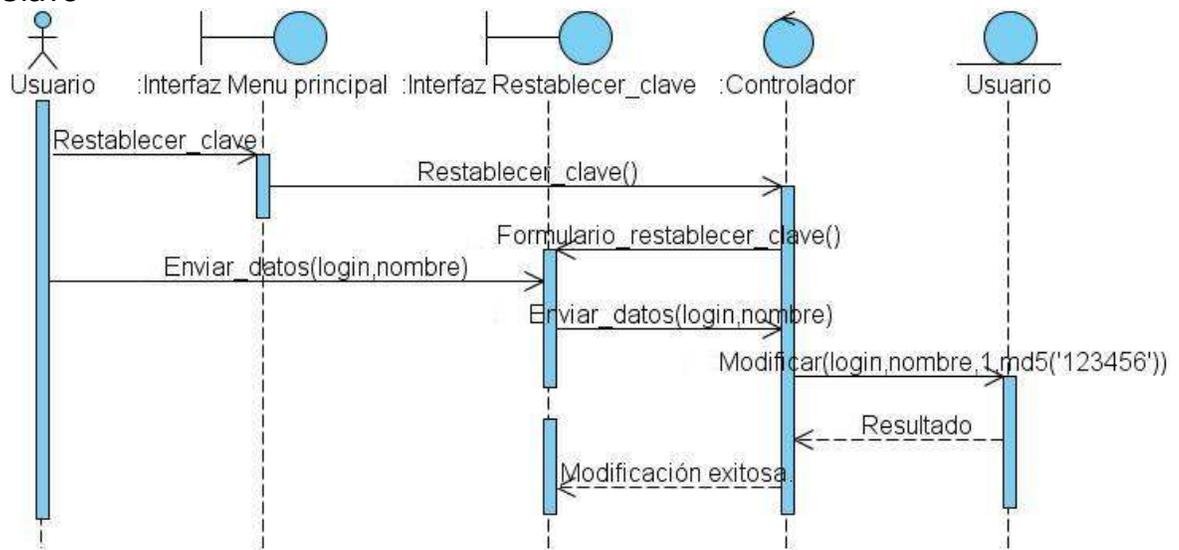


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Usuarios Escenario Restablecer Clave

La figura 44 muestra la secuencia de eventos para el caso de uso Administrar Usuarios en el escenario Restablecer Clave. El administrador elige la opción Restablecer Clave del submenú Administrar Usuarios y el sistema le permite volver asignar la clave a un usuario que se le dio al momento de ser adicionado a la base de datos. Esto, en caso de olvido.

Figura 44. Diagrama de Secuencia Administrar Usuarios Escenario Restablecer Clave

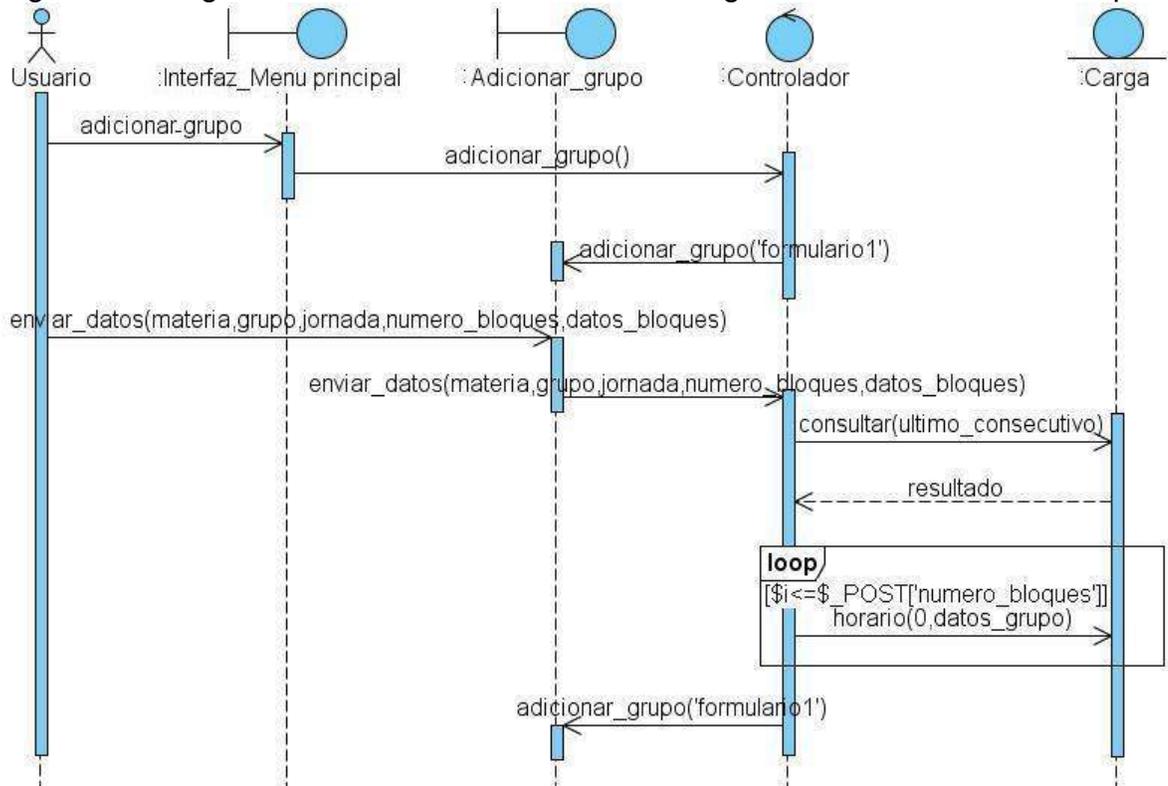


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Adicionar Grupo

La figura 45 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Adicionar Grupo. El administrador elige la opción Adicionar Grupo del submenú Administrar Carga y el sistema le permite ingresar la información correspondiente a un grupo para ser registrada en la base de datos.

Figura 45. Diagrama de Secuencia Administrar Carga Escenario Adicionar Grupo

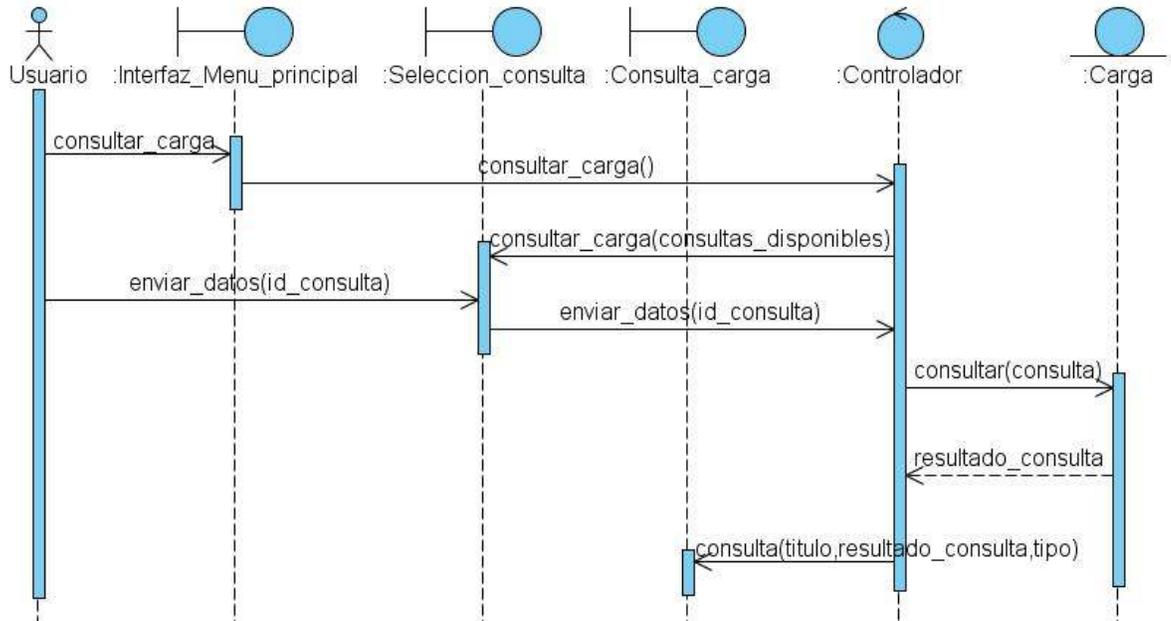


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Consultar

La figura 46 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Consultar. El usuario elige la opción Consultar del submenú Administrar Carga y el sistema le permite consultar información referente a los grupos que tienen asignados los profesores y de los grupos que se han creado de cada asignatura.

Figura 46. Diagrama de Secuencia Administrar Carga Escenario Consultar

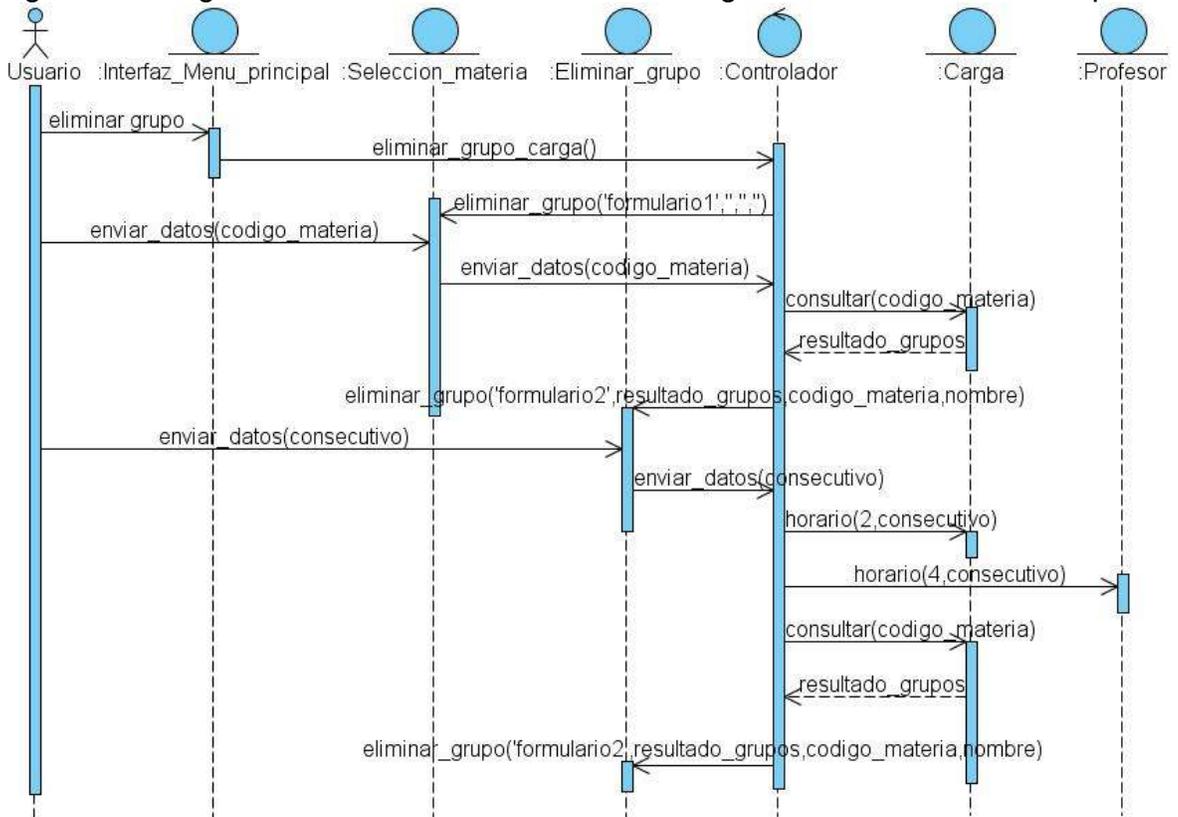


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Eliminar Grupo

La figura 47 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Eliminar Grupo. El administrador elige la opción Eliminar Grupo del submenú Administrar Carga y el sistema le permite borrar un grupo de una asignatura registrado en la base de datos.

Figura 47. Diagrama de Secuencia Administrar Carga Escenario Eliminar Grupo

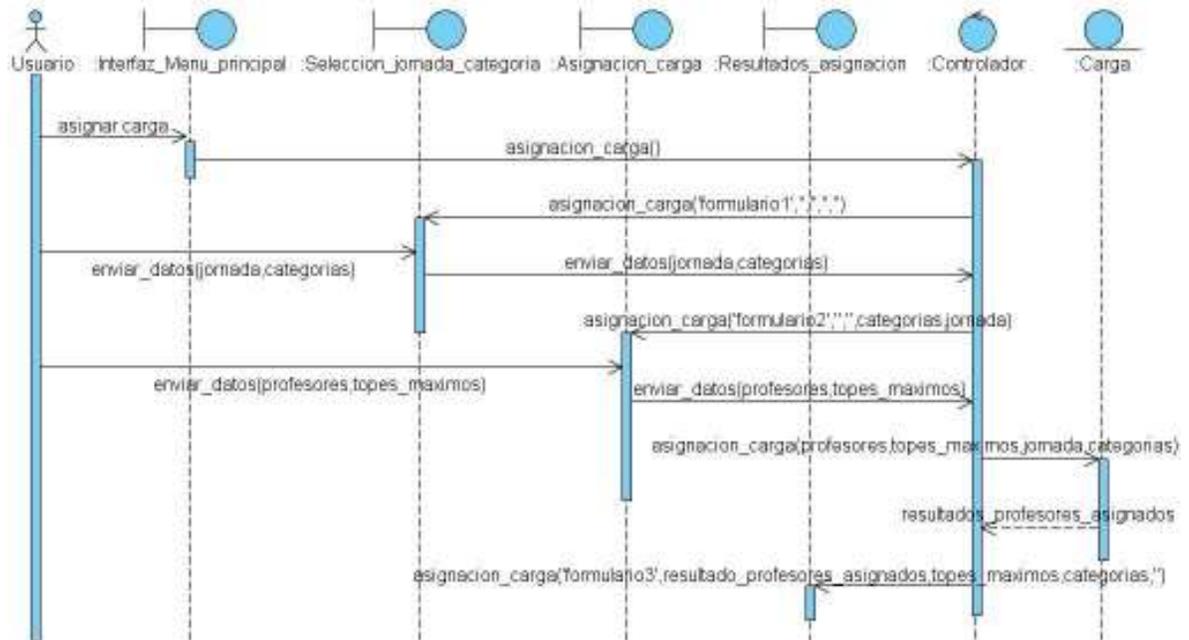


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Asignar Carga

La figura 48 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Asignar Carga. El administrador elige la opción Asignar Carga del submenú Administrar Carga y el sistema le permite elegir la jornada y las categorías de asignación (fases) para realizar la respectiva asignación de los grupos a los docentes en los diferentes horarios dependiendo de la disponibilidad de éstos.

Figura 48. Diagrama de Secuencia Administrar Carga Escenario Reiniciar Carga

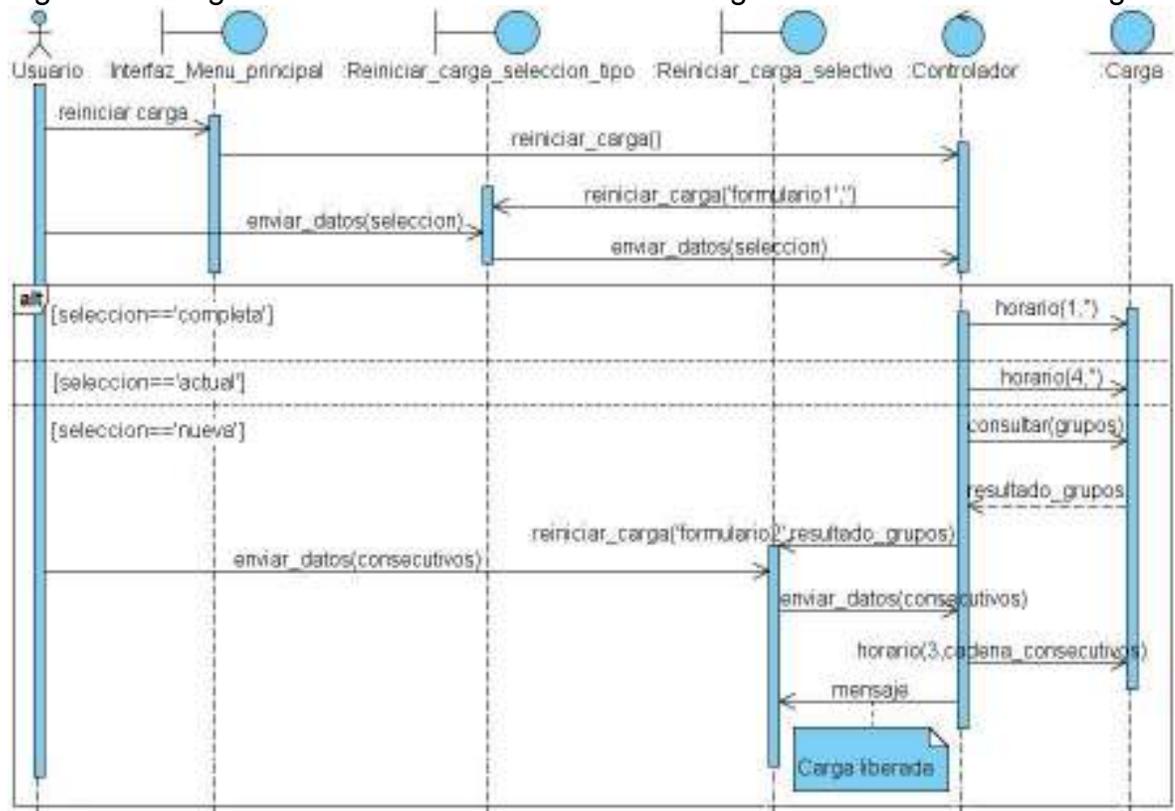


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Reiniciar Carga

La figura 49 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Reiniciar Carga. El administrador elige la opción Reiniciar Carga del submenú Administrar Carga y el sistema le permite elegir el tipo de reinicio (liberación completa, conservar última selección o realizar nueva selección) de la asignaciones previas que se hayan realizado.

Figura 49. Diagrama de Secuencia Administrar Carga Escenario Reiniciar Carga

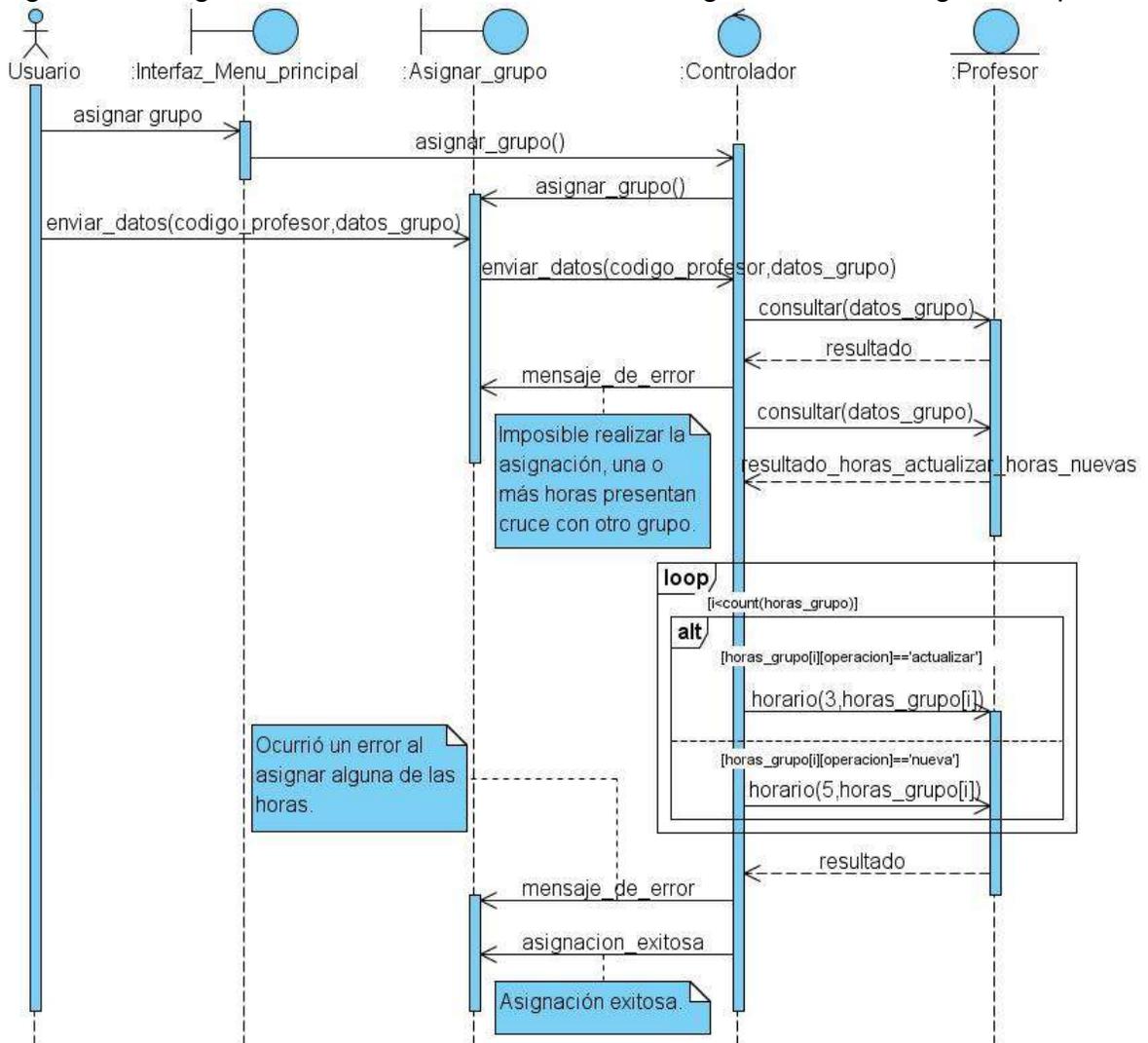


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Asignar Grupo

La figura 50 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Asignar Grupo. El administrador elige la opción Asignar Grupo del submenú Administrar Carga y el sistema le permite elegir el profesor y el grupo para realizar una adición manual sin tener consideraciones de perfil ni de disponibilidad del docente; únicamente se garantiza que dicho grupo no entre en conflicto con otro(s) grupo(s) que ya tenga(n) asignado(s) el profesor.

Figura 50. Diagrama de Secuencia Administrar Carga Escenario Asignar Grupo

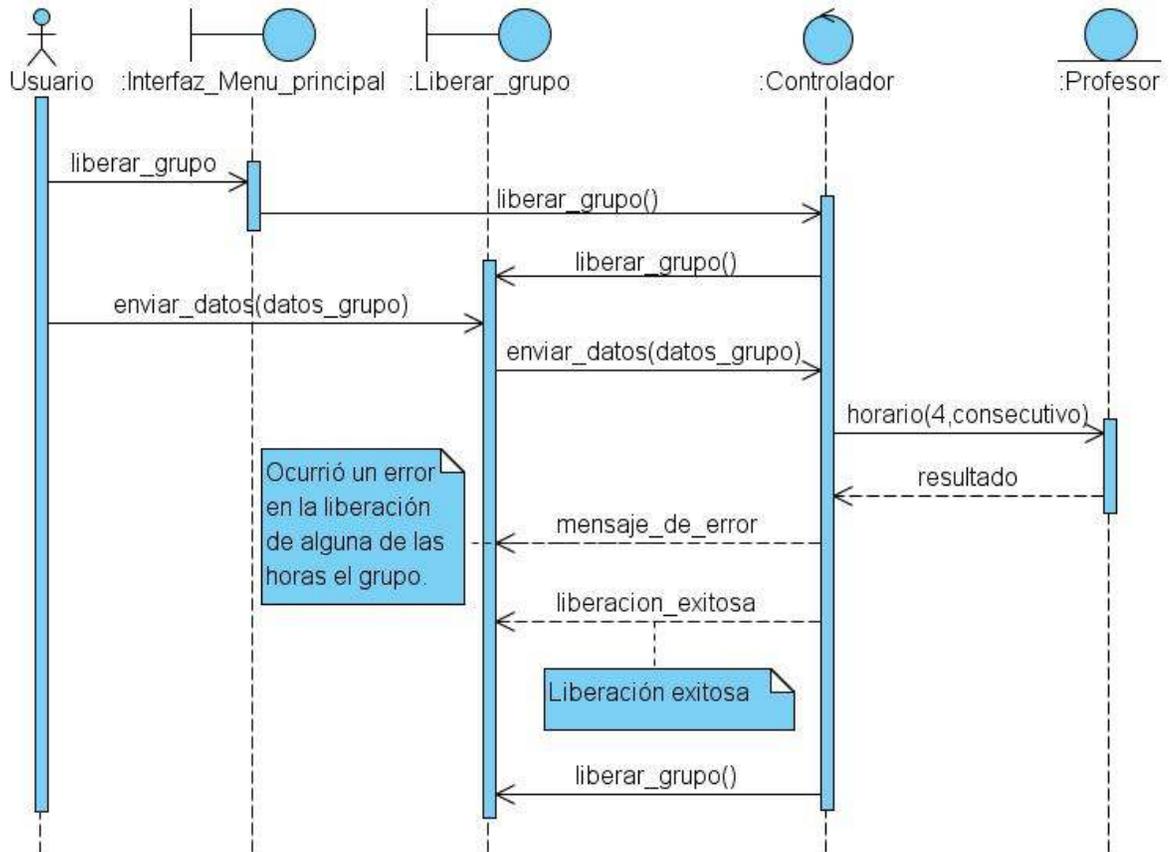


Fuente: los autores.

Diagrama de Secuencia Caso de Uso Administrar Carga Escenario Liberar Grupo

La figura 51 muestra la secuencia de eventos para el caso de uso Administrar Carga en el escenario Liberar Grupo. El usuario elige la opción Liberar Grupo del submenú Administrar Carga y el sistema le permite seleccionar un grupo de una asignatura para eliminarlo de la base de datos.

Figura 51. Diagrama de Secuencia Administrar Carga Escenario Liberar Grupo



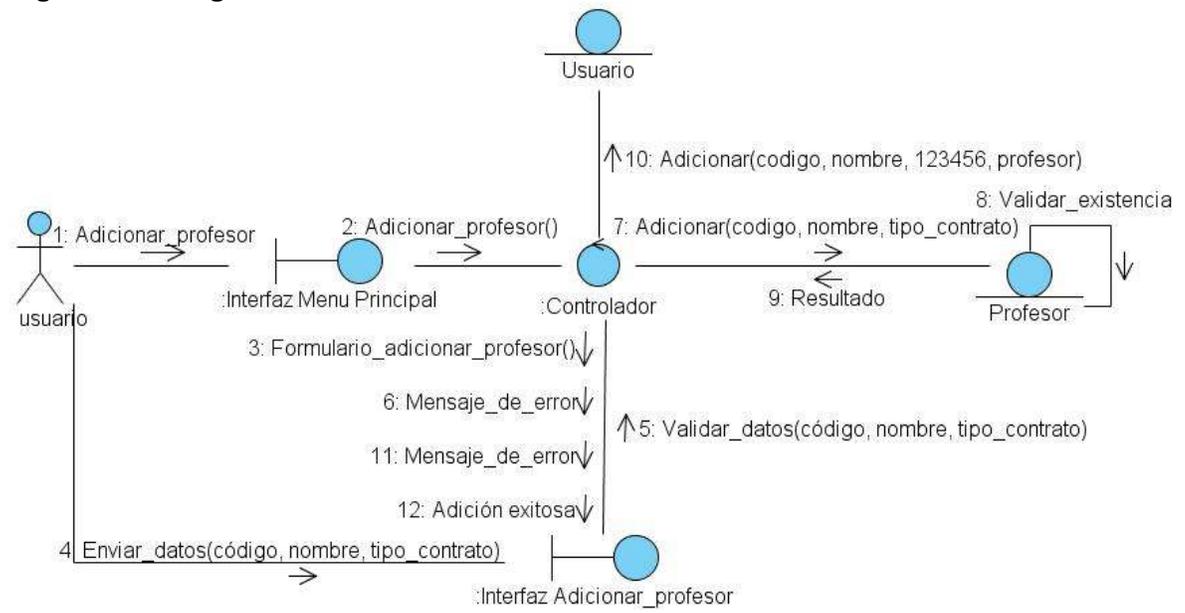
Fuente: los autores.

7.5.2. Diagramas de Colaboración

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Adicionar

La figura 52 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Adicionar.

Figura 52. Diagrama de Colaboración Administrar Profesores Escenario Adicionar

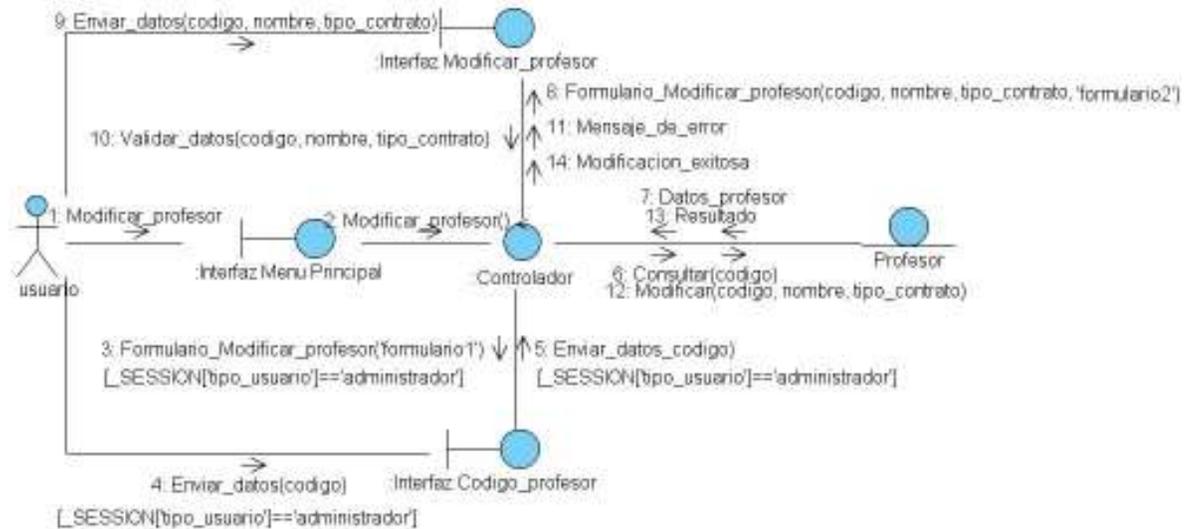


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Modificar

La figura 53 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Modificar.

Figura 53. Diagrama de Colaboración Administrar Profesores Escenario Modificar

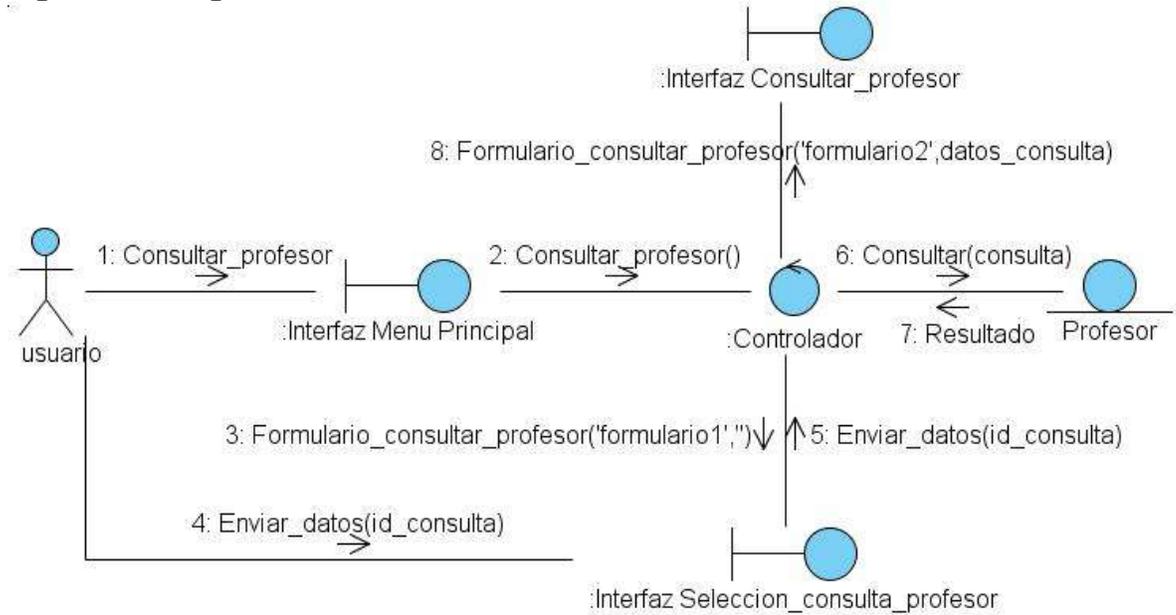


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Consultar

La figura 54 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Consultar.

Figura 54. Diagrama de Colaboración Administrar Profesores Escenario Consultar

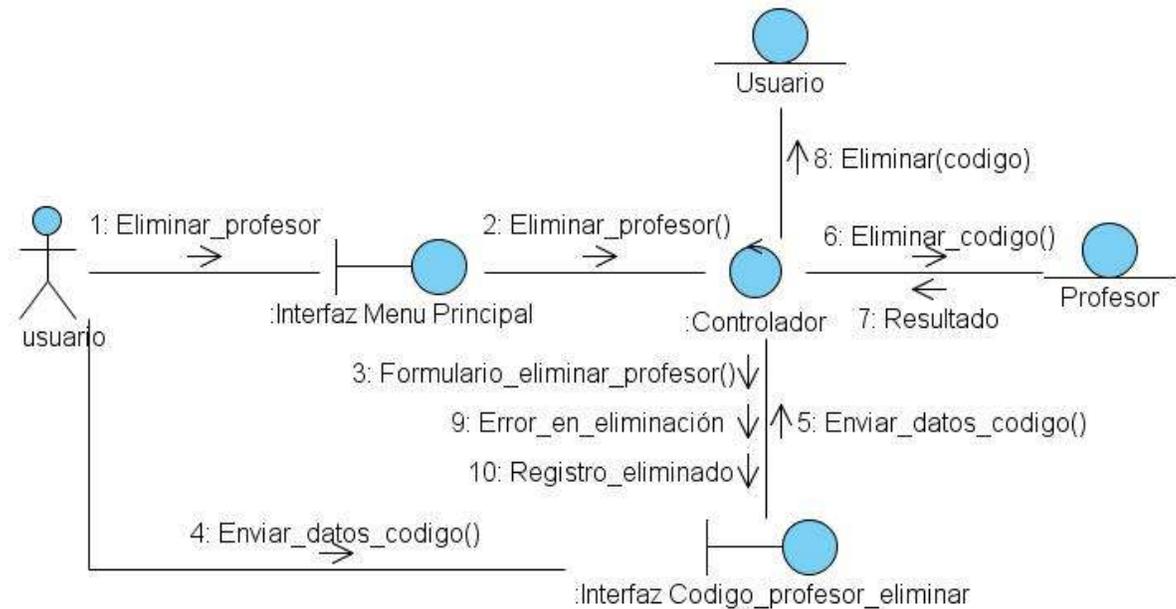


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Eliminar

La figura 55 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Eliminar.

Figura 55. Diagrama de Colaboración Administrar Profesores Escenario Eliminar

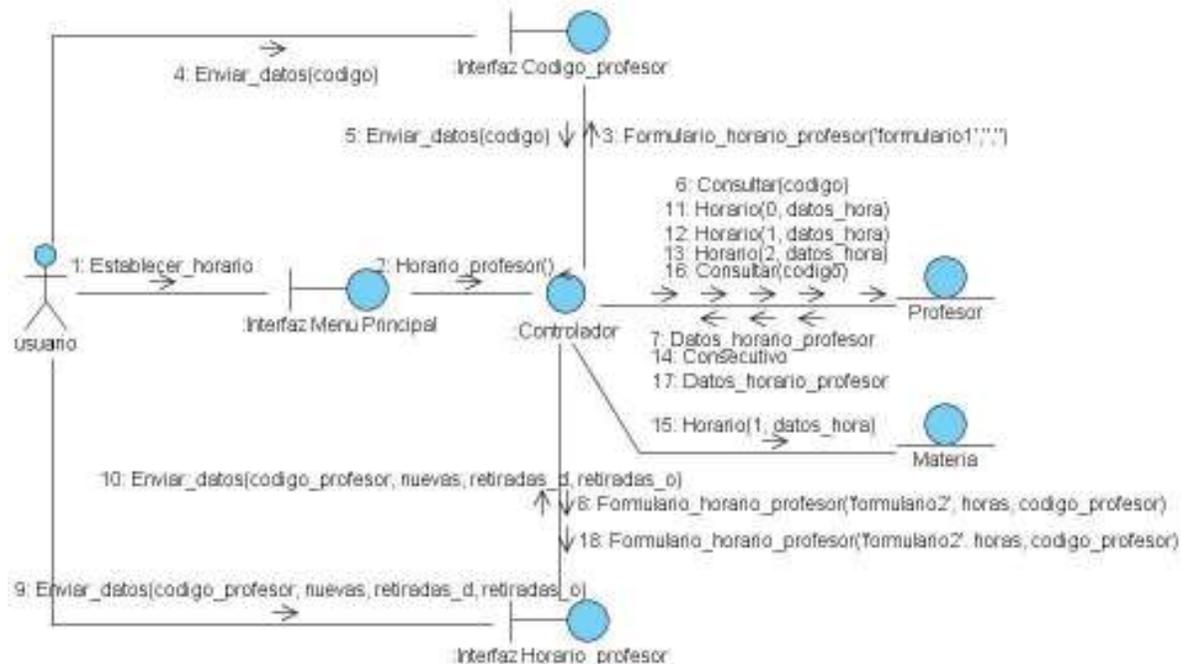


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Establecer Horario

La figura 56 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Establecer Horario.

Figura 56. Diagrama de Colaboración Administrar Profesores Escenario Establecer Horario

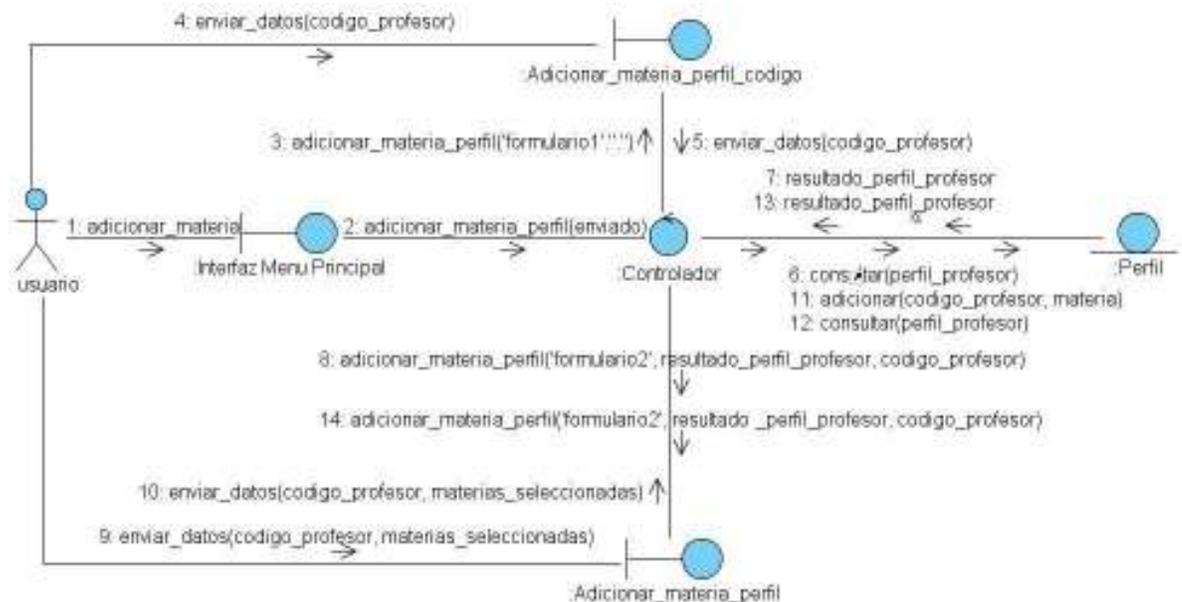


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Adicionar Materia

La figura 57 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Adicionar Materia.

Figura 57. Diagrama de Colaboración Administrar Profesores Escenario Adicionar Materia

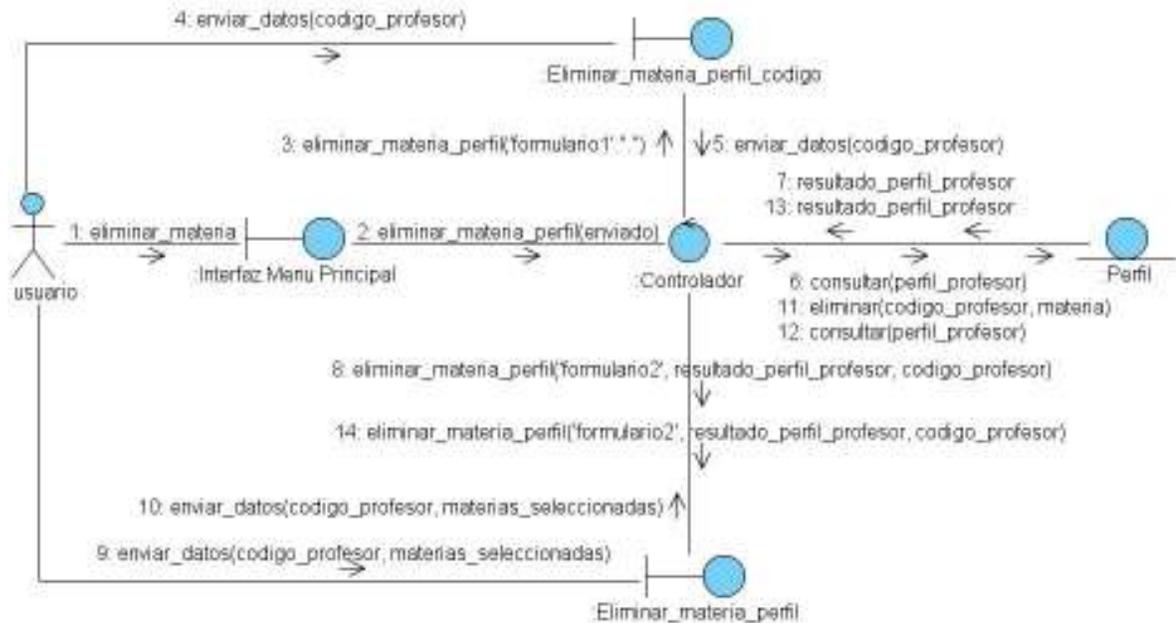


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Profesores Escenario Eliminar Materia

La figura 58 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Profesores en el escenario Eliminar Materia.

Figura 58. Diagrama de Colaboración Administrar Profesores Escenario Eliminar Materia

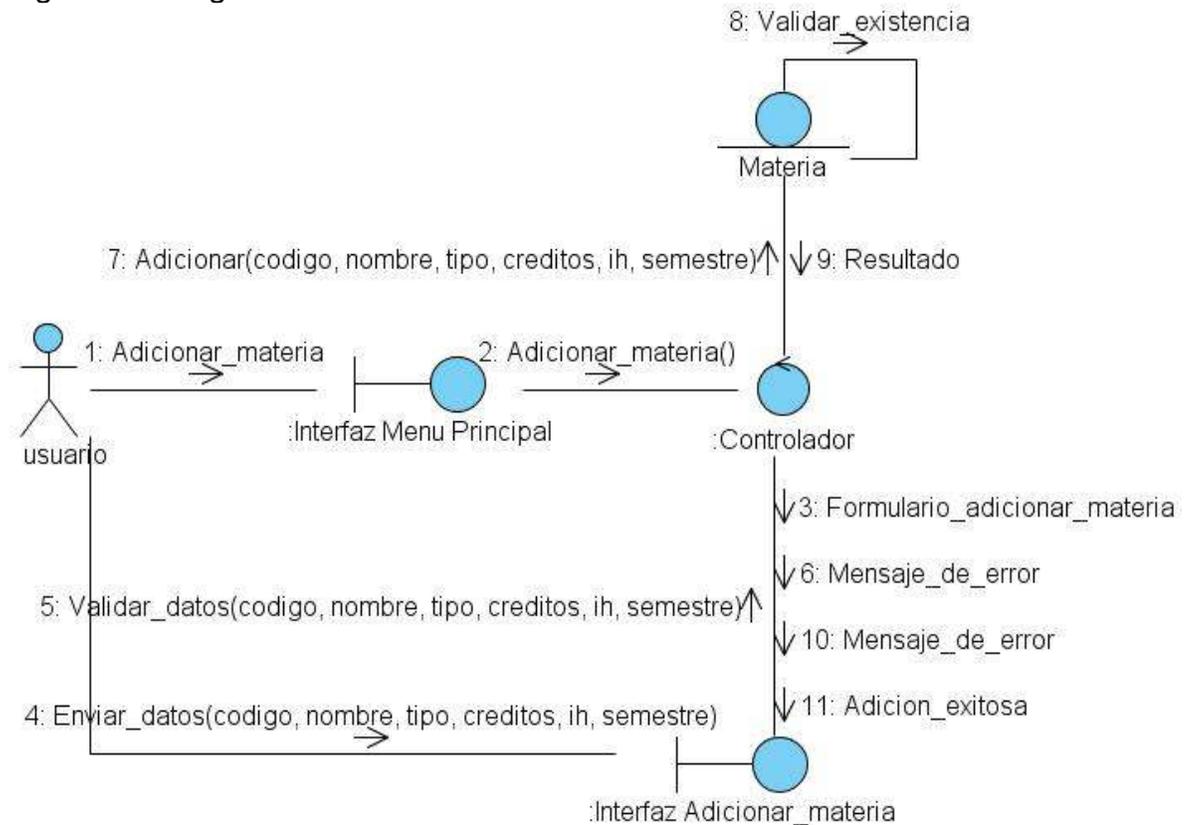


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Materias Escenario Adicionar

La figura 59 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Materias en el escenario Adicionar.

Figura 59. Diagrama de Colaboración Administrar Materias Escenario Adicionar

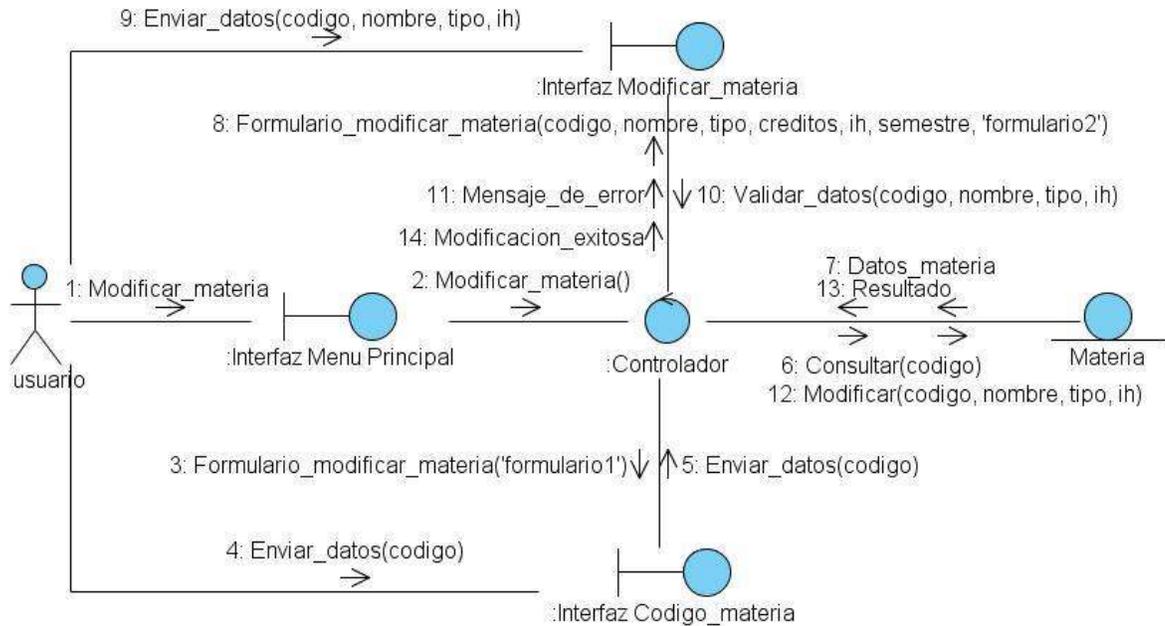


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Materias Escenario Modificar

La figura 60 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Materias en el escenario Modificar.

Figura 60. Diagrama de Colaboración Administrar Materias Escenario Modificar

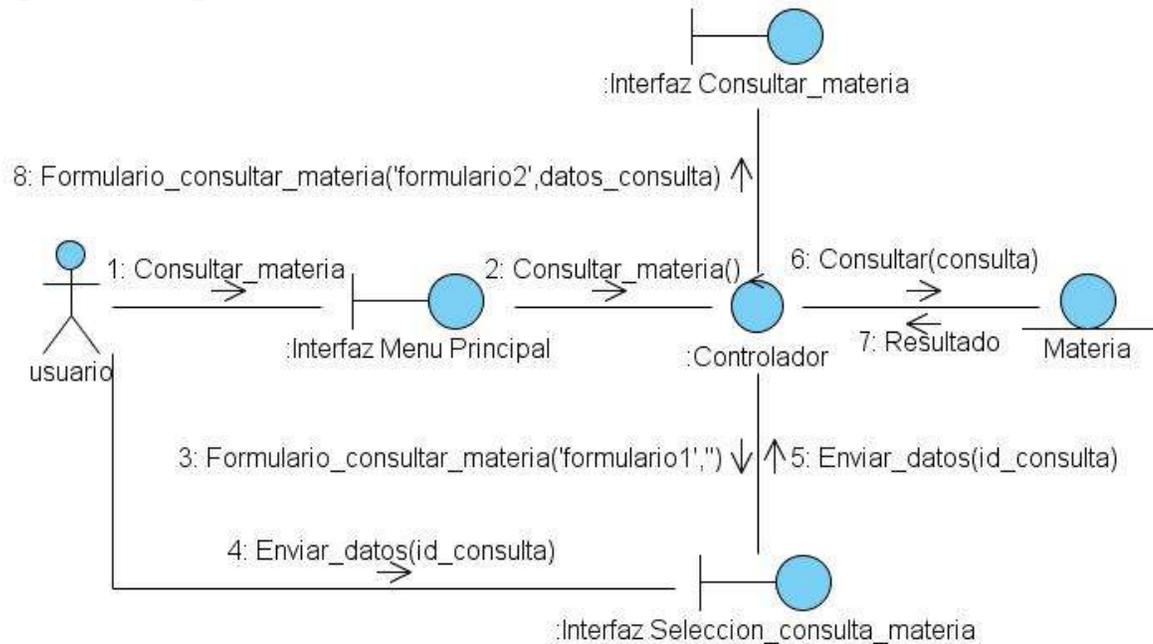


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Materias Escenario Consultar

La figura 61 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Materias en el escenario Consultar.

Figura 61. Diagrama de Colaboración Administrar Materias Escenario Consultar

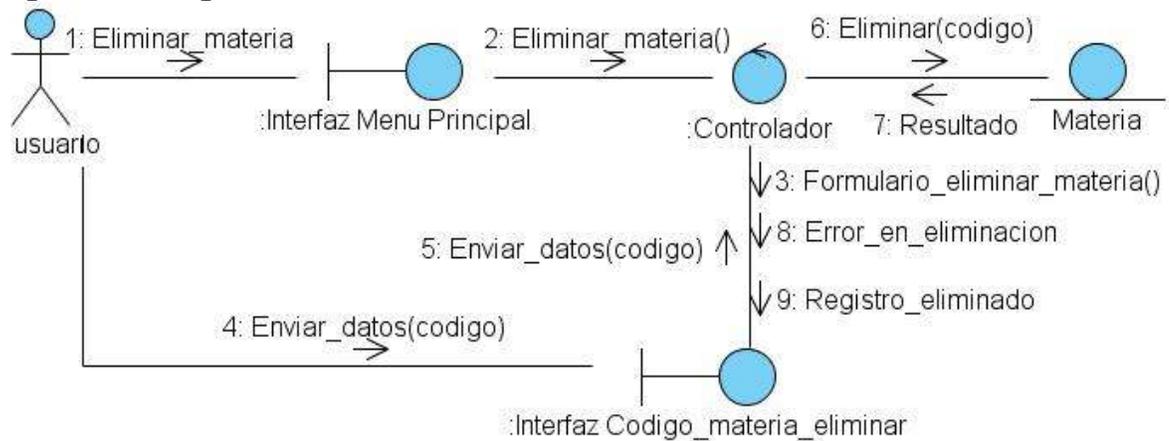


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Materias Escenario Eliminar

La figura 62 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Materias en el escenario Eliminar.

Figura 62. Diagrama de Colaboración Administrar Materias Escenario Eliminar

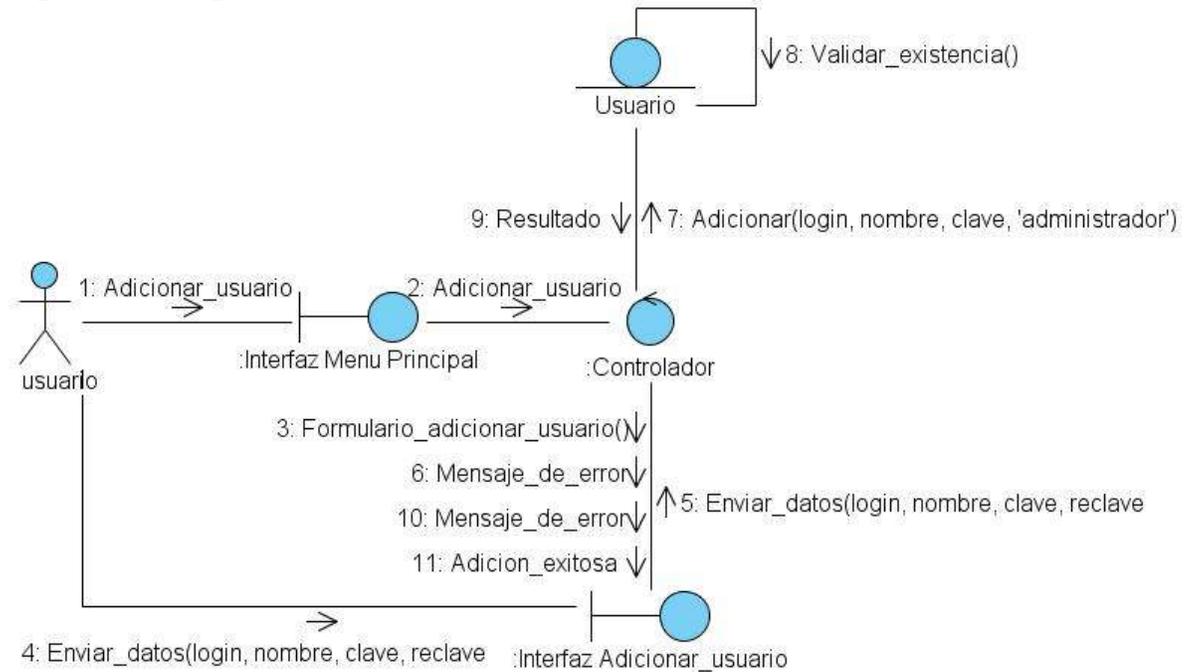


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Usuarios Escenario Adicionar

La figura 63 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Usuarios en el escenario Adicionar.

Figura 63. Diagrama de Colaboración Administrar Usuarios Escenario Adicionar

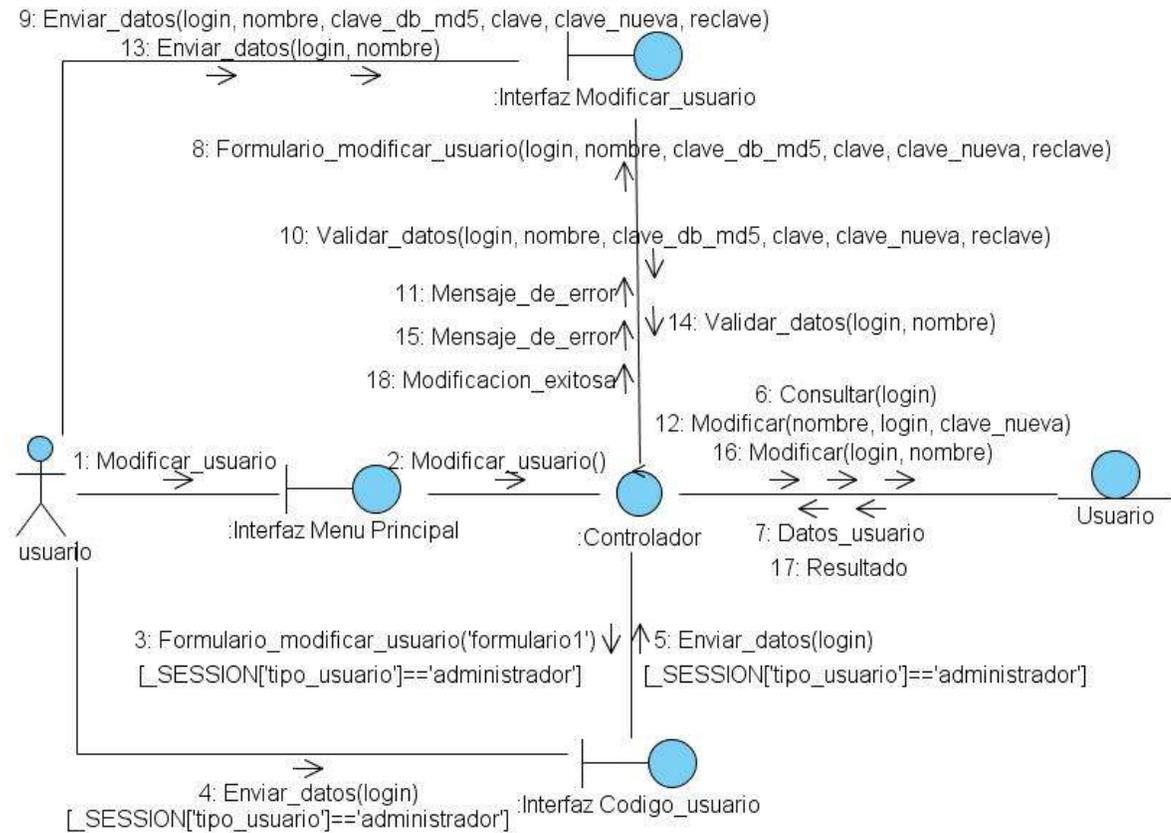


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Usuarios Escenario Modificar

La figura 64 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Usuarios en el escenario Modificar.

Figura 64. Diagrama de Colaboración Administrar Usuarios Escenario Modificar

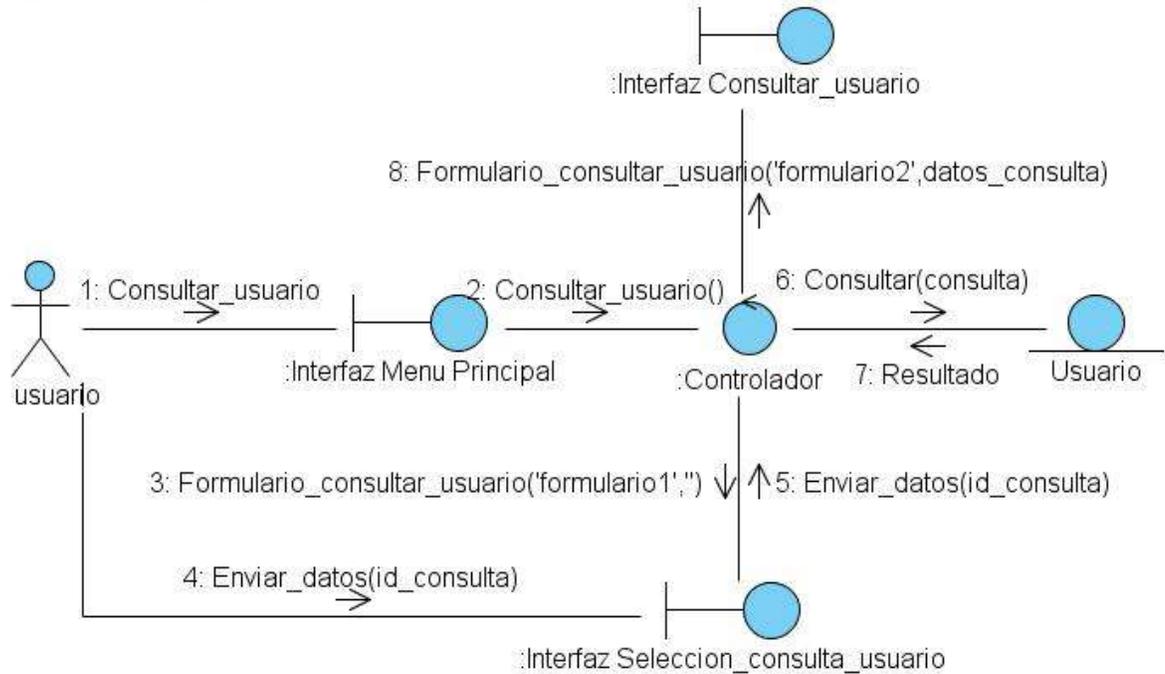


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Usuarios Escenario Consultar

La figura 65 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Usuarios en el escenario Consultar.

Figura 65. Diagrama de Colaboración Administrar Usuarios Escenario Consultar

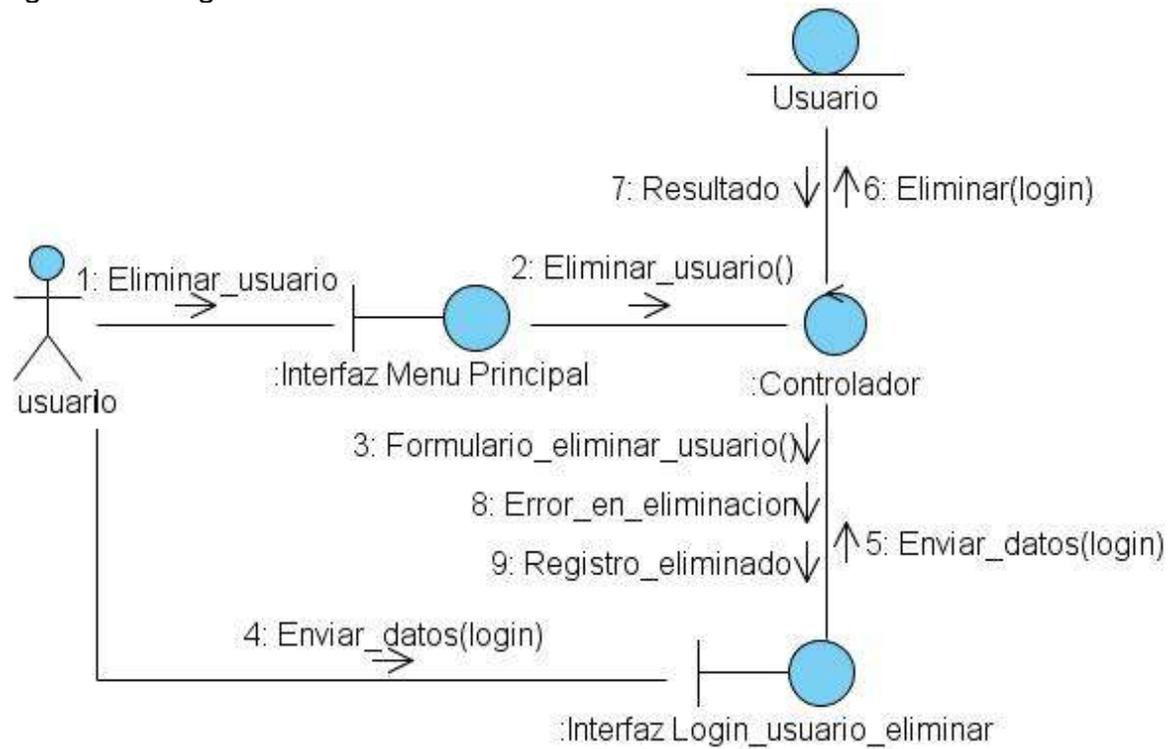


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Usuarios Escenario Eliminar

La figura 66 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Usuarios en el escenario Eliminar.

Figura 66. Diagrama de Colaboración Administrar Usuarios Escenario Eliminar

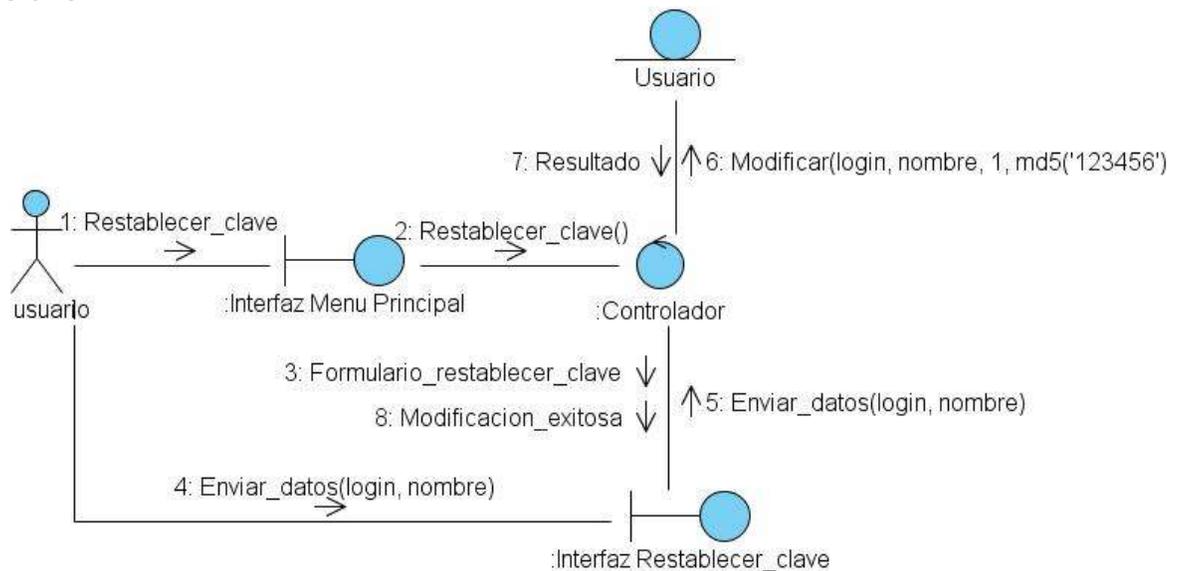


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Usuarios Escenario Restablecer Clave

La figura 67 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Usuarios en el escenario Restablecer Clave.

Figura 67. Diagrama de Colaboración Administrar Usuarios Escenario Restablecer Clave

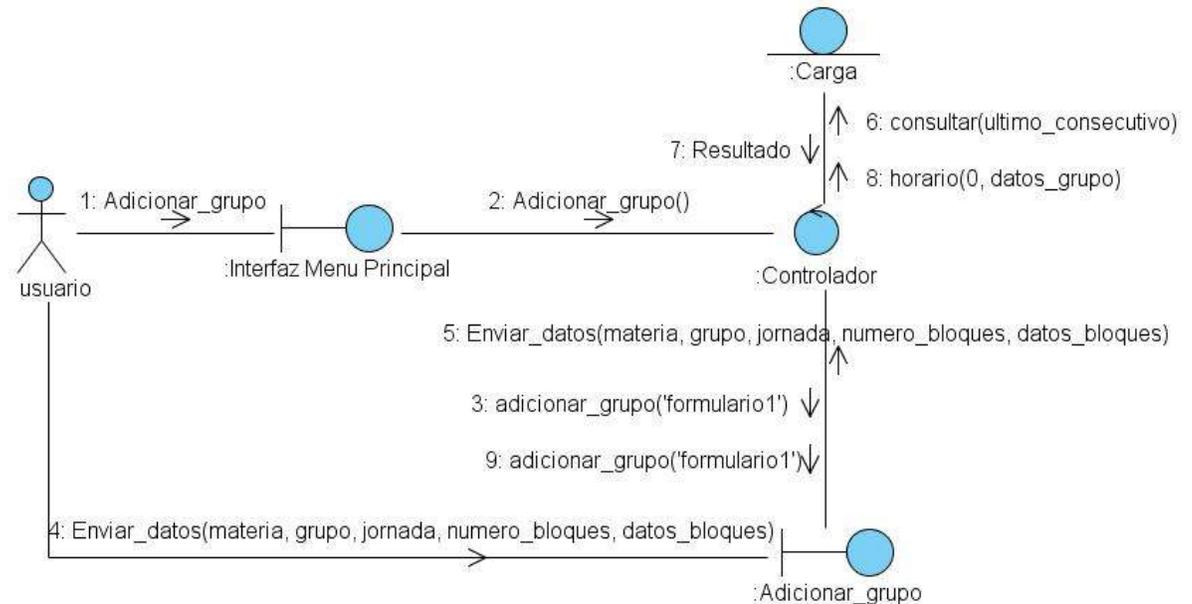


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Adicionar Grupo

La figura 68 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Adicionar Grupo.

Figura 68. Diagrama de Colaboración Administrar Carga Escenario Adicionar Grupo

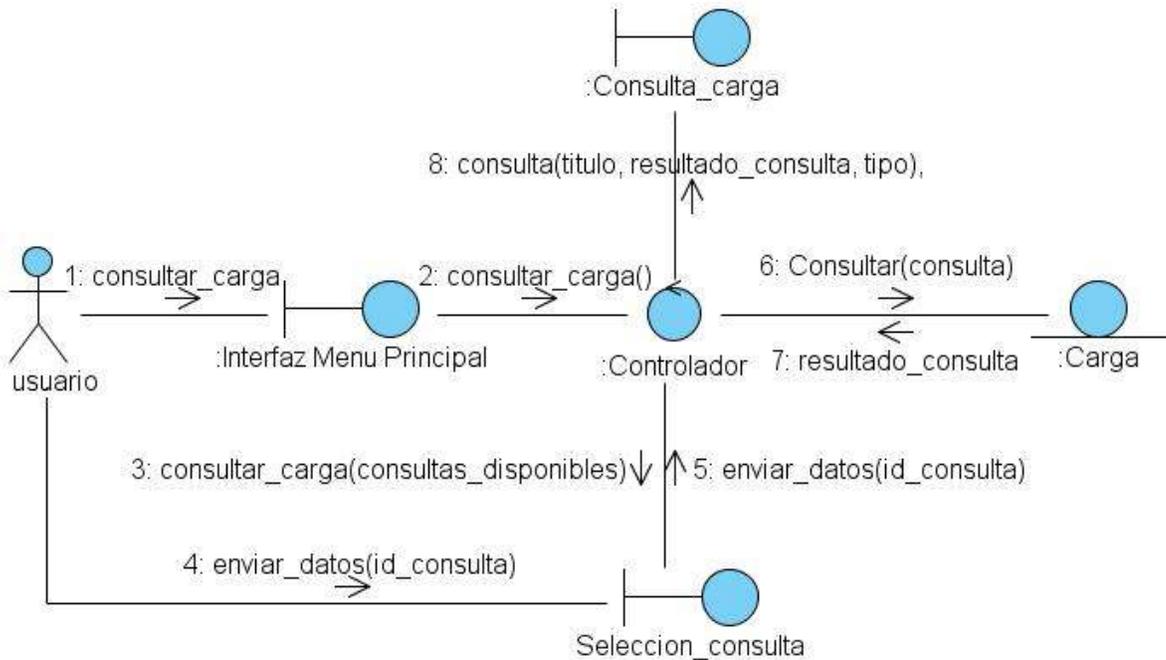


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Consultar

La figura 69 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Consultar.

Figura 69. Diagrama de Colaboración Administrar Carga Escenario Consultar

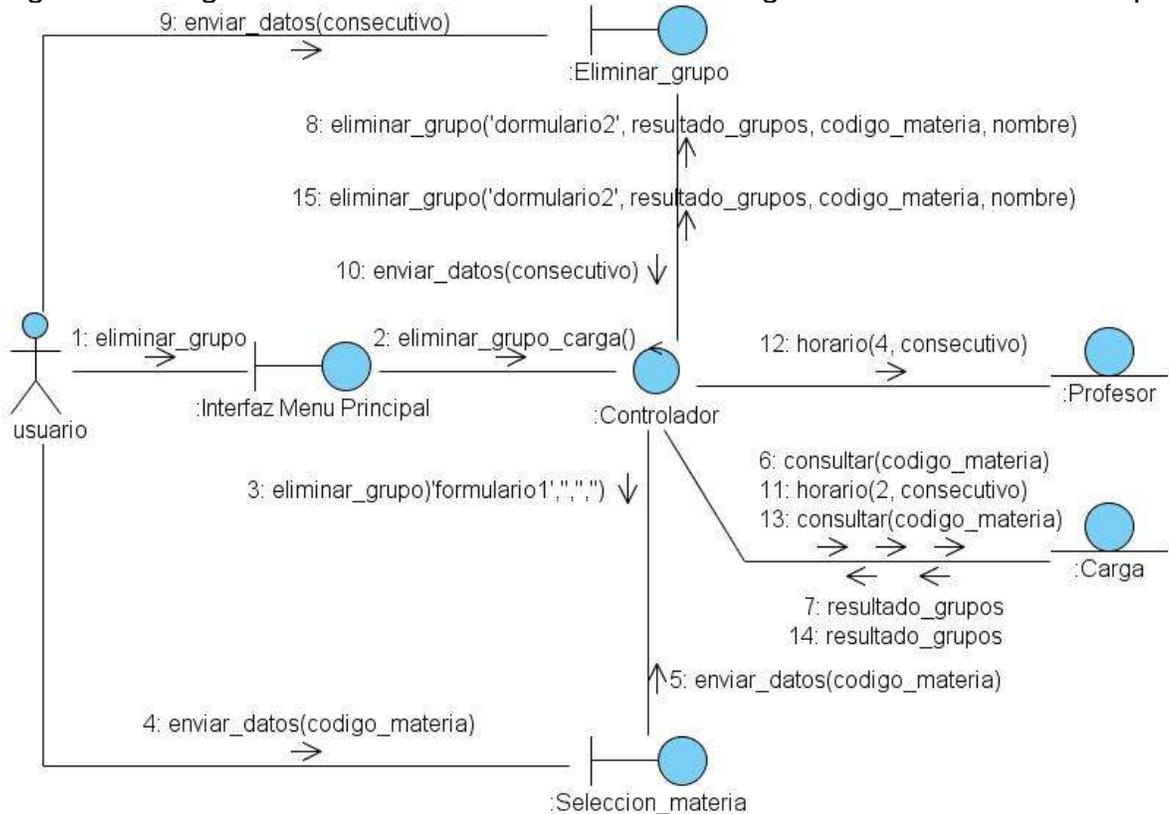


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Eliminar Grupo

La figura 70 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Eliminar Grupo.

Figura 70. Diagrama de Colaboración Administrar Carga Escenario Eliminar Grupo

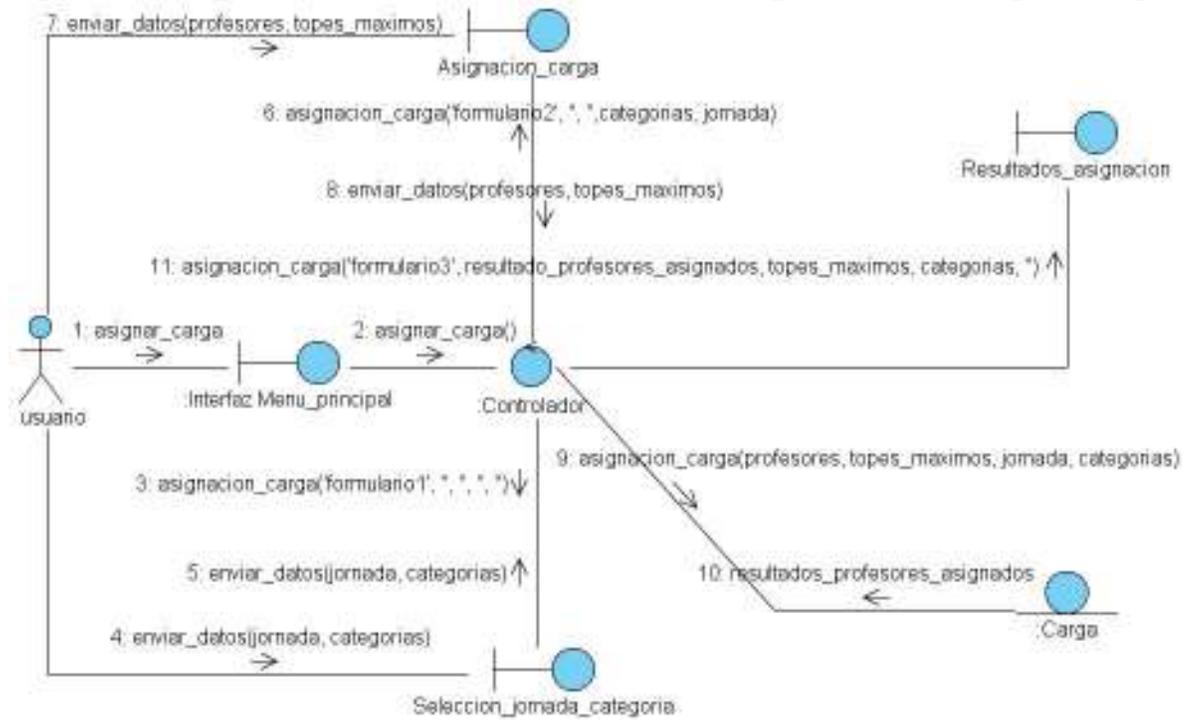


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Asignar Carga

La figura 71 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Asignar Carga.

Figura 71. Diagrama de Colaboración Administrar Carga Escenario Asignar Carga

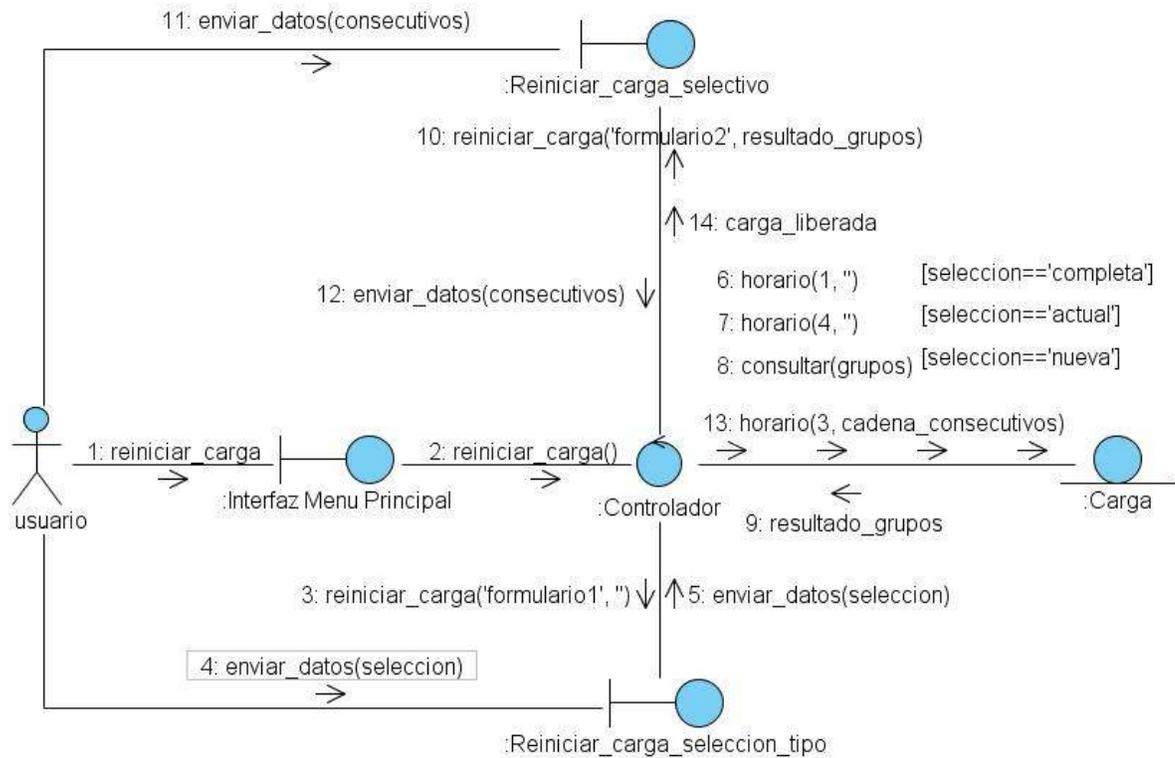


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Reiniciar Carga

La figura 72 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Reiniciar Carga.

Figura 72. Diagrama de Colaboración Administrar Carga Escenario Reiniciar Carga

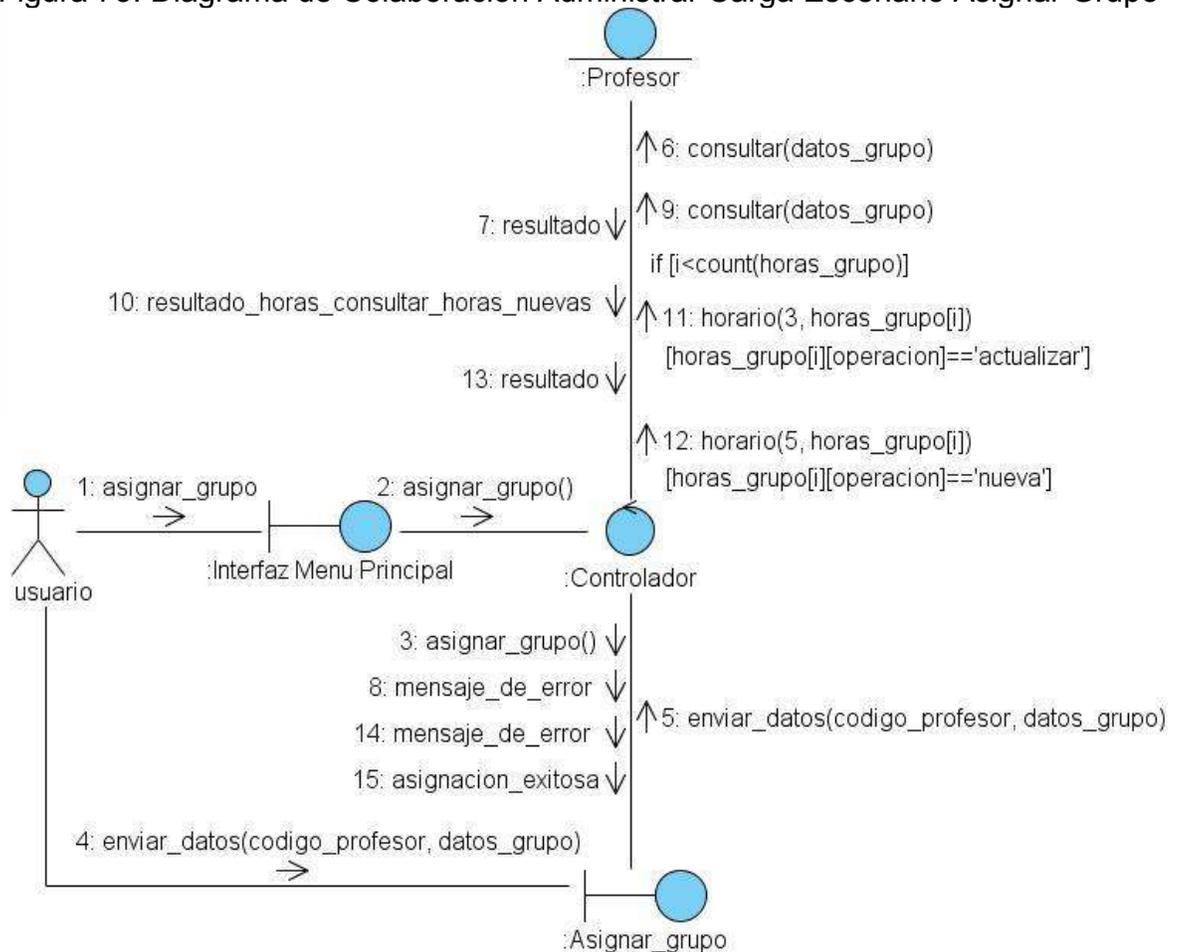


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Asignar Grupo

La figura 73 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Asignar Grupo.

Figura 73. Diagrama de Colaboración Administrar Carga Escenario Asignar Grupo

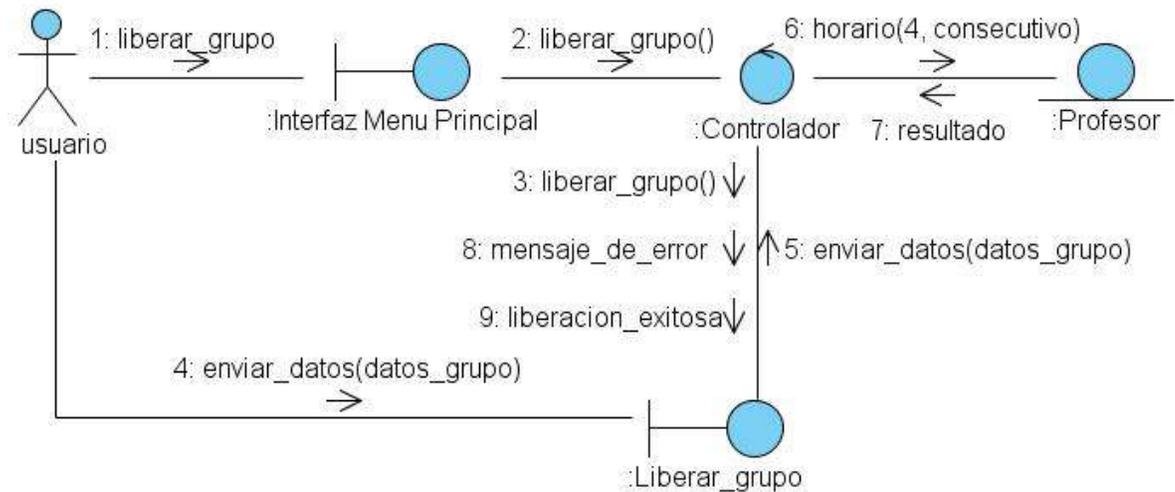


Fuente: los autores.

Diagrama de Colaboración Caso de Uso Administrar Carga Escenario Liberar Grupo

La figura 74 muestra las relaciones entre los roles de los objetos, la secuencia de los mensajes y los flujos de interacción generados por el caso de uso Administrar Carga en el escenario Liberar Grupo.

Figura 74. Diagrama de Colaboración Administrar Carga Escenario Liberar Grupo

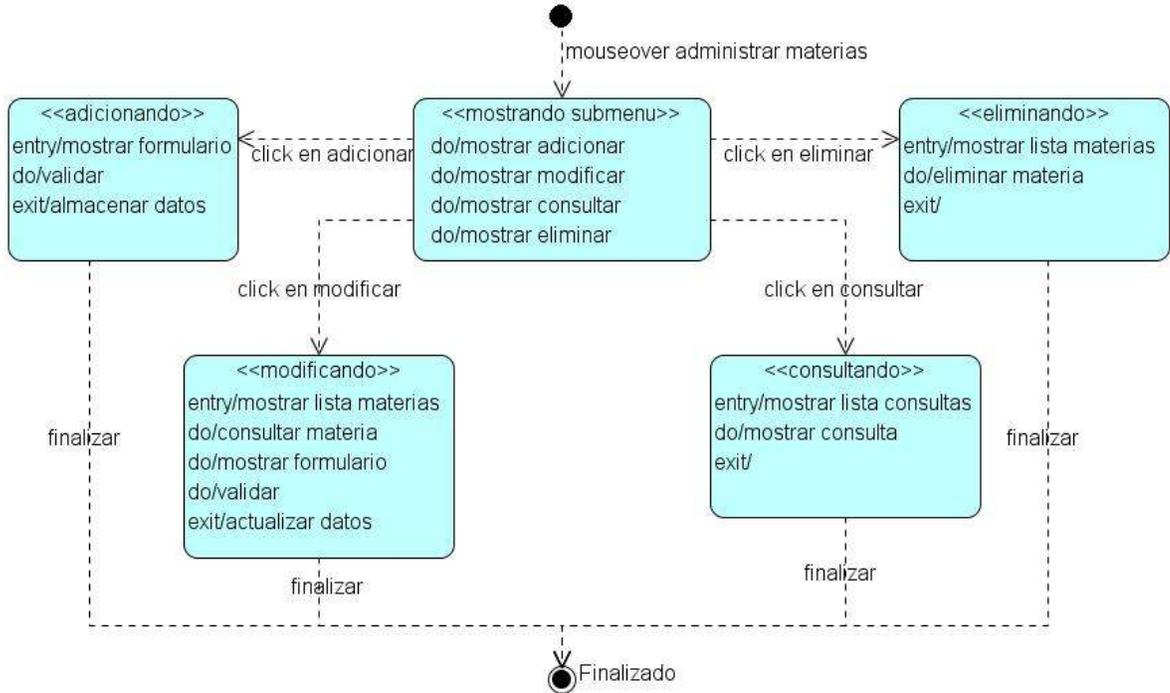


Fuente: los autores.

Diagrama de Estados Caso de Uso Administrar Materias

La figura 76 muestra la secuencia de estados por los que pasa el caso de uso Administrar Materias en sus diferentes escenarios, los eventos que hacen que se pase de un estado a otro y las respuestas y acciones que se generan.

Figura 76. Diagrama de Estados Caso de Uso Administrar Materias

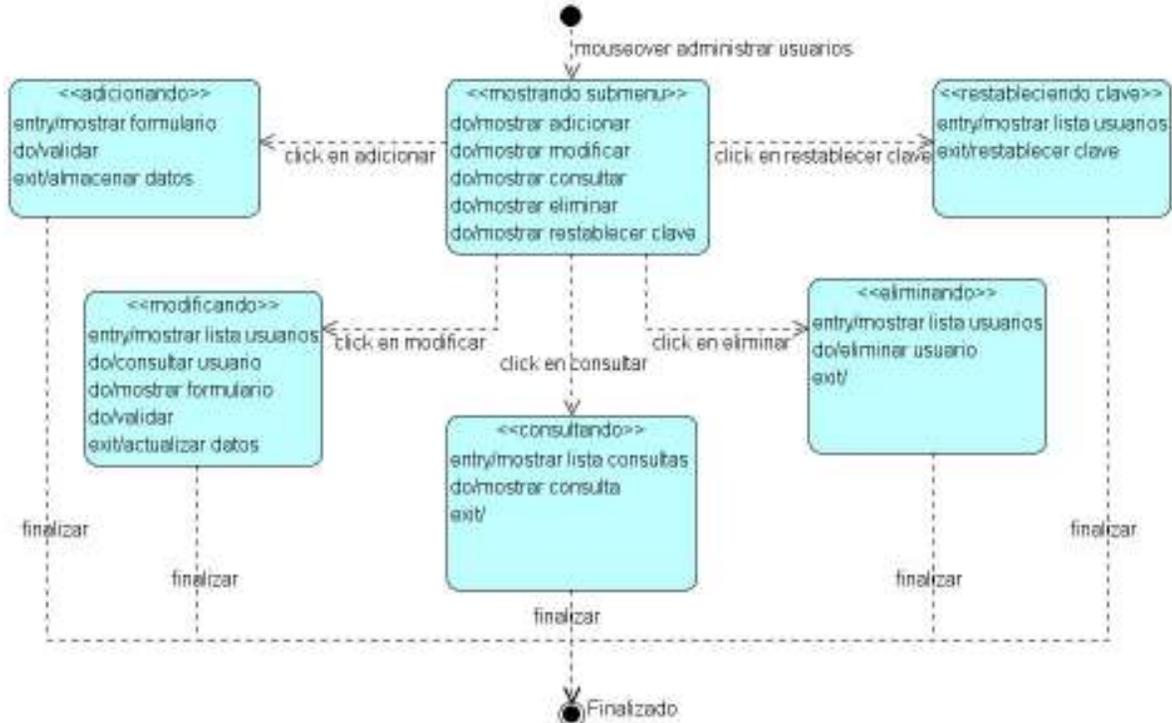


Fuente: los autores.

Diagrama de Estados Caso de Uso Administrar Usuarios

La figura 77 muestra la secuencia de estados por los que pasa el caso de uso Administrar Usuarios en sus diferentes escenarios, los eventos que hacen que se pase de un estado a otro y las respuestas y acciones que se generan.

Figura 77. Diagrama de Estados Caso de Uso Administrar Usuarios

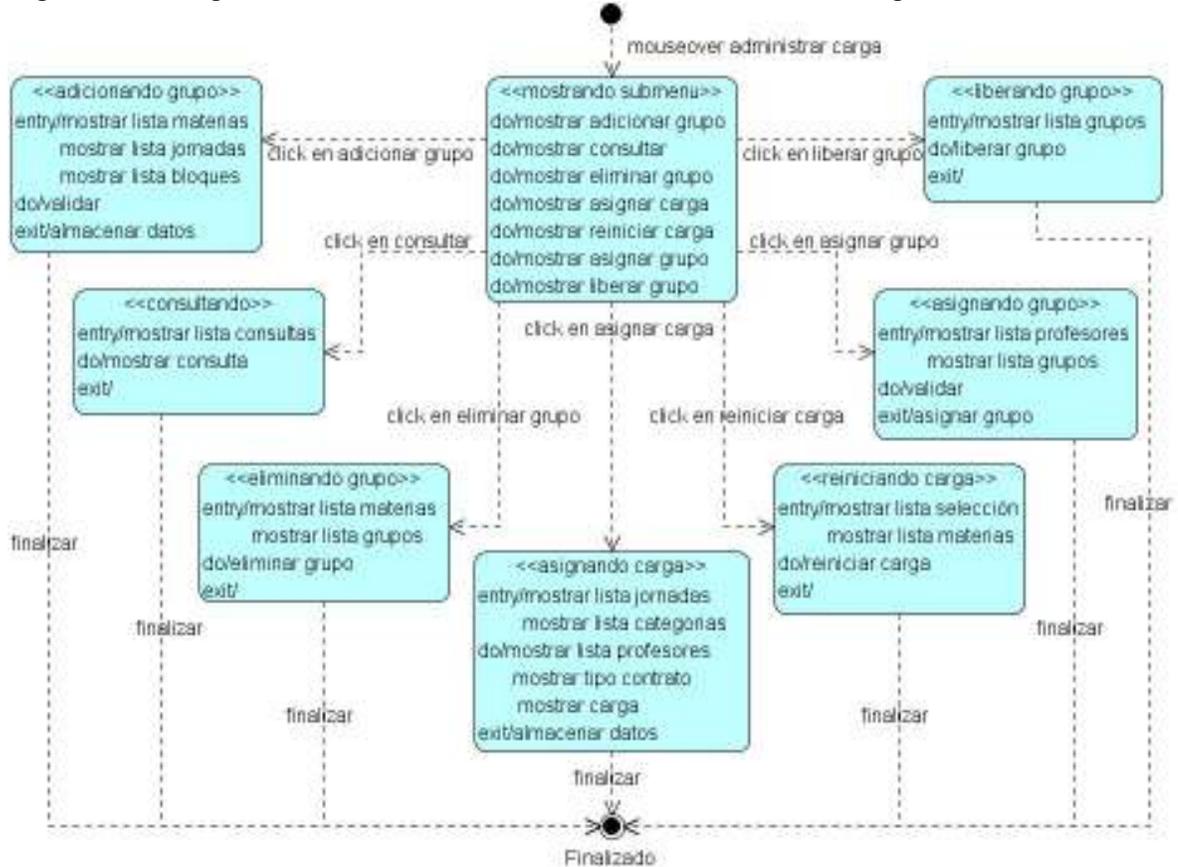


Fuente: los autores.

Diagrama de Estados Caso de Uso Administrar Carga

La figura 78 muestra la secuencia de estados por los que pasa el caso de uso Administrar Carga en sus diferentes escenarios, los eventos que hacen que se pase de un estado a otro y las respuestas y acciones que se generan.

Figura 78. Diagrama de Estados Caso de Uso Administrar Carga



Fuente: los autores.

7.7. MODELO FUNCIONAL

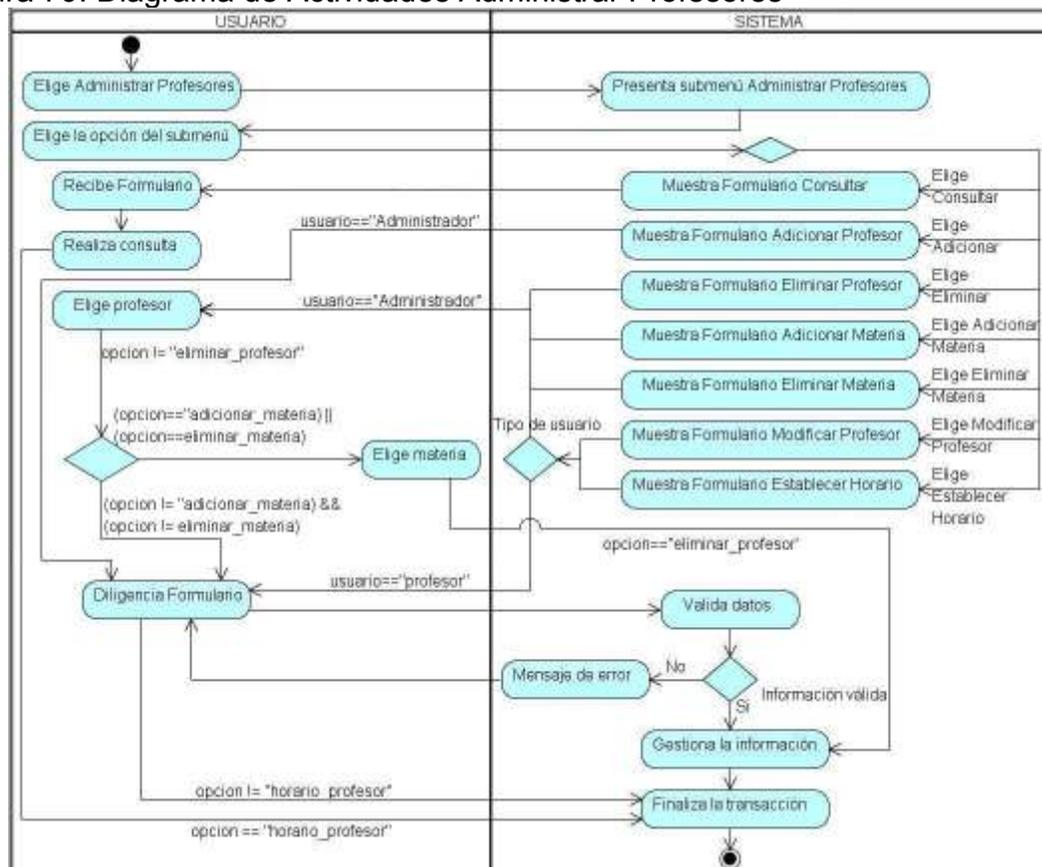
A continuación se ilustrarán la serie de acciones que deben ser realizadas para cada caso de uso y la manera como puede reaccionar el sistema ante determinados eventos.

7.7.1. Diagramas de Actividades

Diagrama de Actividades Caso de Uso Administrar Profesores

La figura 79 muestra la serie de actividades que deben ser realizadas en el caso de uso Administrar Profesores en sus diferentes escenarios, así como las distintas rutas que pueden irse desencadenando en ellos.

Figura 79. Diagrama de Actividades Administrar Profesores

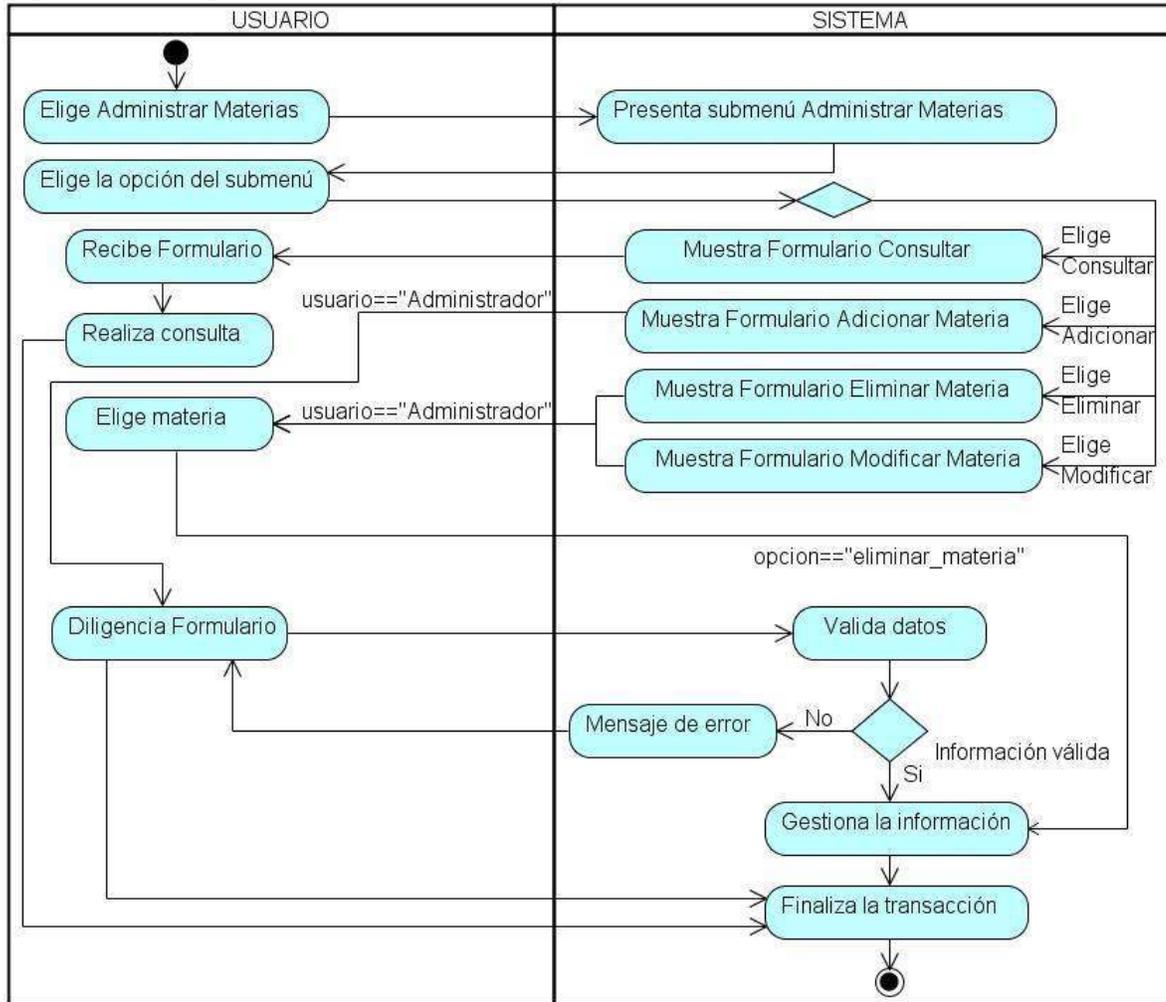


Fuente: los autores.

Diagrama de Actividades Caso de Uso Administrar Materias

La figura 80 muestra la serie de actividades que deben ser realizadas en el caso de uso Administrar Materias en sus diferentes escenarios, así como las distintas rutas que pueden irse desencadenando en ellos.

Figura 80. Diagrama de Actividades Administrar Materias

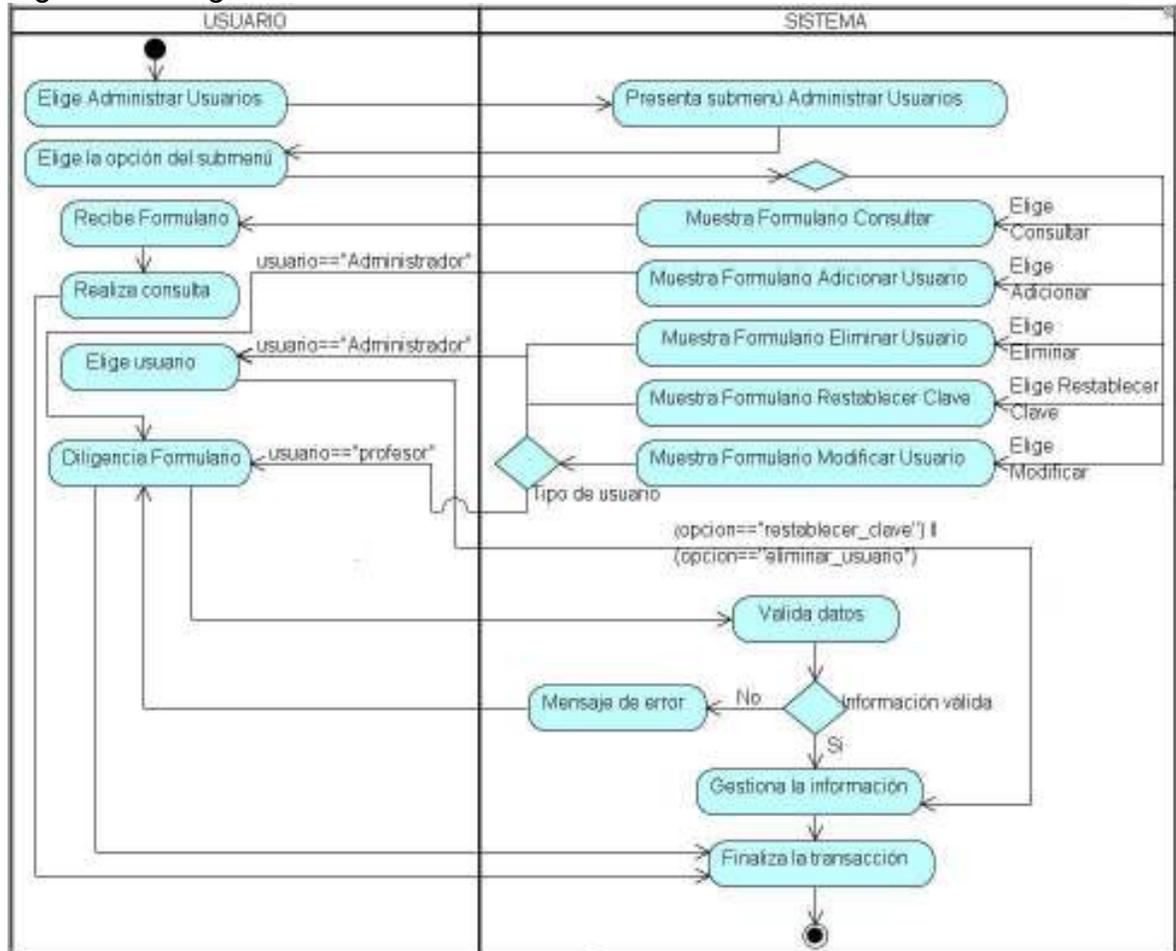


Fuente: los autores.

Diagrama de Actividades Caso de Uso Administrar Usuarios

La figura 81 muestra la serie de actividades que deben ser realizadas en el caso de uso Administrar Usuarios en sus diferentes escenarios, así como las distintas rutas que pueden irse desencadenando en ellos.

Figura 81. Diagrama de Actividades Administrar Usuarios

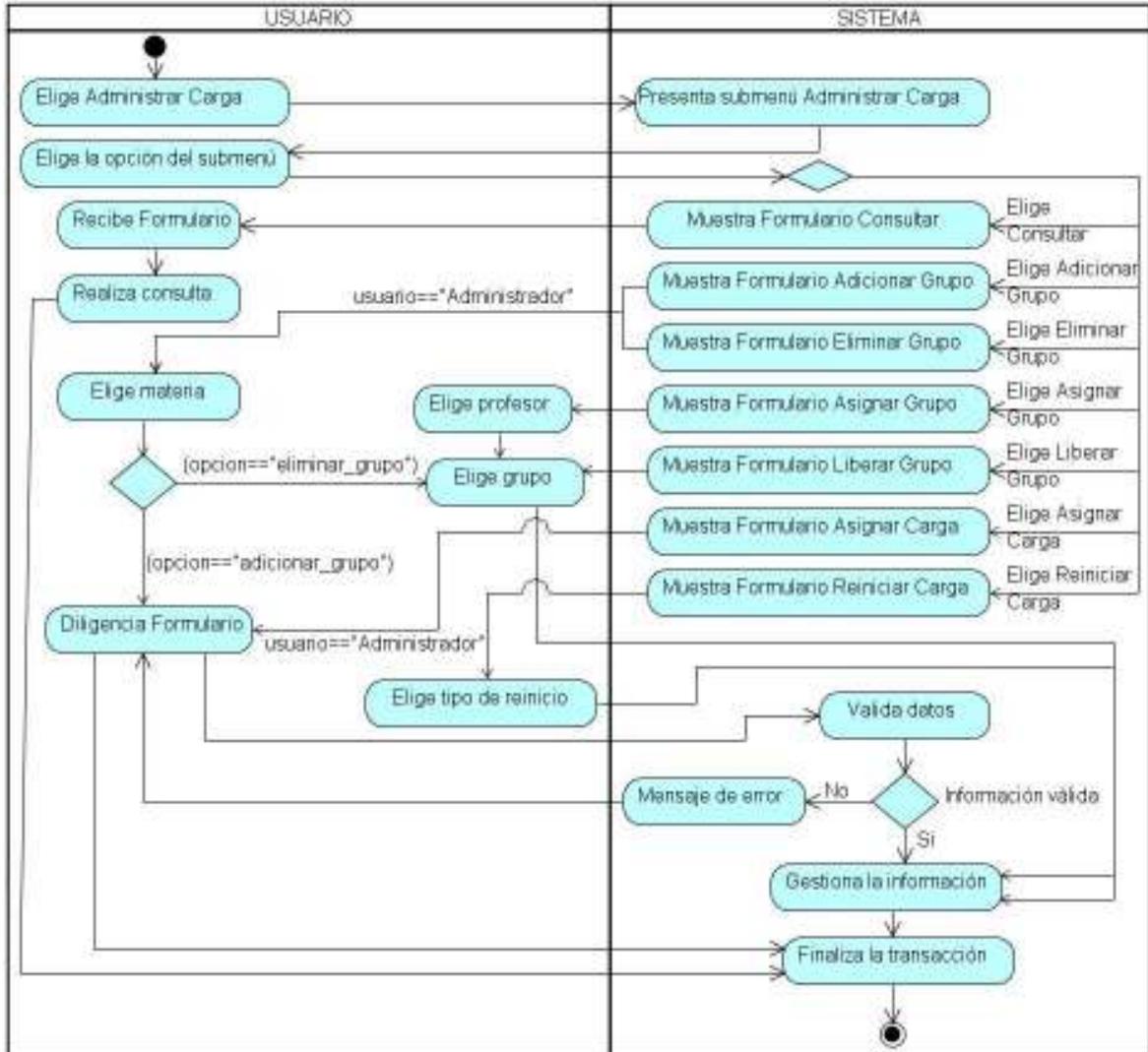


Fuente: los autores.

Diagrama de Actividades Caso de Uso Administrar Carga

La figura 82 muestra la serie de actividades que deben ser realizadas en el caso de uso Administrar Carga en sus diferentes escenarios, así como las distintas rutas que pueden irse desencadenando en ellos.

Figura 82. Diagrama de Actividades Administrar Carga



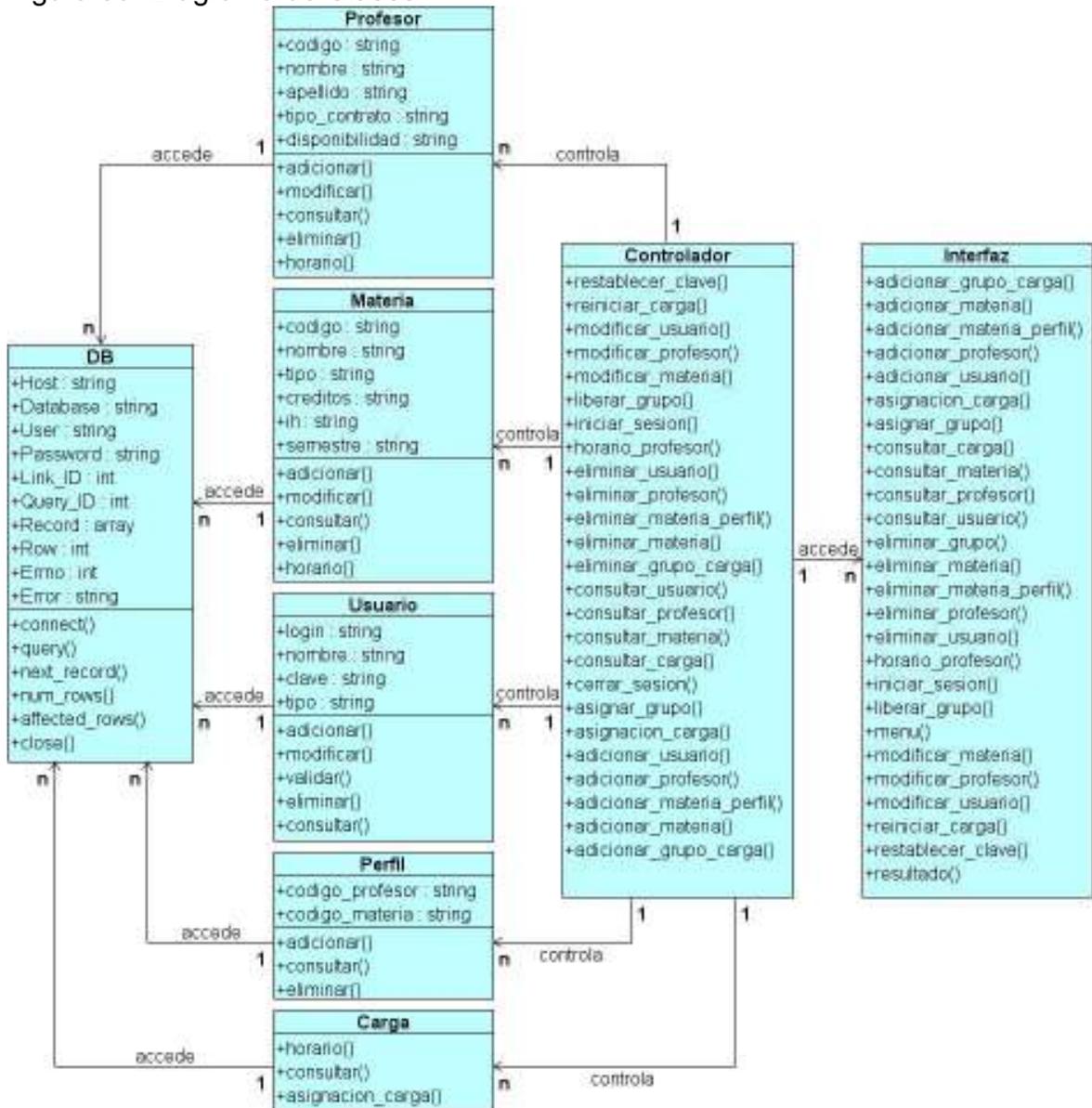
Fuente: los autores.

7.8. MODELO FINAL DE OBJETOS

7.8.1. Diagrama de Clases

La figura 83 muestra las relaciones entre las clases involucradas en el sistema (profesor, materia, usuario, perfil, carga, interfaz, controlador y db).

Figura 83. Diagrama de Clases



Fuente: los autores.

7.9. DICCIONARIO DE DATOS

7.9.1. Diccionario de clases

En esta parte se describen todas las clases que componen el sistema, especificando de cada una su nombre, tipo, y descripción.

Tabla 27. Diccionario de Clases

Nombre	Tipo	Descripción
DB	Pública	Se encarga de gestionar todas las transacciones con la base de datos. Es la que hace uso de los métodos propios del lenguaje de desarrollo para realizar operaciones de conexión y queries.
Profesor	Pública	Hace uso de la clase DB para insertar, modificar, consultar y eliminar registros en lo que a profesores se refiere, como su información personal y su horario.
Materia	Pública	Hace uso de la clase DB para insertar, modificar, consultar y eliminar registros en lo que respecta a materias y a grupos.
Usuario	Pública	Hace uso de la clase DB para insertar, modificar, consultar y eliminar registros referentes a usuarios, tiene participación en la validación de los mismos en el momento de ingreso al sistema.
Perfil	Pública	Hace uso de la clase DB para insertar, consultar y eliminar registros en lo que respecta a perfiles y sus respectivas materias.
Carga	Pública	La clase Carga se encarga de gestionar las operaciones relacionadas con la asignación de la carga.
Controlador	Pública	La clase controlador es la columna vertebral de la aplicación, posee métodos para controlar todos y cada uno de los procesos que se realizan en el sistema.
Interfaz	Pública	Como su nombre lo indica, esta clase se encarga de generar todas las interfaces con las cuales va a interactuar el usuario final, constituye el primer filtro de errores pues cuenta en donde las características lo permiten, con campos en donde solo puede escoger de entre algunas opciones, lo que contribuye a la consistencia, a la coherencia y a la integridad referencial de la información.

Fuente: los autores.

7.9.2. Diccionario de atributos de clase

- **Diccionario de atributos de la clase DB**

Tabla 28. Diccionario de atributos de la clase DB

Nombre	Tipo	Visibilidad	Descripción
Host	String	Público	Este atributo sirve para almacenar el nombre del host del servidor sql que alberga la base de datos, para nuestro caso es el mismo que alberga la aplicación.
Database	String	Público	Este atributo sirve para almacenar el nombre de la base de datos que contiene la información del sistema.
User	String	Público	Este atributo sirve para almacenar el usuario con el cual se va a conectar a la base de datos.
Password	String	Público	Este atributo sirve para almacenar la clave para el usuario y la base de datos antes mencionados.
Link_ID	Integer	Público	Este atributo sirve para almacenar el resultado de la función <code>mysql_connect()</code> .
Query_ID	Integer	Público	Este atributo sirve para almacenar el resultado del último llamado a la función <code>mysql_query()</code> .
Record	Array	Público	Este atributo sirve para almacenar el último resultado de la función <code>mysql_fetch_array()</code> .
Row	Integer	Público	Este atributo sirve para almacenar el actual número de fila o de registro del query actual.
Errno	Integer	Público	Este atributo sirve para almacenar el código de error de query en caso de presentarse.
Error	String	Público	Este atributo sirve para almacenar la Descripción del error correspondiente al código anterior.

Fuente: los autores.

- **Diccionario de atributos de la clase Profesor**

Tabla 29. Diccionario de atributos de la clase Profesor

Nombre	Tipo	Visibilidad	Descripción
codigo	String	Público	Este atributo sirve para almacenar el código del profesor.
nombre	String	Público	Este atributo sirve para almacenar el nombre del profesor.
apellido	String	Público	Este atributo sirve para almacenar el apellido del profesor.
tipo_contrato	String	Público	Este atributo sirve para almacenar el tipo de contrato que el profesor tiene con la universidad.
disponibilidad	String	Público	Ese atributo informa al sistema de la jornada en la cual el profesor esta habilitado para trabajar.

Fuente: los autores.

- **Diccionario de atributos de la clase Materia**

Tabla 30. Diccionario de atributos de la clase Materia

Nombre	Tipo	Visibilidad	Descripción
codigo	String	Público	Este atributo sirve para almacenar el código de la materia según el pénsum vigente.
nombre	String	Público	Este atributo sirve para almacenar el nombre de la materia según el pénsum vigente.
Tipo	String	Público	Este atributo sirve para almacenar de que tipo es la materia según el pénsum vigente.
creditos	String	Público	Este atributo sirve para almacenar el número de créditos de la materia según el pénsum vigente.
lh	String	Público	Este atributo sirve para almacenar la intensidad horaria semanal de la materia según el pénsum vigente.
semestre	String	Público	Este atributo sirve para almacenar a que semestre pertenece la materia según el pénsum vigente.

Fuente: los autores.

- **Diccionario de atributos de la clase Usuario**

Tabla 31. Diccionario de atributos de la clase Usuario

Nombre	Tipo	Visibilidad	Descripción
login	String	Público	Este atributo sirve para almacenar el login del usuario.
nombre	String	Público	Este atributo sirve para almacenar el nombre del usuario.
clave	String	Público	Este atributo sirve para almacenar la clave del usuario.
tipo	String	Público	Este atributo sirve para almacenar el tipo de usuario.

Fuente: los autores.

- **Diccionario de atributos de la clase Perfil**

Tabla 32. Diccionario de atributos de la clase Perfil

Nombre	Tipo	Visibilidad	Descripción
codigo_profesor	String	Público	Este atributo sirve para almacenar el código del profesor.
codigo_materia	String	Público	Este atributo sirve para almacenar el nombre de la materia.

Fuente: los autores.

7.9.3. Diccionario de métodos de clase

En el diccionario de métodos de clase se describen las características de los métodos de cada una de las clases del sistema, al igual que el funcionamiento de cada uno de ellos.

- **Diccionario de métodos de clase BD**

Tabla 33. Diccionario de métodos de clase BD

Nombre	Tipo de retorno	Parámetros	Descripción
connect	Void	Ninguno	Se encarga de establecer una conexión con la base de datos.
query	Boolean	Query_String	Después de haber establecido conexión con la base de datos, esta función se encarga de enviar un query a la base de datos.
next_record	Array	Ninguno	Esta función se encarga de obtener el siguiente registro producto el último query.
num_rows	Integer	Ninguno	Esta función se encarga de informar cuantas filas posee la ultima consulta, es decir el último query de tipo select.
affected_rows	Integer	Ninguno	Esta función se encarga de informar cuantas filas posee el último query de tipo insert, update o delete.
close	Void	Ninguno	Se encarga de cerrar la conexión con la base de datos.

Fuente: los autores.

- **Diccionario de métodos de clase profesor**

Tabla 34. Diccionario de métodos de clase profesor

Nombre	Tipo de retorno	Parámetros	Descripción
adicionar	Integer	String codigo String nombre String tipo_contrato	Por medio de la clase DB, adiciona un profesor en la base de datos.
modificar	Integer	String codigo String nombre String tipo_contrato	Por medio de la clase DB, actualiza los datos de un profesor en la base de datos.
consultar	Array	String consulta Integer tipo	Por medio de la clase DB, realiza consultas relacionadas con información de profesores.
eliminar	Integer	String codigo	Por medio de la clase DB, elimina un profesor de la base de datos.
horario	Integer	Integer operacion Array datos	Por medio de la clase DB y dependiendo de la operación especificada, gestiona la información referente al horario de un profesor en la base de datos.

Fuente: los autores.

- **Diccionario de métodos de clase materia**

Tabla 35. Diccionario de métodos de clase materia

Nombre	Tipo de retorno	Parámetros	Descripción
adicionar	Integer	String codigo String nombre String tipo String creditos String ih String semestre	Por medio de la clase DB, adiciona una materia en la base de datos.
modificar	Integer	String codigo String nombre String tipo String ih	Por medio de la clase DB, actualiza los datos de una materia en la base de datos.

Tabla 35. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
consultar	Array	String consulta Integer tipo	Por medio de la clase DB, realiza consultas relacionadas con información de materias.
eliminar	Integer	String codigo	Por medio de la clase DB, elimina una materia de la base de datos.
horario	Integer	Integer operacion Array datos	Por medio de la clase DB y dependiendo de la operación especificada, gestiona la información referente al horario de un grupo en la base de datos.

Fuente: los autores.

- **Diccionario de métodos de clase usuario**

Tabla 36. Diccionario de métodos de clase usuario

Nombre	Tipo de retorno	Parámetros	Descripción
adicionar	Integer	String login String nombre String clave_md5 String tipo	Por medio de la clase DB, adiciona un usuario en la base de datos.
modificar	Integer	String login String nombre Boolean cambiar_clave String clave_nueva_md5	Por medio de la clase DB, actualiza los datos de un usuario en la base de datos.
validar	Array	String Login String clave_md5	Por medio de la clase DB, valida e informa si los datos proporcionados corresponden a los de un usuario registrado en el sistema.
eliminar	Integer	String login	Por medio de la clase DB, elimina un usuario de la base de datos.
consultar	Array	String consulta Integer tipo	Por medio de la clase DB, realiza consultas relacionadas con información de usuarios.

Fuente: los autores.

- **Diccionario de métodos de clase perfil**

Tabla 37. Diccionario de métodos de clase perfil

Nombre	Tipo de retorno	Parámetros	Descripción
adicionar	Integer	string codigo_profesor string codigo_materia	Por medio de la clase DB, adiciona una materia al perfil de un profesor.
consultar	Array	String consulta Integer tipo	Por medio de la clase DB, realiza consultas relacionadas con información de perfiles.
eliminar	Integer	string codigo_profesor string codigo_materia	Por medio de la clase DB, elimina una materia del perfil de un profesor.

Fuente: los autores.

- **Diccionario de métodos de clase carga**

Tabla 38. Diccionario de métodos de clase carga

Nombre	Tipo de retorno	Parámetros	Descripción
horario	Integer	Integer accion Array datos	Por medio de la clase DB, realiza diversas operaciones en cuanto a la carga académica, inserciones, actualizaciones y eliminaciones.
consultar	Array	String consulta Integer tipo	Por medio de la clase DB, realiza consultas relacionadas con información de grupos.
asignacion_carga	String	Array profesores Array tope_max Boolean sobrecarga Array tope_sobre String jornada Array categoria	Por medio de la clase DB, realiza la asignación de docentes a los grupos de las asignaturas registradas teniendo en cuenta restricciones definidas y grupos existentes.

Fuente: los autores.

- **Diccionario de métodos de clase controlador**

Tabla 39. Diccionario de métodos de clase controlador

Nombre	Tipo de retorno	Parámetros	Descripción
iniciar_sesion	Void	String enviado	Este método se encarga de controlar todo el proceso referente al inicio de sesión; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
cerrar_sesion	Void	Ninguno	Este método se encarga de controlar todo el proceso referente al cierre de sesión.
adicionar_profesor	Void	String enviado	Este método se encarga de controlar todo el proceso para adicionar un profesor; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
Modificar_profesor	Void	String enviado	Este método se encarga de controlar todo el proceso para modificar los datos de un profesor; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
consultar_profesor	Void	String enviado	Este método se encarga de controlar todo el proceso para consultar la información referente a profesores; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
eliminar_profesor	Void	String enviado	Este método se encarga de controlar todo el proceso para eliminar un profesor; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.

Tabla 39. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
horario_profesor	Void	String enviado	Este método se encarga de controlar todo el proceso para gestionar el horario de un profesor; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
adicionar_materia_perfil	Void	String enviado	Este método se encarga de controlar todo el proceso para adicionar una materia a un profesor para establecer su perfil; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
eliminar_materia_perfil	Void	String enviado	Este método se encarga de controlar todo el proceso para eliminar una materia a un profesor para modificar su perfil de asignaturas; haciendo uso de las clases interfaz y profesor, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
adicionar_materia	Void	String enviado	Este método se encarga de controlar todo el proceso para adicionar una materia; haciendo uso de las clases interfaz y materia, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
modificar_materia	Void	String enviado	Este método se encarga de controlar todo el proceso para modificar los datos de una materia; haciendo uso de las clases interfaz y materia, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
consultar_materia	Void	String enviado	Este método se encarga de controlar todo el proceso para consultar información referente a materias y grupos; haciendo uso de las clases interfaz y materia, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.

Tabla 39. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
eliminar_materia	Void	String enviado	Este método se encarga de controlar todo el proceso para eliminar una materia; haciendo uso de las clases interfaz y materia, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
adicionar_usuario	Void	String enviado	Este método se encarga de controlar todo el proceso para adicionar un usuario; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
modificar_usuario	Void	String enviado	Este método se encarga de controlar todo el proceso para modificar los datos de un usuario; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
consultar_usuario	Void	String enviado	Este método se encarga de controlar todo el proceso para consultar información acerca de usuarios; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
eliminar_usuario	Void	String enviado	Este método se encarga de controlar todo el proceso para eliminar un usuario; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
restablecer_clave	Void	String enviado	Este método se encarga de controlar todo el proceso para restablecer la clave por defecto de un usuario profesor; haciendo uso de las clases interfaz y usuario, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.

Tabla 39. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
adicionar_grupo_carga	Void	String enviado	Este método se encarga de controlar todo el proceso para adicionar un grupo de una determinada asignatura; haciendo uso de las clases interfaz y carga, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
consultar_carga	Void	String enviado	Este método se encarga de controlar todo el proceso para consultar información acerca de los grupos; haciendo uso de las clases interfaz y carga, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
eliminar_grupo_carga	Void	String enviado	Este método se encarga de controlar todo el proceso para eliminar un grupo de una determinada asignatura; haciendo uso de las clases interfaz y carga, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
asignacion_carga	Void	String enviado	Este método se encarga de controlar todo el proceso de asignación de la carga; haciendo uso de las clases interfaz y carga, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
reiniciar_carga	Void	String enviado	Este método se encarga de controlar todo el proceso de poner carga en cero o en una carga específica; haciendo uso de las clases interfaz y carga, se muestra la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.

Tabla 39. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
asignar_grupo	Void	String enviado	Este método se encarga de asignar profesor a un grupo de forma manual; haciendo uso de las clases interfaz y carga, se muestra String enviado la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.
liberar_grupo	Void	String enviado	Este método se encarga de controlar todo el proceso de eliminar la asignación de un profesor de forma manual; haciendo uso de las clases interfaz y carga, se muestra String enviado la información necesaria y se interactúa con la base de datos mediante la invocación de sus métodos.

Fuente: los autores.

- **Diccionario de métodos de clase interfaz**

Tabla 40. Diccionario de métodos de clase interfaz

Nombre	Tipo de retorno	Parámetros	Descripción
iniciar_sesion	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios necesarios para el inicio de sesión.
menu	String	Ninguno	Este método se encarga de mostrar el menú principal y capturar la opción escogida por el usuario.
resultado	Void	Integer tipo String mensaje	Este método es una herramienta auxiliar que muestra un mensaje sobre una interfaz.
adicionar_profesor	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para adicionar un profesor.

Tabla 40. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
modificar_profesor	String	String accion Array profesor	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para modificar la información de un profesor.
consultar_profesor	String	Array consultas	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para mostrar información referente a profesores.
eliminar_profesor	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar un profesor.
horario_profesor	String	String accion DB profesor String codigo_profesor	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para gestionar el horario de un profesor.
adicionar_materia_perfil	String	String accion DB perfil String codigo_profesor	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para adicionar una materia a un profesor para establecer un perfil.
eliminar_materia_perfil	String	String accion DB perfil String codigo_profesor	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar una materia del perfil de un profesor.
adicionar_materia	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para adicionar una materia.
modificar_materia	String	String accion Array materia	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para modificar la información de una materia.

Tabla 40. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
consultar_materia	String	Array consultas	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para mostrar información referente a materias y grupos.
eliminar_materia	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar una materia.
adicionar_usuario	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para adicionar un usuario.
modificar_usuario	String	String accion Array usuario	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para modificar la información de un usuario.
consultar_usuario	String	Array consultas	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para mostrar información referente a usuarios.
eliminar_usuario	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar un usuario.
restablecer_clave	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para restablecer la clave de un usuario profesor.
adicionar_grupo_carga	String	String accion	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para adicionar un grupo.
consultar_carga	String	Array consultas	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para mostrar información referente a grupos.

Tabla 40. (Continuación)

Nombre	Tipo de retorno	Parámetros	Descripción
eliminar_grupo	String	String accion DB grupos String codigo String nombre	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar un grupo.
asignacion_carga	String	String accion Array profesores Array tope_max Boolean sobrecarga Array tope_sobre Array categorías String jornada	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para realizar la asignación de la carga.
reiniciar_carga	String	String accion DB grupos	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para reiniciar la carga.
asignar_grupo	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para asignar manualmente un grupo.
liberar_grupo	String	Ninguno	Este método se encarga de mostrar las interfaces y formularios de captura de datos necesarios para eliminar manualmente un grupo asignado.

Fuente: los autores.

7.10. REQUERIMIENTOS NO FUNCIONALES

En esta sección se mencionan las características generales que debe cumplir el sistema, es decir, cómo debe ser el sistema para que pueda adaptarse a las condiciones impuestas por el usuario sin afectar su correcto funcionamiento.

- **Confiabilidad.** Es necesario que la información que se va a almacenar en el sistema permita que la asignación de los horarios a los diferentes profesores registrados se haga de una forma confiable garantizando de esta manera la transparencia del proceso. Por otro lado, debe garantizarse la coherencia e integridad de los datos que están siendo manipulados con el fin de obtener óptimos resultados en la asignación de la carga académica.
- **Comunicabilidad.** Los usuarios registrados deben poder ingresar al sistema desde cualquier lugar con conexión a Internet y tener habilitadas todas las funciones disponibles dependiendo del tipo de usuario.
- **Eficiencia.** La aplicación debe garantizar el buen uso de los recursos que manipula como profesores y salones para lograr mejor optimización y rendimiento.
- **Seguridad.** La información que se va almacenando en el sistema mediante las diferentes opciones solo podrá ser manipulada por los usuarios registrados. Además, habrán operaciones restringidas para los profesores dependiendo del nivel de manipulación que estos puedan llegar a tener sobre los datos.
- **Usabilidad.** La aplicación debe poseer interfaces en las que se vea claramente el menú de cada función o proceso, siendo este simple y comprensible para un mejor desenvolvimiento del usuario dependiendo de la necesidad.

8. DISEÑO DEL SISTEMA

En esta sección se definen la arquitectura del sistema y el entorno tecnológico que le va a dar soporte, además de la especificación detallada de los componentes del sistema de información.

8.1. ARQUITECTURA DEL SISTEMA

La aplicación se diseñó bajo la arquitectura Web-enable, en la cual se requiere un navegador en el cliente, una aplicación estándar que se encuentra en casi todos los sistemas operativos de hoy en día, lo que representa una ventaja frente a la arquitectura Cliente/Servidor pura.

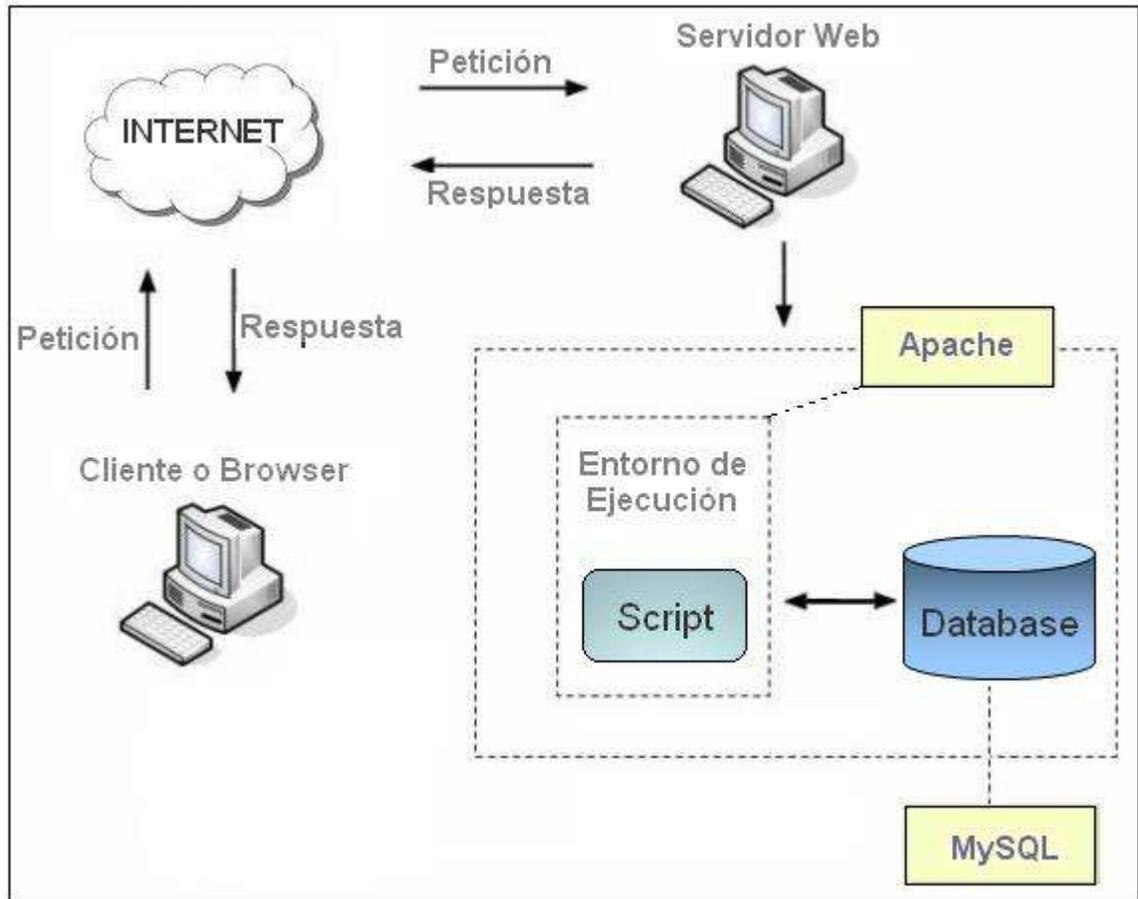
En la arquitectura Web-enable se limita al cliente a recibir y enviar datos dejando la lógica de programación de lado del servidor, lo que permite que la plataforma cliente o browser en este caso, pueda prescindir de propiedades o características particulares propias de la aplicación desarrollada.

Esta arquitectura posee tres niveles o capas:

- La primera capa se sitúa del lado del cliente y está compuesta por el navegador, que se encarga de enviar y recibir información.
- La segunda capa se sitúa del lado del servidor; la componen el entorno de ejecución y los scripts de PHP en este caso, estos se encargan de manipular los datos enviados por el cliente, aplicar las reglas del negocio y enviar la información pertinente de nuevo al cliente.
- La tercera capa se encuentra en la base de datos, en la que se almacena toda la información que el sistema gestiona.⁵⁶

⁵⁶ Kamran H. Enero 2005. Web-Based Enterprise Applications: Architecture, System, Security and Protocols. New York iUniverse. [libro en línea] Disponible desde internet en: <<http://www.iuniverse.com/bookstore/BookDetail.aspx?BookId=SKU-000028785>>

Figura 84. Arquitectura Web-enable



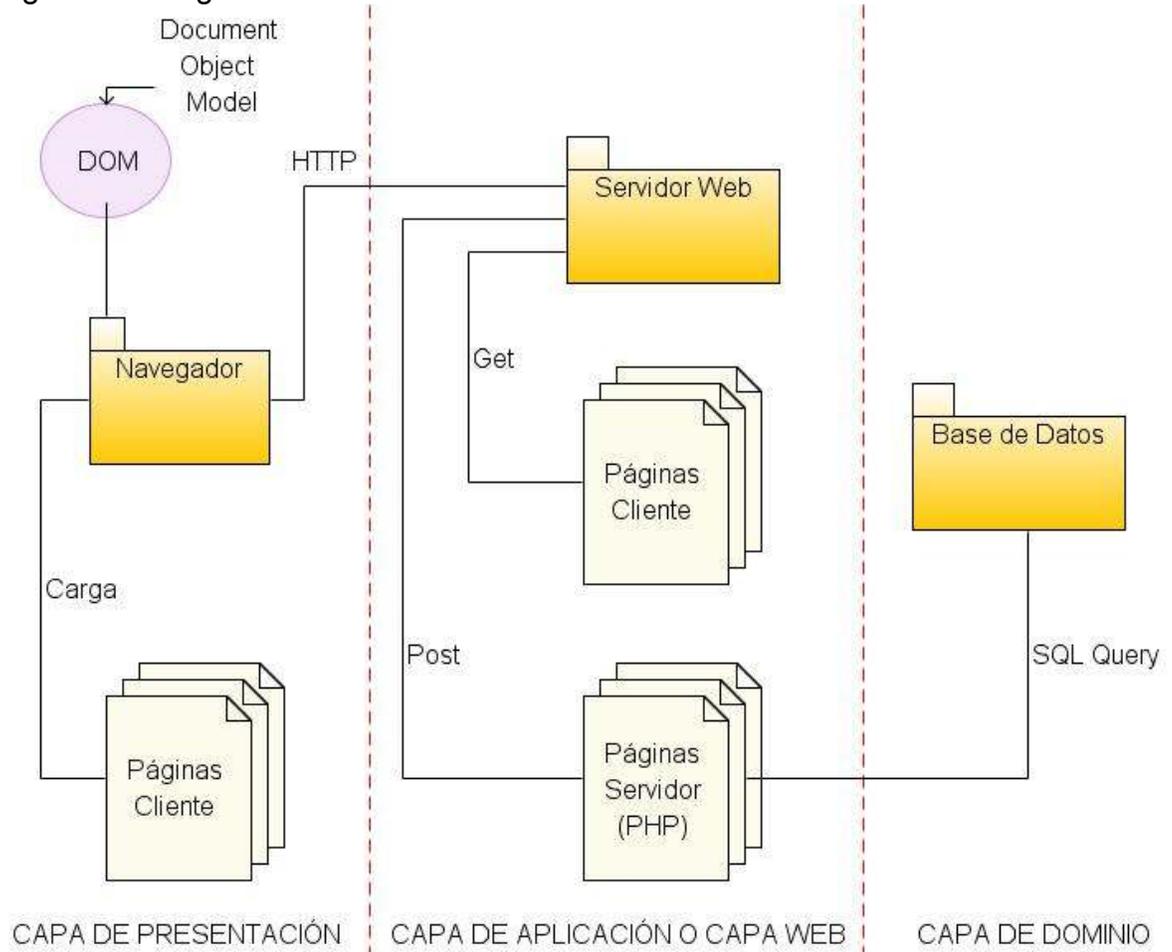
Fuente: los autores.

8.2. IDENTIFICACIÓN DE SUBSISTEMAS

En este apartado se puede ver la descomposición del sistema en subsistemas para facilitar el análisis del mismo.

8.2.1. Diagrama de Subsistemas

Figura 85. Diagrama de Subsistemas



8.2.2. Descripción de cada Subsistema

Aquí se puede ver la identificación de los subsistemas que se hizo con base en la arquitectura bajo la cual se diseñó la aplicación. A continuación se describen cada uno de los subsistemas que hacen parte de las tres capas que constituyen el sistema.

Capa de Presentación

- **Modelo de objetos del documento (DOM).** Es una interfaz orientada a objetos que permite la manipulación dinámica de los elementos del lenguaje en el cual se programan las páginas del cliente haciendo uso de javascript.
- **Navegador.** Es la aplicación que se usa del lado del cliente para que el usuario envíe y reciba información del servidor la cual es interpretada por el navegador y presentada al usuario de una forma legible y por lo general estética.

Capa de Aplicación o capa Web

- **Servidor Web.** Es una aplicación que se encuentra del lado del servidor, en ocasiones se usa este término para hacer referencia a la máquina que aloja dicha aplicación. Se encarga de recibir y servir peticiones http enviadas desde el cliente, el servidor responde con el contenido que el cliente requiere.
- **Páginas Cliente.** Son bloques de código que corresponden a una petición del cliente; el lenguaje usado para estas páginas es básicamente html y javascript, además, el código regularmente viene acompañado de archivos multimedia como fotos, animaciones, videos o música.
- **Páginas Servidor.** Son el código fuente que se ejecuta del lado del servidor, en este caso se utilizó PHP; puede tener además del código php, código html, que sería la porción de información que se enviará al cliente. PHP posee gran cantidad de herramientas entre ellas la posibilidad de conectarse a una base de datos, lo cual es una de las razones por las que se utiliza.

Capa de Dominio

- **Base de Datos.** Es una aplicación que permite guardar información de una forma sistemática y ordenada al igual que provee la manera de manipular la información almacenada y consultarla. PHP se conecta a la base de datos e interactúa con ella proporcionándole queries SQL, que es el lenguaje estándar para Bases de Datos.

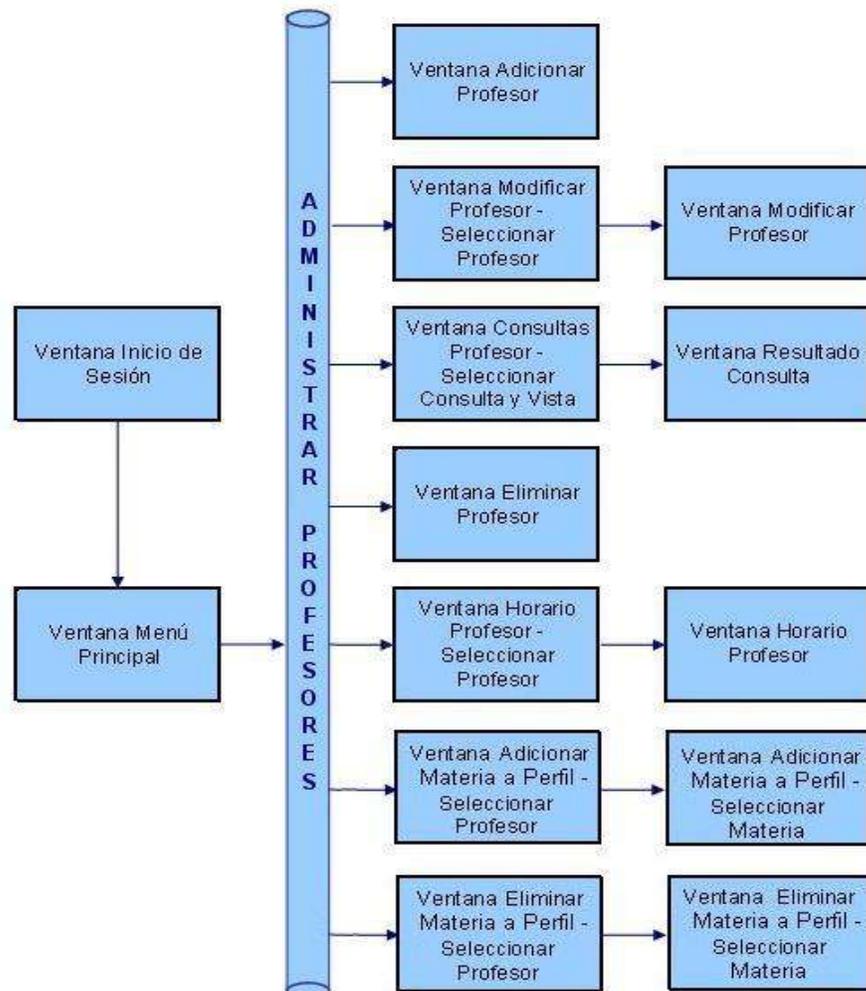
8.3. DESCRIPCIÓN DE SUBSISTEMAS DE APLICACIÓN

8.3.1. Modelo general de las aplicaciones

8.3.1.1 Diagramas de Secuencia de Ventanas

La figura 86 muestra la secuencia en que se abren las ventanas del caso de uso Administrar Profesores, al cual se accede desde la ventana del Menú Principal a través del submenú Administrar Profesores mostrando cada una de las opciones disponibles para el usuario.

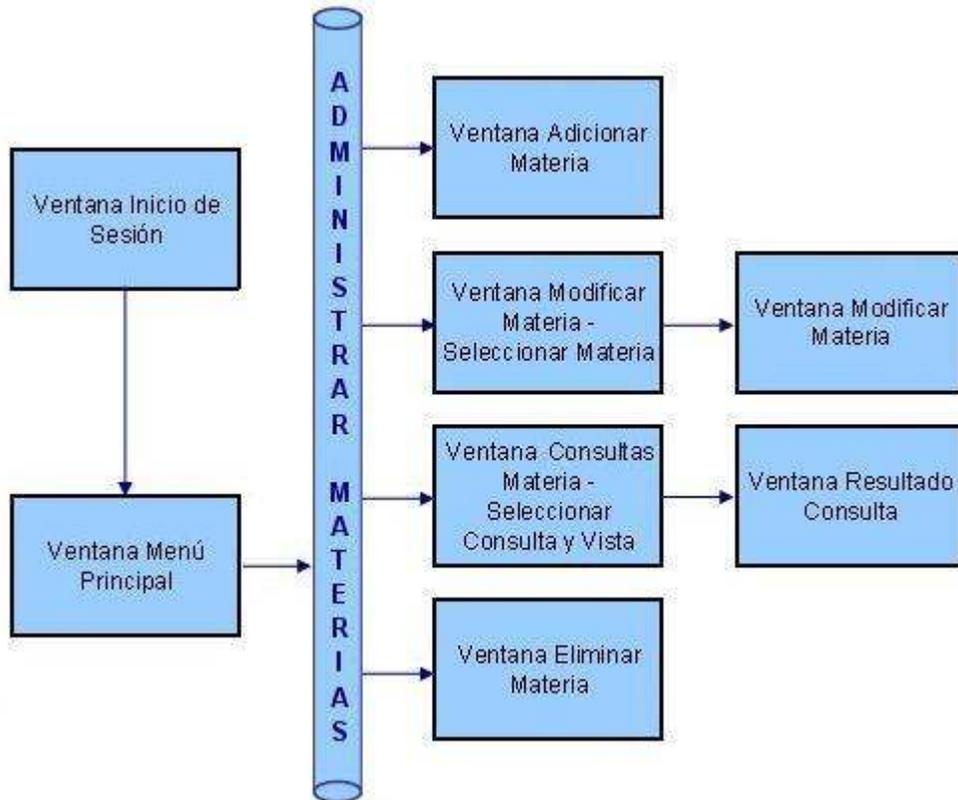
Figura 86. Diagrama de Secuencia de Ventanas – Administrar Profesores



Fuente: los autores.

La figura 87 muestra la secuencia en que se abren las ventanas del caso de uso Administrar Materias, al cual se accede desde la ventana del Menú Principal a través del submenú Administrar Materias mostrando cada una de las opciones disponibles para el usuario.

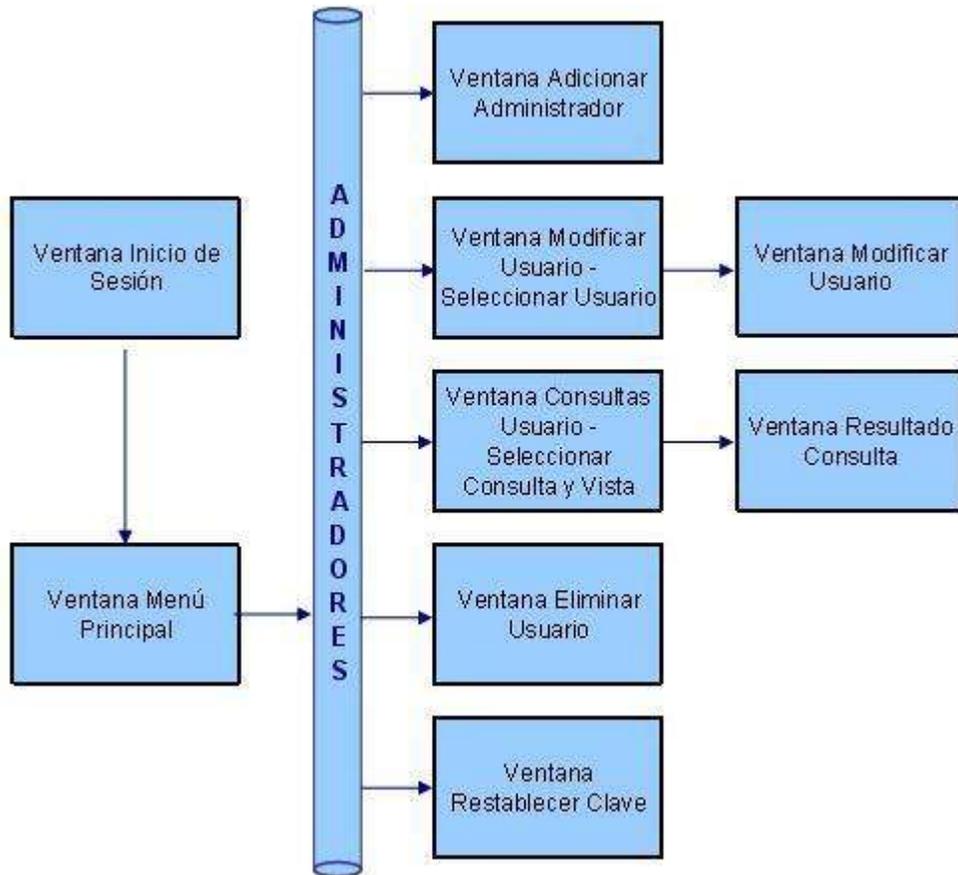
Figura 87. Diagrama de Secuencia de Ventanas – Administrar Materias



Fuente: los autores.

La figura 88 muestra la secuencia en que se abren las ventanas del caso de uso Administrar Usuarios (Administradores), al cual se accede desde la ventana del Menú Principal a través del submenú Administrar Usuarios mostrando cada una de las opciones disponibles para el usuario.

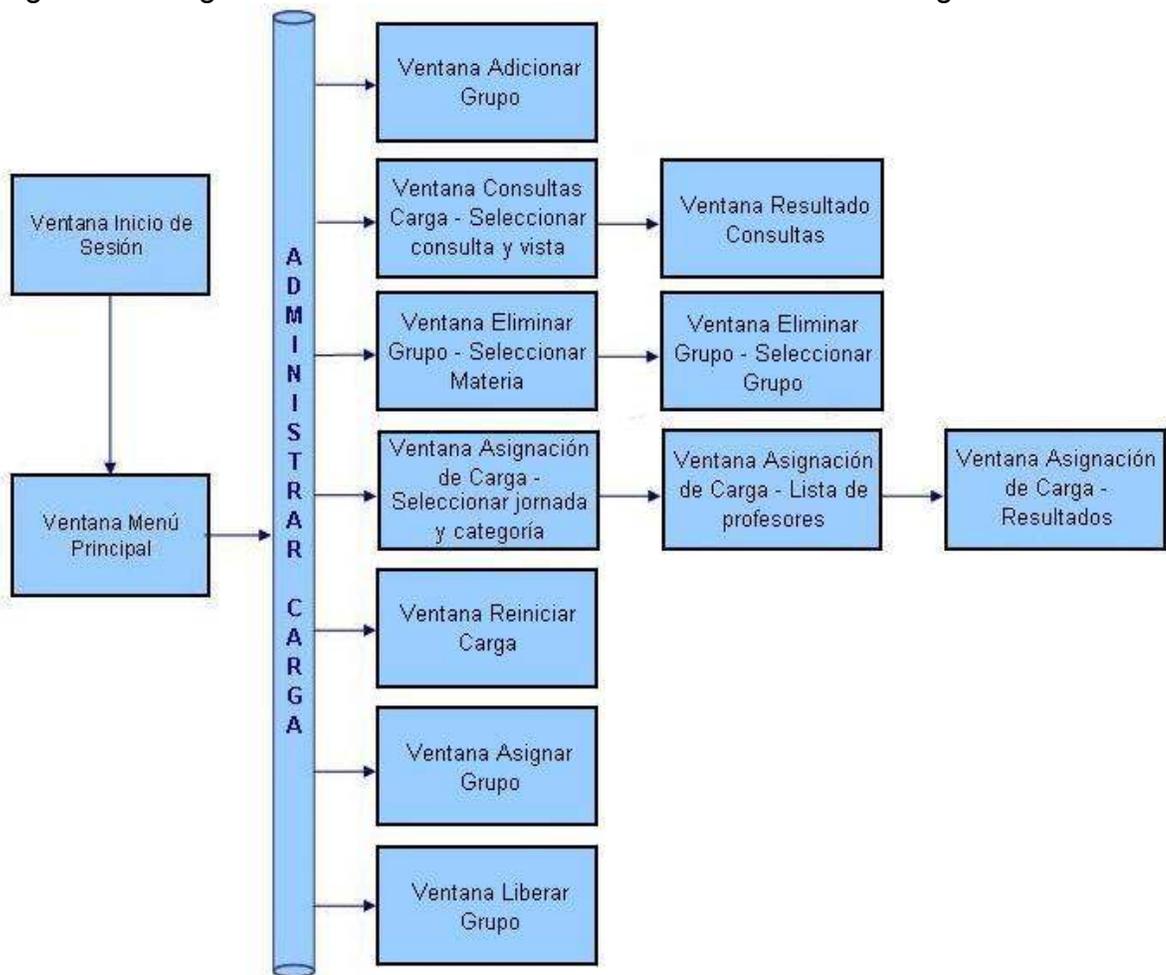
Figura 88. Diagrama de Secuencia de Ventanas – Administrar Usuarios



Fuente: los autores.

La figura 89 muestra la secuencia en que se abren las ventanas del caso de uso Administrar Carga, al cual se accede desde la ventana del Menú Principal a través del submenú Administrar Carga mostrando cada una de las opciones disponibles para el usuario.

Figura 89. Diagrama de Secuencia de Ventanas – Administrar Carga



Fuente: los autores.

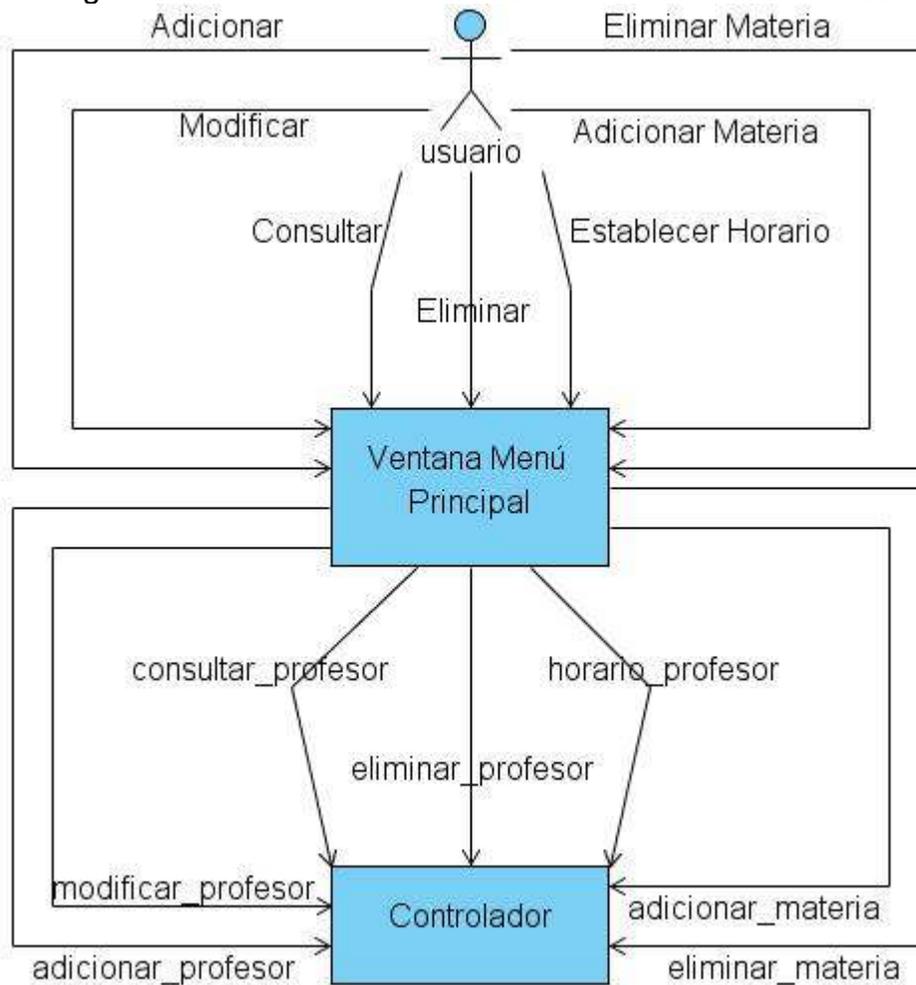
8.3.2. Modelo Dinámico de la aplicación

8.3.2.1. Diagramas de Interacción con el Usuario

Diagrama de Interacción con el Usuario – Administrar Profesores

La figura 90 muestra los eventos que el usuario le genera a la ventana Menú Principal cuando accede al submenú Administrar Profesores y los mensajes que este módulo le envía a la clase núcleo Controlador.

Figura 90. Diagrama de Interacción con el Usuario – Administrar Profesores

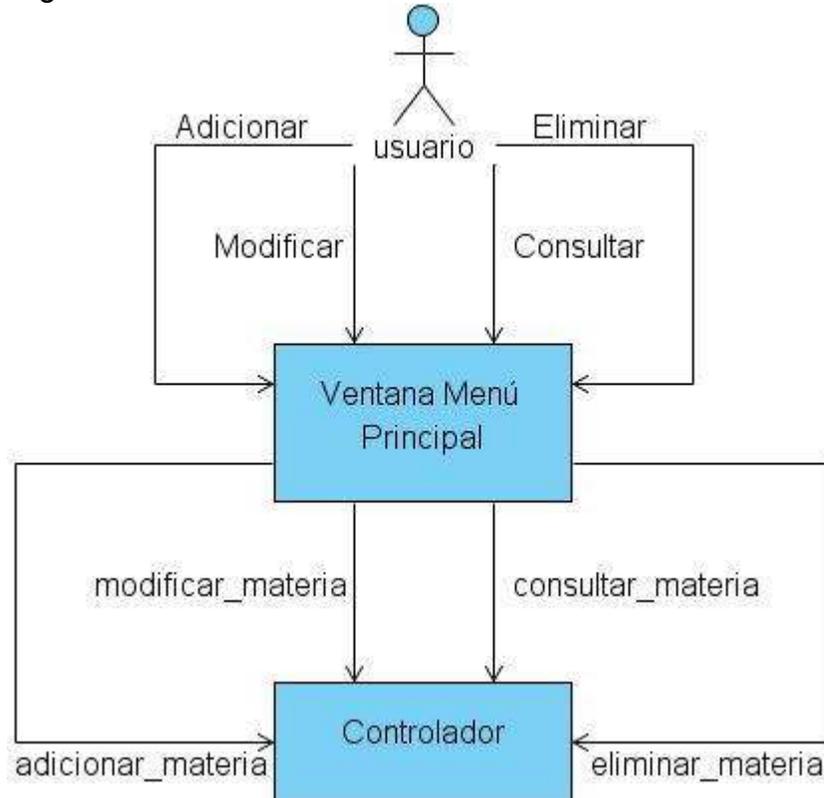


Fuente: los autores.

Diagrama de Interacción con el Usuario – Administrar Materias

La figura 91 muestra los eventos que el usuario le genera a la ventana Menú Principal cuando accede al submenú Administrar Materias y los mensajes que este módulo le envía a la clase núcleo Controlador.

Figura 91. Diagrama de Interacción con el Usuario – Administrar Materias

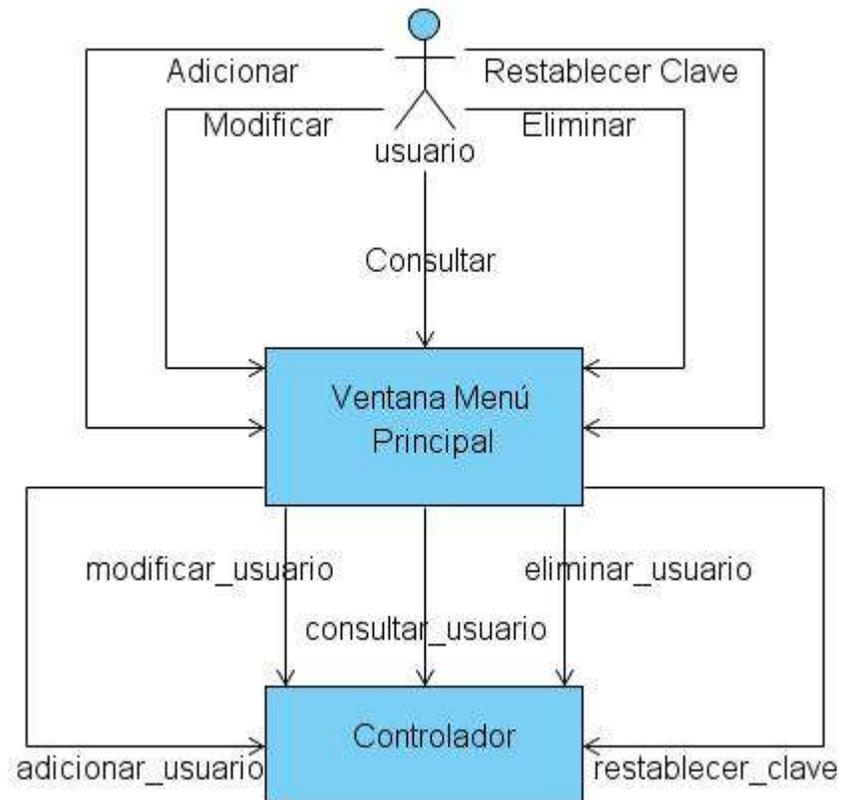


Fuente: los autores.

Diagrama de Interacción con el Usuario – Administrar Usuarios

La figura 92 muestra los eventos que el usuario le genera a la ventana Menú Principal cuando accede al submenú Administrar Usuarios y los mensajes que este módulo le envía a la clase núcleo Controlador.

Figura 92. Diagrama de Interacción con el Usuario – Administrar Usuarios

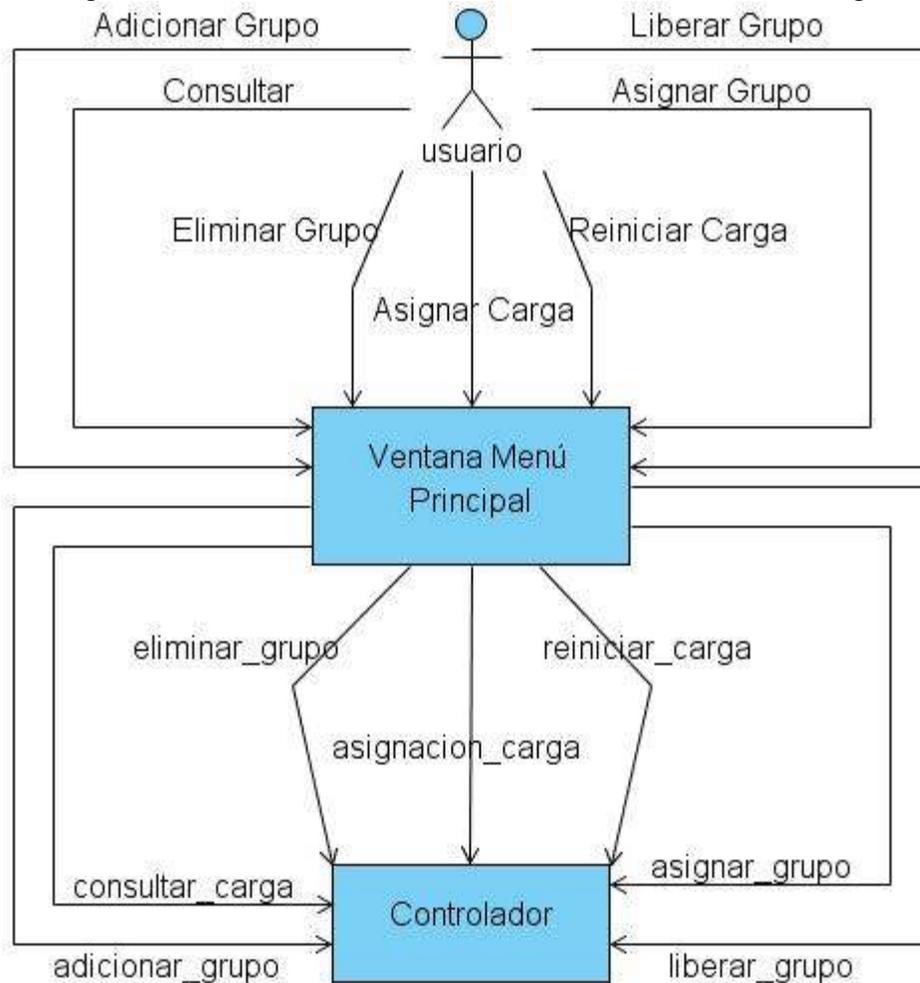


Fuente: los autores.

Diagrama de Interacción con el Usuario – Administrar Carga

La figura 93 muestra los eventos que el usuario le genera a la ventana Menú Principal cuando accede al submenú Administrar Carga y los mensajes que este módulo le envía a la clase núcleo Controlador.

Figura 93. Diagrama de Interacción con el Usuario – Administrar Carga



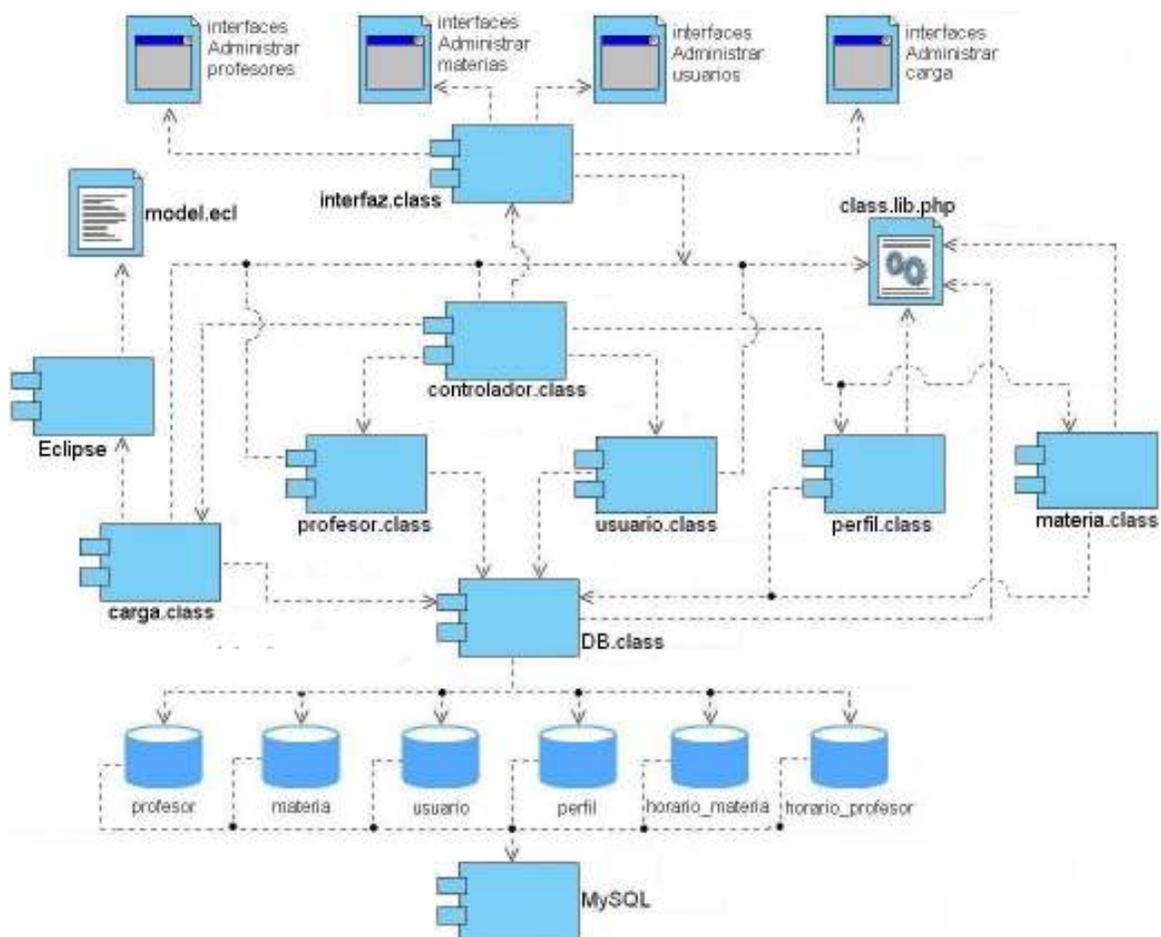
Fuente: los autores.

8.3.3. Modelo de Componentes

8.3.3.1. Diagrama de Componentes

La figura 94 muestra la manera como están organizados los elementos físicos del Sistema y las relaciones entre ellos. Estos elementos o componentes son librería, archivos, ejecutables y documentos.

Figura 94. Diagrama de Componentes



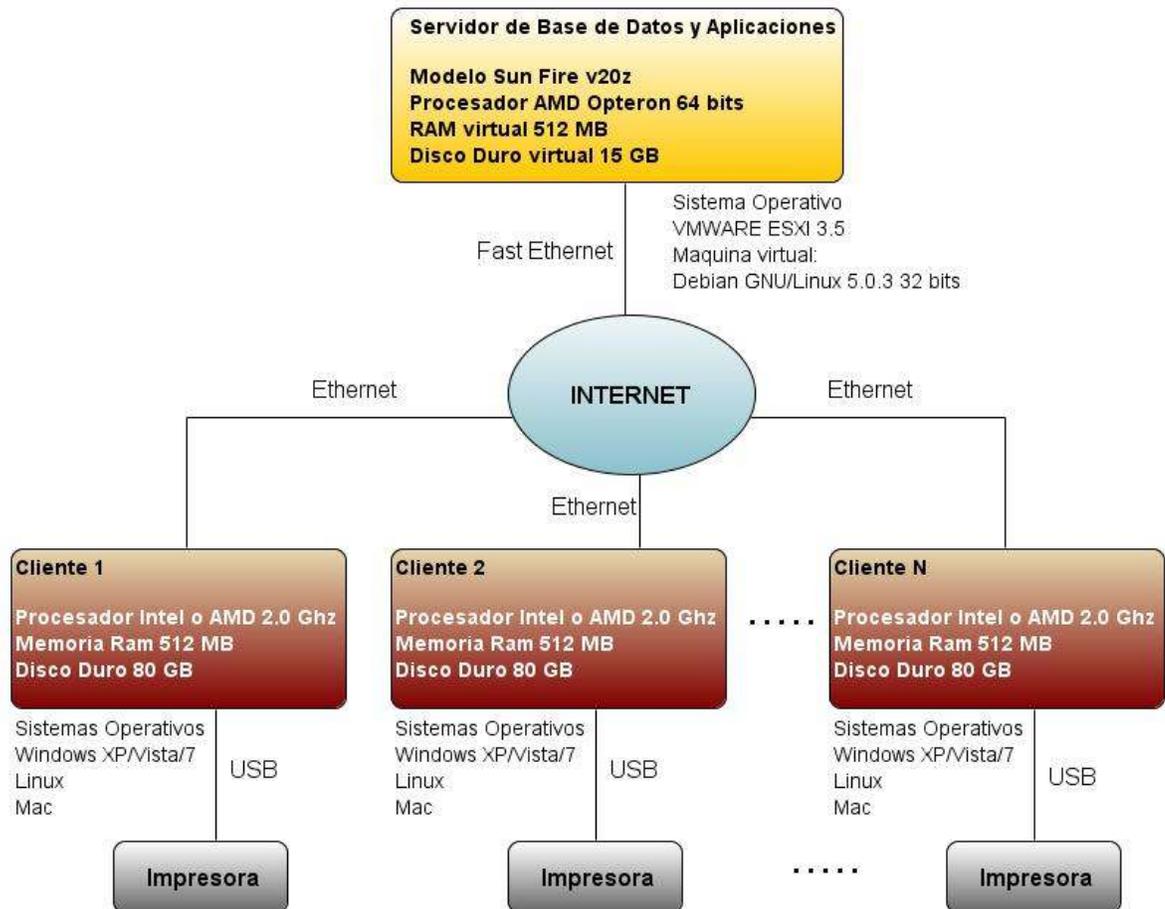
Fuente: los autores.

8.3.4. Modelo de Distribución

8.3.4.1. Diagrama de Despliegue

La figura 95 muestra las relaciones físicas entre los componentes hardware y software del sistema.

Figura 95. Diagrama de Despliegue



Fuente: los autores.

8.4. DISEÑO DE INTERFACES

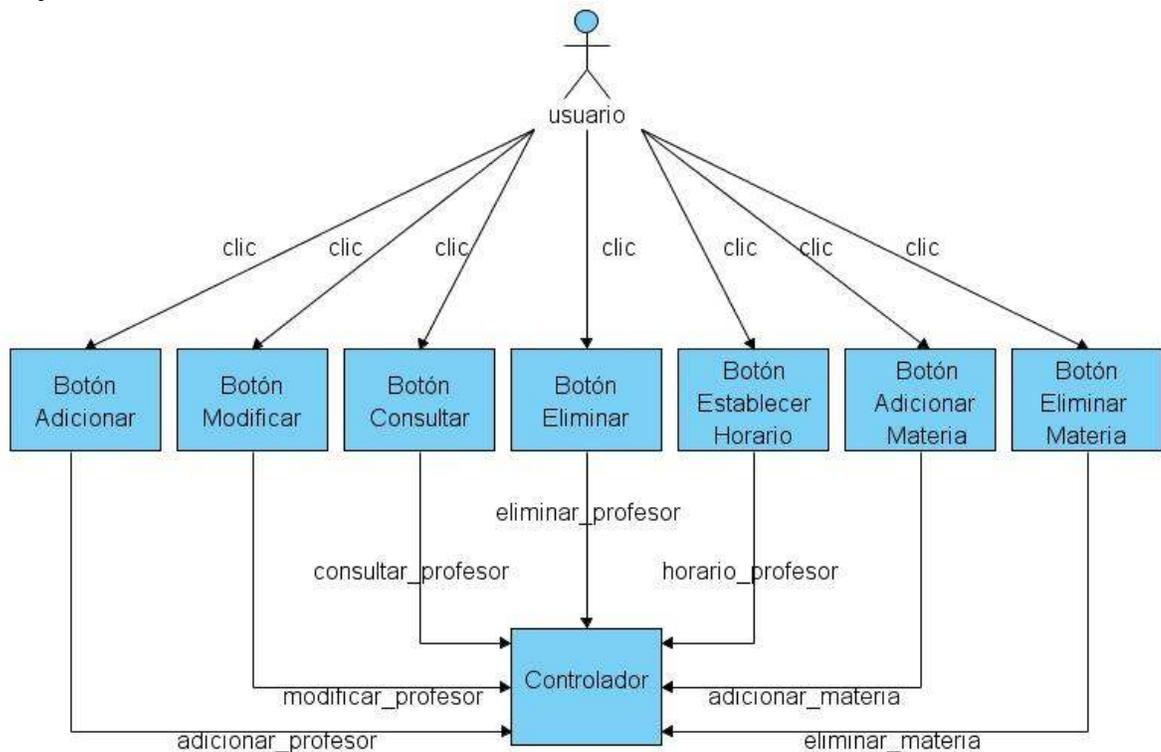
En este capítulo se puede ver la manera como se comportan los diferentes objetos gráficos del sistema y la manera como interactúan con las clases involucradas en el sistema.

8.4.1. Diagramas de Interacción con el usuario de los componentes a nivel de objetos

Diagramas de Interacción con el usuario de los componentes a nivel de objetos – Administrar Profesores

La figura 96 muestra los eventos que le son generados a los objetos gráficos y los mensajes que recibe la clase núcleo Controlador en el módulo Administrar Profesores.

Figura 96. Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Profesores

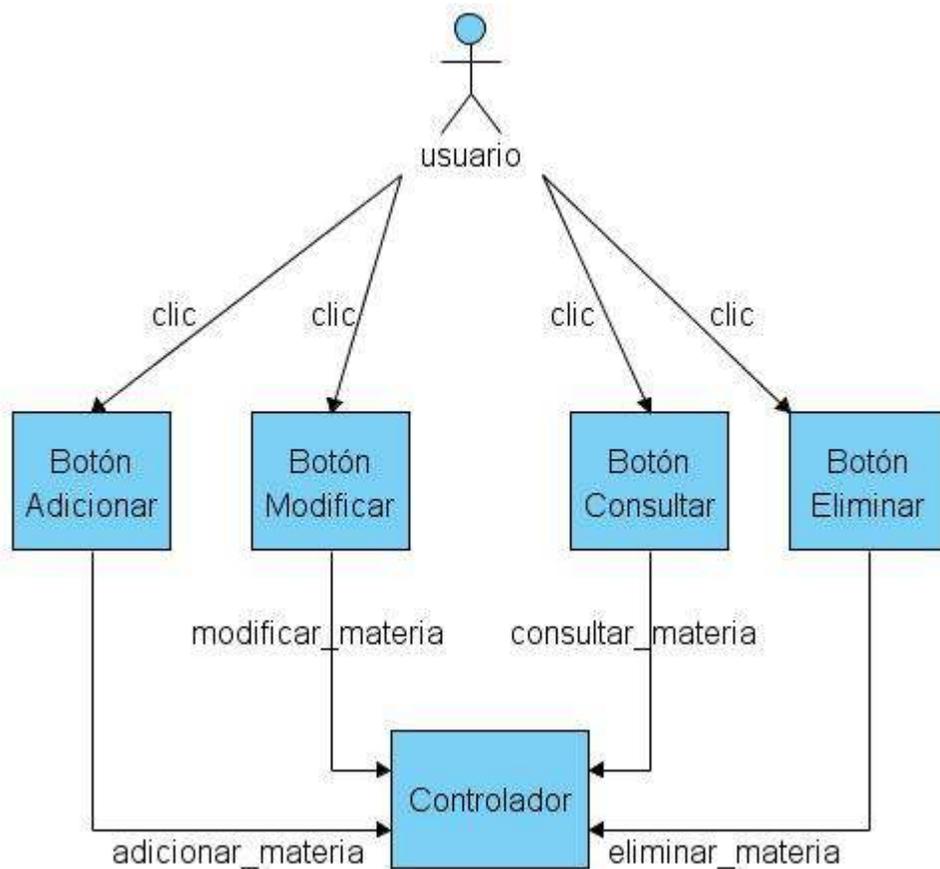


Fuente: los autores.

Diagramas de Interacción con el usuario de los componentes a nivel de objetos – Administrar Materias

La figura 97 muestra los eventos que le son generados a los objetos gráficos y los mensajes que recibe la clase núcleo Controlador en el módulo Administrar Materias.

Figura 97. Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Materias

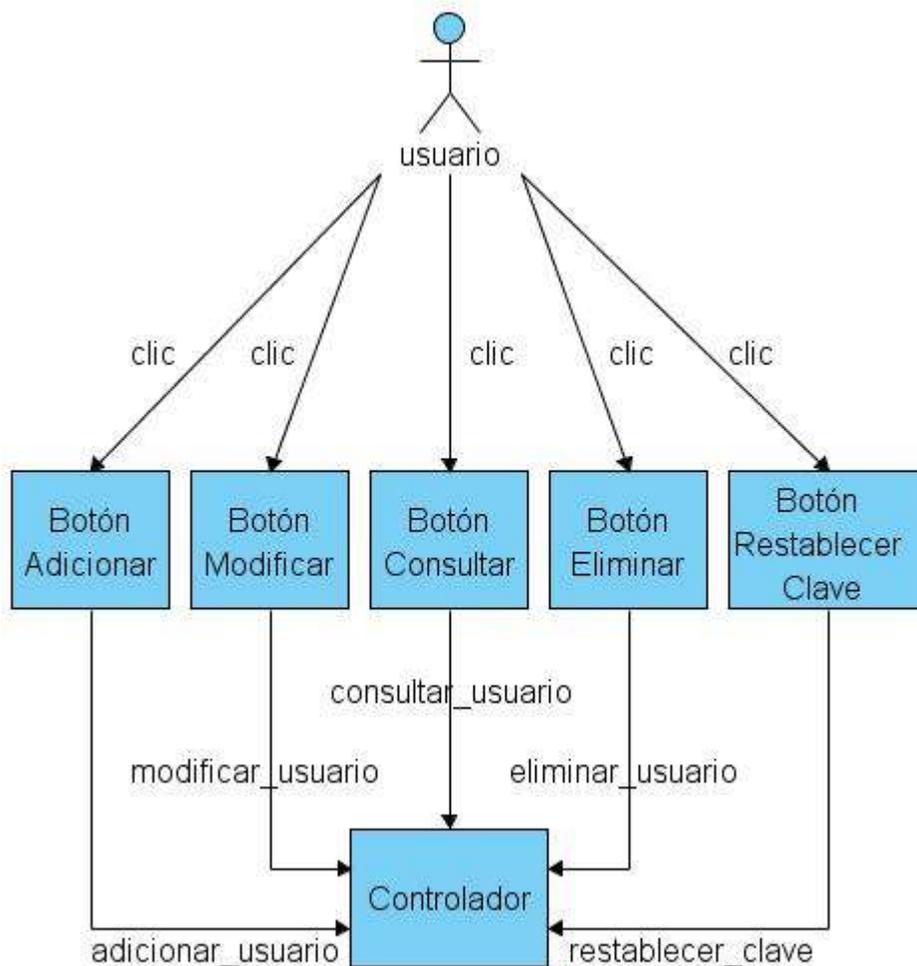


Fuente: los autores.

Diagramas de Interacción con el usuario de los componentes a nivel de objetos – Administrar Usuarios

La figura 98 muestra los eventos que le son generados a los objetos gráficos y los mensajes que recibe la clase núcleo Controlador en el módulo Administrar Usuarios.

Figura 98. Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Usuarios

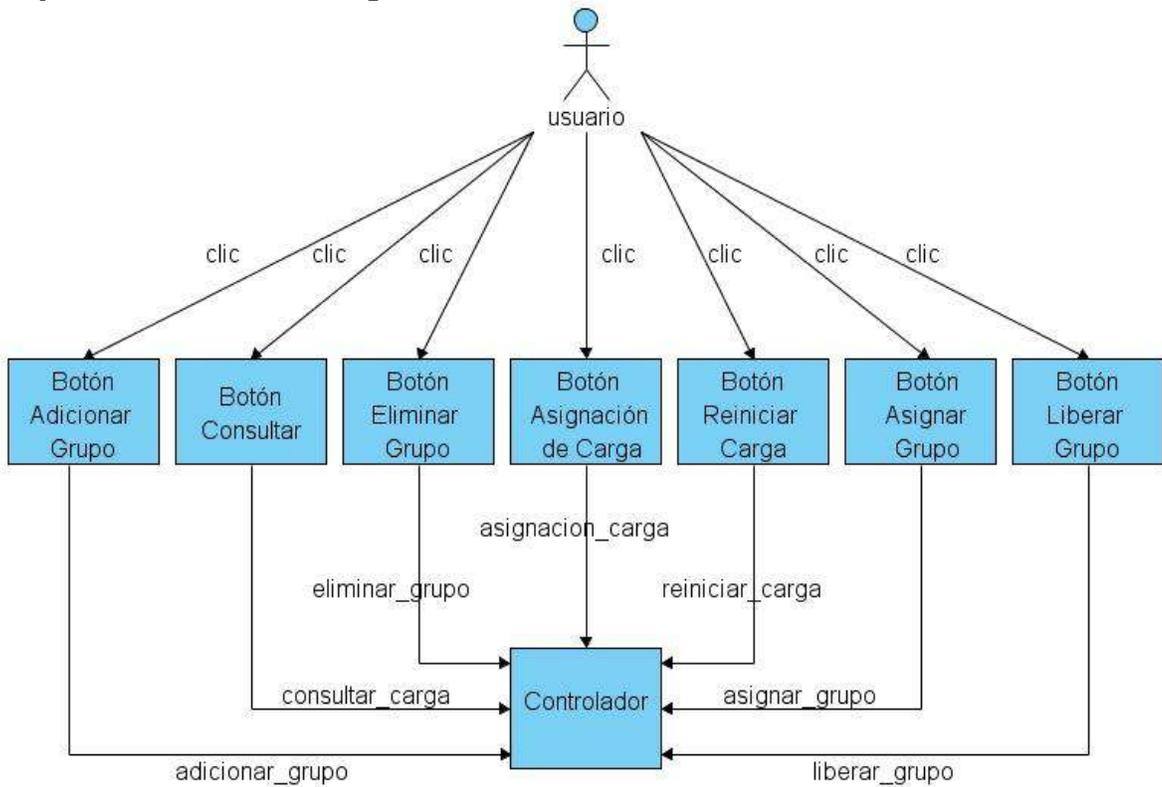


Fuente: los autores.

Diagramas de Interacción con el usuario de los componentes a nivel de objetos – Administrar Carga

La figura 99 muestra los eventos que le son generados a los objetos gráficos y los mensajes que recibe la clase núcleo Controlador en el módulo Administrar Carga.

Figura 99. Diagrama de Interacción con el usuario de los componentes a nivel de Objetos – Administrar Carga



Fuente: los autores.

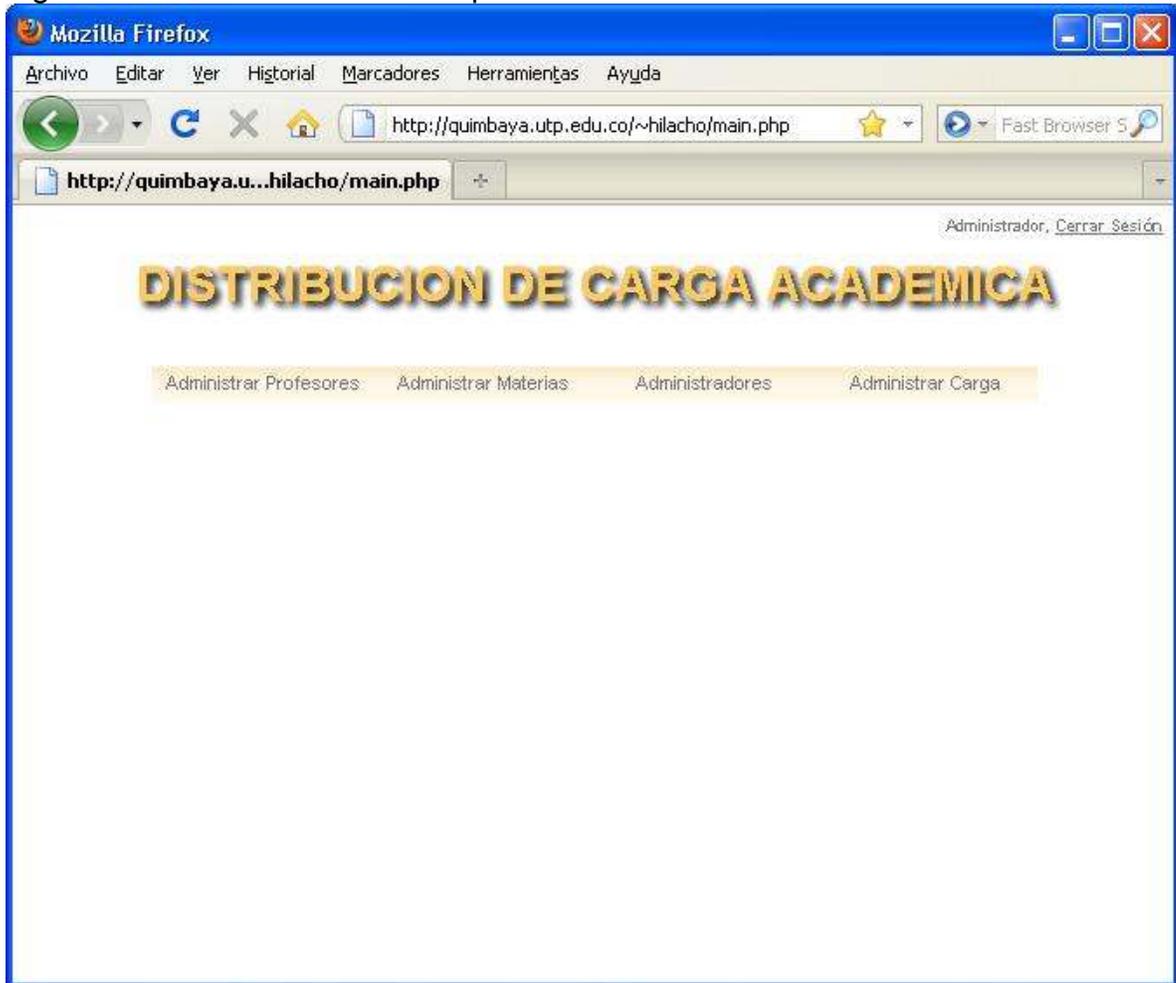
8.4.2. Diseño estético

Figura 100. Ventana Inicio de Sesión



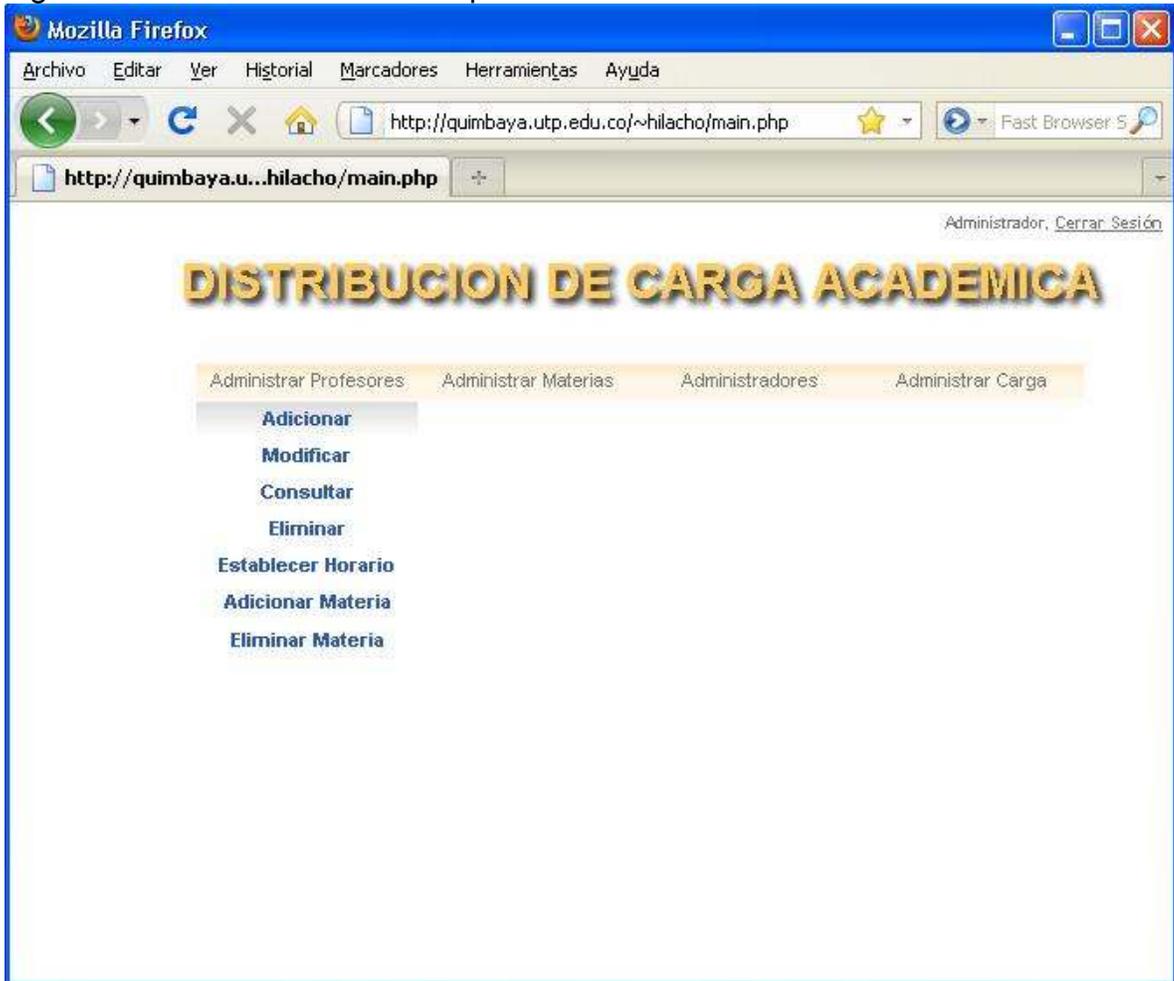
Fuente: los autores.

Figura 101. Ventana Menú Principal



Fuente: los autores.

Figura 102. Ventana Menú Principal – Administrar Profesores



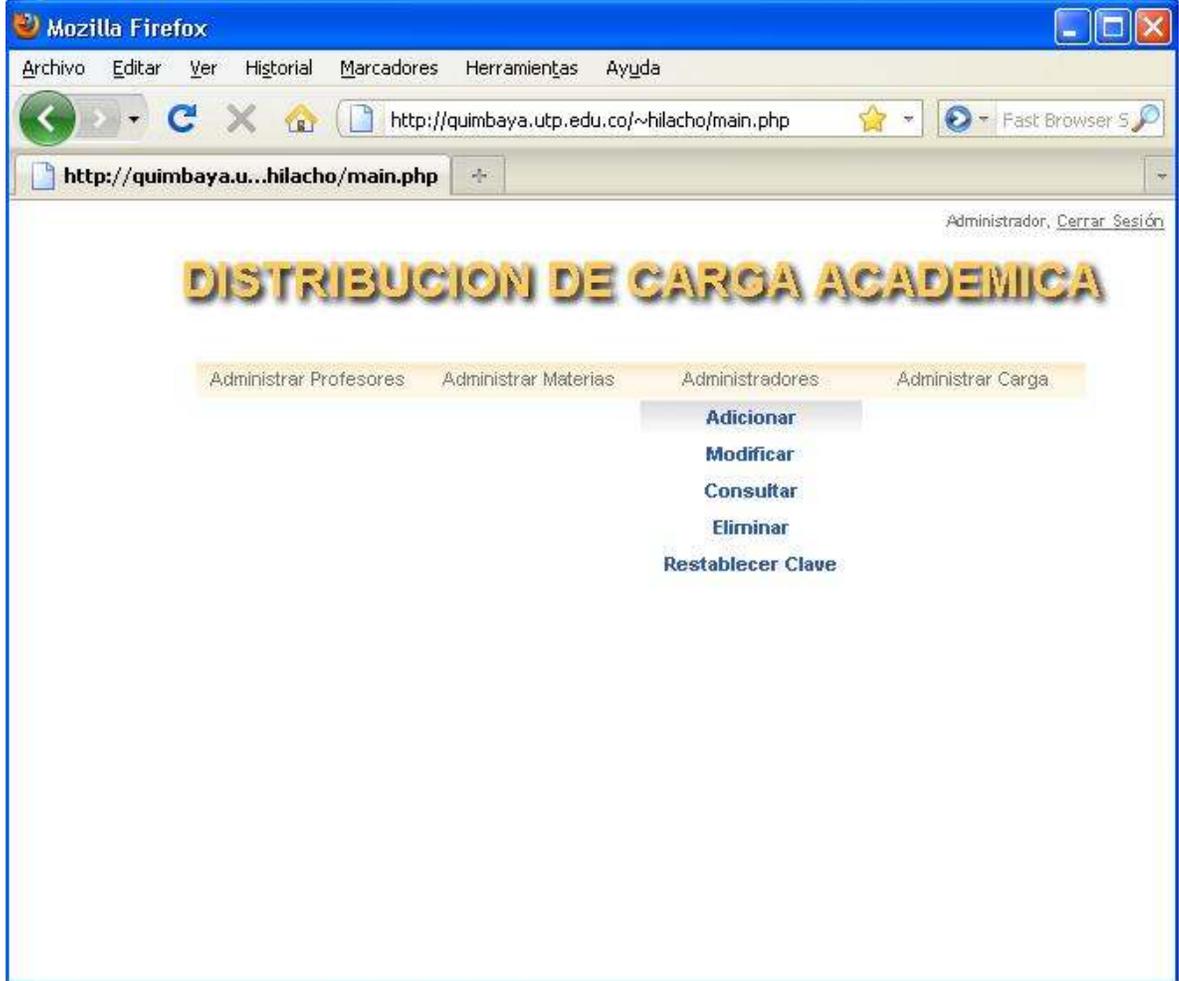
Fuente: los autores.

Figura 103. Ventana Menú Principal – Administrar Materias



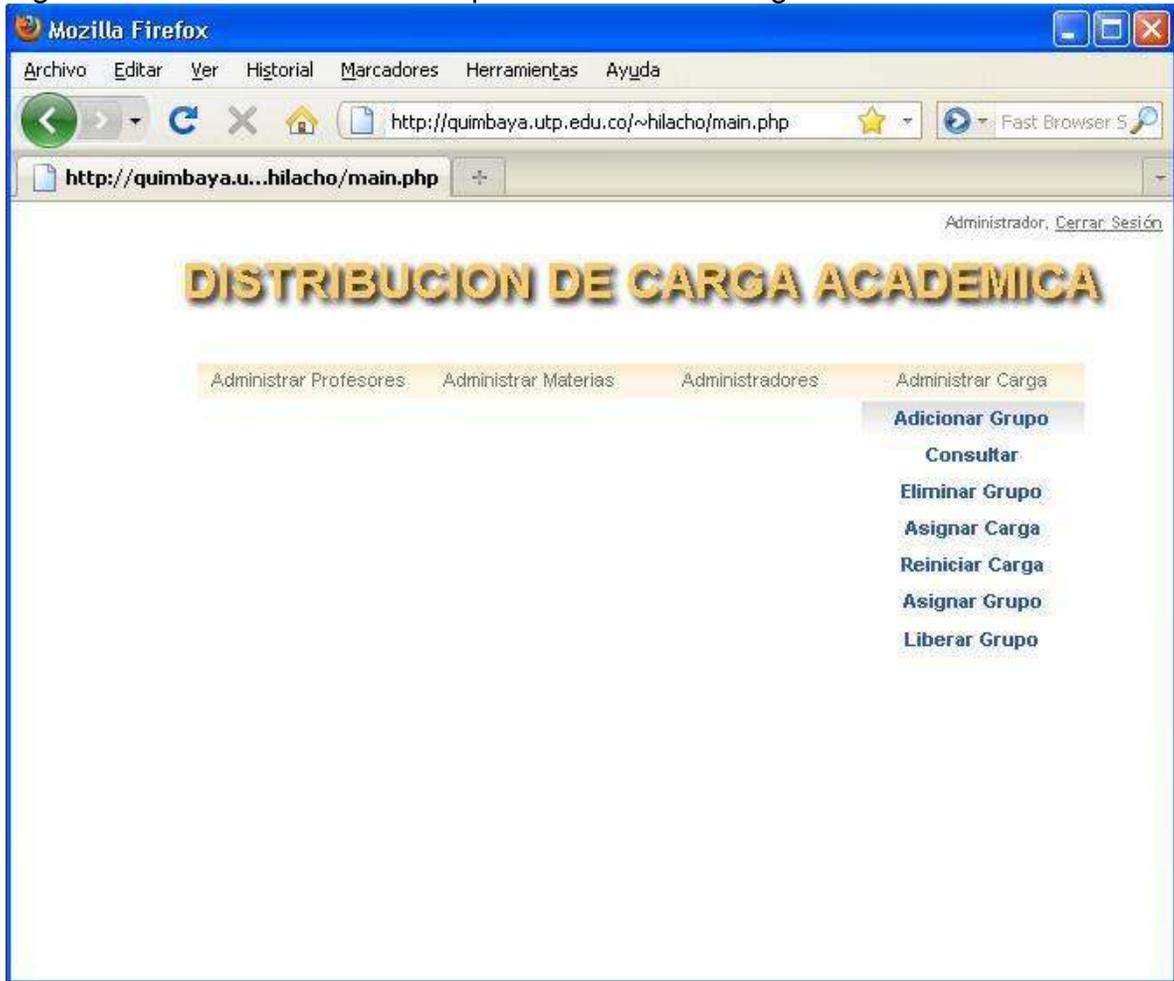
Fuente: los autores.

Figura 104. Ventana Menú Principal – Administrar Usuarios (Administradores)



Fuente: los autores.

Figura 105. Ventana Menú Principal – Administrar Carga



Fuente: los autores.

Figura 106. Ventana Adicionar Profesor

Administrador, [Cerrar Sesión](#)

ADICIONAR PROFESOR

Código:	<input type="text"/>
Nombre:	<input type="text"/>
Tipo de contrato:	<input type="text"/>
Habilitado en:	<input type="text"/>
Cambiar disponibilidad:	<input checked="" type="checkbox"/>

Todos los campos son obligatorios.

Fuente: los autores.

Figura 107. Ventana Modificar Profesor - Seleccionar Profesor



Fuente: los autores.

Figura 108. Ventana Modificar Profesor

Administrador, [Cerrar Sesión](#)

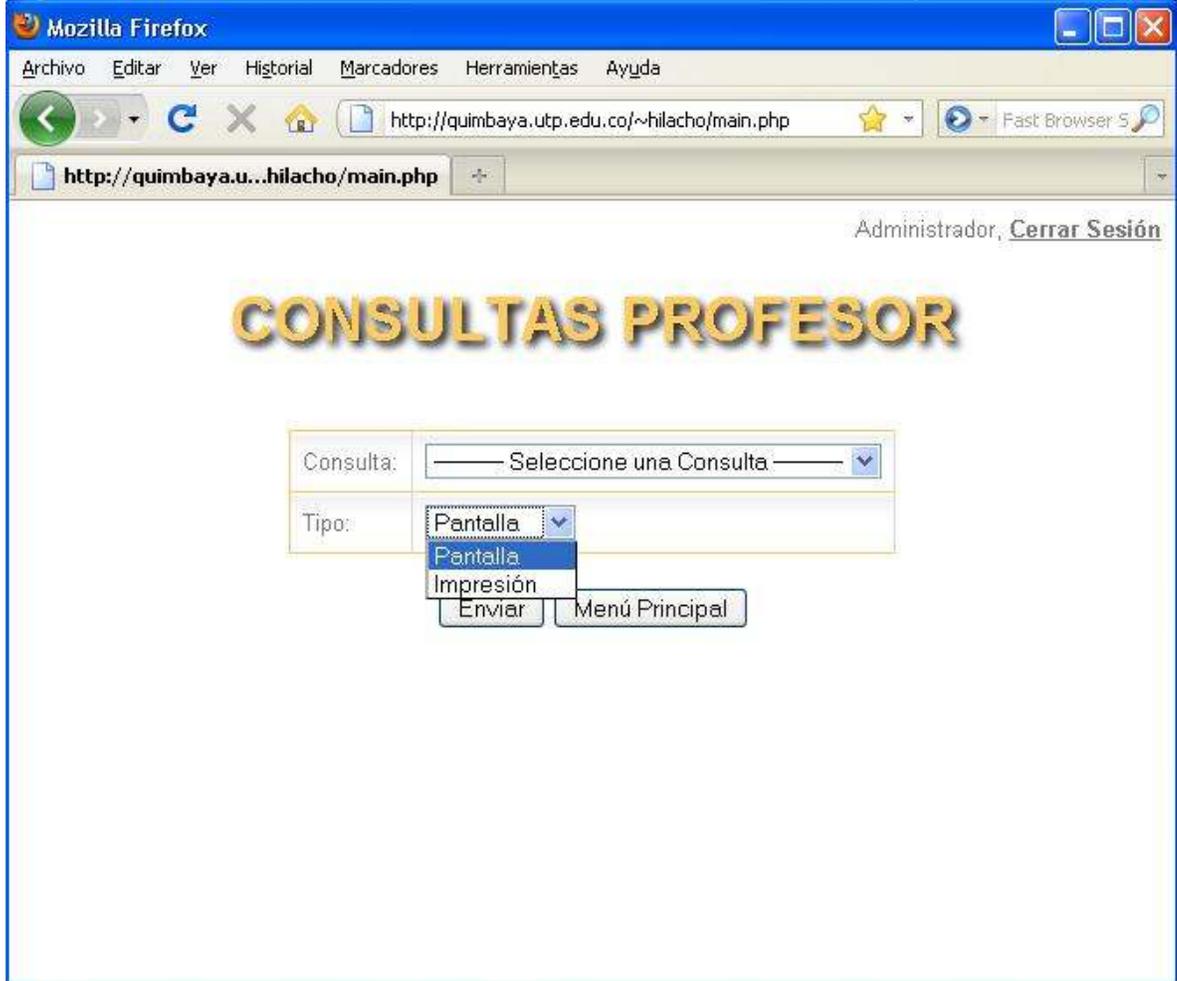
MODIFICAR PROFESOR

Código:	42105115
Nombre:	LOPEZ ECHEVERRY ANA MARIA 
Tipo de contrato:	Transitorio medio tiempo 
Habilitado en:	Jornada diurna 

Todos los campos son obligatorios.

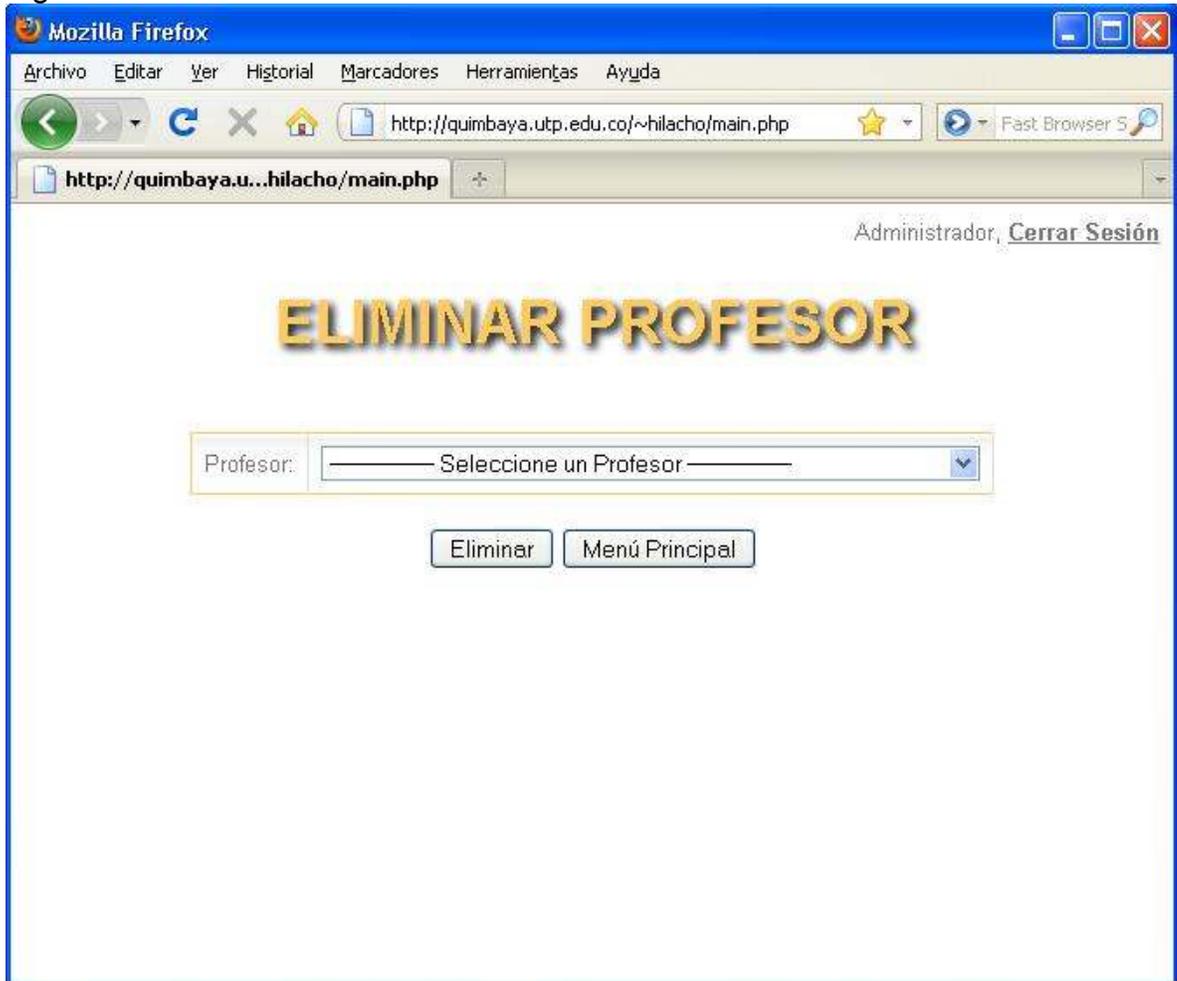
Fuente: los autores.

Figura 109. Ventana Consultas Profesor - Seleccionar Consulta y Vista



Fuente: los autores.

Figura 110. Ventana Eliminar Profesor



Fuente: los autores.

Figura 111. Ventana Horario Profesor - Seleccionar Profesor



Fuente: los autores.

Figura 112. Ventana Horario Profesor – Horario

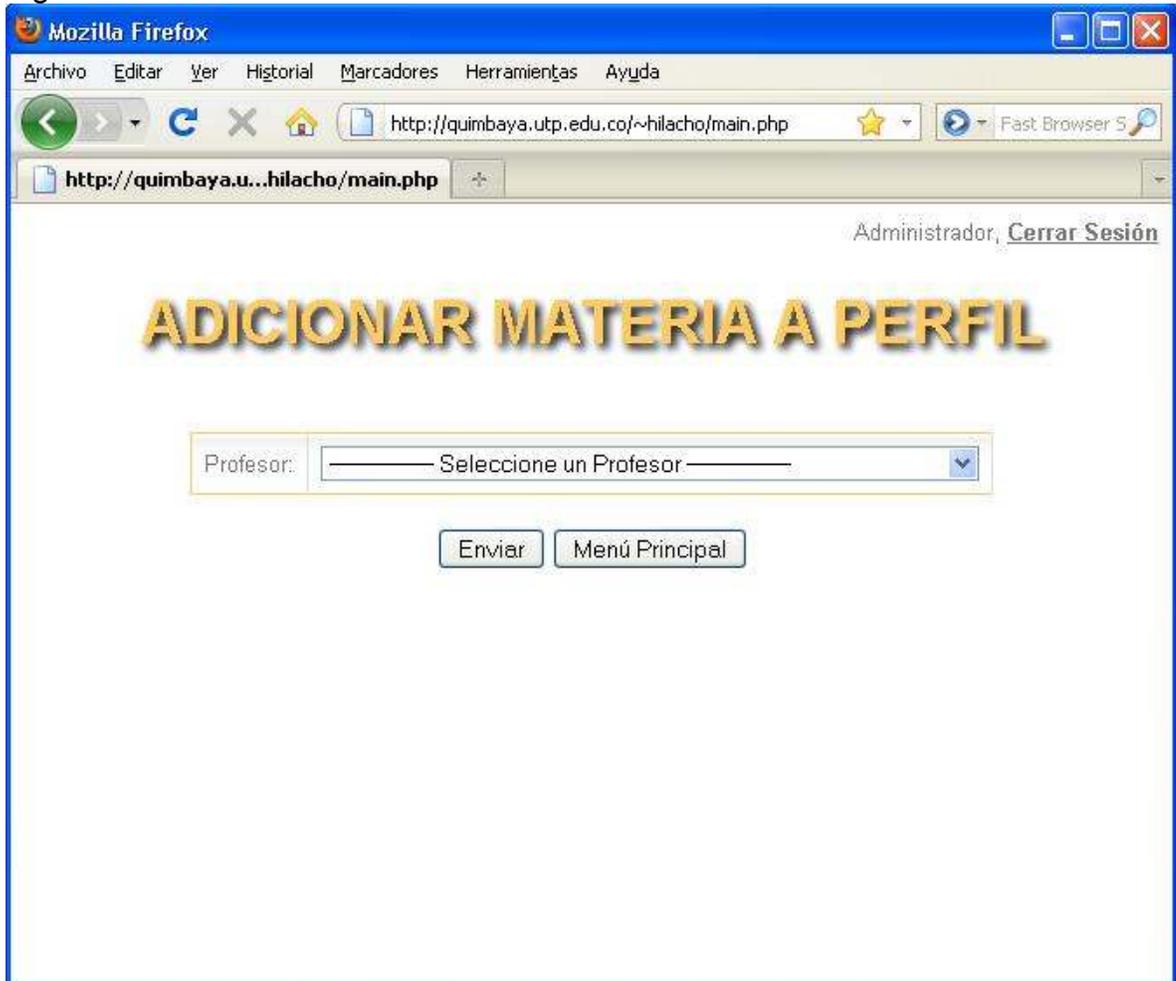
Administrador, [Cerrar Sesión](#)

HORARIO PROFESOR

Profesor	LOPEZ ECHEVERRY ANA MARIA		Horas Jornada Diurna	4		
Dedicación	Transitorio medio tiempo		Horas Jornada Especial	0		
Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
6:00 - 7:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7:00 - 8:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8:00 - 9:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9:00 - 10:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10:00 - 11:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> IS962-2	<input checked="" type="checkbox"/> IS962-2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11:00 - 12:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> IS962-2	<input checked="" type="checkbox"/> IS962-2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12:00 - 13:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13:00 - 14:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14:00 - 15:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15:00 - 16:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:00 - 17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17:00 - 18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18:00 - 19:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> IS066-1	<input checked="" type="checkbox"/> IS066-1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19:00 - 20:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> IS066-1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20:00 - 21:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21:00 - 22:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fuente: los autores.

Figura 113. Ventana Adicionar Materia a Perfil - Seleccionar Profesor



Fuente: los autores.

Figura 114. Ventana Adicionar Materia a Perfil - Seleccionar Materia

Administrador, [Cerrar Sesión](#)

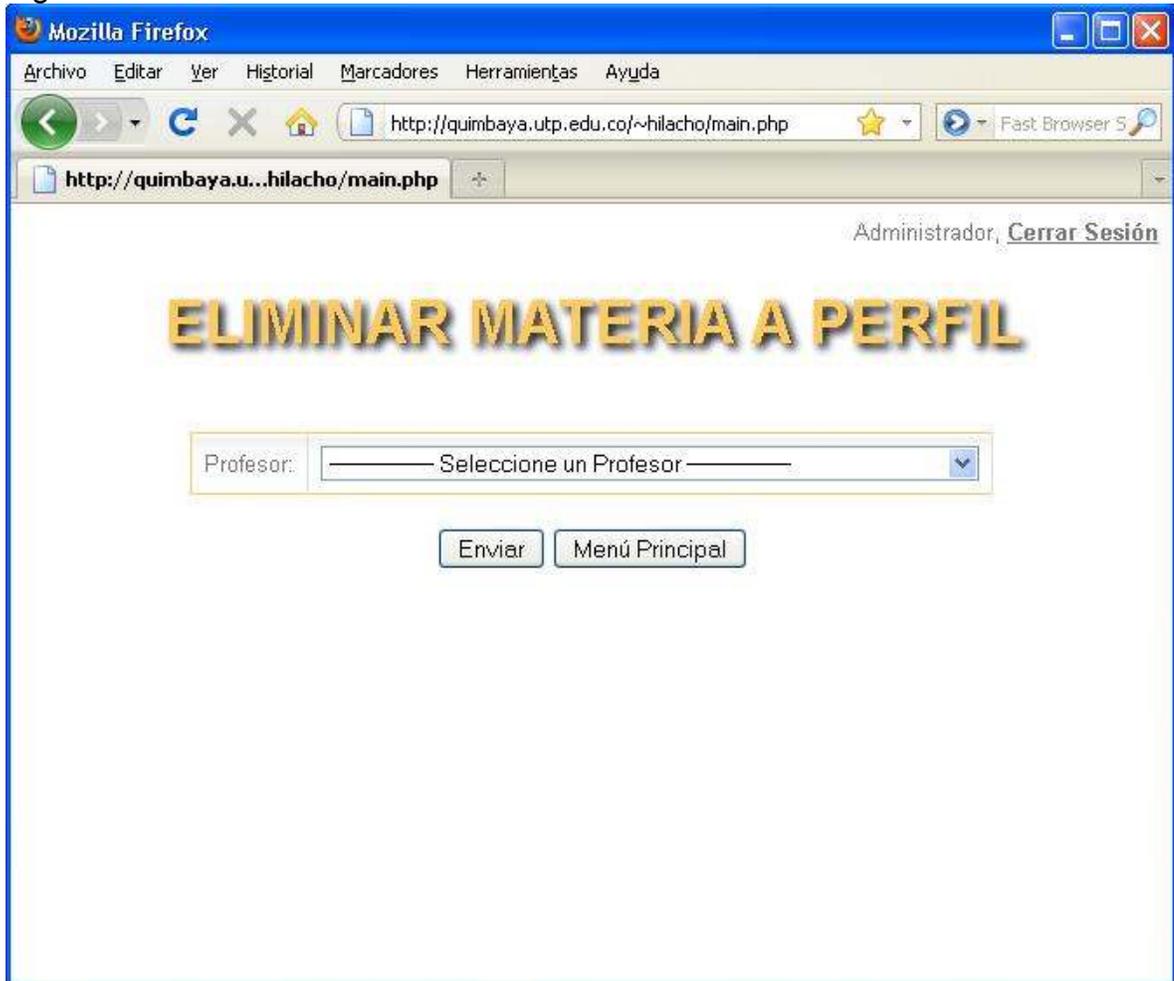
ADICIONAR MATERIA A PERFIL

Profesor	LOPEZ ECHEVERRY ANA MARIA
<input type="checkbox"/>	IS105 - Programación I
<input type="checkbox"/>	IS133 - Informática I
<input type="checkbox"/>	IS142 - Desarrollo del Pensamiento Lógico
<input type="checkbox"/>	IS182 - Técnicas de la Comunicación Oral y Escrita
<input type="checkbox"/>	IS193 - Introducción a la Informática
<input type="checkbox"/>	IS -
<input type="checkbox"/>	IS -
<input type="checkbox"/>	IS -
<input type="checkbox"/>	IS923 - Comunicaciones III
<input type="checkbox"/>	IS924 - Arquitectura Cliente/Servidor
<input type="checkbox"/>	IS933 - Gerencia Institucional
<input type="checkbox"/>	IS943 - Sistemas Expertos
<input type="checkbox"/>	IS013 - Administración de Sistemas de Información
<input type="checkbox"/>	IS023 - Auditoría de Sistemas
<input type="checkbox"/>	IS031 - Constitución Política
<input type="checkbox"/>	IS042 - Planeación Estratégica de Sistemas
<input type="checkbox"/>	IS053 - Gerencia de Proyectos

*Solo se muestran las materias que el profesor no tiene en su perfil.

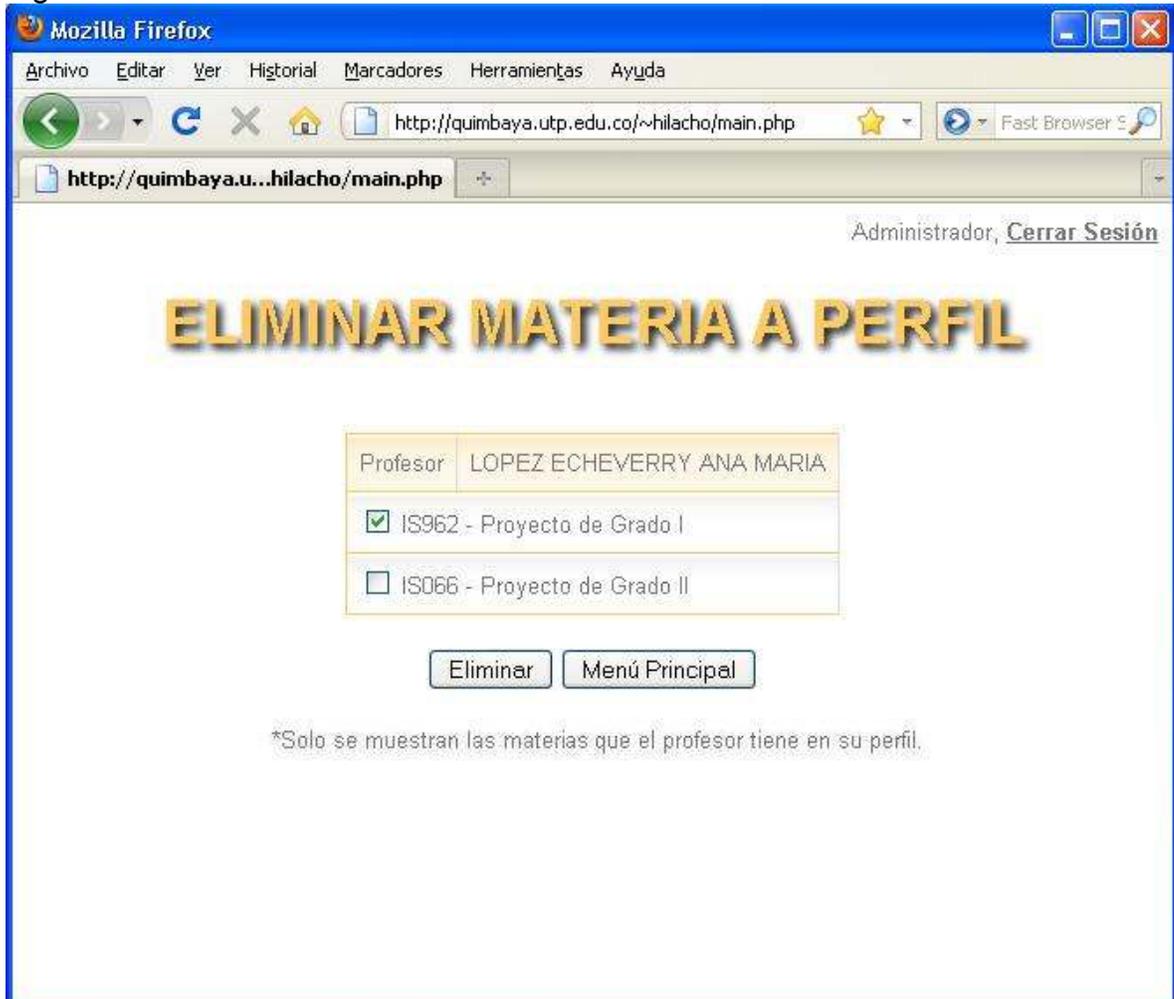
Fuente: los autores.

Figura 115. Ventana Eliminar Materia a Perfil - Seleccionar Profesor



Fuente: los autores.

Figura 116. Ventana Eliminar Materia a Perfil - Seleccionar Materia



Fuente: los autores.

Figura 117. Ventana Adicionar Materia

Administrador, [Cerrar Sesión](#)

ADICIONAR MATERIA

Código:	<input type="text" value="IS"/>
Nombre:	<input type="text"/>
Tipo:	<input type="text" value="Teórica"/>
Créditos:	<input type="text"/>
Intensidad Horaria:	<input type="text"/>
Semestre:	<input type="text"/>

Todos los campos son obligatorios.

Fuente: los autores.

Figura 118. Ventana Modificar Materia - Seleccionar Materia



Fuente: los autores.

Figura 119. Ventana Modificar Materia

Administrador, [Cerrar Sesión](#)

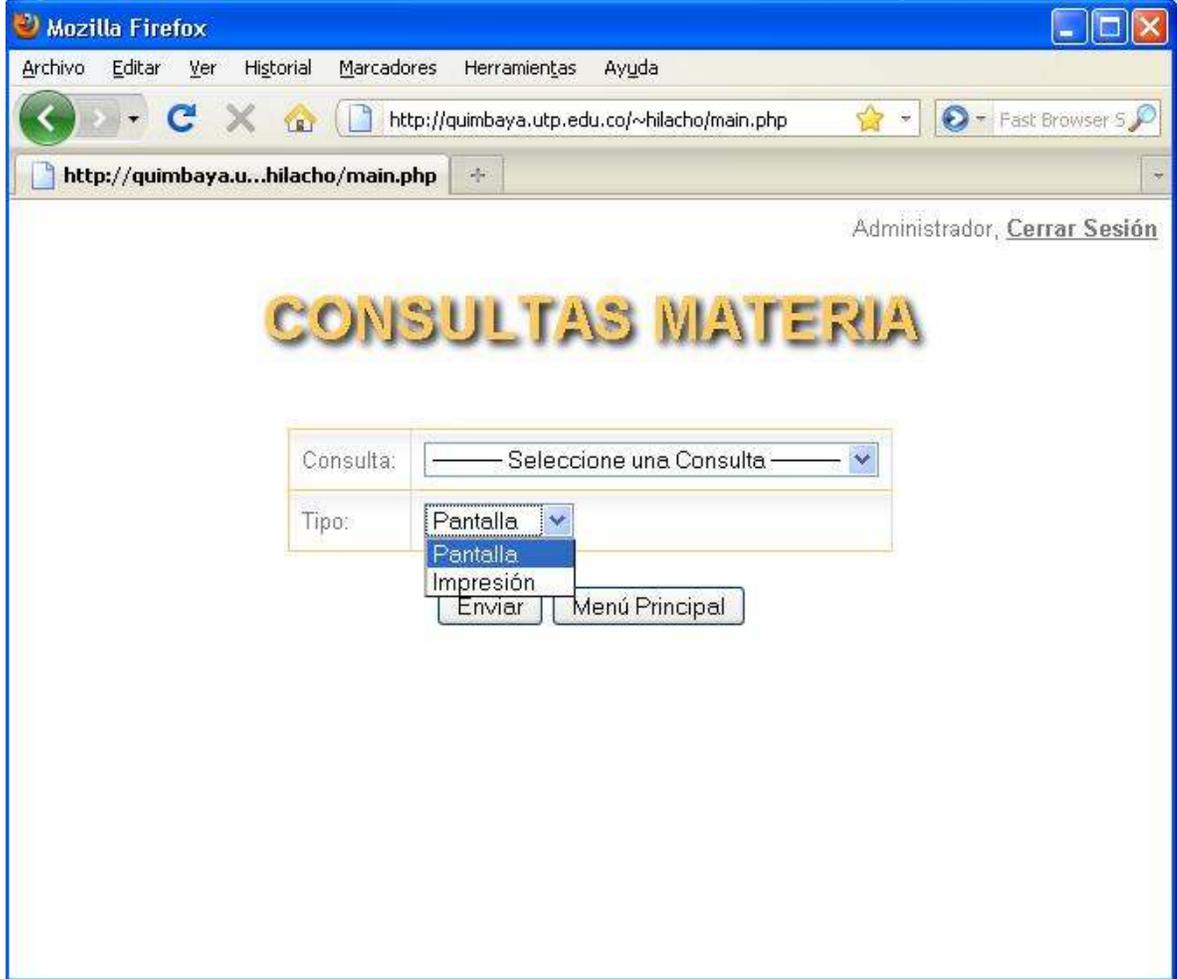
MODIFICAR MATERIA

Código:	IS923
Nombre:	<input type="text" value="Comunicaciones III"/>
Tipo:	<input type="text" value="Teórico-Práctica"/>
Créditos:	3
Intensidad Horaria:	<input type="text" value="4"/>
Semestre:	9

Todos los campos son obligatorios.

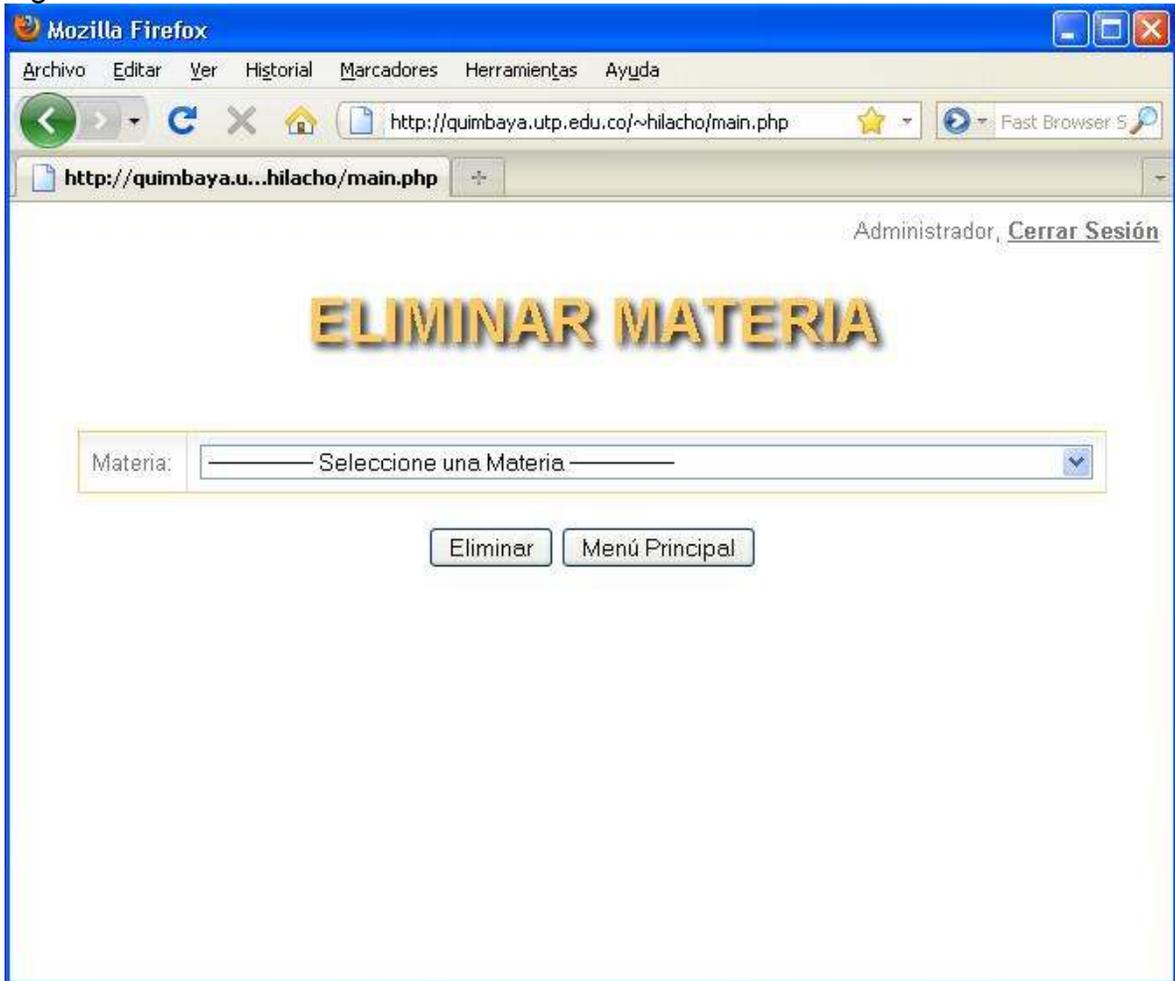
Fuente: los autores.

Figura 120. Ventana Consultas Materia - Seleccionar Consulta y Vista



Fuente: los autores.

Figura 121. Ventana Eliminar Materia



Fuente: los autores.

Figura 122. Ventana Adicionar Administrador

The image shows a screenshot of a Mozilla Firefox browser window. The address bar displays the URL `http://quimbaya.utp.edu.co/~hilacho/main.php`. The page content includes a header with the text "Administrador, [Cerrar Sesión](#)". The main heading is "ADICIONAR ADMINISTRADOR" in large, bold, yellow letters. Below the heading is a form with four input fields, each with a question mark icon to its right:

Login:	<input type="text"/>	?
Nombre:	<input type="text"/>	?
Clave:	<input type="text"/>	?
Confirmar clave:	<input type="text"/>	?

Below the form are two buttons: "Adicionar" and "Menú Principal". At the bottom of the form area, there is a note: "Todos los campos son obligatorios."

Fuente: los autores.

Figura 123. Ventana Modificar Usuario - Seleccionar Usuario



Fuente: los autores.

Figura 124. Ventana Modificar Usuario

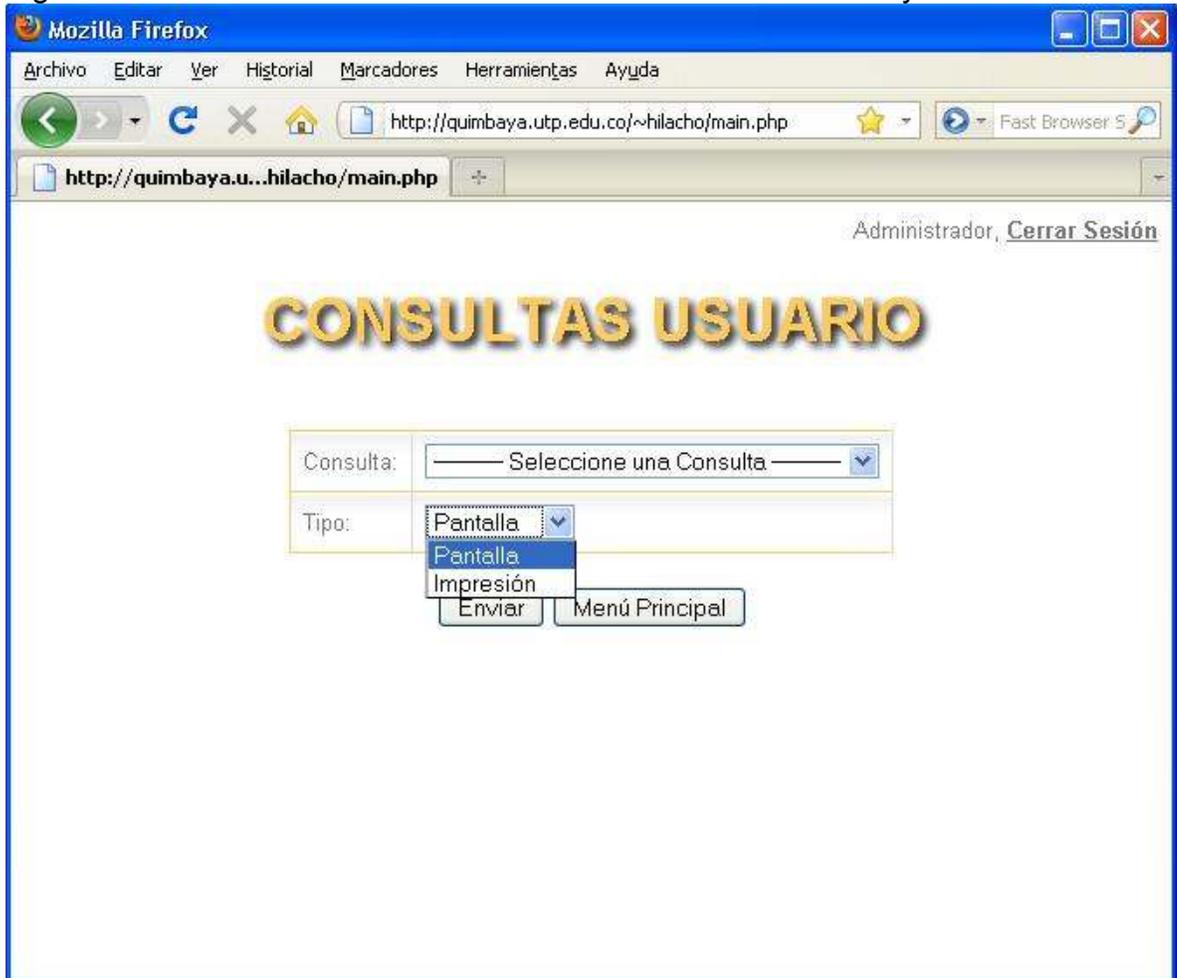
Administrador, [Cerrar Sesión](#)

MODIFICAR USUARIO

Login:	admin
Nombre:	<input type="text" value="Administrador"/>
Cambiar clave?	<input type="checkbox"/>
Clave actual:	<input type="password"/>
Clave nueva:	<input type="password"/>
Confirmar clave:	<input type="password"/>

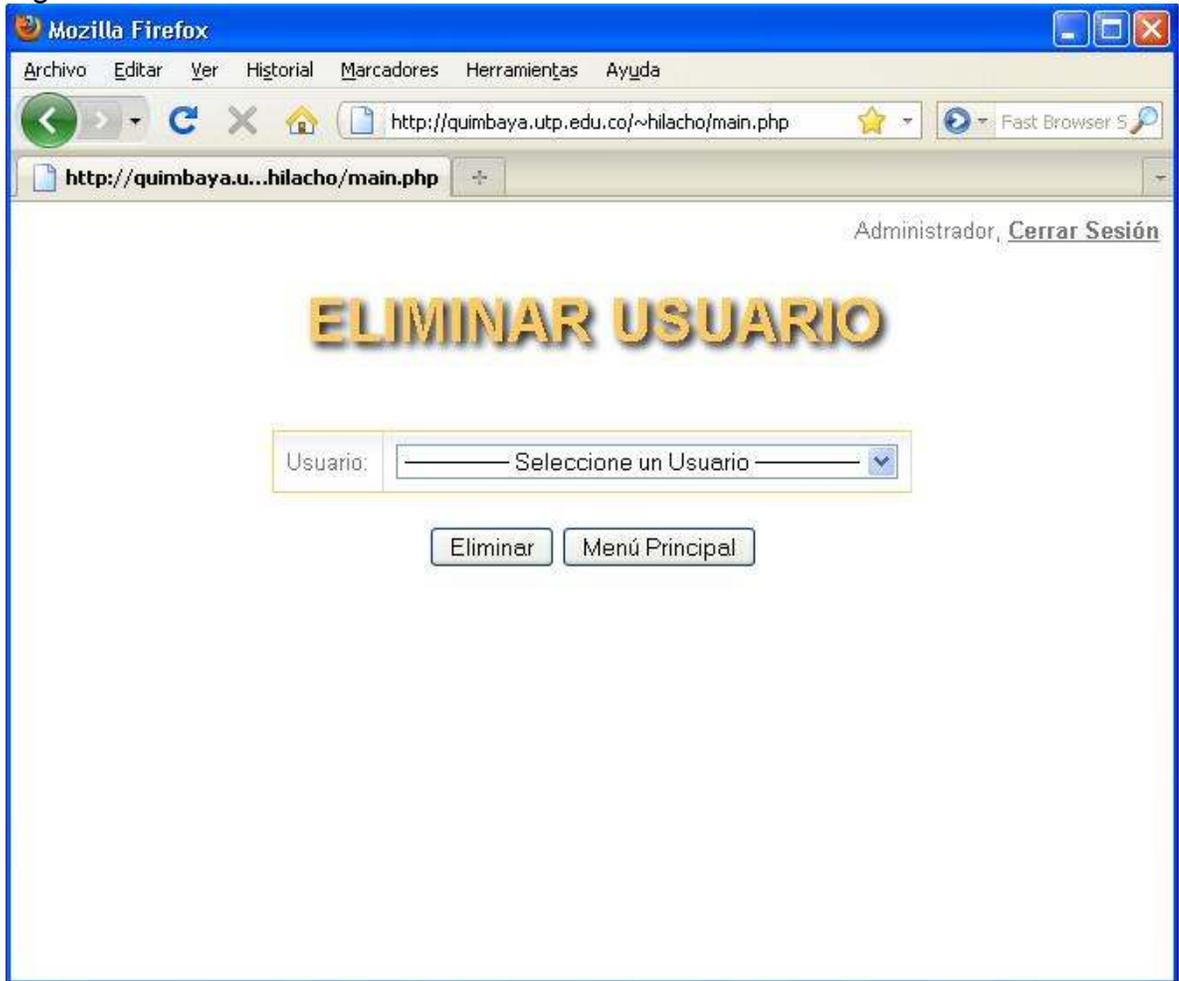
Fuente: los autores.

Figura 125. Ventana Consultas Usuario - Seleccionar Consulta y Vista



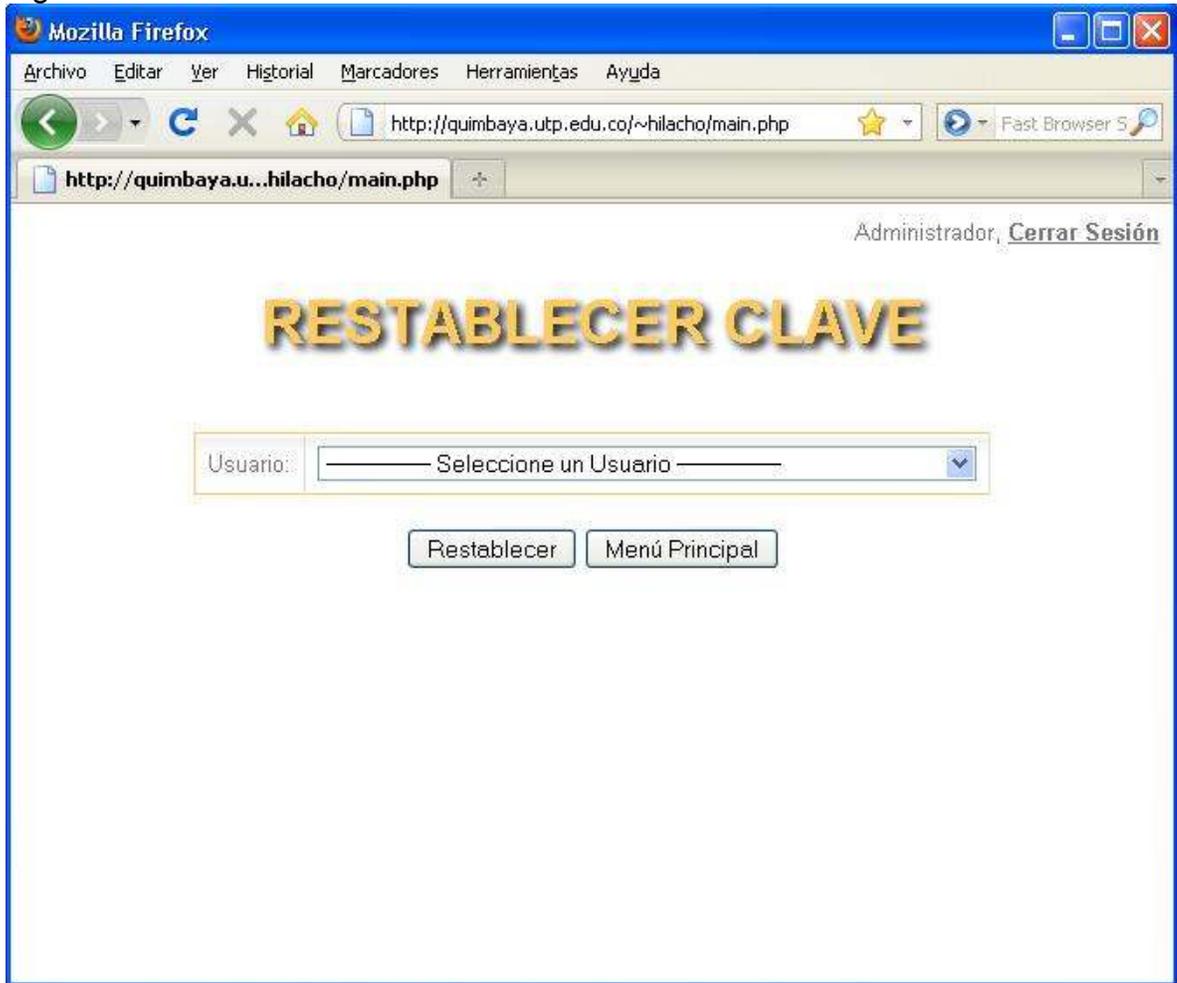
Fuente: los autores.

Figura 126. Ventana Eliminar Usuario



Fuente: los autores.

Figura 127. Ventana Restablecer Clave - Seleccionar Usuario



Fuente: los autores.

Figura 128. Ventana Adicionar Grupo



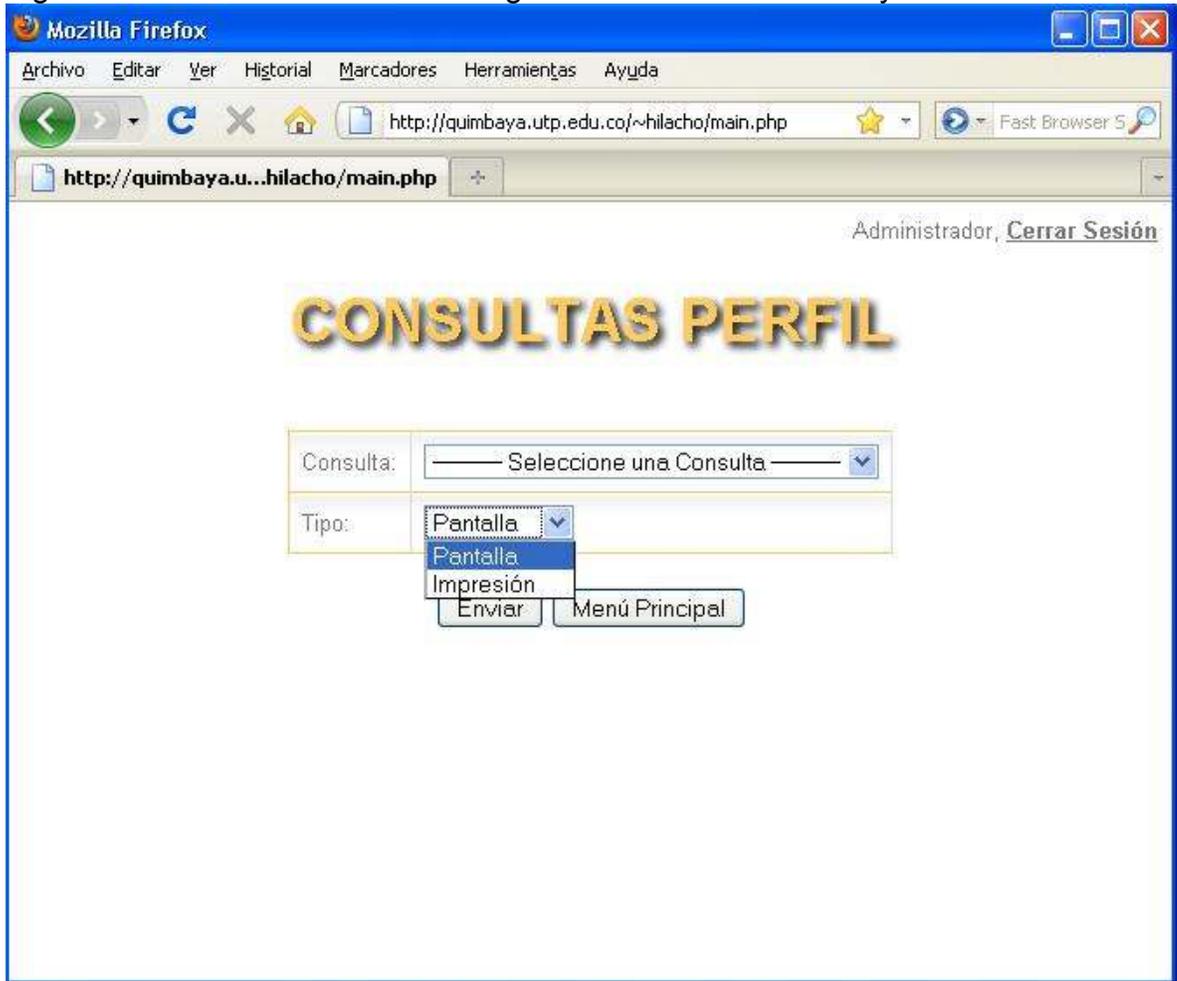
The image shows a screenshot of a Mozilla Firefox browser window displaying a web application interface. The browser's address bar shows the URL `http://quimbaya.utp.edu.co/~hilacho/main.php`. The page title is "ADICIONAR GRUPO" in large, bold, yellow letters. In the top right corner, there is a link for "Administrador" and a "Cerrar Sesión" button. The main content area contains a form with the following fields:

Materia:	IS714 - Ingenieria de Software I
Grupo:	2
Jornada:	Diurna
Número de bloques:	2
Bloque1	Día Hora Duración
Bloque2	Día Hora Duración

At the bottom of the form, there are two buttons: "Adicionar" and "Menú Principal".

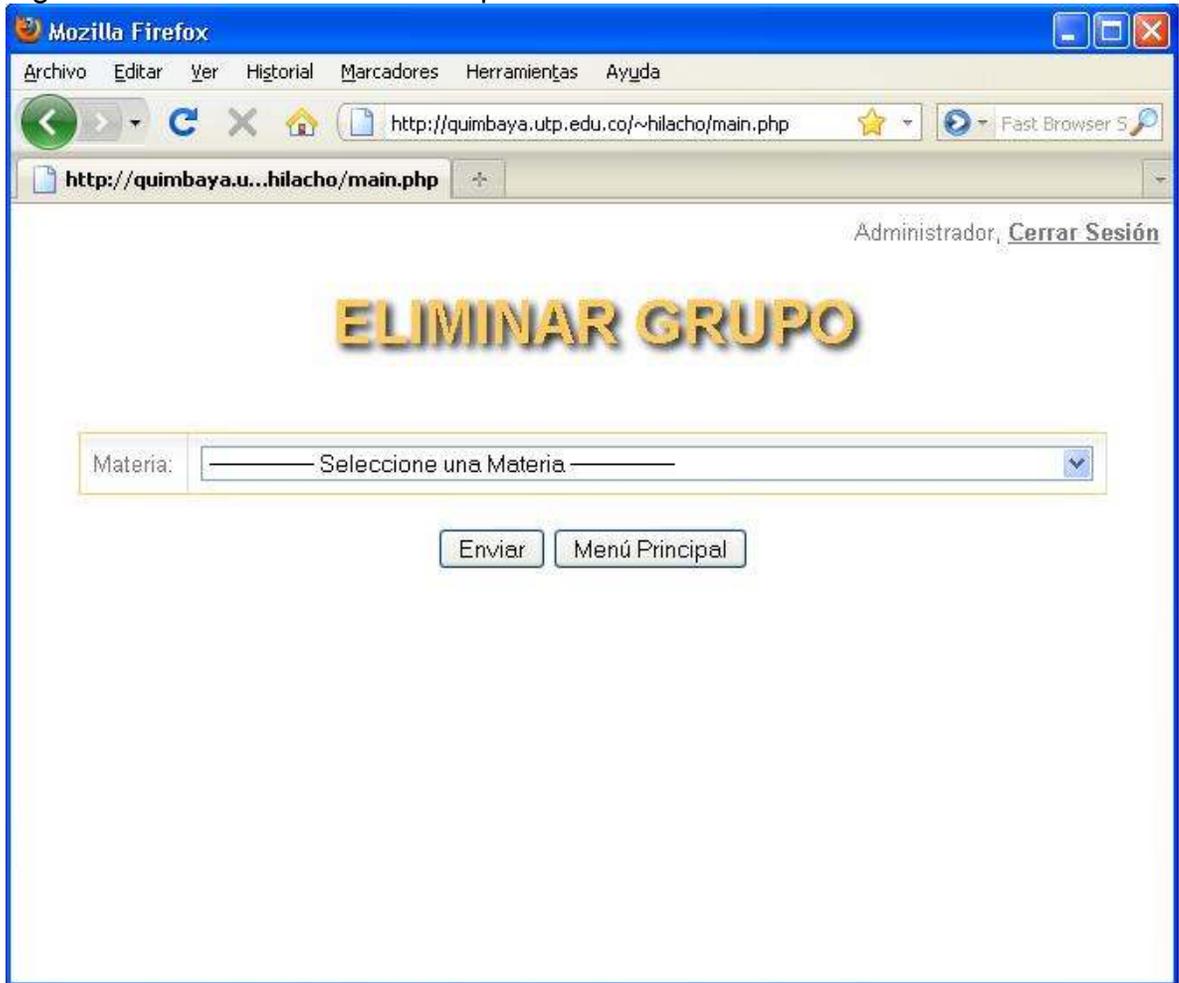
Fuente: los autores.

Figura 129. Ventana Consultas Carga - Seleccionar Consulta y Vista



Fuente: los autores.

Figura 130. Ventana Eliminar Grupo – Seleccionar Materia



Fuente: los autores.

Figura 131. Ventana Eliminar Grupo – Seleccionar Grupo

Administrador, [Cerrar Sesión](#)

ELIMINAR GRUPO

Código	Materia	Grupo	Horario	Profesor	Jornada	Eliminar
IS013	Administración de Sistemas de Información					
1			Martes 6,2 Miércoles 6,2		Diurna	-
2			Martes 14,2 Miércoles 14,2		Diurna	-
101			Miércoles 20,2 Viernes 18,2		Especial	-

[Menú Principal](#) [Regresar](#)

Fuente: los autores.

Figura 132. Ventana Asignación de Carga - Seleccionar jornada y categoría

Administrador, [Cerrar Sesión](#)

ASIGNACION DE CARGA

Jornada:	— Seleccione una Jornada —
Sobrecarga:	<input checked="" type="checkbox"/>

Elija las categorías de asignación.

Categoría
<input type="checkbox"/> Planta
<input type="checkbox"/> Transitorio tiempo completo
<input checked="" type="checkbox"/> Transitorio medio tiempo
<input type="checkbox"/> Catedrático

Fuente: los autores.

Figura 133. Ventana Asignación de Carga – Lista de Profesores

Administrador, [Cerrar Sesión](#)

ASIGNACION DE CARGA

Incluido	Profesor	Contrato	Carga Diurna	Carga Especial	Tope Contrato	Tope Sobrecarga
<input checked="" type="checkbox"/>	JARAMILLO VILLEGAS JOSE ALFREDO	Transitorio medio tiempo	5	0	10	8
<input checked="" type="checkbox"/>	LOPEZ CARLOS ANDRES	Transitorio medio tiempo	10	0	10	8
<input checked="" type="checkbox"/>	LOPEZ ECHEVERRY ANA MARIA	Transitorio medio tiempo	4	0	10	8
<input checked="" type="checkbox"/>	MORALES PEÑA HUGO HUMBERTO	Transitorio medio tiempo	8	0	10	8

Fuente: los autores.

Figura 134. Ventana Asignación de Carga – Resultados

Administrador, [Cerrar Sesión](#)

RESULTADOS ASIGNACION DE CARGA

Profesor	Contrato	Carga diurna	Carga especial	Tope Contrato	Tope Sobrecarga
JARAMILLO VILLEGAS JOSE ALFREDO	Transitorio medio tiempo	5	0	10	8
LOPEZ CARLOS ANDRES	Transitorio medio tiempo	10	0	10	8
LOPEZ ECHEVERRY ANA MARIA	Transitorio medio tiempo	4	0	10	8
MORALES PEÑA HUGO HUMBERTO	Transitorio medio tiempo	18	0	10	8

[Regresar](#) [Menú Principal](#)

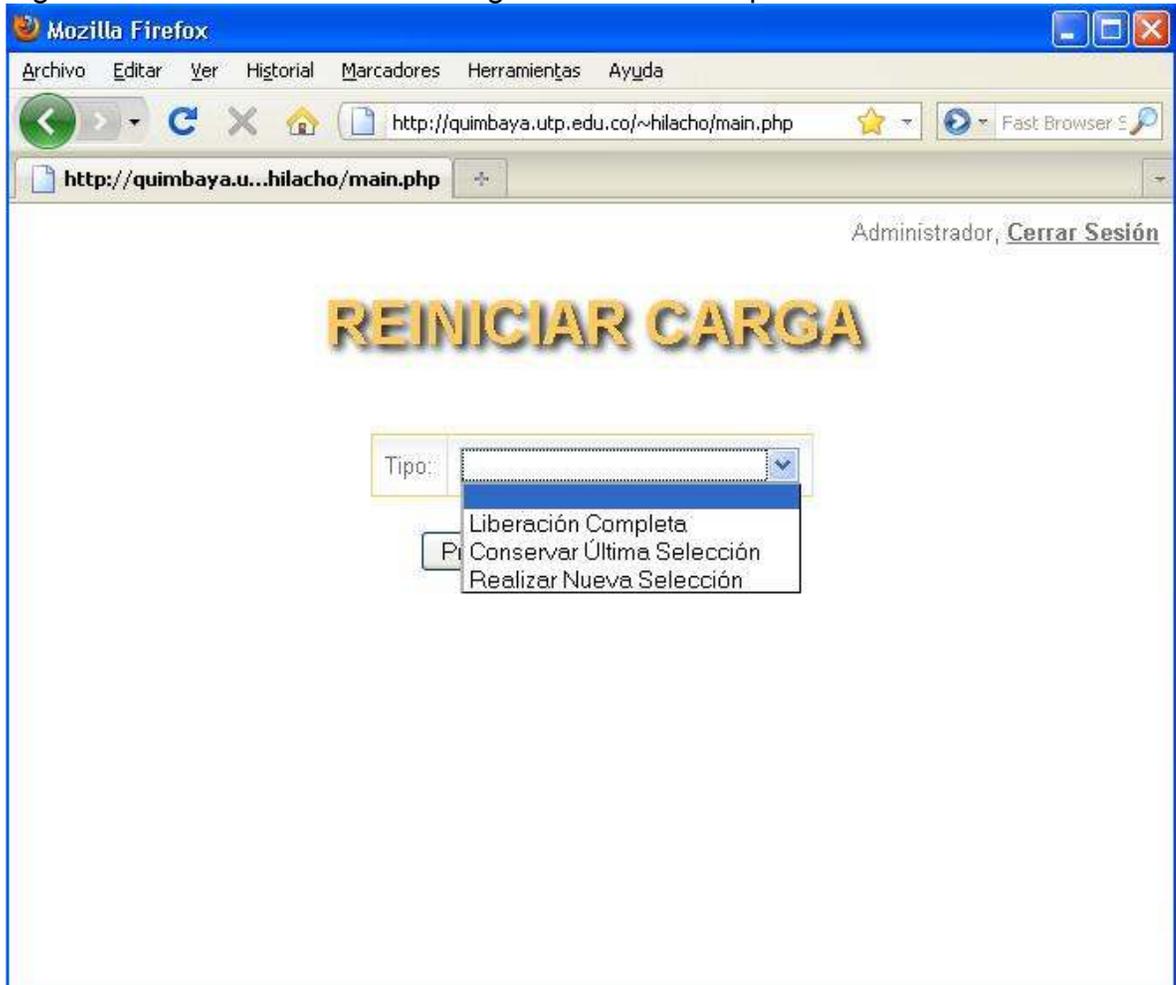
Fuente: los autores.

Figura 135. Ventana Reiniciar Carga



Fuente: los autores.

Figura 136. Ventana Reiniciar Carga – Seleccionar tipo



Fuente: los autores.

Figura 137. Ventana Asignar Grupo



Fuente: los autores.

Figura 138. Ventana Liberar Grupo

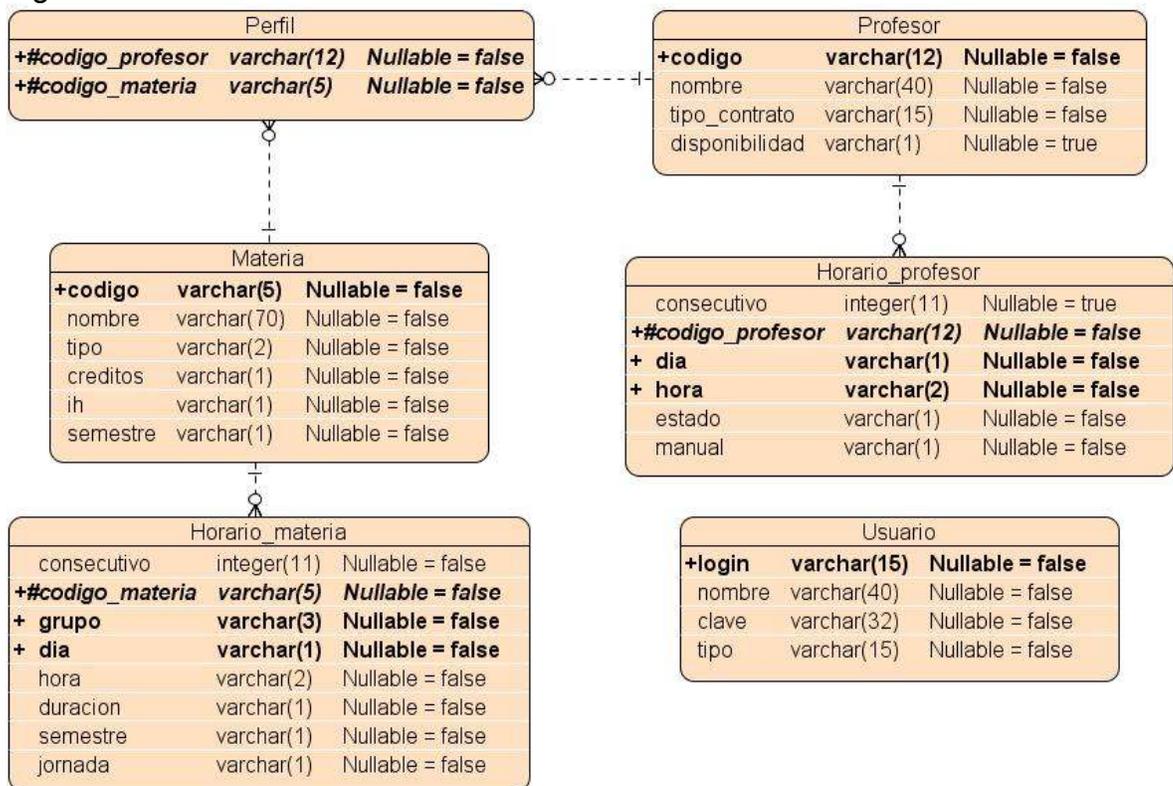


Fuente: los autores.

8.5. DISEÑO DE LA BASE DE DATOS

8.5.1. Modelo Entidad-Relación

Figura 139. Modelo Entidad-relación



Fuente: los autores.

8.5.2. Scripts

- **DB Scripts**

```
create table if not exists usuario(  
login varchar(15) primary key,  
nombre varchar(40) not null,  
clave varchar(32) not null,  
tipo varchar(15) not null  
);
```

```
create table if not exists materia(  
codigo varchar(5) primary key,  
nombre varchar(70) not null,  
tipo varchar(2) not null,  
creditos varchar(1) not null,  
ih varchar(1) not null,  
semestre varchar(1) not null  
)engine=innodb;
```

```
create table if not exists profesor(  
codigo varchar(12) primary key,  
nombre varchar(40) not null,  
tipo_contrato varchar(15) not null,  
disponibilidad varchar(1) not null  
)engine=innodb;
```

```
create table if not exists perfil(  
codigo_profesor varchar(12) not null,  
codigo_materia varchar(5) not null,  
primary key (codigo_profesor,codigo_materia),  
index(codigo_profesor),  
index(codigo_materia),  
foreign key(codigo_profesor) references profesor(codigo),  
foreign key(codigo_materia) references materia(codigo)  
)engine=innodb;
```

```
create table horario_profesor(  
consecutivo int,  
codigo_profesor varchar(12),  
dia varchar(1),  
hora varchar(2),  
estado varchar(1) default '0',  
manual varchar(1) default '0',  
primary key (codigo_profesor,dia,hora),  
index(codigo_profesor),  
foreign key(codigo_profesor) references profesor(codigo)  
)engine=innodb;
```

```
create table if not exists horario_materia(  
consecutivo int not null,
```

```
codigo_materia varchar(5),
grupo varchar(3),
dia varchar(1),
hora varchar(2) not null,
duracion varchar(1) not null,
semestre varchar(1) not null,
jornada varchar(1) not null,
primary key(codigo_materia, grupo, dia),
index(codigo_materia),
foreign key(codigo_materia) references materia(codigo)
)engine=innodb;

insert into usuario
values('admin', 'Administrador', '21232f297a57a5a743894a0e4a801fc3', 'admini
strador');
```

8.5.3. Descripción de las tablas

- **Nombre de la tabla: usuario**

En esta tabla se almacena toda la información referente a los usuarios habilitados para usar la aplicación.

Tabla 41. Descripción de la tabla usuario

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
login	varchar(15)	NO	SI			Almacena el login del usuario.
nombre	varchar(40)	NO				Almacena el nombre del usuario.
clave	varchar(32)	NO				Almacena la clave del usuario.
tipo	varchar(15)	NO				Almacena el tipo de usuario.

Fuente: los autores.

- **Nombre de la tabla: materia**

En esta tabla se almacena toda la información referente a las materias de las cuales se generaran los grupos correspondientes.

Tabla 42. Descripción de la tabla materia

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
codigo	varchar(5)	NO	SI			Almacena el código de la materia.
nombre	varchar(70)	NO				Almacena el nombre de la materia.
tipo	varchar(2)	NO				Almacena el tipo de materia.

Tabla 42. (Continuación)

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
creditos	varchar(1)	NO				Almacena el número de créditos que posee la materia.
lh	varchar(1)	NO				Almacena el número de horas semanales que se dictan de la materia en cuestión.
semestre	varchar(1)	NO				Almacena el semestre al que pertenece la materia.

Fuente: los autores.

- **Nombre de la tabla: perfil**

En esta tabla se almacena toda la información referente a los perfiles. Los perfiles obedecen a una clasificación de las materias.

Tabla 43. Descripción de la tabla perfil

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
codigo_profesor	varchar(12)	NO	SI			Almacena el código del profesor.
codigo_materia	varchar(5)	NO	SI			Almacena el código de la materia.

Fuente: los autores.

- **Nombre de la tabla: profesor**

En esta tabla se almacena toda la información referente a profesores.

Tabla 44. Descripción de la tabla profesor

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
Codigo	varchar(12)	NO	SI			Almacena el código del profesor.
nombre	varchar(40)	NO				Almacena el nombre del profesor.
tipo_contrato	varchar(15)	NO				Almacena el tipo de contrato establecido con el profesor en términos de la dedicación.
disponibilidad	varchar(1)	NO				Sirve para indicarle al sistema la disponibilidad horaria de los docentes.

Fuente: los autores.

- **Nombre de la tabla: horario_profesor**

Esta tabla guarda la parte de la información de un grupo con datos referentes al profesor que la dictara.

Tabla 45. Descripción de la tabla horario_profesor

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
consecutivo	int(11)	SI				Almacena un número que establece que registros pertenecen a un grupo y con cuales registros esta relacionado en la tabla horario_materia.

Tabla 45. (Continuación)

codigo_profesor	varchar (12)	NO	SI		Profesor (código)	Identificador de profesor.
Dia	varchar(1)	NO	SI			Almacena el día en el que fue programada la hora.
Hora	varchar(2)	NO	SI			Almacena la hora.
Estado	varchar(1)	NO		0		Almacena el estado de la hora.
Manual	varchar(1)	NO		0		Sirve para indicar si el registro fue insertado automática o manualmente.

Fuente: los autores.

- **Nombre de la tabla: horario_materia**

Esta tabla guarda la parte de la información de un grupo con datos referentes a la materia a la cual pertenece.

Tabla 46. Descripción de la tabla horario_materia

Campo	Tipo	Null	Llave Primaria	Valor por defecto	Llave Foránea	Descripción
consecutivo	int(11)	NO				Almacena un número que establece que registros pertenecen a un grupo y con cuales registros esta relacionado en la tabla horario_profesor
codigo_materia	varchar(5)	NO	SI		Materia (código)	Identificador de materia.
Grupo	varchar(3)	NO	SI			Almacena el número de grupo programado.

Tabla 46. (Continuación)

dia	varchar(1)	NO	SI			Almacena el día para el que fue programado el bloque.
hora	varchar(2)	NO				Almacena la hora de inicio del bloque.
duracion	varchar(1)	NO				Almacena la duración del bloque.
semestre	varchar(1)	NO				Almacena a que semestre pertenece la materia.
jornada	varchar(1)	NO				Almacena a que jornada pertenece el grupo.

Fuente: los autores.

9. CONCLUSIONES

- Se puso a disposición de la universidad una herramienta útil que permite disminuir la complejidad del proceso de asignación de la carga académica.
- Después de combinar los datos y aplicando técnicas de programación con restricciones sobre ellos, se logró que la aplicación generara una o varias alternativas para que el usuario elija la(s) que le permita(n) una mejor distribución de la carga académica.
- Si por razones de fuerza mayor la disponibilidad de un profesor se ve afectada, el software permite registrar en un punto específico del proceso de ajuste de carga, las horas en las que estará inhábil, por supuesto ceñido al reglamento de la universidad, lo que se traduce en un proceso más subjetivo y participativo por parte del profesorado.
- Al disminuir el tiempo empleado en el proceso de ajuste de carga, el directivo verá aumentada la disponibilidad en su labor administrativa.
- El paradigma de Programación con Restricciones permitió resolver el problema planteado inicialmente.
- La programación con restricciones es una técnica que se ajusta muy bien a la resolución de este tipo de problemas, puesto que las características del modelo matemático formulado guardan una similitud con las del modelo propio de la técnica, lo que se significa una reducción en la complejidad del proceso de traducción de un modelo al otro.
- El hecho de contar con un manual de usuario interactivo y entendible para el usuario, permite una mejor comprensión de la forma como se gestionan los procesos que brinda el software de distribución y una fácil adaptación a él.
- Se aplicaron y afianzaron los conocimientos y las habilidades adquiridas durante toda la carrera de ingeniería de Sistemas y Computación, especialmente de Programación, Investigación de Operaciones, Bases de Datos e Ingeniería del Software.
- Gracias a la investigación y a la consulta de muchas fuentes de información relacionadas con el tema, se logró recopilar la teoría suficiente para sustentar el problema tratado.
- Siguiendo las metodologías especificadas en el capítulo siete (XP y Cascada), se realizaron todas las etapas para el desarrollo del software.

10. RECOMENDACIONES

- Sugerimos que las personas encargadas de la manipulación del software de distribución de la carga académica informen a quien corresponda de las posibles mejoras que puedan realizarse, de esta manera la aplicación se consolida y se va ajustando a los requerimientos del usuario.
- Impulsar el uso de esta aplicación no solo en el programa de Ingeniería de Sistemas y Computación, sino en todos los programas de la Universidad, capacitar al profesorado para que puedan ingresar y extraer información que el sistema actual no proporciona o por lo menos no de una manera clara.
- Estudiar la forma de adicionar la variable de salones, buscando la manera de reducir el número de combinaciones que puedan resultar, a la hora de poner el algoritmo de generación de grupos en funcionamiento, teniendo en cuenta las características de las aulas y laboratorios en cuanto a las herramientas o equipos que posea, para realizar una asignación acorde a los requerimientos de las materias.
- Como recomendación general, buscar la manera de poner en funcionamiento los proyectos de software que realicen los estudiantes de nuestro programa al finalizar su carrera, encontrar un espacio donde puedan ser útiles ya que muchos de ellos no salen del ámbito de tesis de grado y no dejan de representar un requisito de graduación para el estudiante.

11. FUTURAS LÍNEAS DE INVESTIGACIÓN

- La programación con restricciones, como se mencionó en el capítulo siete, es una técnica que puede ser aplicada en muchas áreas del conocimiento, pero guardando la relación con el campo en el que se desarrolla el proyecto, se pueden realizar trabajos en planificación o scheduling con variables como la asignación de recursos o alguna especificación en la precedencia de tareas. Un software con estas características, podría ser usado en una empresa de producción con el fin de optimizar el tiempo, la materia prima, el uso de las máquinas y el personal.
- Consultar en diferentes universidades la forma en que abordan el problema de la distribución de la carga académica, y medir las dificultades para migrar o ajustar el software actual a las necesidades o requerimientos de otras instituciones.
- Emplear herramientas para resolver problemas de programación con restricciones que usen algoritmos de búsqueda diferentes a los tradicionales, como las redes neuronales o los algoritmos genéticos y comparar los resultados; de igual manera estudiar cómo se puede mejorar el modelo matemático y por ende del algoritmo de generación del mismo, para encontrar horarios más subjetivos.

BIBLIOGRAFÍA

- [1] Alvarez, Sara, “Modelo entidad-relación”. Internet: (<<http://www.desarrolloweb.com/articulos/modelo-entidad-relacion.html> >).
- [2] Apt, Krzysztof R. Wallace, M. Diciembre 2006; Constraint Logic Programming Using Eclipse. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780511345852>
- [3] Apt, Krzysztof. 2003; Principles of Constraint Programming. Nueva York. Cambridge University Press. [libro en línea] Disponible desde Internet en: <http://www.cambridge.org/9780521825832>
- [4] aSc Time Tables. Disponible en <<http://www.asctimetables.com/>> [última versión 2008]
- [5] Association for Constraint Programming [en línea] <<http://slash.math.unipd.it/acp/>> [Marzo de 2005]
- [6] Barber, Federico. Salido A. Miguel. Introducción a la programación con restricciones [en línea]. <<http://iajournal.aepia.org/aepia/Uploads/20/72.pdf>> [2003]
- [7] Booch Grady. Jacobson Ivar. Rumbaugh James. El lenguaje unificado de modelado. Manual de referencia. Edición en español. España. Addison Wesley, 2000.
- [8] Gruber & Petters de Austria. gpUntis. Disponible en < <http://www.gpuntis.com/timetabler.htm>> [última versión 2009]
- [9] Grupo CF Developer de Colombia. Software de Gestión Escolar (DocCF). Disponible en <<http://www.grupocfdeveloper.com/>> [última versión 2008]
- [10] Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial. N° 20, pp:13-29, ISSN:1137-3601, [2003]
- [11] INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Trabajos escritos: presentación y referencias bibliográficas. Sexta actualización. Bogotá: ICONTEC, 2008 110.
- [12] Instituto para la sostenibilidad computacional <<http://www.computational-sustainability.org/crocs09>>, [Septiembre 20 de 2009]

- [13] Introducción a la programación con restricciones
<<http://iajournal.aepia.org/aepia/Uploads/20/72.pdf>>,[2003]
- [14] Hamilton K., Miles R. Abril 2006. Learning UML 2.0. Ed. O'Reilly. Sebastopol. [Libro en línea] Disponible desde Internet en: <http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>
- [15] Organización Nicta y su proyecto
<http://nicta.com.au/research/projects/constraint_programming_platform>,
[última actualización Agosto 27 de 2009]
- [16] Peñalara Software de España. Generador de Horarios para Centros de Enseñanza (GHC).
Disponible en < <http://www.penalara.com/index.asp>> [última versión 2008]
- [17] Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill.
- [18] Rossi F. Van Beek P. Walsh T. Agosto 2006; Handbook of Constraint Programming. Holanda. Elsevier. [libro en línea] Disponible desde Internet en: http://www.elsevier.com/wps/find/bookaudience.cws_home/708863/description
- [19] The 14th International conference on principles and practice of constraint programming – Program [en línea]
<<http://www.cs.mu.oz.au/cp2008/program.html>> [Septiembre de 2008]
- [20] Unified Modeling Language. UML: Resource Page. Internet:
(<<http://www.uml.org/>>)