



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

**"SISTEMA DE PLANEACIÓN AVANZADA Y ADMINISTRACIÓN DE RECURSOS
DE UNIDADES ACADÉMICAS"**

TRABAJO DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

ESPECIALIZACIÓN

SISTEMAS TECNOLÓGICOS

Y SISTEMAS DE INFORMACIÓN

Presentado por

CHRISTIAN SALAZAR GONZÁLEZ

KATHERINE VITERI AYALA

GUAYAQUIL - ECUADOR

2005

AGRADECIMIENTO

Agradecemos a cada una de las personas que hicieron posible la realización de esta tesis, de manera muy especial:

A nuestro Padre celestial, nuestras familiares y amigos fuente de nuestra inspiración.

Al Ing. Fabricio Echeverría, nuestro Director de Tesis, por su presión oportuna y la confianza depositada en nosotros para el desarrollo y fin exitoso de este proyecto de tesis, sus valiosos comentarios y dedicación a la hora de compartir sus experiencias y conocimiento con nosotros.

DEDICATORIA

Dedicamos la obtención de nuestros títulos a Dios, por su infinita bondad, por brindarnos el privilegio de tener a nuestros padres principales forjadores de nuestra identidad que con sus invaluable esfuerzos y sacrificios sirvieron como pilares fundamentales para nuestro crecimiento académico y desarrollo personal.

TRIBUNAL DE GRADUACIÓN

.....
ING. CARLOS MONSALVE
DECANO FIEC

.....
ING. FABRICIO ECHEVERRÍA
DIRECTOR DE TESIS

.....
ING. CARLOS JORDÁN
MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta tesis de grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica Del Litoral"

(Reglamento de Graduación de La Espol)

.....
CHRISTIAN SALAZAR GONZÁLEZ

.....
KATHERINE VITERI AYALA

RESUMEN

Los algoritmos genéticos son parte de la computación evolutiva, un área creciente de la inteligencia artificial que esta basada en la evolución natural biológica, y pueden adaptarse a la resolución de múltiples problemas. Como es de esperarse, los algoritmos genéticos están basados en la teoría de la evolución de Darwin, y son apropiados para resolver problemas donde el dominio de la solución pueda resultar extenso. Dada la dificultad de encontrar una solución óptima en la planificación de horarios de entidades académicas que cumpla con los requerimientos iniciales como el uso restrictivo de la disponibilidad de horarios por profesor, aulas y la asignación de recursos utilizando un procedimiento manual, se plantea un sistema lo suficientemente flexible, capaz de generar un conjunto de posibles soluciones clasificadas por un nivel de aptitud, que darán paso a la obtención de una solución óptima que transforme los datos de manera eficiente y confiable.

ÍNDICE GENERAL

RESUMEN	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE TABLAS	XIV
ÍNDICE DE ILUSTRACIONES.....	XVI
INTRODUCCIÓN	XXVIII
1. ANÁLISIS DEL PROBLEMA	32
1.1 PLANTEAMIENTO DEL PROBLEMA	32
1.2 SITUACIÓN ACTUAL DE LA PLANIFICACIÓN.....	35
1.3 RESULTADOS ESPERADOS.....	37
1.4 DEFINICIÓN DEL PROBLEMA.....	38
1.5 DIFERENTES ASPECTOS DE LA SOLUCIÓN.....	40
2. SISTEMAS DE PLANEACIÓN AVANZADA.....	41

2.1	¿QUE ES UN SISTEMA DE PLANEACIÓN AVANZADA?	42
2.2	TECNOLOGÍAS RELACIONADAS CON LOS APS.....	43
2.2.1	INVENTARIO JUSTO A TIEMPO- JUST IN TIME INVENTORY (JIT)	43
2.2.2	SISTEMAS DE REQUERIMIENTOS DE MATERIALES - MATERIAL REQUIREMENTS PLANNING (MRP)	45
2.2.3	PLANEACIÓN DE RECURSOS DE MANUFACTURA - MANUFACTURING RESOURCE PLANNING (MRP II).....	46
2.2.4	SOFTWARE DE PLANEACIÓN DE CAPACIDADES FINITAS - FINITE CAPACITY SCHEDULING SOFTWARE (FCS)	46
2.2.5	SISTEMAS DE PLANEACIÓN DE RECURSOS EMPRESARIALES - ENTERPRISE RESOURCE PLANNING (ERP)	47
2.2.6	CADENAS DE SUMINISTROS - SUPPLY CHAIN MANAGEMENT (SCM).....	51
2.3	VENTAJAS DE LOS APS SOBRE LOS ERP	52
3.	ALGORITMOS GENÉTICOS	53
3.1	INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS.....	53
3.2	ELEMENTOS BÁSICOS	55
3.2.1	FUNCIÓN DE ADAPTACIÓN (FITNESS)	56
3.2.2	CRITERIO DE INICIALIZACIÓN	56
3.2.3	CRITERIO DE PARADA.....	56
3.2.4	ESPACIO DE SOLUCIÓN.....	57
3.2.5	PORCENTAJE DE CRUCE.....	57
3.2.6	PORCENTAJE DE MUTACIÓN	58

3.2.7	TAMAÑO DE LA POBLACIÓN.....	59
3.2.8	NÚMERO DE GENERACIONES.....	59
3.3	ANÁLISIS DEL ALGORITMO.....	59
3.3.1	MÉTODOS DE REPRESENTACIÓN.....	60
3.3.2	MÉTODOS DE SELECCIÓN.....	61
3.3.2.1	SELECCIÓN ELITISTA	62
3.3.2.2	SELECCIÓN PROPORCIONAL A LA APTITUD	62
3.3.2.3	SELECCIÓN DE LA RULETA	62
3.3.2.4	SELECCIÓN ESCALADA.....	62
3.3.2.5	SELECCIÓN POR TORNEO.....	63
3.3.2.6	SELECCIÓN POR RANGO.....	63
3.3.2.7	SELECCIÓN GENERACIONAL	63
3.3.2.8	SELECCIÓN POR ESTADO ESTACIONARIO.....	64
3.3.2.9	SELECCIÓN JERÁRQUICA.....	64
3.3.3	CRUZAMIENTO	64
3.3.3.1	CRUCE BÁSICO	65
3.3.3.2	CRUCE MULTIPUNTO	65
3.3.3.3	CRUCE UNIFORME.....	66
3.3.4	MUTACIÓN.....	66
3.3.4.1	MUTACIÓN DE BITS.....	67
3.3.4.2	MUTACIÓN MULTIBIT	67

3.3.4.3	MUTACIÓN DE GEN.....	68
3.3.4.4	MUTACIÓN MULTIGEN.....	68
3.3.4.5	MUTACIÓN DE INTERCAMBIO	68
3.3.5	LIMITACIONES	69
3.4	MÍNIMOS LOCALES Y SOLUCIÓN ÓPTIMA	71
3.5.	APLICACIÓN DE ALGORITMOS GENÉTICOS EN LA SOLUCIÓN DEL PROBLEMA	72
3.6.	ANÁLISIS COMPARATIVO Y SELECCIÓN DE HEURÍSTICA PARA LA SOLUCIÓN DEL PROBLEMA.	73
3.6.1	RECOCIDO SIMULADO.	74
3.6.2	BÚSQUEDA TABÚ.....	75
3.6.3	GRASP.	76
3.6.4	REDES NEURONALES.	77
4.	DESARROLLO DE LA APLICACIÓN.....	79
4.1	RECOPIACIÓN DE DATOS	80
4.2	REPRESENTACIÓN HEURÍSTICA	82
4.2.1	SELECCIÓN DE LA GENERACIÓN INICIAL	88
4.2.2	SELECCIÓN DE LOS CROMOSOMAS PADRES.....	93
4.2.3	REPRESENTACIÓN DE UN STRING	93
4.2.4	OPERACIONES DE CRUZAMIENTO.....	94
4.2.5	OPERACIONES DE MUTACIÓN	97
4.3.6	CÁLCULO DEL FITNESS.....	98

4.3.7	TRATAMIENTO DE EXCEPCIONES	98
4.3	MODELO DE ANÁLISIS	99
4.3.1	MODELOS ESTÁTICOS	99
4.3.1.1	DISEÑO CONCEPTUAL DE LA BASE DE DATOS.....	99
4.3.1.2	MODELO DE ANÁLISIS DE OBJETOS.....	100
4.3.1.3	DESCRIPCIÓN DE CLASES	102
4.3.2	MODELOS DINÁMICOS	110
4.3.2.1	DESCRIPCIÓN DE ACTORES	110
4.3.2.2	LISTADO DE CASOS DE USO	111
4.3.2.3	DIAGRAMA DE CASOS DE USOS.....	120
4.3.2.4	ESCENARIOS.....	121
4.3.2.5	DIAGRAMAS DE INTERACCIÓN DE OBJETOS	138
4.3.2.6	FLUJO DE VENTANA Y LAYOUT.....	140
4.4	MODELOS DE DISEÑOS.....	156
4.4.1	MODELOS ESTÁTICOS	156
4.4.1.1	DISEÑO DE LA BASE DE DATOS	157
4.4.2	MODELO DE DISEÑO DE OBJETOS	158
5.	DEFINICIÓN DEL PRODUCTO	199
5.1	ANÁLISIS FODA.....	199
5.2	SEGMENTACIÓN DE MERCADO	204
5.3	COSTO DE LA INVERSIÓN.....	205

5.3.1	COSTO DEL LICENCIAMIENTO	205
5.3.1.1	LICENCIA DEL SISTEMA	205
5.3.1.2	LICENCIAMIENTO DE LA BASE DE DATOS	207
5.3.1.3	LICENCIAMIENTO SISTEMA OPERATIVO DE SERVIDOR	207
5.3.2	REQUERIMIENTOS MÍNIMOS DEL SISTEMA	209
5.4	COSTOS RELACIONADOS	210
6.	ANÁLISIS DE RESULTADOS	212
6.1.	PLAN DE PRUEBAS	212
6.2.	ANÁLISIS DE RESULTADOS OBTENIDOS.....	214
6.3	MEJORAS AL DESEMPEÑO.....	214
6.4	CONVERGENCIA DE RESULTADOS.....	215
	APÉNDICES	CCXXX
	APÉNDICE A.	DICCIONARIO DE DATOS CCXXXI
	APÉNDICE B.	CODIFICACIÓN CCXXXVII
	APÉNDICE C.	MANUAL DE USUARIO CCLVII
	APÉNDICE D.	MANUAL DE INSTALACIÓN DEL SISTEMA CCLXXXIII
	BIBLIOGRAFÍA	CCLXXXVI

ÍNDICE DE TABLAS

TABLA 2-1: DIFERENCIAS FUNDAMENTALES EN PLANTA TRADICIONAL Y LA PROPUESTA POR JIT[XXXV]	44
TABLA 5-1: SIMULACIÓN CONSIDERANDO UNA GENERACIÓN CON DIEZ CROMOSOMAS	204
TABLA 5-2: COSTO DE LICENCIAMIENTO DEL SISTEMA	206
TABLA 5-3 : COSTOS DE LICENCIAMIENTO BASE DE DATOS PROGRAMA DE LICENCIAMIENTO ACADÉMICO ABIERTO.....	207
TABLA 5-4: COSTOS DE LICENCIAMIENTO SBS SERVER PROGRAMA DE LICENCIAMIENTO COMERCIAL.....	208
TABLA 5-5: COSTOS DE LICENCIAMIENTO WINDOWS SERVER 2003 PROGRAMA DE LICENCIAMIENTO ACADÉMICO. *CONSIDERANDO UN TOTAL DE DOS ACCESOS ...	208
TABLA 5-6: COSTOS DE LICENCIAMIENTO WINDOWS XP PROFESSIONAL PROGRAMA DE LICENCIAMIENTO ACADÉMICO.....	209
TABLA 5-7: REQUERIMIENTOS MÍNIMOS DE HARDWARE.....	210
TABLA 5-8: TOTAL DE COSTOS INCURRIDOS O TCO (TOTAL COST OWNERSHIP)...	210
TABLA 5-9: COSTO MANO DE OBRA CON EL USO DE LA HERRAMIENTA	211
TABLA 5-10: ROI: RETORNO SOBRE LA INVERSIÓN	211
TABLA 5-11: COSTO ACTUAL DE LA MANO DE OBRA.....	211

TABLA 6-1: RESULTADO DEL ALGORITMO CON UNA GENERACIÓN Y UN CROMOSOMA	219
TABLA 6-2: RESULTADO DEL ALGORITMO CON UNA GENERACIÓN Y DIEZ CROMOSOMAS	221
TABLA 6-3: RESULTADO DEL ALGORITMO CON UNA GENERACIÓN Y VEINTE CROMOSOMAS	222
TABLA 6-4: RESULTADO DEL ALGORITMO CON CINCO GENERACIONES Y CUATRO CROMOSOMAS	223
TABLA 6-5: RESULTADO DEL ALGORITMO CON CINCO GENERACIONES Y VEINTE CROMOSOMAS	224
TABLA 6-6: RESULTADO DEL ALGORITMO CON VEINTE GENERACIONES Y VEINTE CROMOSOMAS	225
TABLA 6-7: RESULTADO DEL ALGORITMO CON DIEZ GENERACIONES Y CIENTO CROMOSOMAS	226

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 2-1 RED LOGÍSTICA DE LA CADENA DE SUMINISTROS [XXXVI]	52
ILUSTRACIÓN 3-1: CRUCE BÁSICO	65
ILUSTRACIÓN 3-2: CRUCE MULTIPUNTO	66
ILUSTRACIÓN 3-3: CRUCE UNIFORME.....	66
ILUSTRACIÓN 3-4: MUTACIÓN DE BITS	67
ILUSTRACIÓN 3-5: MUTACIÓN MULTIBITS.....	68
ILUSTRACIÓN 3-6: MUTACIÓN DE INTERCAMBIO	69
ILUSTRACIÓN 3-7: MUTACIÓN DE BARAJADO	69
ILUSTRACIÓN 4-1 ESQUEMA BÁSICO DE LOS ALGORITMOS GENÉTICOS [XXXVII] ...	87
ILUSTRACIÓN 4-2: REPRESENTACIÓN DE LA GENERACIÓN INICIAL.....	88
ILUSTRACIÓN 4-3: REPRESENTACIÓN DE UN CROMOSOMA	89
ILUSTRACIÓN 4-4: MATRIZ DE HORARIO DE DISPONIBILIDAD DE AULAS I/O PROFESORES.	90

ILUSTRACIÓN 4-5: BLOQUES DE HORARIO PREDEFINIDOS	92
ILUSTRACIÓN 4-6: REPRESENTACIÓN EN LA BASE DE DATOS DE LOS BLOQUES DE HORARIO.....	92
ILUSTRACIÓN 4-7: CROMOSOMA 1 PARA DOS PUNTOS DE CRUCE.	94
ILUSTRACIÓN 4-8: CROMOSOMA 2 PARA DOS PUNTOS DE CRUCE.	95
ILUSTRACIÓN 4-9: CROMOSOMA RESULTADO CON DOS PUNTOS DE CRUCE.	95
ILUSTRACIÓN 4-10: CROMOSOMA 1 PARA CRUZAMIENTO HORIZONTAL	96
ILUSTRACIÓN 4-11: CROMOSOMA 2 PARA CRUZAMIENTO HORIZONTAL	96
ILUSTRACIÓN 4-12: CROMOSOMA RESULTADO CON UN PUNTO DE CRUCE.	97
ILUSTRACIÓN 4-13: DISEÑO CONCEPTUAL DE LA BASE DE DATOS	100
ILUSTRACIÓN 4-14: MODELO DE ANÁLISIS DE OBJETOS.....	101
ILUSTRACIÓN 4-15:CLASE CASIGNACION.....	102
ILUSTRACIÓN 4-16: CLASE CCONEXIÓN	102
ILUSTRACIÓN 4-17: CLASE CAULA.....	103
ILUSTRACIÓN 4-18: CLASE CMATERIA	104
ILUSTRACIÓN 4-19: CLASE CHORARIO	105
ILUSTRACIÓN 4-20: CLASE CPARALELO	105
ILUSTRACIÓN 4-21: CLASE CPARAMETROS.....	105
ILUSTRACIÓN 4-22: CLASE CPROFESOR.....	106
ILUSTRACIÓN 4-23: CLASE CPLANIFICACION.....	107
ILUSTRACIÓN 4-24: CLASE CPROFESORMATERIA.....	107
ILUSTRACIÓN 4-25: CLASE CUSUARIO.....	108

ILUSTRACIÓN 4-26: CLASE CGEN	108
ILUSTRACIÓN 4-27: CLASE CCROMOSOMA.....	109
ILUSTRACIÓN 4-28: ACTOR PRIMARIO ADMINISTRADOR.....	110
ILUSTRACIÓN 4-29: ACTOR PRIMARIO DIGITADOR	110
ILUSTRACIÓN 4-30: ACTOR SECUNDARIO INVITADO.....	110
ILUSTRACIÓN 4-31: CASO DE USO 1: CREACIÓN DE NUEVA PLANIFICACIÓN.....	112
ILUSTRACIÓN 4-32: CASO DE USO 2: APERTURA DE PLANIFICACIÓN.....	113
ILUSTRACIÓN 4-33: CASO DE USO 3: ELIMINACIÓN DE PLANIFICACIÓN.....	113
ILUSTRACIÓN 4-34: CASO DE USO 4: GUARDAR COMO DE PLANIFICACIÓN	113
ILUSTRACIÓN 4-35: CASO DE USO 5: CAMBIO DE USUARIO.....	113
ILUSTRACIÓN 4-36: CASO DE USO 6: CAMBIO DE CONTRASEÑA.....	114
ILUSTRACIÓN 4-37: CASO DE USO 7: INGRESO DE USUARIO.....	114
ILUSTRACIÓN 4-38: CASO DE USO 8: MODIFICACIÓN DE USUARIO.....	114
ILUSTRACIÓN 4-39: CASO DE USO 9: ELIMINACIÓN DE USUARIO.....	114
ILUSTRACIÓN 4-40: CASO DE USO 10: INGRESO DE ASIGNACIONES ESPECIALES.....	115
ILUSTRACIÓN 4-41: CASO DE USO 11: INGRESO DE PROFESOR.....	115
ILUSTRACIÓN 4-42: CASO DE USO 12: MODIFICACIÓN DE PROFESOR EXITOSO.....	115
ILUSTRACIÓN 4-43: CASO DE USO 13: ELIMINACIÓN DE PROFESOR.....	115
ILUSTRACIÓN 4-44: CASO DE USO 14: INGRESO DE MATERIA.....	116
ILUSTRACIÓN 4-45: CASO DE USO 15: MODIFICACIÓN DE MATERIA.....	116
ILUSTRACIÓN 4-46: CASO DE USO 16: ELIMINACIÓN DE MATERIA.....	116

ILUSTRACIÓN 4-47: CASO DE USO 17: INGRESO DE AULA.	116
ILUSTRACIÓN 4-48: CASO DE USO 18: MODIFICACIÓN DE AULA.	116
ILUSTRACIÓN 4-49: CASO DE USO 19: ELIMINACIÓN DE AULA.....	117
ILUSTRACIÓN 4-50: CASO DE USO 20: INGRESO DE PARALELO.....	117
ILUSTRACIÓN 4-51: CASO DE USO 21: ASIGNACIÓN DE PARALELO.....	117
ILUSTRACIÓN 4-52: CASO DE USO 22: ELIMIACIÓN DE PARALELO.	117
ILUSTRACIÓN 4-53: CASO DE USO 23: INGRESO DE HORARIO PROFESOR.....	117
ILUSTRACIÓN 4-54: CASO DE USO 24: INGRESO DE HORARIO AULA.	118
ILUSTRACIÓN 4-55: CASO DE USO 25: OBTENER LISTA DE PROFESORES.....	118
ILUSTRACIÓN 4-56: CASO DE USO 26: OBTENER LISTA DE MATERIAS	118
ILUSTRACIÓN 4-57: CASO DE USO 27: OBTENER LISTA DE AULAS.	118
ILUSTRACIÓN 4-58: CASO DE USO 28: OBTENER HORARIOS POR AULA EXITOSO .	118
ILUSTRACIÓN 4-59: CASO DE USO 29: OBTENER HORARIOS POR MATERIA EXITOSO.	
119	
ILUSTRACIÓN 4-60: CASO DE USO 30: OBTENER HORARIOS POR PROFESOR EXITOSO.	119
ILUSTRACIÓN 4-61: CASO DE USO 31: OBTENER REPORTE DE CONFLICTOS.	119
ILUSTRACIÓN 4-62: CASO DE USO 32: INGRESO DE PARÁMETROS INICIALES.....	119
ILUSTRACIÓN 4-63: CASO DE USO 33: OPTIMIZACIÓN.	119
ILUSTRACIÓN 4-64: DIAGRAMA DE CASOS DE USO.....	120
ILUSTRACIÓN 4-65: ESCENARIO 1.1: CREACIÓN DE NUEVA PLANIFICACIÓN EXITOSA.	

ILUSTRACIÓN 4-66: ESCENARIO 2.1: APERTURA DE PLANIFICACIÓN EXITOSA.	123
ILUSTRACIÓN 4-67: ESCENARIO 3.1: ELIMINACIÓN DE PLANIFICACIÓN EXITOSA. ...	124
ILUSTRACIÓN 4-68: ESCENARIO 4.1: GUARDAR COMO DE PLANIFICACIÓN EXITOSA. 124	
ILUSTRACIÓN 4-69: ESCENARIO 5.1: CAMBIO DE USUARIO EXITOSO.	124
ILUSTRACIÓN 4-70: ESCENARIO 6.1: CAMBIO DE CONTRASEÑA EXITOSA.	125
ILUSTRACIÓN 4-71: ESCENARIO 7.1: INGRESO DE USUARIO EXITOSO.	125
ILUSTRACIÓN 4-72: ESCENARIO 8.1: MODIFICACIÓN DE USUARIO EXITOSO.	125
ILUSTRACIÓN 4-73: ESCENARIO 9.1: ELIMINACIÓN DE USUARIO EXITOSO.	126
ILUSTRACIÓN 4-74: ESCENARIO 10.1: INGRESO DE ASIGNACIONES ESPECIALES EXITOSA.	126
ILUSTRACIÓN 4-75: ESCENARIO 11.1: INGRESO DE PROFESOR EXITOSO.	127
ILUSTRACIÓN 4-76: ESCENARIO 11.2: IDENTIFICACIÓN DEL PROFESOR INVALIDA	127
ILUSTRACIÓN 4-77: ESCENARIO 12.1: MODIFICACIÓN DE PROFESOR EXITOSO	127
ILUSTRACIÓN 4-78: ESCENARIO 13.1: ELIMINACIÓN DE PROFESOR EXITOSA.	128
ILUSTRACIÓN 4-79: ESCENARIO 14.1: INGRESO DE MATERIA EXITOSO	128
ILUSTRACIÓN 4-80: ESCENARIO 14.2: ERROR EN EL CÓDIGO DE LA MATERIA.	128
ILUSTRACIÓN 4-81: ESCENARIO 15.1: MODIFICACIÓN DE MATERIA EXITOSO	129
ILUSTRACIÓN 4-82: ESCENARIO 15.2: ERROR EN EL NIVEL DE LA MATERIA.	129
ILUSTRACIÓN 4-83: ESCENARIO 16.1: ELIMINACIÓN DE MATERIA EXITOSO.	129
ILUSTRACIÓN 4-84: ESCENARIO 17.1: INGRESO DE AULA EXITOSO.	130
ILUSTRACIÓN 4-85: ESCENARIO 18.1: MODIFICACIÓN DE AULA EXITOSO.	130

ILUSTRACIÓN 4-86: ESCENARIO 19.1: ELIMINACIÓN DE AULA EXITOSO.	130
ILUSTRACIÓN 4-87: ESCENARIO 20.1: INGRESO DE PARALELO EXITOSO.	131
ILUSTRACIÓN 4-88: ESCENARIO 21.1: ASIGNACIÓN DE PARALELO EXITOSO.	131
ILUSTRACIÓN 4-89: ESCENARIO 22.1: ELIMINACIÓN DE PARALELO EXITOSO.....	132
ILUSTRACIÓN 4-90: ESCENARIO 23.1: INGRESO DE HORARIO PROFESOR EXITOSO. 132	
ILUSTRACIÓN 4-91: ESCENARIO 24.1: INGRESO DE HORARIO AULA EXITOSO.	133
ILUSTRACIÓN 4-92: ESCENARIO 25.1: OBTENER LISTA DE PROFESORES EXITOSA. 133	
ILUSTRACIÓN 4-93: ESCENARIO 26.1: OBTENER LISTA DE MATERIAS EXITOSA.	133
ILUSTRACIÓN 4-94: ESCENARIO 27.1: OBTENER LISTA DE AULAS EXITOSA.	134
ILUSTRACIÓN 4-95: ESCENARIO 28.1: OBTENER HORARIOS POR AULA EXITOSO ..	134
ILUSTRACIÓN 4-96: ESCENARIO 29.1: OBTENER HORARIOS POR MATERIA EXITOSO. 134	
ILUSTRACIÓN 4-97: ESCENARIO 30.1: OBTENER HORARIOS POR PROFESOR EXITOSO.	134
ILUSTRACIÓN 4-98: ESCENARIO 31.1: OBTENER REPORTE DE CONFLICTOS EXITOSO.	135
ILUSTRACIÓN 4-99: ESCENARIO 32.1: INGRESO DE PARÁMETROS INICIALES EXITOSO.	135
ILUSTRACIÓN 4-100: ESCENARIO 33.1: OPTIMIZACIÓN EXITOSA.	136
ILUSTRACIÓN 4-101: ESCENARIO 33.2: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS MAYORES QUE HORAS DISPONIBLES DE AULAS.	136

ILUSTRACIÓN 4-102: ESCENARIO 33.3: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS MAYORES QUE HORAS DISPONIBLES DE LABORATORIOS.....	137
ILUSTRACIÓN 4-103: ESCENARIO 33.4: OPTIMIZACIÓN FALLIDA - HORAS DISPONIBLES DE AULAS MAYOR A LAS HORAS DE ASIGNACIÓN SEMANAL.....	137
ILUSTRACIÓN 4-104: ESCENARIO 33.5: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS A PROFESORES ES MAYOR QUE SUS DISPONIBILIDAD	138
ILUSTRACIÓN 4-105: PANTALLA DE VALIDACIÓN DE USUARIOS	140
ILUSTRACIÓN 4-106: SELECCIÓN O APERTURA DE NUEVAS PLANIFICACIONES.....	141
ILUSTRACIÓN 4-107: INGRESO DE LOS DATOS PARA LA NUEVA PLANIFICACIÓN...	141
ILUSTRACIÓN 4-108: PARÁMETROS ORGANIZACIONALES Y DEL ALGORITMO GENÉTICO.....	142
ILUSTRACIÓN 4-109: VISTA DE PARALELOS ESPECIALES	142
ILUSTRACIÓN 4-110: INGRESO DE PARALELOS ESPECIALES.....	143
ILUSTRACIÓN 4-111: INGRESAR PROFESOR.....	143
ILUSTRACIÓN 4-112: INGRESAR MATERIA.....	144
ILUSTRACIÓN 4-113: INGRESAR AULA	144
ILUSTRACIÓN 4-114: INGRESAR HORARIO PROFESOR.....	145
ILUSTRACIÓN 4-115: INGRESAR HORARIO AULA	145
ILUSTRACIÓN 4-116: INGRESAR USUARIO	146
ILUSTRACIÓN 4-117: MODIFICAR PROFESOR	146
ILUSTRACIÓN 4-118: MODIFICAR MATERIA	147
ILUSTRACIÓN 4-119: MODIFICAR AULA	147

ILUSTRACIÓN 4-120: MODIFICAR USUARIO	148
ILUSTRACIÓN 4-121: ELIMINAR PROFESOR	148
ILUSTRACIÓN 4-122: ELIMINAR MATERIA.....	149
ILUSTRACIÓN 4-123: ELIMINAR AULA	149
ILUSTRACIÓN 4-124: ELIMINAR USUARIO	150
ILUSTRACIÓN 4-125: ELIMINACIÓN DE UN PARALELO.....	150
ILUSTRACIÓN 4-126: ASIGNACIÓN DE PARALELOS	151
ILUSTRACIÓN 4-127: OPTIMIZAR	151
ILUSTRACIÓN 4-128: CREACIÓN DE NUEVO PARALELO, POSTERIOR A LA OPTIMIZACIÓN	152
ILUSTRACIÓN 4-129: REPORTE DE PROFESORES	153
ILUSTRACIÓN 4-130: REPORTE DE MATERIAS.....	153
ILUSTRACIÓN 4-131: REPORTE DE AULAS	154
ILUSTRACIÓN 4-132: REPORTE DE HORARIOS POR AULA.....	154
ILUSTRACIÓN 4-133: REPORTE DE HORARIOS POR PROFESOR.....	155
ILUSTRACIÓN 4-134: REPORTE DE HORARIOS POR MATERIA.....	155
ILUSTRACIÓN 4-135: REPORTE DE CONFLICTOS.....	156
ILUSTRACIÓN 4-136: MODELO LÓGICO DE LA BASE DE DATOS	157
ILUSTRACIÓN 4-137: MODELO DE DISEÑO DE OBJETOS	158
ILUSTRACIÓN 4-138: DIO 1.1: CREACIÓN DE NUEVA PLANIFICACIÓN EXITOSA.	159
ILUSTRACIÓN 4-139: DIO 2.1: APERTURA DE PLANIFICACIÓN EXITOSA.	160
ILUSTRACIÓN 4-140: DIO 3.1: ELIMINACIÓN DE PLANIFICACIÓN EXITOSA.	161

ILUSTRACIÓN 4-141: DIO 4.1: GUARDAR COMO DE PLANIFICACIÓN EXITOSA.	162
ILUSTRACIÓN 4-142: DIO 5.1: CAMBIO DE USUARIO EXITOSO.	163
ILUSTRACIÓN 4-143: DIO 6.1: CAMBIO DE CONTRASEÑA EXITOSA.	164
ILUSTRACIÓN 4-144: DIO 7.1: INGRESO DE USUARIO EXITOSO.	165
ILUSTRACIÓN 4-145: DIO 8.1: MODIFICACIÓN DE USUARIO EXITOSO.	166
ILUSTRACIÓN 4-146: DIO 9.1: ELIMINACIÓN DE USUARIO EXITOSO.	167
ILUSTRACIÓN 4-147: DIO 10.1: INGRESO DE ASIGNACIONES ESPECIALES EXITOSA.	
168	
ILUSTRACIÓN 4-148: DIO 11.1: INGRESO DE PROFESOR EXITOSO.	169
ILUSTRACIÓN 4-149: DIO 11.2: IDENTIFICACIÓN DEL PROFESOR INVALIDA.	170
ILUSTRACIÓN 4-150: DIO 12.1: MODIFICACIÓN DE PROFESOR EXITOSO.	171
ILUSTRACIÓN 4-151: DIO 13.1: ELIMINACIÓN DE PROFESOR EXITOSA.	172
ILUSTRACIÓN 4-152: DIO 14.1: INGRESO DE MATERIA EXITOSO.	173
ILUSTRACIÓN 4-153: DIO 14.2: ERROR EN EL CÓDIGO DE LA MATERIA	174
ILUSTRACIÓN 4-154: DIO 15.1: MODIFICACIÓN DE MATERIA EXITOSO.	175
ILUSTRACIÓN 4-155: DIO 15.2: ERROR EN EL NIVEL DE LA MATERIA.	176
ILUSTRACIÓN 4-156: DIO 16.1: ELIMINACIÓN DE MATERIA EXITOSO.	177
ILUSTRACIÓN 4-157: DIO 17.1: INGRESO DE AULA EXITOSO.	178
ILUSTRACIÓN 4-158: DIO 18.1: MODIFICACIÓN DE AULA EXITOSO.	179
ILUSTRACIÓN 4-159: DIO 19.1: ELIMINACIÓN DE AULA EXITOSO.	180
ILUSTRACIÓN 4-160: DIO 20.1: INGRESO DE PARALELO EXITOSO.	181
ILUSTRACIÓN 4-161: DIO 21.1: ASIGNACIÓN DE PARALELO EXITOSO.	182

ILUSTRACIÓN 4-162: DIO 22.1: ELIMINACIÓN DE PARALELO EXITOSO.	183
ILUSTRACIÓN 4-163: DIO 23.1: INGRESO DE HORARIO PROFESOR EXITOSO.	184
ILUSTRACIÓN 4-164: DIO 24.1: INGRESO DE HORARIO AULA EXITOSO.	185
ILUSTRACIÓN 4-165: DIO 25.1: OBTENER LISTA DE PROFESORES EXITOSA.....	186
ILUSTRACIÓN 4-166: DIO 26.1: OBTENER LISTA DE MATERIAS EXITOSA.	187
ILUSTRACIÓN 4-167: DIO 27.1: OBTENER LISTA DE AULAS EXITOSA.	188
ILUSTRACIÓN 4-168: DIO 28.1: OBTENER HORARIOS POR AULA EXITOSO.....	189
ILUSTRACIÓN 4-169: DIO 29.1: OBTENER HORARIOS POR MATERIA EXITOSO.	190
ILUSTRACIÓN 4-170: DIO 30.1: OBTENER HORARIOS POR PROFESOR EXITOSO. ...	191
ILUSTRACIÓN 4-171: DIO 31.1: REPORTE DE CONFLICTO EXITOSO.....	192
ILUSTRACIÓN 4-173: DIO 32.1: INGRESO DE PARÁMETROS INICIALES EXITOSO. ...	193
ILUSTRACIÓN 4-174: DIO 33.1: OPTIMIZACIÓN EXITOSA.	194
ILUSTRACIÓN 4-175: DIO 33.2: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS MAYORES QUE HORAS DISPONIBLES DE AULAS.	195
ILUSTRACIÓN 4-176: DIO 33.3: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS MAYORES QUE HORAS DISPONIBLES DE LABORATORIOS.....	196
ILUSTRACIÓN 4-177: DIO 33.4: OPTIMIZACIÓN FALLIDA - HORAS DISPONIBLES DE AULAS MAYOR A LAS HORAS DE ASIGNACIÓN SEMANAL.....	197
ILUSTRACIÓN 4-178: DIO 33.5: OPTIMIZACIÓN FALLIDA - HORAS ASIGNADAS A PROFESORES ES MAYOR QUE SUS DISPONIBILIDAD.....	198
ILUSTRACIÓN C-1: PANTALLA DE VALIDACIÓN DE USUARIOS.....	CCLVII
ILUSTRACIÓN C-2: INGRESAR USUARIO.....	CCLVIII

ILUSTRACIÓN C-3: MODIFICAR USUARIO	CCLVIII
ILUSTRACIÓN C-4: ELIMINAR USUARIO.....	CCLVIII
ILUSTRACIÓN C-5: CAMBIO DE CONTRASEÑA DESDE USUARIO ADMINISTRADOR. CCLIX	
ILUSTRACIÓN C-6: CAMBIO DE CONTRASEÑA DESDE USUARIO DIGITADOR.....	CCLIX
ILUSTRACIÓN C-7: PANTALLA DE CREACIÓN DE UNA NUEVA PLANIFICACIÓN.....	CCLX
ILUSTRACIÓN C-8: PANTALLA DE SELECCIÓN DE PLANIFICACIÓN	CCLX
ILUSTRACIÓN C-9: PARÁMETROS DEL SISTEMA Y DEL ALGORITMO GENÉTICO.	CCLXI
ILUSTRACIÓN C-10: OPTIMIZACIÓN USANDO ALGORITMOS GENÉTICOS	CCLXII
ILUSTRACIÓN C-11: INGRESO DE PROFESORES	CCLXIII
ILUSTRACIÓN C-12: SELECCIÓN DE PROFESORES PARA MODIFICACIÓN.	CCLXIV
ILUSTRACIÓN C-13: MODIFICACIÓN DE PROFESORES	CCLXV
ILUSTRACIÓN C-14: SELECCIÓN DE PROFESORES PARA MODIFICAR SU HORARIO DE DISPONIBILIDAD.	CCLXVI
ILUSTRACIÓN C-15: HORARIO DE DISPONIBILIDAD POR PROFESOR	CCLXVII
ILUSTRACIÓN C-16: ELIMINAR PROFESOR.....	CCLXVIII
ILUSTRACIÓN C-17: INGRESO DE MATERIAS	CCLXIX
ILUSTRACIÓN C-18: SELECCIÓN DE MATERIA PARA SU MODIFICACIÓN	CCLXX
ILUSTRACIÓN C-19: MODIFICACIÓN DE MATERIA.	CCLXXI
ILUSTRACIÓN C-20: ELIMINAR MATERIA.....	CCLXXII
ILUSTRACIÓN C-21: INGRESO DE AULA.	CCLXXIII
ILUSTRACIÓN C-22: SELECCIÓN DE AULA PARA SU MODIFICACIÓN	CCLXXIV

ILUSTRACIÓN C-23: MODIFICACIÓN DE AULA.....	CCLXXV
ILUSTRACIÓN C-24: SELECCIÓN DE AULA PARA MODIFICAR SU HORARIO DE DISPONIBILIDAD	CCLXXVI
ILUSTRACIÓN C-25: MODIFICACIÓN DE HORARIO DE DISPONIBILIDAD DE AULA CCLXXVII	
ILUSTRACIÓN C-26: ELIMINAR AULA.....	CCLXXVIII
ILUSTRACIÓN C-27: ASIGNACIÓN DE MATERIAS POR PROFESOR.....	CCLXXIX
ILUSTRACIÓN C-28: ELIMINACIÓN DE UN PARALELO.	CCLXXX
ILUSTRACIÓN C-29: ASIGNACIÓN DE PARALELOS.....	CCLXXXI
ILUSTRACIÓN C-30: CREACIÓN DE NUEVO PARALELO, POSTERIOR A LA OPTIMIZACIÓN	CCLXXXII
ILUSTRACIÓN D-1: BIENVENIDA AL PROGRAMA DE INSTALACIÓN	CCLXXXIII
ILUSTRACIÓN D-2: INICIO DEL PROCESO DE INSTALACIÓN.....	CCLXXXIV
ILUSTRACIÓN D-3: SELECCIÓN DE DIRECTORIO DE DESTINO DE LA INSTALACIÓN CCLXXXIV	
ILUSTRACIÓN D-4: SELECCIÓN DE GRUPO DE PROGRAMAS	CCLXXXV
ILUSTRACIÓN D-5: PROGRESO DE LA INSTALACIÓN.....	CCLXXXV

INTRODUCCIÓN

En un enfoque minucioso sobre como se lleva de forma manual los procesos de planificación de horarios en las entidades educativas siempre surge más de un rompecabezas de difícil resolución, convirtiéndose en procesos que una vez planteados difícilmente pueden someterse a modificaciones o actualizaciones debido a que una alteración mínima podría resultar en una nueva planificación, afectar los horarios de profesores, o lo que es aún más riesgoso caer en una mala distribución de recursos.

Son cada una de estas problemáticas las que nos llevan a buscar mecanismos automatizados y de manera más enfocada de optimización heurística que nos ayuden a solucionar los problemas reales por las que pasan millares de instituciones académicas a nuestro alrededor.

La utilización de algoritmos genéticos para este tipo de problemas nos permite realizar búsquedas ciegas en diferentes entornos evitando el desgaste de los recursos. El poder de cómputo del que disfrutamos en la actualidad nos admite costear un sinnúmero de iteraciones hasta llegar a una solución óptima. Lo que es aún más interesante es la permisibilidad que nos dan los algoritmos genéticos de agregar recursos, asignaciones, nuevos paralelos, horarios o profesores sin que esto afecte a lo ya planteado.

El grado de dispersión con el que cuentan los algoritmos genéticos nos dan pauta para iniciar el proceso con una búsqueda dispersa, recombinar y mutar las soluciones, evitar recaer en mínimos locales, usar distintas representaciones para el problema, y recurrir a métodos intrínsecos de la evolución natural explotándolos para encontrar un conjunto de soluciones que satisfagan los requerimientos de el problema planteado.

En el capítulo 1, se da una introducción general al problema de la planificación de horarios, su estado actual y la justificación del desarrollo de este proyecto de tesis, las condiciones iniciales y se analizan las posibles restricciones del sistema.

En el capítulo 2, se proporcionan los fundamentos teóricos y las tecnologías relacionadas con los Sistemas de Planeación Avanzada (APS) que son las herramientas de asignación de recursos que se utilizará para la resolución

del problema y se realizará un análisis comparativo con su tecnología antecesora, los Sistemas de Planeación de Recursos Empresariales (ERP)

En el capítulo 3, se describen los algoritmos genéticos, elementos básicos y operaciones inmersas, se concluye con la comparación de los algoritmos genéticos con otros métodos heurísticos y los fundamentos para su selección en la resolución del problema.

Con las bases del capítulo 3, en el capítulo 4 se analiza ya la representación heurística del problema, se describe el análisis y diseño de los requerimientos funcionales y operativos clasificados en modelos estáticos y dinámicos. Se incluyen descripciones detalladas de las restricciones que el sistema debe permitir definir y cumplir, así como los procesos que se deberán llevar a cabo, esto en su arquitectura tanto física como lógica.

El capítulo 5, es un estudio financiero de costos y retornos sobre la inversión para las instituciones que implementen el sistema. Se realiza además un análisis de mercado sobre el impacto del uso de la aplicación en el mercado local.

En el capítulo 6, se incluye un plan de pruebas con los usuarios del sistema, análisis y convergencia de los resultados obtenidos y las posibles mejoras al desempeño del sistema.

Finalmente se presentan Conclusiones y Recomendaciones acerca de los siguientes pasos a seguir para mejorar el desempeño del sistema así como los valores agregados que pueden incluirse para brindar mayor precisión en la resolución del problema. Adicionalmente, se incluyen apéndices que comprenden: El Manual del Usuario, Diccionario de Datos, Manual de Instalación del Sistema y la Codificación del mismo.

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

La problemática trazada nos permite enfocar de forma general la resolución del problema de planificación de horarios, el producto terminado que aunque esta enfocado a resolver este problema de forma muy general (pudiendo ser aplicado en cualquier entidad educativa), provee de múltiples alternativas al establecerse como un sistema multiusuario y multiempresas, tomando datos e información de muestra de la Facultad de Ingeniería en Electricidad y Computación (FIEC) de la Escuela Superior Politécnica del Litoral (ESPOL).

Entrevistas con el coordinador de la planificación de horarios permitieron determinar los inconvenientes que actualmente posee el proceso de planificación de horarios de materias a dictarse en cada semestre, debido a que no se cuenta con herramientas computacionales para realizar dicha tarea, por lo que deben realizar un arduo procedimiento manual, que actualmente toma una semana en el mejor de los escenarios, proceso que debe repetirse en cada semestre.

El mayor inconveniente no es solo es el hecho de llevar un proceso manual para obtener una solución (que no necesariamente es la más óptima), sino que además existe la necesidad de incluir variables exógenas que influyan directamente sobre la planificación, cuyo comportamiento es muy irregular, y en ciertos casos inflexible, para acomodar otra solución que satisfaga este nuevo escenario.

Un claro ejemplo sobre esta variabilidad mencionada la encontramos en el horario de disponibilidad de un profesor y en sus requerimientos para dictar una materia, debido a que en un semestre en particular un profesor puede estar totalmente disponible, y al siguiente dictar clases a ciertas horas y en determinados días de la semana, aún tratándose de un profesor con nombramiento. En otros casos el

profesor durante un semestre puede estar disponible en un bloque de horario limitado, en un día específico, creando así muchas más restricciones y limitantes en la distribución de los horarios.

Con estas inconsistencias y un sin número de condiciones de igual aleatoriedad, bajo un procedimiento manual, es extremadamente difícil encontrar una solución óptima que satisfaga todas las condiciones iniciales o en su mayoría.

Dados los antecedentes descritos, se plateó un sistema que sea lo suficientemente flexible, para obtener a partir de un listado de posibles soluciones, la solución más óptima que cumpla con los requerimientos iniciales como la disponibilidad de horarios por profesor (con nombramiento y sin nombramiento), material didáctico o condiciones físicas solicitadas por el profesor para dictar una materia (proyector, laboratorio, etc.), disponibilidad de horarios de aulas, etc.

Nuestro objetivo es dotar a la FIEC de una eficiente herramienta de planificación, que trabaje de forma ágil y ordenada para proporcionar la planeación de horarios a nivel FIEC, siendo esta una herramienta multiempresas de fácil adaptación para las demás facultades de la

ESPOL, que con mínimos parámetros de ajuste pueda ser implementada por diversas entidades académicas de diferente nivel.

1.2 SITUACIÓN ACTUAL DE LA PLANIFICACIÓN

La serie de entrevistas previas realizadas al área de coordinación de la planificación de horarios de la FIEC, permitió analizar los inconvenientes de la planificación; entre otras cosas se encontró que la forma manual de obtención de la planificación demanda excesivo tiempo y recursos, lo que hace urgente la implementación de una herramienta que agilite este engorroso proceso.

Uno de los puntos principales sujetos de análisis al momento de la planificación es la coordinación de los horarios de disponibilidad de profesores; la divergencia en cuanto a tiempo disponible por profesor requiere ser organizada por ciertos factores claves como la disponibilidad de horarios, condiciones prioritarias por profesor de acuerdo a tiempo de contratación y escalas directivas o salariales, cantidad de materias a dictar, cantidad de paralelos asignados, etc.

El factor logística se torna un tanto más flexible en esta perspectiva debido a que en nuestro caso de análisis la oferta de aulas supera lo

requerimientos, con un horario de disponibilidad de las mismas entre las 07H0 y las 22H00; los factores determinantes están dados por la falta de recursos didácticos en las aulas tales como: proyectores, retro proyectores, aire acondicionado, pizarra de tiza líquida, etc., mismos que son relevantes para un correcto desenvolvimiento académico. Este tipo de requerimientos puntuales se dan lugar en las materias relacionadas con el área de computación o en aquellas en las que el profesor por iniciativa propia solicita recursos adicionales. En lo referente a los laboratorios los inconvenientes antes descritos normalmente no afectan a los mismos, dado que son aulas con equipamiento especial y de tipo práctico.

Uno de los grandes problemas de consenso para la coordinación de la planeación, es la combinación que permita que materias del mismo nivel y de la misma especialización no se dicten a la misma hora, de tal forma que un estudiante tenga la oportunidad de elegir la mayor; sino la total cantidad de materias de un nivel según indique su pènsum académico.

Ajustarse a los horarios de disponibilidad, prioridades y requerimientos solicitados por los profesores nos conlleva a realizar un sin número de iteraciones manuales con plazos promedios de una

semana, tiempo que en definitiva pueden ser mejor empleado, debido al innecesario desgaste de los recursos humanos que se dedica a esta labor.

1.3 RESULTADOS ESPERADOS

En base a la información obtenida se plantea de forma general el requerimiento del cliente, en el capítulo 4 retomaremos este tema con la finalidad de ofrecer una solución factible.

CONDICIONES INICIALES

- Cada profesor propone un horario de disponibilidad.
- El horario de disponibilidad de aulas es de 7h00 a 22h00 de lunes a viernes, sábados hasta las 14h00.
- Dos materias del mismo nivel no deben cruzarse o este porcentaje de cruce de materias debe cumplir un mínimo especificado.
- Ciertas materias se dictan en aulas predeterminadas.
- Las materias de laboratorio se asignan a su laboratorio correspondiente.
- Cargas horarias distribuidas.

- De preferencia las horas de clases de un paralelo deben dictarse en una misma aula, en el mismo horario, pasando un día, disponiendo de bloques de horario predeterminados.
- Existen casos privilegiados de asignación de recursos, aulas u horarios de clases que deben ser manejados por la administración.

RESULTADOS ESPERADOS

- Horarios de clases por aula.
- Horarios de clases por profesor.
- Horarios de clases por materia.
- Penalizaciones (violación de condiciones iniciales o reporte de conflictos).

1.4 DEFINICIÓN DEL PROBLEMA

FACTORES A OPTIMIZAR:

1. Número de Profesores.
2. Horarios disponibles por profesor
3. Aulas disponibles.
4. Recursos disponibles por aula.

5. Materias (por paralelo) a dictarse en el presente ciclo académico.
6. Materias (del mismo nivel y misma especialización) a dictarse en el presente ciclo académico.
7. Laboratorios.
8. Horas hábiles de clases.

INFORMACIÓN PROPORCIONADA:

Se tiene la siguiente información para dar cumplimiento a las restricciones:

- Número de profesores disponibles por materia, y por especialidad.
- Número de aulas disponible para el semestre en curso.
- Número de laboratorios disponibles por materia.
- Horario de disponibilidad por cada profesor incluido los que tienen y no nombramiento.
- Número de materias por especialidad.
- Número de paralelos distribuidos de acuerdo a las materias.
- Carga de la materia, la cual representa el número de horas semanales y diarias que se dicta una materia en particular.

- Recursos solicitados por profesores para dictar su materia.
- Recursos necesarios para dictar cada materia en el presente semestre, si la materia lo requiere.

1.5 DIFERENTES ASPECTOS DE LA SOLUCIÓN

No debe existir cruce de horarios entre materias de la misma especialidad.

No debe existir cruce de horarios entre materias del mismo nivel.

Una materia de laboratorio no puede dictarse en un aula común.

Las materias de computación solo podrán ser dictadas en las aulas de computación.

Se deberá respetar los horarios de disponibilidad de cada profesor.

Existen casos especiales de profesores con horarios restringidos que son inamovibles y tendrán prioridad frente a otras asignaciones.

CAPÍTULO 2

2. SISTEMAS DE PLANEACIÓN AVANZADA

Los Sistemas de Planeación Avanzada (APS) son la herramienta de asignación de recursos que se utilizará para la resolución del problema, pueden ser para muchas empresas una fuerte inversión humana y de capital donde los procesos de implementación pueden llegar a tomar entre dos y diez años o más^[1], debido en gran parte a que la implementación de estos sistemas requiere cambios en todos los niveles de la organización, ya sea en los procesos de manufactura, la cultura de la organización, restricciones impositivas, etc. Sin embargo, es aquel retorno sobre la inversión el que nos muestra la potencialidad de su adopción. ¹

Estos sistemas de planeación avanzada facilitan la búsqueda de información, permiten obtener reportes históricos, mejoran el control de las políticas y

¹ Refiérase a la sección 5.4 de esta tesis donde se obtiene el Retorno Sobre la Inversión de la implementación del APS

procesos de negocio, mejoran la productividad, calidad y satisfacción de la fuerza de trabajo, constituyen una herramienta predictiva en control de inventarios y tiempos de envío, además de proveer información en tiempo real ^[iii]. No obstante el mayor problema de las compañías es la tendencia a pensar que un APS llevara el control de sus negocios, cuando estos solo proveen información para mejorar la administración de los negocios.

2.1 ¿QUE ES UN SISTEMA DE PLANEACIÓN AVANZADA?

Un Sistema de Planeación Avanzada - Advanced Planning and Scheduling Software (APS) se define típicamente como un proceso de optimización y asignación de recursos de producción. Los principales recursos de un APS son:

- Capacidad de maquinaria y labores de producción: En esta etapa toma las riendas un Software de Planeación de Capacidades Finitas - Finite Capacity Scheduling Software (FCS), el que permite manejar los recursos de la producción, dependencias, reglas comerciales y cuellos de botella. ^[iiii]
- Procesos de producción de materiales: En esta fase, los Sistemas de Planeación de Recursos Empresariales - Enterprise Resource Planning(ERP) y/o las Cadenas de Suministros - Supply Chain Management (SCM), aseguran los esfuerzos necesarios para que la materia prima este

disponible durante todo el proceso de producción y se minimice la cantidad de inventario disponible. ^[III]

Un APS busca eliminar todo tipo de pérdidas ya sean estas de materiales, tiempo, transportación, energía, retardos, espacio, sobreproducción, reparaciones o trabajo duplicado.

2.2 TECNOLOGÍAS RELACIONADAS CON LOS APS

Los sistemas de ERP han evolucionado de los antiguos sistemas MRP (Sistemas de Requerimientos de Materiales) proporcionando valor agregado a los negocios. Muchas organizaciones, sin embargo, han encontrado que ellos necesitan una visión más profunda en ciertos aspectos funcionales de las compañías, y al parecer muestran soluciones demasiado puntuales es allí donde aparecen los APS a manera de complemento para proveer control detallado y visibilidad de los procesos productivos. ^[IV]

2.2.1 INVENTARIO JUSTO A TIEMPO- JUST IN TIME INVENTORY (JIT)

El concepto "Just in Time" fue creado en el año 1954 por el ejecutivo de Toyota Motor Co., el señor Taiichi Ohno, en el que supone una nueva forma de gestión constituida por un conjunto de técnicas y

prácticas de organización de la producción, que pretende que el cliente sea servido cuando lo precise (justo a tiempo) y en la cantidad y calidad requeridas.^[VI] Las dos estrategias básicas de este enfoque consisten en la eliminación de todas las funciones innecesarias en las operaciones industriales (llamadas desperdicios) y en producir los distintos productos y componentes en el momento en que se necesiten, en la cantidad en que se precise y con la máxima calidad, siendo estos sus objetivos: reducir a cero la cantidad de defectos, averías, stock, tiempo de ocio y burocracia.^[VI]

SISTEMAS TRADICIONALES	SISTEMA JUST IN TIME
<ul style="list-style-type: none"> - Secciones por tipo de máquina (por funciones). - Líneas de proceso independiente para cada gama de producto. - Personal especializado en un tipo de máquina. 	<ul style="list-style-type: none"> - Secciones por proceso con diversas máquinas. - Agrupación de procesos comunes para diversas gamas de producto. - Personal polivalente especializado en un proceso con utilización de diversas máquinas.

Tabla 2-1: Diferencias Fundamentales en Planta Tradicional y la propuesta por JIT^[XXXVI]

2.2.2 SISTEMAS DE REQUERIMIENTOS DE MATERIALES - MATERIAL REQUIREMENTS PLANNING (MRP)

Los MRP se iniciaron en las empresas de manufactura, con el objetivo de planificar los componentes, partes y materiales involucrados en los procesos de manufactura, creando sistemas modulares que intentan abarcar todas las entidades de una empresa mejorando los procesos de planeación, control y operación. ^[VI]

Estos sistemas eran capaces de detectar nuevas órdenes de materia prima y materiales, programar el número de unidades necesarias en cada etapa del proceso, brindar respuestas oportunas a las demandas del mercado, manejar de forma centralizada e integrada la contabilidad, finanzas, cuentas por pagar, cuentas por cobrar, comisiones, inventario, ventas, recursos humanos, servicio al cliente y planeación gerencial. ^[VII]

A pesar de su cúmulo de beneficios, la falta de educación del área ejecutiva sobre el énfasis de la importancia de la MRP como instrumento de planeación estratégica, no reconocer que el MRP es solo una herramienta de software que no genera toma de decisiones, de igual manera, la falta de exactitud en los datos ingresados

pueden producir trastornos a los beneficios que puedan obtenerse con el uso de esta herramienta.

2.2.3 PLANEACIÓN DE RECURSOS DE MANUFACTURA - MANUFACTURING RESOURCE PLANNING (MRP II).

Los cambios tecnológicos hicieron posible la transición entre los MRP que originalmente solo planeaban el uso de materiales, pasando a denominarse MRP II o Planeación de Recursos de Manufactura, cuyo objetivo inicial fue que los sistemas de fabricación fueran simulados y se pudiera monitorear todos los recursos de una empresa de manufactura, tales como mercadeo, finanzas e ingeniería de procesos, a través de un sistema de ciclo cerrado que generaba cifras financieras. ^[VIII]

2.2.4 SOFTWARE DE PLANEACIÓN DE CAPACIDADES FINITAS - FINITE CAPACITY SCHEDULING SOFTWARE (FCS)

Es la unión de todos los modelos modernos de administración y operación de sistemas. Es un modelo de predicciones de cargas de trabajo que permite detectar cuellos de botella. Creando una secuencia operacional que usa la disponibilidad de tiempo real para producir recursos de una orden de trabajo.

En los FCS están involucrados factores como:

- Patrones de falla
- Trabajos en proceso
- Planeación de mantenimiento
- Herramienta de disponibilidad para el operador.^[IX]

FCS es una herramienta para manejar cambios, minimiza tiempos de despacho, brinda mayor visibilidad a los procesos productivos, permite a los fabricantes comparar varias estrategias diferentes antes de generar el horario de producción. El resultado es una planificación de la producción más exacta y realista.

2.2.5 SISTEMAS DE PLANEACIÓN DE RECURSOS EMPRESARIALES - ENTERPRISE RESOURCE PLANNING (ERP)

Las ERP se inician en los años 70, cuando Gartner Group y AMR dan el primer paso para definir las bases de las E.R.P., cuando las organizaciones estaban orientadas a las funciones y solo se cubría las áreas de manufactura y finanzas.^[X]

En los años 80, se cambio el paradigma tomando ahora una orientación hacia los procesos, identificando los procesos críticos de las empresas (Logística, Inventarios, Compras, Ventas, Distribución,

Producción, Recursos Humanos) y su integración entre sí utilizando el modelo Cliente – Servidor. ^[X]

Asimismo, existen varios autores que explican que una E.R.P. no es un proceso evolutivo de dicha metodología, tal es el caso de Don Ralston y David Turbide que explican respectivamente: "El E.R.P. no es un paso mayor, pero si una expansión del MRP II, con nuevas aplicaciones a través de la incorporación de tecnología moderna a la empresa, como el concepto Cliente-Servidor y EDI (Intercambio de Información Electrónica), que han provisto de mayor alcance al uso de la misma filosofía." ^[XI]

2.2.5.1 FUNCIONALIDADES DE LOS ERP

FINANCIERA Y RRHH: Contabilidad General, Cuentas por Cobrar, Cuentas por Pagar, Presupuestos y Prorratesos, Reconciliaciones Bancarias, Multimoneda, Activos Fijos y Administración de Activos, Recursos Humanos (básico), Administración y Control de Proyectos, Contabilidad de Costos, Tesorería, Presupuesto, Facturación. ^[XII]

SOLUCIONES ESPECIALIZADAS: Solicitud Avanzada de Cotizaciones, Administración Avanzada de Maquila, Administración

Avanzada de RMA, Administración Punto de Venta, TMS - Administración del Transporte, WMS - Administración de Almacenes.

[XII]

LOGÍSTICA Y DISTRIBUCIÓN: Administración de Inventarios, Administración de Compras, Precios(Requisiciones, Cotizaciones, Contratos, Calendarios de compra, Calificación de proveedores), Administración de Órdenes de Venta (Contratos, Calendario de Venta), Administración Avanzada de Precios (Precios, Márgenes, Cotizaciones), Pronósticos (Planificación de artículos, Planificación de recursos, Planificación de materiales, Planificación de capacidad, Planificación de compras, Planificación de distribución y análisis de planificación).^[XII]

PAQUETES NACIONALES: Impuestos en base a Flujo, Reevaluación de Inventarios, Control de Importaciones y Exportaciones, Nota de Embarque.^[XII]

FUNDAMENTOS: Herramientas de Personalización, Reportes, Herramienta para Incrementar Inteligencia a Procesos (Flujos de Trabajo).^[XII]

MANUFACTURA: Administración de Datos de Producto (PDM), Planeación Maestra de la Producción (MPS), Planeación de Materiales (MRP), Control de Producción en Piso (SFC), Costeo basado en Órdenes de Trabajo y/o Periodos; Contabilidad de Manufactura, Sistema Avanzado de Planeación y Programación (APS) (Programación Finita), Rutas, Cálculo de costos, Control de proyectos de manufactura, Control de manufactura, Requerimientos de herramientas, Control de configuración y clasificación de productos. ^[XII]

ALMACENAMIENTO – WAREHOUSING: Planificación y manejo de inventarios, Análisis de inventarios. ^[XXI]

GESTIÓN DE SERVICIOS: Gestión de contratos de servicio, Gestión de configuraciones, gestión de subcontrataciones, Gestión de llamadas, planificación de órdenes de servicio, Gestión de órdenes de servicio, Programación, Reparación de instalaciones. ^[XII]

ADMINISTRACIÓN DE CALIDAD: Calidad del Proveedor, Planificación y ejecución de inspecciones, Planes de calibración y ejecución, Calidad del Proceso. (5) E-CIS Soluciones, S.A. 24 de Abril de 2005. ^[XII]

2.2.6 CADENAS DE SUMINISTROS - SUPPLY CHAIN MANAGEMENT (SCM).

La gestión de la cadena de suministros (SCM por sus siglas en inglés, Supply Chain Management), sugiere una vez lograda una mejora en sus operaciones lograr mayores ahorros y beneficios al mejorar los procesos y el intercambio de información, englobando aquellas actividades asociadas con el movimiento de bienes desde el suministro de materias primas hasta el consumidor final. ^[XIII] Esto incluye la selección, compra, programación de producción, procesamiento de órdenes, control de inventarios, transportación, almacenamiento y servicio al cliente.

En resumen: "Una exitosa cadena de suministros entrega al cliente final el producto apropiado, en el lugar correcto y en el tiempo exacto, al precio requerido y con el menor costo posible." ^[XIII]

Algunos de los beneficios destacados de las SCM son: mayor productividad, aumento en las ventas, segmentación y conocimiento de los clientes, optimización de procesos internos, conocimiento permanente de la situación del mercado al interior de la organización (precios, comercialización, promociones, inventarios, entre otros), mejor servicio, fidelidad de los clientes. ^[XIV]

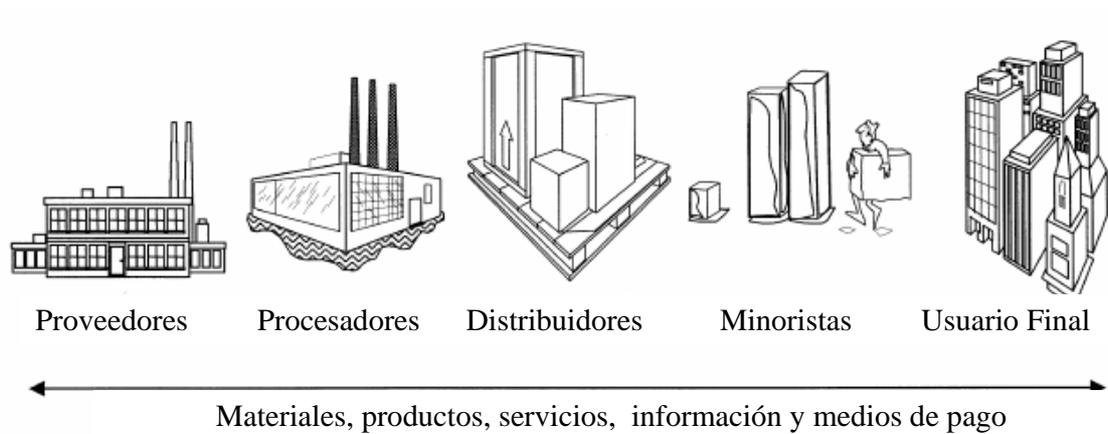


Ilustración 2-1 Red logística de la Cadena de Suministros [XXXVI]

2.3 VENTAJAS DE LOS APS SOBRE LOS ERP

- Los ERP buscan mejorar el uso de los recursos de la empresa, integrándolo y eliminando islas de información, sin embargo los APS requieren de menores inversiones que los ERP y no afectan la forma de actuar de la organización, ni requieren largos periodos de implementación.
- Los APS pueden trabajar en ambientes no predictivos utilizando herramientas de optimización heurística, minería de datos o simulación.
- Los APS son capaces de integrar al ambiente externo que no tiene relación implícita con la empresa, en su lógica de negocio, aplicando una verdadera cadena de suministros.

CAPÍTULO 3

3. ALGORITMOS GENÉTICOS

Dentro de la perfecta naturaleza ocurren cambios a cada instante, cambios en los individuos y todas las especies existentes en ella, muchas veces nos preguntamos el porque de las cosas e intentamos encontrarle una explicación lógica, muchas veces errada y en ocasiones un poco cercana a la realidad, sea cercana o no, en la naturaleza los individuos evolucionan y se adaptan al tiempo y al espacio en los cuales se encuentran y a los cuales migran. Esta evolución que permite al individuo mejorar o adaptarse al medio fue utilizada en el desarrollo de los algoritmos genéticos, los cuales son métodos de solución de problemas basados en la evolución biológica.^[XVI]

3.1 INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos (AG) son métodos utilizados en la solución de problemas de búsqueda y optimización que se basa en la evolución biológica como estrategia para resolver problemas. Estos

algoritmos fueron introducidos en 1970 por Jhon Holland, tomando como bases el proceso en la evolución de los seres vivos.^[XVII]

En la naturaleza, donde el más fuerte sobrevive, se realiza una selección natural, la cual consiste en que una población, la cual es un conjunto de individuos, en la cual el individuo que tiene las mejores características puede seguir viviendo y le permite reproducirse; si un individuo no puede adaptarse al medio en que vive, perece. Esta selección hace posible que cada vez existan individuos con mejoras características y que puedan adaptarse al medio, porque de lo contrario morirían.

Para la siguiente generación de individuos que puedan reproducirse tendrán la oportunidad de cruzarse y dar paso a una generación mejor que la anterior, el cruzamiento de los mejores individuos es una de las técnicas de evolución biológica.

Dentro de la naturaleza se pueden producir alteraciones aleatorias en el código genético de los individuos, estas alteraciones son las llamadas mutaciones, las mutaciones pueden ser de dos tipos: las generativas, las cuales son mutaciones que heredan y las somáticas las cuales son mutaciones que no heredan.^[XVII]

Estos dos conceptos, el cruzamiento y la mutación, son tomados de la naturaleza por los algoritmos genéticos para la búsqueda y optimización de soluciones.

El algoritmo genético es independiente del problema, es un algoritmo ciego, constituyendo esta su mayor fortaleza y debilidad a su vez, ya que puede ser utilizado para la resolución de cualquier problema, pero no permite especializarse en la resolución de un problema determinado.

3.2 ELEMENTOS BÁSICOS

Partiendo de bases biológicas sabemos que los seres vivos están formados por células, en cada célula existe el mismo número de cromosomas. Los cromosomas son cadenas de ADN (Ácido Desoxirribonucleico) y esto define como es un ser vivo. Un cromosoma se divide en genes los cuales son bloques de ADN, cada gen codifica una característica de un ser vivo.

3.2.1 FUNCIÓN DE ADAPTACIÓN (FITNESS)

La función de adaptación es la función que evalúa un individuo o una generación, que a su vez asigna un peso al individuo de acuerdo al criterio elegido, esto ayuda a decidir que individuo es seleccionado para que pase a la siguiente generación.

Para el problema de planificación de horarios, la función fitness depende de los cruces de materias o de su agrupación por bloques.

3.2.2 CRITERIO DE INICIALIZACIÓN

El criterio de inicialización nos indica como se va a crear la primera generación de individuos, el criterio más utilizado es la generación aleatoria en base a los parámetros ingresados, criterio que se utilizará para la planificación de horarios; existe también la aplicación de diversas funciones o algoritmos para crear la generación inicial.

[XVIII]

3.2.3 CRITERIO DE PARADA

Los criterios de parada son las normas utilizadas para indicarle al algoritmo el momento en el que debe parar, uno los criterios de

parada generalmente utilizado es el porcentaje de error, o el número de generaciones en el algoritmo, este último será el utilizado en la resolución del problema.

El **genotipo** es la representación interna que se utilizará para trabajar con la solución, mientras que el **fenotipo** es la solución en sí misma. Los operadores del algoritmo trabajarán con el genotipo.^[XIX]

3.2.4 ESPACIO DE SOLUCIÓN

El espacio de solución es el conjunto de posibles soluciones, son todas las posibles soluciones a un problema en particular.

Cada punto dentro del espacio de soluciones es una posible solución. En base a la función fitness el algoritmo genético devolverá la mejor solución dentro del espacio de soluciones. La Solución óptima es la mejor solución dentro del espacio de solución

3.2.5 PORCENTAJE DE CRUCE

El porcentaje de cruce es la frecuencia con la que se cruzaran los individuos, la decisión sobre el porcentaje de cruce a utilizar, esta

basada en el criterio de un experto o puede ser producto de diversas pruebas.

Solo se realizará mutaciones en caso que el porcentaje de cruzamiento sea mayor a 0. Con el cruzamiento se generarán nuevos individuos a partir de la generación anterior, una de las técnicas utilizadas es que los mejores individuos pasen a la siguiente generación.

3.2.6 PORCENTAJE DE MUTACIÓN

El porcentaje de mutación es la probabilidad de mutar un individuo, es decir, la probabilidad de modificar alguna de las características del individuo. Si el porcentaje de mutación es 0 los individuos generados no sufrirán cambios después del cruzamiento, si el porcentaje es 100 todos los individuos sufrirán cambios.

La mutación se aplica para que el algoritmo genético no caiga en mínimos locales, por lo que es conveniente que aplique con el criterio de un experto en genética, algoritmos genéticos y optimización avanzada.

3.2.7 TAMAÑO DE LA POBLACIÓN

Es importante establecer cuantos individuos habrá en cada una de las generaciones, a esto se le llama tamaño de la población. Si el tamaño de la población es muy bajo, existirá muchas similitudes entre los padres y los hijos, si el tamaño de la población es muy alto notaremos que los resultados no presentan mejoras significativas.

3.2.8 NÚMERO DE GENERACIONES

Una vez definido el tamaño de la población, necesitamos establecer el número de generaciones, con la ayuda del criterio de experto, pudiendo utilizarse como un criterio de parada.

3.3 ANÁLISIS DEL ALGORITMO

En la resolución de un problema podemos encontrarnos con un gran conjunto de soluciones, si utilizamos búsquedas secuenciales tardaríamos mucho tiempo en encontrar la mejor solución a dicho problema.

Los algoritmos genéticos proporcionan una solución aceptable, que posiblemente no es la mejor dentro del amplio universo de

soluciones, pero nos da una buena aproximación en un tiempo mucho más prudencial que si se analizara el universo de soluciones completo.^[XX]

Es muy importante tener en cuenta los requerimientos del algoritmo para una correcta ejecución del mismo, ya que si no se utiliza una buena representación del problema, este no funcionará de la forma deseada y la solución obtenida no alcanzará un óptimo, más aún durante el proceso de cruzamiento no se generarán soluciones válidas.

3.3.1 MÉTODOS DE REPRESENTACIÓN

Uno de los requerimientos de los algoritmos genéticos, es que el problema se codifique de manera que el algoritmo genético y las computadoras puedan procesar los datos del problema. Existen tres formas para codificar un problema para que pueda ser interpretado por las computadoras.

El primer enfoque es representar las soluciones como cadenas de números binarios, es decir secuencias de 0s o 1s, en el cual cada posición representa el valor de un aspecto de la solución.^[XXI]

El segundo método de representación consiste en utilizar cadenas de enteros o de números decimales que al igual que el anterior cada posición representa el valor de un aspecto de la solución. ^[XXI]

El tercer método consiste en utilizar cadenas de caracteres para representar a los individuos en un Algoritmo genético. ^[XXI]

Sea cual fuere el método que elija para realizar la representación, lo importante es que se realice una correcta representación de los datos reales para que el algoritmo pueda realizar los procesos de cruzamiento, mutación y que los resultados que se generen sean válidos.

3.3.2 MÉTODOS DE SELECCIÓN

La forma en la que se selecciona a los individuos que pasan a la siguiente generación es muy importante, ya que de ello depende el fitness de la nueva generación. Existen diferentes técnicas para seleccionar a los individuos que van a pasar a la siguiente generación. Algunos de estos métodos son excluyentes, pero otros se pueden combinar, la combinación de métodos es un método efectivo y ampliamente utilizado. ^[XXII]

3.3.2.1 SELECCIÓN ELITISTA

Esta selección garantiza que los miembros más aptos pasen a la siguiente generación, en este método los mejores individuos pasan a la siguiente generación, la mayoría de los AG no utilizan elitismo puro, sino formas combinadas de este.^[XXI]

3.3.2.2 SELECCIÓN PROPORCIONAL A LA APTITUD

Los individuos más aptos tienen más probabilidad de ser seleccionados, pero no asegura su paso a la siguiente generación.

[XXII]

3.3.2.3 SELECCIÓN DE LA RULETA

Este tipo de selección funciona como una ruleta, cada individuo posee una porción de la ruleta de acuerdo a su aptitud, los más aptos poseen mayores porciones que los no aptos, se hace girar la ruleta y se elige al individuo que posea esa porción de la ruleta.^[XXIII]

3.3.2.4 SELECCIÓN ESCALADA

A medida que se van creando las nuevas generaciones, la aptitud de los individuos va mejorando por lo que cada vez se debe realizar una mayor discriminación, este aumento en la discriminación es la

selección escalada, este tipo de selección se aplica para seleccionar después de algunas generaciones.^[XXIII]

3.3.2.5 SELECCIÓN POR TORNEO

Se eligen subgrupos de los individuos de la población, los miembros de cada subgrupo compiten entre ellos. Solo se elige a un individuo del subgrupo para la reproducción.^[XXIII]

3.3.2.6 SELECCIÓN POR RANGO.

En esta selección a cada individuo se le asigna un rango numérico basado en su aptitud. La ventaja de este método es que logra que los individuos más aptos no ganen el dominio y reduzcan la diversidad genética.^[XXIII]

3.3.2.7 SELECCIÓN GENERACIONAL

En esta selección no se conservan individuos entre generaciones, pasan a la siguiente generación la descendencia de los individuos más aptos.^[XXIII]

3.3.2.8 SELECCIÓN POR ESTADO ESTACIONARIO

Se conservan algunos individuos para la siguiente generación, la descendencia de los individuos seleccionados en cada generación vuelve al banco de selección.^[XXI]

3.3.2.9 SELECCIÓN JERÁRQUICA

Los individuos pasan por diferentes rondas de selección, en este método las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que son evaluados en los niveles más altos son evaluados más rigurosamente.^[XXIV]

3.3.3 CRUZAMIENTO

El cruzamiento consiste en el intercambio genético entre dos individuos, este es el principal operador de los algoritmos genéticos, es tan importante que si no existe cruzamiento no podemos hablar de AG, pero si no existe mutación si lo podemos hacer.

La técnica de cruce es la forma de calcular el genoma del nuevo individuo a partir de sus padres, esta técnica es responsable del

Algoritmo Genético y determinará la evolución de la población, a continuación presentamos algunas de estas técnicas.

3.3.3.1 CRUCE BÁSICO

Se selecciona un punto aleatorio dentro de la cadena, la parte superior es copiada del padre y la parte inferior de la madre

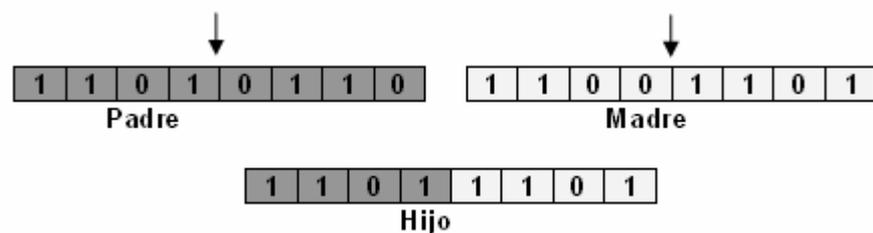


Ilustración 3-1: Cruce Básico

Como se muestra en la figura 3.1, se selecciona un punto, que para este caso es la cuarta posición, y se toma la primera parte del padre y la segunda parte de la madre, para la creación del hijo.

3.3.3.2 CRUCE MULTIPUNTO

Se seleccionan dos puntos aleatorios y se toma dos porciones del padre y una de la madre o viceversa.

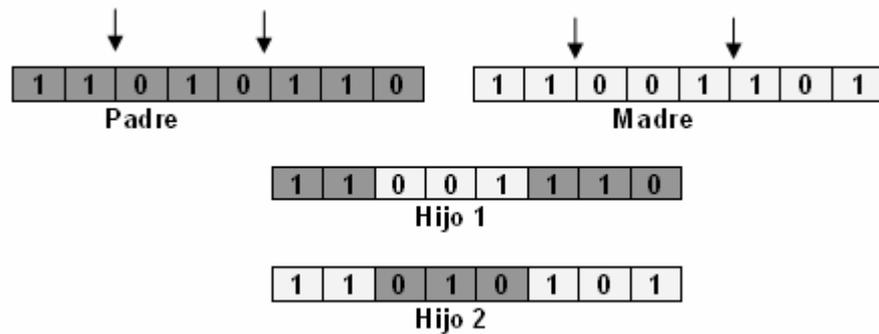


Ilustración 3-2: Cruce Multipunto

3.3.3.3 CRUCE UNIFORME

Para cada gen del descendiente existe la misma probabilidad de que el gen pertenezca al padre o a la madre.

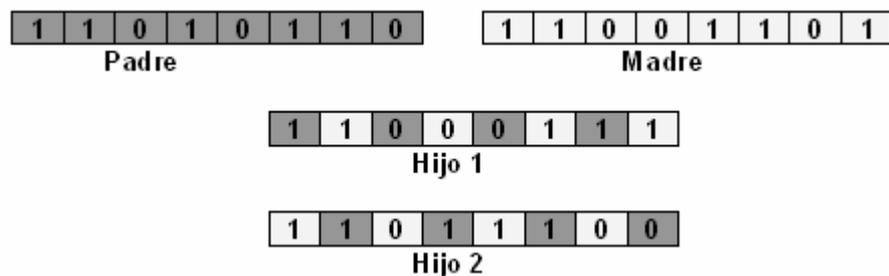


Ilustración 3-3: Cruce Uniforme

3.3.4 MUTACIÓN

Muchas veces cuando escuchamos la palabra mutación, lo primero que se nos viene a la mente son deformaciones o características no comunes de los individuos, lo cierto es que las mutaciones están

presentes en nuestro ambiente y no solo se producen artificialmente sino también naturalmente.

La mutación es la alteración del código genético, las mutaciones son totalmente aleatorias y es un mecanismo de variedad genética.

Existen varias técnicas de mutación que se indicarán a continuación.

3.3.4.1 MUTACIÓN DE BITS

En esta técnica solo uno de los bits puede cambiar, la elección del bit que se va a cambiar es aleatoria. ^[xxv]

0	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---

Gen sin mutar

1	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---

Gen mutado

Ilustración 3-4: Mutación de Bits

3.3.4.2 MUTACIÓN MULTIBIT

Cada bit tiene la probabilidad de mutarse o no, esta probabilidad es calculada en cada paso del operador de mutación. ^[xxv]



Ilustración 3-5: Mutación Multibits

3.3.4.3 MUTACIÓN DE GEN

Funciona igual que la mutación de bits, pero no a nivel de bits, sino que se cambia el gen completo, aquí se puede sumar un número constante, una variable o generar un gen nuevo.^[XXV]

3.3.4.4 MUTACIÓN MULTIGEN

La mutación multigen funciona igual que la mutación multibits, en lugar de intercambiar un conjunto de bits cambia un conjunto de genes.^[XXV]

3.3.4.5 MUTACIÓN DE INTERCAMBIO

Es esta tipo de mutación se toma dos bits/genes y se los intercambia entre si.^[XXVI]



Ilustración 3-6: Mutación de Intercambio

3.3.4.6 MUTACIÓN DE BARAJADO

Toma dos bits o dos genes y los baraja entre si aleatoriamente. ^[XXV]

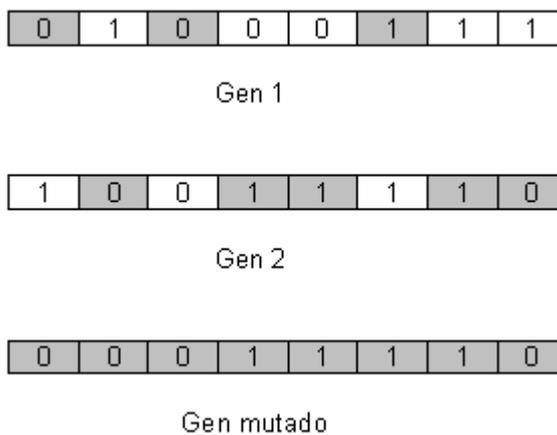


Ilustración 3-7: Mutación de Barajado

3.3.5 LIMITACIONES

La consideración más importante cuando vamos a utilizar AG, es la definición de la representación del problema. La correcta interpretación

por parte del AG del planteamiento del problema es muy importante ya que de ello dependerá la validez de la solución propuesta por el AG, debe ser capaz de tolerar cambios aleatorios, sin que esto represente la generación de errores fatales o resultados sin sentido.

Una vez realizada una correcta representación del problema, lo siguiente es escribir una buena función de aptitud o función fitness. Si la función fitness se elige de manera incorrecta, el algoritmo genético podría ser incapaz de encontrar una solución al problema o puede terminar resolviendo un problema equivocado.

Un problema muy conocido es la convergencia prematura, esta puede surgir en un AG si un individuo es más apto que la mayoría de sus competidores ^[xxvii], este individuo emergerá muy pronto en el curso de la ejecución, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo, en lugar de buscar en el espacio de solución lo bastante a fondo para encontrar el óptimo global. Esto es un problema especialmente común en las poblaciones pequeñas, donde incluso una variación aleatoria en el ritmo de reproducción puede provocar que un individuo se haga dominante sobre los otros.

3.4 MÍNIMOS LOCALES Y SOLUCIÓN ÓPTIMA

Un mínimo local es una buena solución dentro del espacio de soluciones, pero que no es la mejor solución. Un mínimo local es una buena solución dentro de una porción del espacio de soluciones. La Solución Óptima es la mejor solución dentro del espacio de solución. Podemos caer en un mínimo local si un individuo es más apto que la mayoría de sus competidores, pudiendo llegar a una convergencia prematura, que es presentar una solución como óptima cuando solo es una buena solución dentro de una porción del espacio de solución, a esto se conoce como mínimo local.

El cruzamiento nos permite salir de los mínimos locales, debido a que la combinación que se produce entre un padre y una madre da como resultado un hijo con características de los dos, pero diferente a ellos, el cual puede estar muy alejado de una porción de espacio de solución del cual ya tenemos una posible solución, pero que es un mínimo local.

La mutación también colabora, es la búsqueda de la solución óptima o salida de un mínimo local ya que altera ciertas características de los

individuos con la probabilidad de alejarlos de una porción del espacio de solución el cual ya ha presentado un mínimo local.

3.5. APLICACIÓN DE ALGORITMOS GENÉTICOS EN LA SOLUCIÓN DEL PROBLEMA

Para realizar la organización de los horarios de una facultad han sido utilizados múltiples métodos, dentro de ellos la búsqueda exhaustiva, la cual consiste recorrer todo el espacio de solución, este método nos proporcionará la solución óptima pero en un tiempo muy largo.

El objetivo de la organización de los horarios es que con los recursos que se posee, como profesores, materias, aulas, laboratorios y disponibilidades se cree la mejor forma de distribuirlos. El algoritmo genético generará soluciones, que diversifiquen y exploren el conjunto solución, obteniendo un conjunto de soluciones que satisfagan las condiciones iniciales. De este acumulado de planificaciones obtenidas se seleccionará la que mejor se acomode a los requerimientos.

En la sección 4.1 de esta tesis, se podrá verificar como se formará la generación inicial, los cromosomas y la mínima unidad celular representada por los genes.

3.6. ANÁLISIS COMPARATIVO Y SELECCIÓN DE HEURÍSTICA PARA LA SOLUCIÓN DEL PROBLEMA.

“Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución” [XXVIII]

A principio de los 80 surgió una serie de métodos denominados Meta heurísticos, con el propósito de obtener mejores resultado que los métodos Heurísticos, ya que estos métodos se limitaban a encontrar una buena solución pero no la solución óptima, los métodos meta heurísticos surgen con la influencia de las Matemáticas, Biología, Ingeniería y demás ciencias. [LIV]

Los procedimientos metaheurísticos son métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los métodos clásicos de búsqueda, métodos matemáticos, no son efectivos. [XXX]

Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la

inteligencia artificial, la evolución biológica y los mecanismos estadísticos.

Procederemos a analizar algunos métodos meta heurísticos con el objetivo de demostrar por las características propias de cada algoritmo, cual de estos algoritmos proporciona un mejor resultado en la solución nuestro problema.

Uno de los métodos meta heurísticos son los Algoritmos Genéticos los cuales ya hemos revisado a lo largo de este capítulo.

3.6.1 RECOCIDO SIMULADO.

El Recocido Simulado es una de las meta heurísticas más clásicas. Es un algoritmo de aproximación a la solución óptima, que se basa en una analogía del comportamiento de sistemas termodinámicos simples y ha sido muy utilizado en ciertos problemas de ingeniería. Uno de los inconvenientes de este método es la lentitud de convergencia.

Fundamentado en el campo de la termodinámica estadística, simula los cambios energéticos en un sistema de partículas conforme decrece la temperatura, hasta que converge en un estado estable, que se lo llama congelado. ^[XXXI]

3.6.2 BÚSQUEDA TABÚ.

“La Búsqueda Tabú, es una estrategia basada en el uso de estructuras de memoria para escapar de los óptimos locales, en los que se puede caer al moverse de una solución a otra en el espacio de soluciones.

[XXXII]

Este algoritmo posee memoria donde se almacenan los últimos movimientos realizados, y que puede ser utilizada para marcar aquellos movimientos que hacen caer de nuevo en soluciones ya exploradas. Esta memoria, que puede ser una matriz de movimientos, serviría para impedir la evolución hacia esas soluciones.

Para evitar caer en óptimos locales, este algoritmo considera ciertos movimientos “tabú”, de forma que no puedan ser aceptados durante un cierto tiempo o un cierto número de iteraciones.

La estrategia de movimientos rechazados del algoritmo de búsqueda tabú en su forma más simple se denomina de memoria a corto plazo, debido a que la búsqueda que se realiza es local, utilizando la memoria con los movimientos prohibidos en la lista tabú. Para problemas más complejos se pueden considerar estrategias de memorias a plazos más largos. [XXVII]

La memoria a medio plazo tiene como objetivo registrar los atributos más comunes de un subconjunto de soluciones seleccionadas durante un cierto período de búsqueda que con más probabilidad lleven hacia mejores zonas para explorar. Estos atributos sirven de modelos para intensificar la búsqueda de soluciones. La memoria a largo plazo diversifica la búsqueda sobre regiones poco exploradas.

3.6.3 GRASP.

“GRASP (Greedy Randomized Adaptive Search Procedure), Procedimiento de Búsqueda miope, aleatorizada y adaptativa, este método consiste en generar múltiples soluciones iniciales aleatorias aplicándole a cada una de ellas una búsqueda local”^[XXVI]

Cada iteración GRASP consiste en dos fases: una fase constructiva la cual se genera una solución buena, aunque no necesariamente un óptimo local, en la segunda fase se realiza una búsqueda en la vecindad con el objetivo de encontrar un óptimo local.^[XXVII] Con esto obtenemos un óptimo local por iteración, así continúan las iteraciones guardando la mejor solución encontrada, hasta alcanzar el criterio de finalización.

3.6.4 REDES NEURONALES.

"Las redes neuronales Artificiales (RNA) son modelos matemáticos multiparamétricos no-lineales, el procesamiento de la información se realiza de manera tal de emular los sistemas neuronales biológicos" [XXVIII]

Las RNA trabajan con pesos asignados a cada una de las neuronas, estos pueden ser fijos o adaptables, las que tienen pesos adaptables emplean métodos de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas.

Las RNA adaptables tienen dos fases en su operación, la primera es la fase de entrenamiento en la cual se especifican las entradas y salidas y se otorgan los pesos de manera que la salida sea lo suficientemente cercana al valor de salida. La segunda fase es la de evaluación en la cual se especifican las entradas y se obtiene la salida en base a los pesos obtenidos en la fase de entrenamiento.

En base a la revisión de los métodos meta heurísticos analizados, el algoritmo genético cumple con las necesidades para la resolución del problemas de planificación de horarios, esto sin desmerecer ninguno de los otros métodos, consideramos que con la creación de individuo

y de poblaciones que nos ofrecen los algoritmos genéticos podemos realizar una buena representación del problema como se especifica en la sección 3.5.

CAPÍTULO 4

4. DESARROLLO DE LA APLICACIÓN

Una visión general del análisis de sistemas nos permite, a partir de la comprensión de los requisitos, el análisis de sus implicaciones y el entorno en el que va a instaurarse, obtener un modelo comprensible y correcto del mundo real, indicando lo que es preciso hacer sin limitar la forma en que ha de hacerse.^[XXXI]

El diseño por su parte, es una estrategia de alto nivel para resolver el problema planteado y ofrecer una solución^[XXXII], en esta etapa, se realiza un enfoque más general del problema, se definen los algoritmos necesarios para cumplir con los requerimientos, se plantean estructuras, arquitecturas y factores que ayudaran en la optimización del sistema^[XXXIII]. En adelante detallaremos el ciclo de desarrollo del APS de Administración de Recursos a nivel FIEC

4.1 RECOPIACIÓN DE DATOS

La ESPOL nos facilita los datos de profesores, materias, paralelos, asignación de profesores por materia, horarios de disponibilidad y aulas.

- Los horarios de aula y laboratorios son desde las 7:30 a las 21:30
- Los horarios de clases aula y laboratorios los días Sábados son de 7.30 hasta las 14.30
- Los sábados puede dictarse clases de todas las materias y laboratorios.
- Los profesores se categorizan por antigüedad, aunque esta no sea una norma, pero se considera siempre a la hora de elaborar los horarios.
- Los profesores se clasifican como: antiguos con y sin nombramiento y nuevos con o sin nombramiento.
- Los profesores con nombramiento tienen todo su horario disponible salvo excepciones.
- Las materias del mismo nivel no pueden ser dictadas en el mismo horario.
- Para cada profesor se obtiene su disponibilidad de horario.

- Materia del mismo nivel y de la misma especialidad tienen un porcentaje límite de cruzamiento del 50%, superior a ello no debe entrar en consideración.
- Las sesiones de clases generalmente son de dos horas.
- Existen excepciones de 1 y 3 horas de clases.
- Existen bloques predefinidos hasta ahora intuitivamente de horarios de clases. Por ejemplo un paralelo debe recibir una materia a la misma hora pasando un día en la misma aula.
- El escenario ideal para dictar una materia sería que para la misma todos sus paralelos se dicten a la misma hora.
- Con la excepción de las materias de laboratorios, todas las materias pueden ser dictadas en todas las aulas.
- Se dará preferencia en la asignación de aulas a las materias que pertenezcan a dicha especialización. Por ejemplo: si un profesor es de computación, preferiblemente usará las aulas de computación.
- Dependiendo de la cantidad de alumnos registrados en una materia dependerá el aula a asignarse.
- Los profesores especificarán sus requerimientos de material didáctico tales como: proyector, pizarra de tiza

liquida, pizarra de tiza, proyector de diapositivas, aire acondicionado.

- Para los profesores de computación es preferible asignar aulas con proyectores, o por lo menos el 50% de sus horas deberán dictarse en estas aulas.
- El sistema debe ser capaz de almacenar antiguas planificaciones.
- El sistema debe permitir reanudar una planificación con datos actualizados por ejemplo agregar paralelos a la planificación sin que se altere la planificación generada.

4.2 REPRESENTACIÓN HEURÍSTICA

Dados los diversos tipos de algoritmos de búsqueda heurística como recocido simulado, redes neuronales, búsqueda tabú, grafos y algoritmos genéticos; se trata de buscar una representación heurística dada la estructura del problema, tomando en cuenta la necesidad de soluciones flexibles que cumplan si no con todos, con la mayoría de los requerimientos iniciales.

Como resultado, se llegó a determinar que los algoritmos que brindaban la representación más próxima y que cumplen con los parámetros determinantes del problema son los algoritmos genéticos.

De los métodos antes mencionados se escogieron tres para ser evaluados con el problema a resolver, entre ellos recocido simulado, redes neuronales y algoritmos genéticos. Los dos primeros no dieron la suficiente apertura para obtener una representación que se ajustara de manera eficiente al problema de planificación de horarios, razones que se detallan a continuación:

El recocido simulado por su parte recibe cuatro parámetros como lo definimos a continuación:

Recocido (TemperaturaMínima, TemperaturaMáxima, MáximoNum
Iteraciones, SoluciónInicial)

El punto clave es la determinación de lo que sería nuestra solución inicial por procesos iterativos recursivos de las variaciones de temperatura entre la solución inicial y las soluciones subsiguientes, el costo de procesamiento y el tiempo de proceso resultan como una de sus mayores desventajas pese a que sus resultados puedan ser muy eficaces.

Las temperaturas mínima y máxima serán función de los costos de obtención de las soluciones donde:

$$T_{\min} = \left(\frac{\Delta E}{\ln(\text{maIteraciones})} \right)$$

$$T_{\max} = \left(\frac{-0,3 * \Delta E}{\ln(\text{max Iteraciones})} \right)$$

$$\Delta E = \text{costoSolucionActual} - \text{costoSolucionAnterior}$$

La base de este algoritmo es calcular el cambio de energía hasta llegar a un punto de congelación, en el proceso de evitar mínimos locales se debe determinar cuantas iteraciones deben realizarse en cada temperatura y en cuanto decrementar la temperatura en cada paso.

En el caso de las redes neuronales, podemos observar que su orden de su algoritmo en el nivel más interno es igual a:

$$O(\text{numSoluciones}) * O(\text{numCapas}) * O(\text{numNeuronas}^2)$$

Mismo que queda representado por el siguiente pseudo código:

```

For m = 1 To numSoluciones
  For i = 0 To numCapas
    For j = 1 To numNeuronas
      For k = 1 To numNeuronas
        net(i + 1, j) = net(i + 1, j) + Y(i, k) * pesos(i, j, k)
        Y(i + 1, j) = 1 / (1 + 2.71828182 ^ (net(i + 1, j)))
      Next k
    Next j
  Next i
Next m

```

En esta representación se consideraron únicamente dos capas, durante la cual se busca minimizar el error de prueba del algoritmo y la estabilización de los pesos, aunque este proceso nos permite converger más rápido a una posible solución, el proceso de búsqueda de la solución es al igual que en recocido simulado un proceso lineal lo que reduce y limita significativamente su eficiencia.

Por su parte el proceso de búsqueda del algoritmo genético es ciego, esto quiere decir que no sometemos el proceso de búsqueda a restricciones previas, solo se plantea una heurística, misma sobre la que se trabajara para buscar soluciones factibles; al final del proceso,

es la solución de menor costo la óptima, se asegura que este proceso puede detenerse durante una ejecución y tener soluciones previas que aunque con ciertas limitantes nos ayudarían a vislumbrar una solución lo suficientemente próxima a la óptima.

La solución deberá ser expresada por un espacio de soluciones, hallado a partir de operaciones de exploración y explotación hasta encontrar los individuos más óptimos.

Lo anterior nos conlleva a generar espacios de soluciones que no necesariamente deben ser soluciones óptimas y se determinarán las soluciones que cumplan con los requerimientos solicitados.

El uso de algoritmos genéticos nos permitirá encontrar cada vez mejores cromosomas (soluciones) mediante manipulación “ciega” (cruzamiento y mutación) de sus contenidos.

“Un Algoritmo Genético es una estructura de control que dirige un conjunto de transformaciones y operaciones diseñadas para simular estos procesos de evolución” [XX]

Esta manipulación de su contenido, se basa en tres pasos:

- a.- Escoger una población inicial de soluciones de prueba elegidas en forma aleatoria.
- b.- Cada solución es replicada hacia un nuevo espacio de salida.
- c.- A cada recurso se calcula su costo dentro del conjunto solución llamado “costo óptimo” o “fitness”. (Este valor dependerá, no solo de cumplir las condiciones iniciales, sino además de la optimización de la distribución de los recursos concernientes al problema).

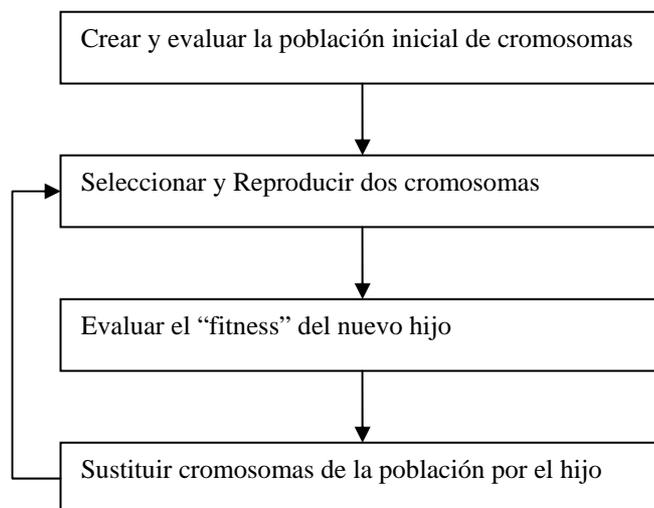


Ilustración 4-1 Esquema básico de los algoritmos genéticos [XXXVII]

4.2.1 SELECCIÓN DE LA GENERACIÓN INICIAL

Realizamos un proceso iterativo entre uno y el número de cromosomas definido por el usuario.

```

Function generacionInicial(NumCromosomas As Integer) As cCromosoma()
    For i = 1 To NumCromosomas
        generarCromosoma(i) = New cCromosoma
        generarCromosoma(i).calcularFitness
    next i
End function

```

generarCromosoma		cCromosoma(1 to 10)	Operaciones.generacionInicial
generarCromosoma(1)		cCromosoma/cCromosoma	Operaciones.generacionInicial
C_status	"CGI"	String	Operaciones.generacionInicial
fitness	423,6	Double	Operaciones.generacionInicial
gen		cGen(1 to 912)	Operaciones.generacionInicial
gen(1)		cGen/cGen	Operaciones.generacionInicial
X	1	Integer	Operaciones.generacionInicial
y	4	Integer	Operaciones.generacionInicial
z	9	Integer	Operaciones.generacionInicial
gen(2)		cGen/cGen	Operaciones.generacionInicial
X	2	Integer	Operaciones.generacionInicial
y	4	Integer	Operaciones.generacionInicial
z	10	Integer	Operaciones.generacionInicial
gen(3)		cGen/cGen	Operaciones.generacionInicial
X	3	Integer	Operaciones.generacionInicial
y	4	Integer	Operaciones.generacionInicial
z	37	Integer	Operaciones.generacionInicial
gen(4)		cGen/cGen	Operaciones.generacionInicial
X	4	Integer	Operaciones.generacionInicial
y	4	Integer	Operaciones.generacionInicial
z	38	Integer	Operaciones.generacionInicial
gen(5)		cGen/cGen	Operaciones.generacionInicial
gen(6)		cGen/cGen	Operaciones.generacionInicial
gen(7)		cGen/cGen	Operaciones.generacionInicial
gen(8)		cGen/cGen	Operaciones.generacionInicial
gen(9)		cGen/cGen	Operaciones.generacionInicial
gen(10)		cGen/cGen	Operaciones.generacionInicial
gen(11)		cGen/cGen	Operaciones.generacionInicial

Ilustración 4-2: Representación de la Generación Inicial

La ilustración 4-2 muestra un número total de cromosomas de 10, un valor de fitness de 423,6 y una dimensión del cromosoma de 912.

Cada cromosoma tiene una dimensión igual al número de combinaciones profesor materia, cada una de estas combinaciones multiplicada por su carga horaria semanal, valor representado en las ilustraciones 4-2 y 4-3 como 912:

```

For i = 1 To numProfesorMateria
    For j = 1 To cargaMat
        generarGen (i, horasAsig(j), aula)
    next j
next i
    
```

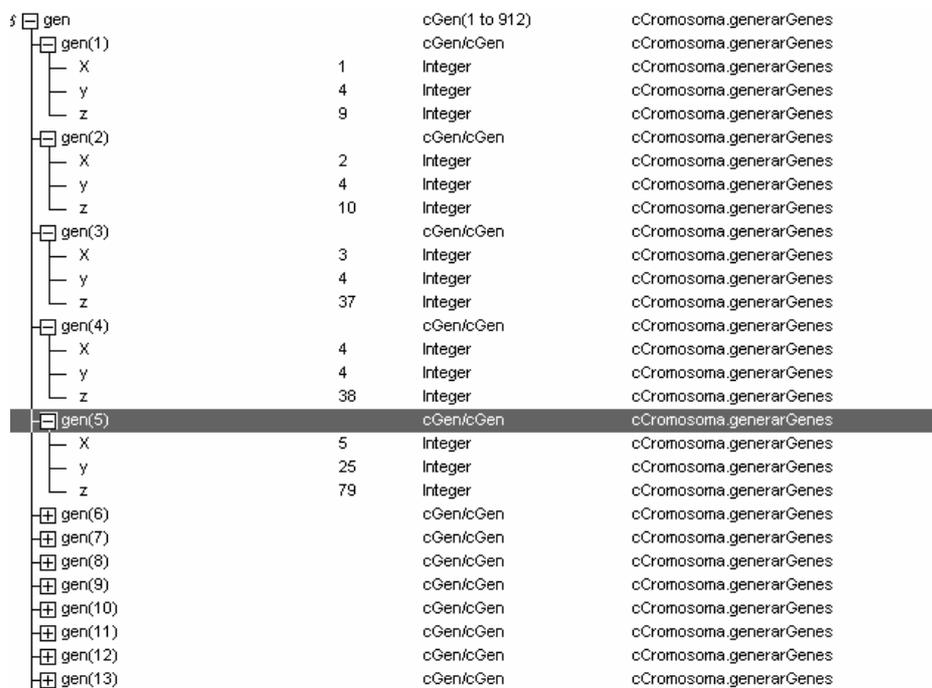


Ilustración 4-3: Representación de un Cromosoma

En la ilustración 4-3 podemos ver que cada gen esta representado por el índice de la combinación profesor materia (X), un bloque de horario (Y) y el índice del aula donde será dictada la materia (Z).

Para asignar un bloque de horarios al gen debe respetarse el horario de disponibilidad del profesor que posee la estructura mostrada en la ilustración 4-4.

`validaDispProfesor(codProfesor, horasAsig)`

Por ejemplo para el profesor con cedula de identidad 091765102-3 se verifica si tiene disponibles las horas pertenecientes al bloque 1-2-29-30.

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
07:30 - 08:30	1	15	29	43	57	71
08:30 - 09:30	2	16	30	44	58	72
09:30 - 10:30	3	17	31	45	59	73
10:30 - 11:30	4	18	32	46	60	74
11:30 - 12:30	5	19	33	47	61	75
12:30 - 13:30	6	20	34	48	62	76
13:30 - 14:30	7	21	35	49	63	0
14:30 - 15:30	8	22	36	50	64	0
15:30 - 16:30	9	23	37	51	65	0
16:30 - 17:30	10	24	38	52	66	0
17:30 - 18:30	11	25	39	53	67	0
18:30 - 19:30	12	26	40	54	68	0
19:30 - 20:30	13	27	41	55	69	0
20:30 - 21:30	14	28	42	56	70	0

Ilustración 4-4: Matriz de Horario de disponibilidad de Aulas i/o Profesores.

De la misma forma para asignar un aula al gen debe respetarse el horario de disponibilidad del aula que posee la estructura mostrada en la ilustración 4-4.

pedirAula(horasAsig, cargaMat, codmateria, tipoMat, capacidad,
recursos)

Por ejemplo para el la materia FIEC02530 se busca un aula que tenga disponibles las horas pertenecientes al bloque 1-2-29-30, cuya capacidad sea mayor o igual al requerimiento de la materia y que disponga de los recursos especificados por el profesor para dictar dicha materia.

Los bloques de horario predefinido podemos observarlos en la ilustración 4-5, la ilustración 4-6 muestra esta representación en la base de datos.

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
07:30 - 08:30	1	15	29	43	57	71
08:30 - 09:30	2	16	30	44	58	72
09:30 - 10:30	3	17	31	45	59	73
10:30 - 11:30	4	18	32	46	60	74
11:30 - 12:30	5	19	33	47	61	75
12:30 - 13:30	6	20	34	48	62	76
13:30 - 14:30	7	21	35	49	63	
14:30 - 15:30	8	22	36	50	64	
15:30 - 16:30	9	23	37	51	65	
16:30 - 17:30	10	24	38	52	66	
17:30 - 18:30	11	25	39	53	67	
18:30 - 19:30	12	26	40	54	68	
19:30 - 20:30	13	27	41	55	69	
20:30 - 21:30	14	28	42	56	70	

Ilustración 4-5: Bloques de Horario Predefinidos

id_bloque	hora1	hora2	hora3	hora4
1	1	2	29	30
2	3	4	31	32
3	5	6	33	34
4	7	8	35	36
5	9	10	37	38
6	11	12	39	40
7	13	14	41	42
8	15	16	43	44
9	17	18	45	46
10	19	20	47	48
11	21	22	49	50
12	23	24	65	66
13	25	26	67	68
14	27	28	55	56
15	57	58	71	72
16	59	60	73	74
17	51	52	61	62
18	53	54	63	64
19	69	70	75	76

Ilustración 4-6: Representación en la Base de Datos de los Bloques de Horario

Posterior a ello se controlan los cruces de materia del mismo nivel y se reducen los horarios de disponibilidad del profesor y aula involucrada.

Finalmente, para cada cromosoma generado se calcula su valor fitness.

4.2.2 SELECCIÓN DE LOS CROMOSOMAS PADRES

Los padres se seleccionarán aleatoriamente de la generación inicial.

4.2.3 REPRESENTACIÓN DE UN STRING

Un string que contiene índices de matrices, hasta ubicarse todos los valores de k:

Cromosoma = (materia- profesor((1..k)*n), horario asignado (1..76), aulas(1..n))

$$\forall k, n \in \pi^+, k = 1, 2, 3, \dots, n$$

Horario asignación = bloque de horario de disponibilidad del profesor evaluado en el horario de disponibilidad del aula seleccionado de los bloques de horario predefinidos. Refiérase a la ilustración 4-3.

4.2.4 OPERACIONES DE CRUZAMIENTO

Existen dos operaciones de cruzamiento utilizando uno o dos puntos de cruce, el proceso de elección entre una y otra es aleatorio.

Para el método de dos puntos de cruzamiento (two point crossover), los puntos de cruzamiento son seleccionados aleatoriamente, los cromosomas se dividirán en tres partes de las cuales el nuevo cromosoma estará formado de dos partes del primer cromosoma y una parte del segundo cromosoma. El cruzamiento se realiza por paralelo de manera que si los puntos de cruzamiento separan las horas del paralelo, se avanzará hacia abajo hasta que el punto de cruce se ubique en un gen con una relación profesor-materia-aula diferente.

	Materia-Profesor	Horario	Aula
	1	1	12
Crossover point 1 →	1	2	12
	1	15	12
	1	16	12
	2	8	7
Crossover point 2 →	2	9	7
	2	22	7
	2	23	7
	3	2	2
	K	5	5

Ilustración 4-7: Cromosoma 1 para dos puntos de cruce.

	Materia- Profesor	Horario	Aula
Crossover point 1	1	5	20
	1	6	20
	1	19	20
	1	20	20
Crossover point 2	2	10	3
	2	11	3
	2	24	3
	2	25	3
	3	2	30
	k	5	5

Ilustración 4-8: Cromosoma 2 para dos puntos de cruce.

	Materia- Profesor	Horario	Aula
Crossover point 1	1	1	12
	1	2	12
	1	15	12
	1	20	20
Crossover point 2	2	10	3
	2	11	3
	2	24	3
	2	23	7
	3	2	2
	K	5	5

Ilustración 4-9: Cromosoma resultado con dos puntos de cruce.

```

For j = 1 To puntoCruce1
    CromosomaCruzado.SetGen j, genInicial(cromo1).GetGen(j)
Next j

```

```

For j = puntoCruce1 + 1 To puntoCruce2
    CromosomaCruzado.SetGen j, genInicial(cromo2).GetGen(j)
Next j

```

```

For j = puntoCruce2 + 1 To numProfMatPar
    CromosomaCruzado.SetGen j, genInicial(cromo1).GetGen(j)
Next j

```

Cuando se tiene un solo punto de cruce, también llamado cruzamiento horizontal (one point crossover), el punto de cruzamiento se selecciona aleatoriamente, los cromosomas se dividirán en dos partes de las cuales el nuevo cromosoma estará formado en su parte superior por la información genética primer cromosoma hasta llegar al punto de cruce y en su parte inferior por la información genética obtenida del segundo cromosoma posterior del punto de cruce.

Materia-Profesor	Horario	Aula
1	1	12
1	2	12
1	15	12
1	16	12
2	8	7
2	9	7
2	22	7
2	23	7
3	2	2
k	5	5

Crossover point →

Ilustración 4-10: Cromosoma 1 para cruzamiento horizontal

Materia-Profesor	Horario	Aula
1	5	20
1	6	20
1	19	20
1	20	20
2	10	3
2	11	3
2	24	3
2	25	3
3	2	30
k	5	5

Crossover point →

Ilustración 4-11: Cromosoma 2 para cruzamiento horizontal

Materia- Profesor	Horario	Aula
1	1	12
1	2	12
1	15	12
1	20	20
2	10	3
2	11	3
2	24	3
2	25	3
3	2	30
k	5	5

Ilustración 4-12: Cromosoma resultado con un punto de cruce.

4.2.5 OPERACIONES DE MUTACIÓN

La cantidad de cromosomas a mutar es función del número de cromosomas a cruzar y constituye un segmento de la generación de cromosomas donde:

$$\text{GeneracionMutados} = \text{Operaciones.Mutacion}(\text{porcentajeMutacion}, \\ \text{GeneracionCruzados})$$

$$\text{numCromoCruce} = \text{Int}(\text{porcentajeCruzamiento} * \text{NumCromosomas} / 100)$$

$$\text{numCromoMutar} = \text{Int}(\text{porcentajeMutacion} * \text{numCromoCruce} / 100)$$

Aleatoriamente se escoge el cromosoma que será mutado con un proceso desde uno al número de cromosomas a mutar numCromoMutar, que en realidad en la generación de un nuevo cromosoma, por el cambio total en el mismo.

4.3.6 CÁLCULO DEL FITNESS

El fitness es la valoración que se le da a un individuo o cromosoma por cada violación a los requerimientos. Como ejemplo de estas violaciones a los requerimientos tenemos: cruces de horario, falta de recursos en las aulas, asignación de aulas según especialización de la materia, materias que no se dictan en su laboratorio respectivo, etc.

De ocurrir casos como los especificados se penalizará al individuo. La penalización consiste en aumentar un valor determinado al fitness, el mejor individuo será el que posea el menor fitness.

4.3.7 TRATAMIENTO DE EXCEPCIONES

Si dentro de la organización de los horarios se requiere un horario diferente al de medias horas, como 7:30, 08:30. etc., o algún profesor desea un aula en particular, esta asignación se realizará antes de la ejecución del algoritmo y no entrará en el proceso de optimización.

4.3 MODELO DE ANÁLISIS

4.3.1 MODELOS ESTÁTICOS

4.3.1.1 DISEÑO CONCEPTUAL DE LA BASE DE DATOS

El diseño conceptual de la base de datos buscará modelar la realidad y almacenar los datos que son relevantes para los usuarios del sistema, existirán datos que no aporte información a la solución del problema y que por ende se obviarán con la única finalidad de comprender la realidad de los usuarios que utilizarán el sistema.

Modelo Conceptual de la Base de Datos

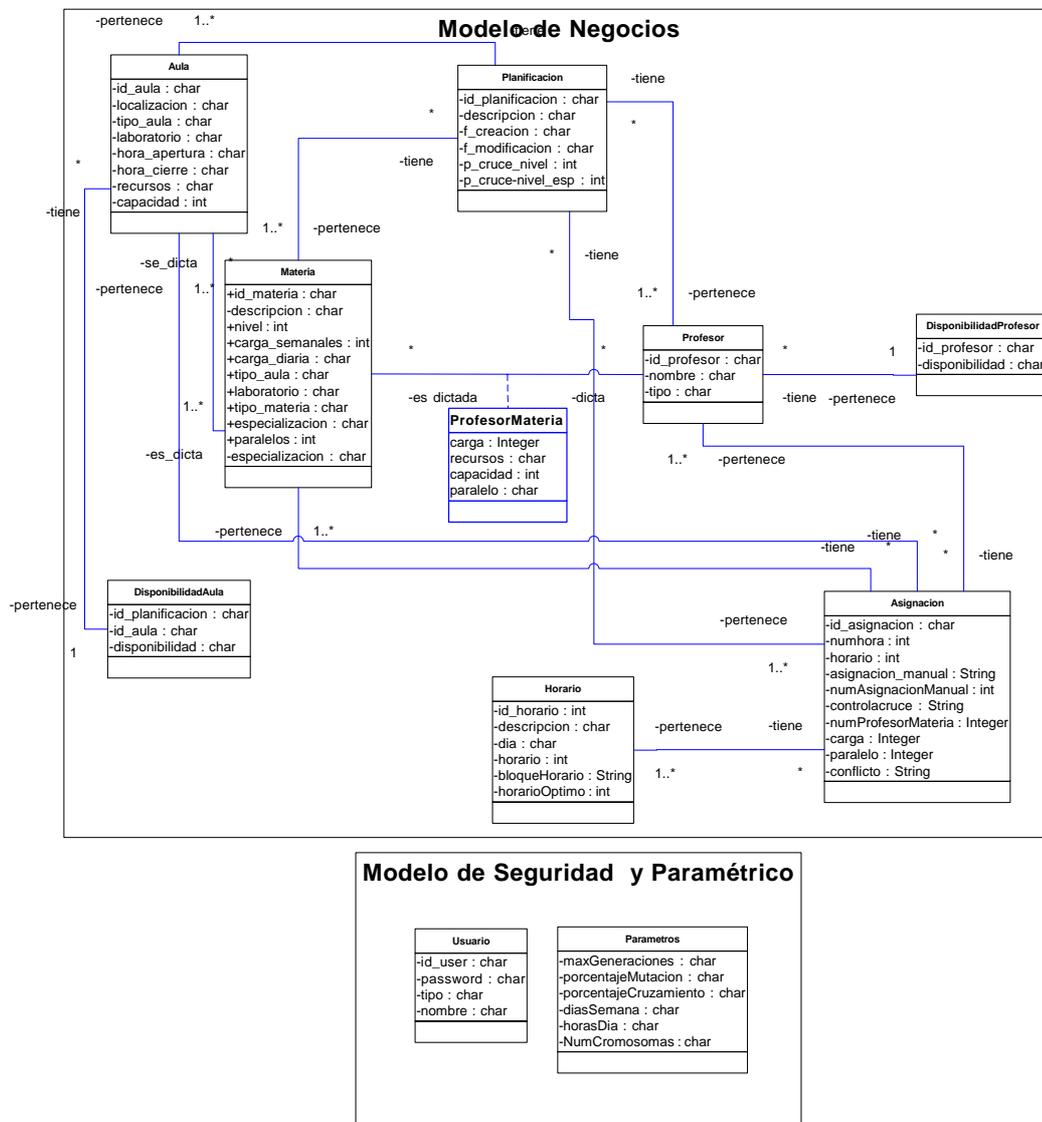


Ilustración 4-13: Diseño Conceptual de la Base de Datos

4.3.1.2 MODELO DE ANÁLISIS DE OBJETOS

Modelo de Análisis de Objetos

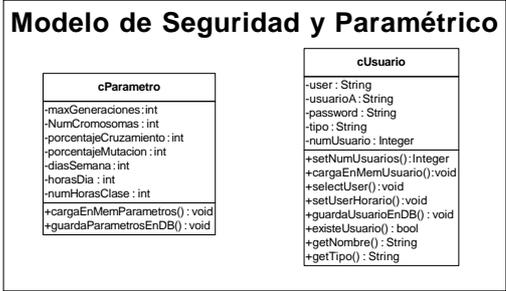
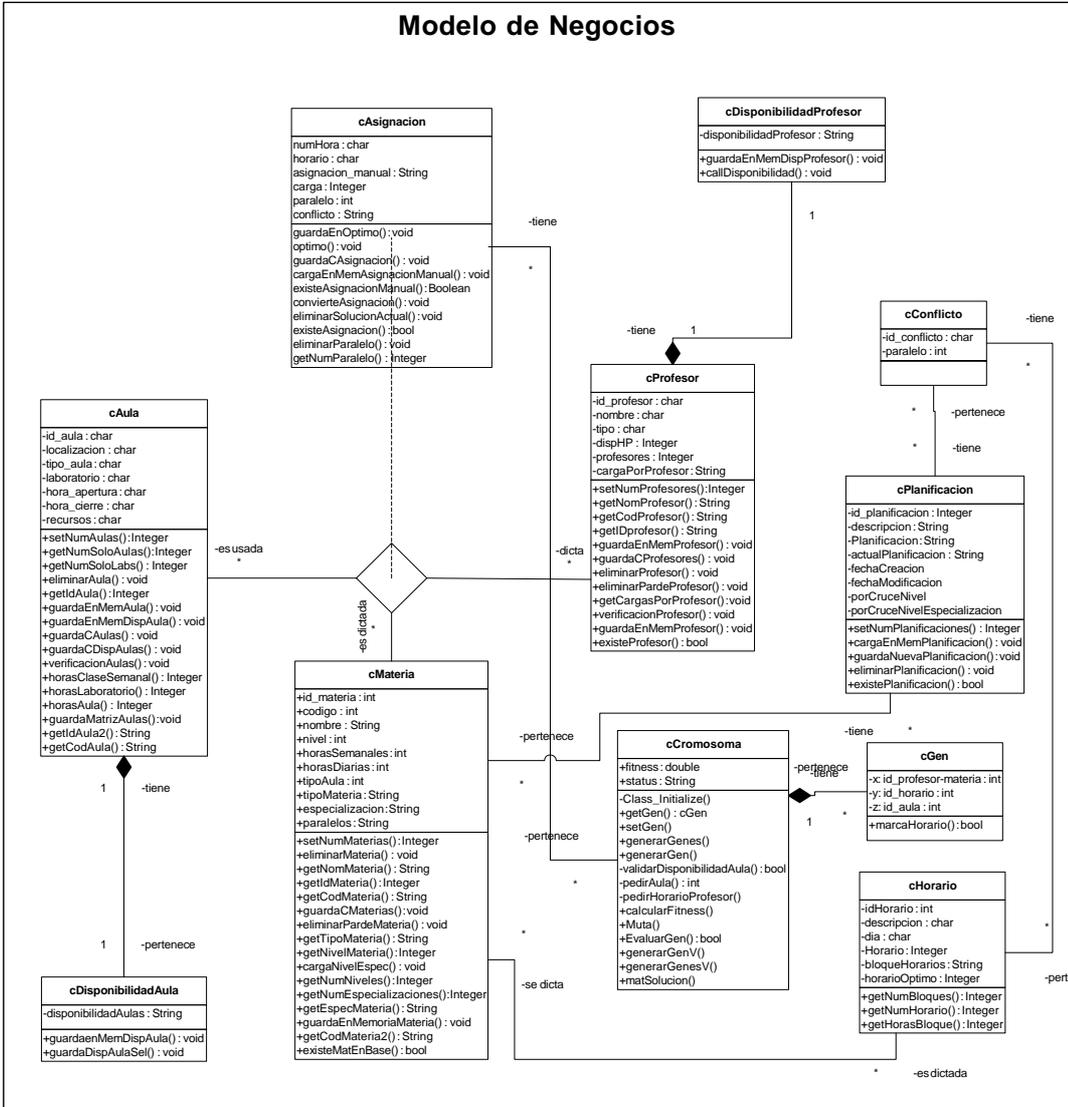


Ilustración 4-14: Modelo de Análisis de Objetos

4.3.1.3 DESCRIPCIÓN DE CLASES

<p>Nombre: cAsignación.</p> <p>Descripción: Almacena las asignaciones profesor materia aula.</p> <p>Atributos:</p> <ul style="list-style-type: none"> ▪ Public asignacion_manual() As String 'matriz que contiene los ingresos especiales. ▪ Public numAsignacionManual As Integer 'numero de registros de tabla asignacion_manual. ▪ Public controlaCruce() As String 'matriz que contiene las asignaciones de horarios a materias para evitar el cruce de materias del mismo nivel. <p>Operaciones:</p> <ul style="list-style-type: none"> ▪ Public Sub optimoDB()' Transforma la solución óptima en la matriz matSolucion, ▪ ' lista para ser almacenada en la base de datos. ▪ Public Sub guardaEnDBOptimo()' Guarda en la base de datos la mejor solución que el algoritmo genere. ▪ Public Sub guardaCAsignacion(planiGc As String)' Guarda las asignaciones profesor-materia de la actual planificación a la nueva. ▪ Public Sub cargaEnMemAsignacionManual()' Almacena la información de la tabla asignacion_manual' en la matriz asignacion_manual. ▪ Public Sub setNumAsignacionManual()' Función que obtiene el numero de registros de la tabla asignacion_manual. ▪ Public Function existeAsignacionManual(id_prof As String, id_mat As String, id_aula As String, paralelo As Integer, dia As String, horaIni As String, horaFin As String)' Verifica en la base si la asignación que se desea realizar ya ha sido ingresada en la opción ingreso de paralelos especiales. ▪ Public Sub convierteAsignacion()'Procedimiento que transforma las asignaciones manuales en asignaciones normales' las almacena en la tabla asignación, y coloca las disponibilidad de aula y profesor a 1 'esto debe ser antes de grabar los resultados del algoritmo. ▪ Public Function materiaRelacionada(codmateria) As Boolean' Dado el código de la materia se verifica si esta registrada en la tabla profesor-materia. ▪ Public Function existeAsignacionPMDB(codmateria As String, paralelo As Integer) As Boolean ' Función que verifica si el paralelo ya ha sido asignado a otro profesor. ▪ Public Function existeAsignacionManualDb(codmateria As String, paralelo As Integer) As Boolean 'Función que verifica si el paralelo ya ha sido asignado a otro profesor. ▪ Public Function existeHorarioEnAsignacionManualDb(aula As String, dia As String, horaIni As String) As Boolean 'Función que verifica si el horario existe en asignación manual. ▪ Public Function existeAsignacionManualDb2(codprof As String, codmateria As String, codAula As String, paralelo As Integer, dia As String, horaIni As String, horaFin As String) As Boolean'Funcion que verifica si el paralelo ya ha sido asignado a otro profesor. ▪ Public Sub eliminarSolucionActual() ▪ Public Function existeAsigdePProfPM(codprof) As Boolean. ▪ Public Function existeAsigdePProfManual(codprof) As Boolean 'Función que verifica si el profesor tiene asignada alguna materia en asignación manual. ▪ Public Function existeAsignacion(materia, paralelo) As Boolean' Verifica si el paralelo ya ha sido asignado ▪ Public Sub eliminarAsignacion_manual()' Eliminar los registros de la tabla asignacion_manual. ▪ Public Sub setDispAsigManual().
--

Ilustración 4-15:Clase cAsignacion

<p>Nombre: cConexion.</p> <p>Descripción: Conexión con la base de datos.</p> <p>Atributos:</p> <ul style="list-style-type: none"> ▪ Public Conexion As adodb.Connection. ▪ Public status. ▪ Public Registro As New adodb.Recordset. <p>Operaciones:</p> <p>Public Sub abrir_base()' Procedimiento para abrir la conexión con la base de datos.</p> <p>Public Sub Cerrar_base()' Procedimiento para cerrar la base de datos.</p>
--

Ilustración 4-16: Clase cConexión

Nombre: cAula.

Descripción: Contiene información de las aulas, su codificación, ubicación, recursos disponibles y el tipo de aula (laboratorio o aula).

Atributos:

- Public numhorasAula As Integer.
- Public numhorasLaboratorio As Integer.
- Public totalhorasLaboratorio As Integer.
- Public totalhorasAula As Integer.
- Public aulas() As Integer.
- Public numAulas As Integer.
- Public disponibilidadAulas() As String.
- Public disponibilidadAulasT() As String.
- Public numSoloLabs As Integer ' numero de laboratorios.
- Public numSoloAulas As Integer ' numero de aulas.
- Public mSoloAulas() As String ' matriz que contiene solo las aulas.
- Public mSoloLabs() As String ' matriz que contiene solo los laboratorios.
- Public DispAula() As String.
- Public numHorasTotAulas As Integer ' numero total de disponibilidad de todas las aulas.
- Public aulaMem() As String.
- Public tipo_aula As String.
- Public tipo_aulaA As String.
- Public tipoAula As Integer.
- Public proy, info, aire As Integer.
- Public capacidad As String.

Operaciones:

- Public Sub setNumAulas() ' Estable el numero de las aulas y laboratorios en la variable numAulas.
- Public Sub getNumSoloAulas() ' Asigna el número de aulas en la variables numSoloAulas.
- Public Sub getNumSoloLabs() ' Asigna el número de laboratorios en la variables numSoloAulas.
- Public Sub eliminarAula(codAula As String) 'Procedimiento que elimina un aula.
- Public Function getIdAula(idAula) ' devuelve el índice de la matriz aulaMem ,recibe código del aula.
- Public Sub guardaEnMemAula() 'Almacena la información de la tabla aula a la matriz "aulaMem".
- Public Sub guardaEnMemDispAula() ' Función que guarda en memoria los datos de la tabla Disponibilidad de aula.
- Public Sub guardaDispAulasEnDB() ' Guarda en la base de las disponibilidades de las aulas.
- Public Sub actDispAulasEnDB() 'actualiza la tabla de disponibilidad de aulas en base a la matriz matDispAulaDb.
- Public Sub guardaDispDAulaSel(codAula As String) ' Procedimiento que actualiza la disponibilidad del profesor seleccionado, al aceptar.
- Public Sub guardaCAulas(planiGc As String) ' Guarda las Aulas de la actual planificación a la nueva en base a la planificación.
- Public Sub guardaCDispAulas(planiGc As String) ' Guarda las Disponibilidad de las Aulas de la actual planificacion a la nueva.
- Public Sub cargaSoloAulas() ' Carga en memoria solo las aulas en la matriz mSoloAulas.
- Public Sub cargaSoloLabs() ' Procedimiento que carga en memoria solo los laboratorios.
- Public Sub verificacionAulas() ' Almacena en numHorasTotAulas las horas disponibles de todas las aulas.
- Public Sub horasClaseSemanal() ' Almacena en totalhoraSemanales el numero total de hora clase de todas las materias, tomada de la tabla profesormateria ' Almacena en totalhorasLaboratorio.- el numero total de horas de laboratorio asignadas ' y en totalhorasAula el numero total de horas d aulas asignadas.
- Public Sub horasLaboratorio() ' Asigna las horas asignadas de los laboratorios para esta planificación ' en la variables totalhorasLaboratorio.
- Public Sub horasAula() ' Asigna el total de horas de todas las aulas disponibles para la planificación.
- Public Function guardaMatrizAulas(indiceAula As String, da() As String) ', dimda As Integer) ' Guarda la disponibilidad de las aulas en la matriz disponibilidadAulas.
- Public Function getIdAula2(idAula) ' Obtiene el la posición del aula dentro de la matriz disponibilidadAulas.
- Public Function getCodAula(id As Integer) ' Obtiene el código del aula de la matriz aulaMem.
- Public Sub matrizAulas() ' Establece la disponibilidad de las aulas.

Ilustración 4-17: Clase cAula

Nombre: cMateria.**Descripción:** Contiene información sobre las materias pertenecientes al pensum académico de la institución.**Atributos:**

- Public codigo As Integer.
- Public nombre As String.
- Public nivel As Integer.
- Public horasSemanales As Integer.
- Public horasDiarias As Integer.
- Public tipoMateria As Integer.
- Public Materias() As Integer.
- Public numMaterias As Integer.
- Public materia() As String ' matriz que contiene las materias.
- Public numEspec As Integer ' numero de especializaciones.
- Public tipo_materia As String.
- Public numEspecializaciones As Integer ' contador de la tabla especialización.
- Public especializacion() As String ' matriz que contiene las especializaciones.

Operaciones:

- Public Sub setNumMaterias()' Guarda en memoria el numero de aulas de la planificación actual.
- Public Sub eliminarMateria(codMat As String)' Procedimiento que elimina una materia.
- Public Function getNomMateria(codMateria As String)' Devuelve el nombre de una materia, recibe el código de la materia.
- Public Function existeMateria(codMat As String) As Boolean' verifica la existencia de una materia, recibe código de la materia.
- Public Function getIdMateria(codMat As String)' Obtiene el Id de la materia.
- Public Function getCodMateria(id As Integer)' Obtiene el código de la materia en base a la posición en la matriz materia.
- Private Sub llenaMaterias0()' Llena el como l_todas con las materias existentes.
- Public Sub guardaCMaterias(planiGc As String)' Guarda las Materias de la actual planificación a la nueva.
- Public Sub guardaCEspecializaciones(planiGc As String)' Guarda las Especializaciones de la actual planificación a la nueva.
- Public Sub eliminarPardeMateria(codMat As String)' Eliminar los registros de la tabla profesorMateria.
- Public Function getTipoMateria(codMat As String)' Función que devuelve el tipo de materia "A" o "L", recibe código de la materia.
- Public Function getNivelMateria(codMat As String)' Función que devuelve el tipo de materia "A" o "L", recibe código de la materia.
- Public Sub cargaNivelEspec()' Carga en memoria la el arreglo de de niveles y especializaciones para poder calcular en fitness el porcentaje de cruces.
- Public Sub getNumNiveles()' Obtiene el número de niveles y los almacena en la variable numNiveles.
- Public Sub getNumEspecializaciones()' Obtiene el número de especializaciones y los almacena en la variable numEspec.
- Public Function getEspecMateria(codMat As String)' Función que devuelve la especialización de una materia, recibe código de la materia.
- Public Sub guardaEnMemoriaMateria()' guarda en memoria los datos de la tabla materia' de la planificación actual.
- Public Function getCodMateria2(nomMat As String)' Función que devuelve el código de la materia y recibe el nombre de la materia.
- Public Function existeMatEnBase(matSel)' Verifica la existencia de la materia dentro de la planificación en la base de datos.
- Public Function existeEspecializacion(espec As String) As Boolean' Verifica la existencia de la especialización.
- Public Function guardaNuevaEspecializacion(idEsp As String, espec As String) As Boolean' Guarda una nueva planificación.
- Public Sub setNumEspecializaciones()' Guarda en memoria el numero de especializaciones de la planificación actual.
- Public Sub cargaEspecializacion()' Guarda en memoria los datos de la tabla especialización' de la panificación actual.

Ilustración 4-18: Clase cMateria

Nombre: Horario.

Descripción: Almacena los bloques de horario predefinidos e información relacionada.

Atributos:

- Public horaInicio As String ' variable que contiene la hora de inicio de un clase.
- Public Horario(NumHorario) As Integer.
- Public bloqueHorarios() As String ' matriz que contiene los bloques de horarios.
- Public horarioOptimo() As Integer.

Operaciones:

- Public Sub getNumBloques()' Obtiene el número de bloques que existen en la base.
- Public Function getNumHorario(día As String, hora As String)' Función de devuelve el idhorario de la tabla horario, recibe día y hora
- Public Sub cargaBloquehorario()' Carga en memoria bloques predefinidos de horarios de 4 horas.
- Public Function getHorasBloque(bloque As Variant) As Variant' Función que devuelve las horas correspondientes al bloque solicitado.

Ilustración 4-19: Clase cHorario**Nombre:** cParalelo.

Descripción: Almacena información de los paralelos de una materia.

Atributos:

- Public numparalelo As Integer.

Operaciones:

- Public Sub eliminarParalelo(codMat As String, paralelo As Integer)' Procedimiento que elimina un paralelo, recibe materia y paralelo.
- Public Function getNumParalelo2(codmateria As String) As Integer 'Obtiene el número de paralelos de la materia.

Ilustración 4-20: Clase cParalelo**Nombre:** cParametros.

Descripción: Almacena los parámetros del sistema, sus valores por defecto y las modificaciones hechas por el administrador.

Atributos:

- Public maxGeneraciones As Integer.
- Public tmaxEjecucion As Integer.
- Public NumCromosomas As Integer.
- Public porcentajeCruzamiento As Integer.
- Public porcentajeMutacion As Integer.
- Public modaSexual As Integer.
- Public modaAsexual As Integer.
- Public numdiasSemana As Integer.
- Public numhorasDia As Integer.
- Public numHorasClase As Integer ' numero de horas semanales.

Operaciones:

- Public Sub cargaEnMemParametros()' Guarda en memoria los datos de los parámetros iniciales.
- Public Sub guardaParametrosEnDB()' Guarda en la base los parámetros iniciales.

Ilustración 4-21: Clase cParametros

Nombre: cProfesor.

Descripción: Contiene información básica de los profesores como su identificación, nombre, tipo de contrato, se asocia con la disponibilidad del profesor.

Atributos:

Public dispHP As Integer.

Public profesores() As Integer.

Public numProfesores As Integer.

Public disponibilidadProfesor() As String.

Public disponibilidadProfesorT() As String.

Public profesores2() As String ' matriz que contiene los profesores.

Public dispProfesorOptima() As String ' matriz que contiene el numero de horas disponibles del profesor.

Public dispProfesorOptimaT() As String ' matriz temporal que contiene el numero de horas disponibles del profesor.

Public DispProfesor() As String.

Public numHorasTotProfesor As Integer ' número total de disponibilidades de todos los profesores.

Public dispPorProfesor() As String ' matriz que contiene las horas disponibles por Profesor.

Public cargaPorProfesor() As String ' matriz que contiene la carga por profesor.

Public tipo_profesor As String.

Operaciones:

- Public Sub setNumProfesores() ' Guarda en profesor el número de aulas de la planificación actual.
- Public Function getNomProfesor(codprof As String) ' Devuelve el nombre de un profesor, recibe el código del profesor.
- Public Function getCodProfesor2(nomProfesor As String) ' Devuelve el código de un profesor, recibe el nombre del profesor.
- Public Function getCodProfesor(id As Integer) ' Devuelve el código del profesor dado su índice de ubicación.
- Public Function getIDprofesor(idprof) ' profesores2 ' Devuelve el índice de la tabla profesor en base al código del profesor.
- Public Sub actDispProfEnDB() ' Procedimiento que actualiza la tabla de disponibilidad de profesor en la base al aceptar.
- Public Sub guardaDispProfEnDB() ' Almacena la disponibilidad de los profesores en la base de datos.
- Public Sub guardaEnMemProfesor() ' Guarda en memoria los datos de la tabla profesor de la planificación actual.
- Public Sub guardaEnMemDispProfesor() ' Funcion que guarda en memoria los datos de la tabla Disponibilidad de profesor' los guarda en la matriz disponibilidadProfesor con un campo para cada hora.
- Public Sub guardaDispDProfSel(codprof As String) ' Procedimiento que actualiza la disponibilidad del profesor seleccionado, al aceptar.
- Public Sub guardaCProfesores(planiGc As String) ' guarda los profesores de la actual planificación a la nueva.
- Public Sub guardaCDispProf(planiGc As String) ' Guarda la Disponibilidad de los profesores en la nueva planificación.
- Public Sub eliminarProfesor(codprof As String) ' Procedimiento que elimina un Profesor, recibe código de profesor.
- Public Sub eliminarPardeProfesor(codprof As String) ' Elimina paralelos del profesor.
- Public Sub actualizaDispOptimaProfT(codprof As String) ' Procedimiento que actualiza la disponibilidad Optima temporal de un profesor en la matriz.
- Public Sub disponibilidadPorProfesor() ' Almacena en la matriz disponibilidadProfesor , las disponibilidades por profesor.
- Public Sub getCargasPorProfesor() ' Almacena en la matriz cargaPorProfesor las cargas horarias por profesor.
- Public Sub verificacionProfesor() ' Almacena en numHorasTotProfesor el numero total de horas disponibles de todos los profesores.
- Public Function existeProfesor(codprof) ' Verifica la existencia de un profesor.
- Public Function guardaMatrizProfesor(indiceProfesor As String, dp() As String) ' Almacena en la base la disponibilidad del profesor.
- Public Function callDisponibilidad() As String ' Presenta de modo gráfico la disponibilidad de los profesores.
- Public Sub guardaEnMatrizP() ' Almacena en la matriz profesores2 los datos de la tabla profesor de la base de datos.
- Public Sub MatrizProfesor() ' Crea matriz de disponibilidad de profesores con todas las horas valoradas como no disponibles.
- Public Sub setNumProfesores() ' guarda en profesor el numero de aulas de la planificación actual.

Ilustración 4-22: Clase cProfesor

Nombre: cPlanificación.

Descripción: Almacena cada una de las planificaciones que se obtienen con cada ejecución, las diferentes versiones y las soluciones multi-instituciones que puedan irse generando.

Atributos:

- Public Planificacion() As String 'matriz que contiene las planificaciones.
- Public idPlani As String.
- Public descPlani As String.
- Public actualPlanificacion As String 'variable que contiene el identificador de la planificación actual.
- Public numPlanificaciones As Integer 'numero de planificaciones.
- Public Planicacion() As String 'matriz que contiene las planificaciones.
- Public porcentajeCruceNivel As Integer 'porcentaje de cruce por nivel (total).
- Public porcentajeCruceNivelEsp As Integer 'porcentaje de cruce por nivel-Especialización.

Operaciones:

- Public Sub setNumPlanificaciones() 'Guarda en memoria el numero de aulas de la planificación actual.
- Public Sub cargaEnMemPlanificacion() 'Guarda en memoria los datos de la tabla profesor de la planificación actual.
- Public Function guardaNuevaPlanificacion(id As String, desc As String) As Boolean 'Guarda una nueva planificación.
- Public Sub eliminarPlanificacion(planiElim As String) 'Elimina una planificación.
- Public Function existePlanificacion(plani As String) As Boolean 'función que verifica la existencia de una planificación.
- Public Sub grabaPorcCruce(cruceNivel As Integer, cruceNivelEsp As Integer) 'Guarda los porcentajes de cruce de la solución.

Ilustración 4-23: Clase cPlanificacion

Nombre: cProfesorMateria..

Descripción: Asociación de profesores con las materias que dicta.

Atributos:

- Public ProfesorMateria() As String.
- Public numProfesorMateria As Integer.
- Public profesorMatPar() As String 'matriz que contiene la tabla profesor materia.
- Public numProfMatPar As Integer 'numero profesor materia paralelo (máximo del gen).

Operaciones:

- Public Sub guardaEnMemProfMatPar() 'Almacena la información de la tabla profesorMateria a la matriz "profesorMatPar".
- Public Sub llenaMatrizX() 'Llena la matriz ProfesorMateria que contiene el Contiene los datos para el X del gen.
- Public Function existeMatParEnBase(matSel, matPar As Integer) 'Función que verifica si la el paralelo-materia ya esta creado, si lo esta no lo guarda.

Ilustración 4-24: Clase cProfesorMateria

Nombre: cUsuario.

Descripción: Usuarios, tipo de usuarios, nombre, privilegios.

Atributos:

- Public tipoA As String ' tipo del usuario actual.
- Public idUsuarioA As String ' iduser actual
- Public nombreUsuarioA As String ' nombre del usuario actual
- Public Usuario() As String ' matriz que contiene los usuarios
- Public numUsuario As Integer ' cantidad de usuarios creados

Operaciones:

- Public Sub setNumUsuarios() ' Almacena en la variable numUsuario el número de usuarios
- Public Sub cargaEnMemUsuario() ' Guarda en memoria los datos de la tabla usuario de la planificación actual
- Public Sub selectUser() ' Establece las habilitación según el tipo de usuario
- Public Sub setUserHorario() ' Habilitación de componentes según usuario
- Public Sub guardaClaveEnDB() ' Guarda la clave en la base de datos
- Public Sub guardaUsuarioEnDB(idUsr As String, nombre As String, tipo As String) ' Guarda la información de los usuarios en la base de datos
- Public Sub guardaUsuarioEnMem(idUsr As String, nombre As String, tipo As String) ' Almacena la información de los usuario en la matriz Usuario
- Public Function existeUsuario(codUsr As String) As Boolean ' Verifica la existencia de un usuario
- Public Function getNombre(codUsr As String) As String ' Obtiene el nombre de un usuario
- Public Function getTipo(codUsr As String) As String ' Obtiene el tipo de usuario
- Public Sub eliminarUsuario(codUsr As String) ' Procedimiento que elimina un usuario

Ilustración 4-25: Clase cUsuario

Nombre: cGen

Descripción: Constituye la mínima unidad celular, en conjunto crean los cromosomas, cada alelo almacena información de la representación.

Atributos:

Public X As Integer 'profesorMateria().

Public y As Integer 'aulas().

Public z As Integer 'horarioProfesor().

Operaciones:

- Public Function marcarHorario() As Boolean 'marca los horarios que se van asignando como ocupados en un aula como no disponibles.
- Public Function numAula() ' Asigna el número de aulas en numAula.

Ilustración 4-26: Clase cGen

Nombre: cCromosoma

Descripción: Almacena información de cromosomas que luego formaran la población inicial que pasara a evaluarse en el algoritmo genético.

Atributos:

- Private gen() As cGen.
- Public fitness As Double.
- Public C_status As String.
- Public porcentajeCruce As Integer.
- Public porcCruceTotNivel As Integer .

Operaciones:

- Public Function GetGen(ByVal indice As Integer) As cGen ' Obtiene el los valores del gen , x = relación Materia-Profesor, y = aula, z = horario del profesor.
- Public Sub SetGen(ByVal indice As Integer, G As cGen) ' Establece los valores del Gen, x = relación Materia-Profesor, y = aula, z = horario del profesor.
- Public Sub generarGenes() ' Procedimiento que crea los genes de la generación, inicialmente trata de asignar las horas en bloques, cuando los bloques han sido cubiertos por materias del mismo nivel , solita horas disponibles del profesor y trata de ubicarlas en bloques y en la misma aula. Si ya no hay aulas disponibles en los horarios de los profesores , asigna un aula aleatoriamente y registra en la tabla conflictos.
- Private Function pedirAula2(tipoMat As String) ' Función que devuelve identificador de Aula, esta función se invoca cuando ya no hay aula disponible para en las horas disponibles del profesor.
- Public Function validaDispProfesor(codprof As String, horas As Variant) As Boolean ' Función que valida la disponibilidad de un profesor en base a un vector de 4 horas. Si el profesor está disponible es esas horas devuelve verdadero de lo contrario falso.
- Public Function validaDispAula(codAula As String, horas As Variant) As Boolean ' Función que valida la disponibilidad de un aula en base a un vector de 4 horas, si el aula está disponible es esas horas devuelve verdadero de lo contrario falso.
- Public Function controlarCruceMaterias(codMat As String, codprof As String, nivel As Integer, bloque As Integer) As Variant ' Función que controla el cruce de materias y devuelve códigos de resultados y bloques, esta función controla el cruce por bloques de materias del mismo nivel trata dentro de lo posible que materias iguales se asignen en un mismo horario.
- Public Sub generarGen(idGen As Integer, horaProf As Variant, aula As Integer) ' Asigna los valores del gen y llama a la función set para establecerlos.
- Private Function pedirAula(horasAsignadas As Variant, numHoras As Integer, codMat As String, tipoMat As String, capaRequerida As Integer, recursos As String) As Integer ' Solicita un aula de acuerdo a un conjunto de horas y asigna el respectivo laboratorio de acuerdo a la relación con la materia y al tipo de materia.
- Private Function pedirHorarioProfesor(codProfesor As String, numHoras As Integer) As Variant ' Devuelve un vector con las horas que se pueden asignar al profesor esto en base a la disponibilidad ingresada y a las horas que ya han sido asignadas.
- Private Function validarDisponibilidadAula(aula As Integer, Horario As Integer, disponibilidadAulasT() As String) As Boolean ' Se le entrega un aula y horario y retorna verdadero si puede utilizar un aula dependiendo de la disponibilidad del aula.
- Private Sub Class_Initialize() ' Inicialización de la clase.
- Public Sub calcularFitness() ' Procedimienbto verificar que no haya cruce de horarios, si existen condena a la solución sumandole un valor a la variable fitness, la mejor solución será la que contenga el menor valor en la función fitness.
- Public Sub matSolucion(j As Integer) ' Lleva la información del cromosoma a la matriz horarioOptimo

Ilustración 4-27: Clase cCromosoma

4.3.2 MODELOS DINÁMICOS

4.3.2.1 DESCRIPCIÓN DE ACTORES

4.3.2.1.1 ACTORES PRIMARIOS

Nombre: Administrador.

Descripción: Tiene acceso a hacer todo tipo de consultas, ingresos o modificaciones en el sistema, puede cambiar o agregar parámetros y colocar restricciones, no hace las veces de digitador aunque podría hacerlo. Define los parámetros iniciales del sistema tanto organizacionales como del algoritmo genético.

Es la persona responsable de hacer la planificación de horarios, tiene todos los privilegios administrativos.

Notas: Cuando hablamos de administrador puede tratarse del coordinador académico, del sub-coordinador académico o del administrador de la base de datos, este último además de lo ya indicado puede realizar los ingresos en batch de profesores, materias o aulas.

Ilustración 4-28: Actor Primario Administrador

Nombre: Digitador.

Descripción: Sus procesos solo son de digitación y actualización, se convierte en un colaborador del coordinador académico.

Posee la mayoría de privilegios del administrador a excepción de:

No puede crear, modificar o eliminar usuarios.

No puede cambiar la contraseña de otros usuarios, solo la propia.

No puede crear paralelos especiales.

No puede ingresar, asignar o eliminar paralelos.

No puede modificar los parámetros iniciales del sistema.

Notas: Este cargo está reservado para un colaborador del coordinador académico.

Ilustración 4-29: Actor Primario Digitador

4.3.2.1.2 ACTORES SECUNDARIOS

Nombre: Invitado.

Descripción: Ejecuta procesos solo son consulta sobre los reportes previos y posteriores a la optimización.

Notas: Puede tratarse de un estudiante o profesor que no cumple ningún papel administrativo sobre la planificación.

Ilustración 4-30: Actor Secundario Invitado

4.3.2.2 LISTADO DE CASOS DE USO

Caso de Uso 1: Creación de Nueva Planificación

Caso de Uso 2: Apertura de Planificación

Caso de Uso 3: Eliminación de Planificación

Caso de Uso 4: Guardar Como de Planificación

Caso de Uso 5: Cambio de Usuario

Caso de Uso 6: Cambio de Contraseña

Caso de Uso 7: Ingreso de Usuario

Caso de Uso 8: Modificación de Usuario

Caso de Uso 9: Eliminación de Usuario

Caso de Uso 10: Ingreso de Asignaciones Especiales

Caso de Uso 11: Ingreso de Profesor

Caso de Uso 12: Modificación de Profesor

Caso de Uso 13: Eliminación de Profesor

Caso de Uso 14: Ingreso de Materia

Caso de Uso 15: Modificación de Materia

Caso de Uso 16: Eliminación de Materia

Caso de Uso 17: Ingreso de Aula

Caso de Uso 18: Modificación de Aula

Caso de Uso 19: Eliminación de Aula

Caso de Uso 20: Ingreso de Paralelo

Caso de Uso 21: Asignación de Paralelo

Caso de Uso 22: Eliminación de Paralelo

Caso de Uso 23: Ingreso de Horario Profesor

Caso de Uso 24: Ingreso de Horario Aula

Caso de Uso 25: Obtener Lista de Profesores

Caso de Uso 26: Obtener Lista de Materias

Caso de Uso 27: Obtener Lista de Aulas

Caso de Uso 28: Obtener Horarios por Aula

Caso de Uso 29: Obtener Horarios por Materia

Caso de Uso 30: Obtener Horarios por Profesor

Caso de Uso 31: Obtener Reporte de Conflictos

Caso de Uso 32: Ingreso de Parámetros Iniciales

Caso de Uso 33: Optimización

4.3.2.1.3 DESCRIPCIÓN DE LOS CASOS DE USO

Caso de Uso 1: Creación de Nueva Planificación.

Actores que participan:

Administrador, Digitador.

Descripción: Administrador o Digitador crea una nueva planificación.

Notas: La nueva planificación es una planificación vacía que no dispone de datos de materias profesores o aulas.

Ilustración 4-31: Caso de Uso 1: Creación de Nueva Planificación.

Caso de Uso 2: Apertura de Planificación.

Actores que participan:

Administrador, Digitador, Invitado.

Descripción: Actor abre una planificación existente.

Notas: Al abrir una planificación se cargan todos los datos de profesores, materias, aulas, paralelos y asignaciones para la planificación actual.

Ilustración 4-32: Caso de Uso 2: Apertura de Planificación.

Caso de Uso 3: Eliminación de Planificación.

Actores que participan:

Administrador, Digitador.

Descripción: Actor elimina una planificación existente.

Notas: Al eliminar una planificación se desechan todos los datos almacenados de la misma, incluidas sus asignaciones.

Ilustración 4-33: Caso de Uso 3: Eliminación de Planificación.

Caso de Uso 4: Guardar Como de Planificación.

Actores que participan:

Administrador, Digitador.

Descripción: Actor guardar la planificación existente, decide entre guardar toda la planificación o personalizar los datos a guardar, sean estos de profesores, materias, aulas, asignaciones especiales y asignaciones de profesor-materia.

Notas: Se guarda la planificación actual según selección del actor.

Ilustración 4-34: Caso de Uso 4: Guardar Como de Planificación

Caso de Uso 5: Cambio de Usuario.

Actores que participan:

Administrador, Digitador, Invitado.

Descripción: Actor decide ingresar con un nuevo usuario al sistema.

Notas: Con el nuevo ingreso al sistema se definen privilegios de usuario.

Ilustración 4-35: Caso de Uso 5: Cambio de Usuario.

Caso de Uso 6: Cambio de Contraseña.

Actores que participan:

Administrador, Digitador.

Descripción: Usuario cambia contraseña actual.

Notas: En el caso del administrador puede cambiar contraseña de todos los usuarios sin contar con la contraseña anterior, los digitadores solo pueden modificar la contraseña propia.

Ilustración 4-36: Caso de Uso 6: Cambio de Contraseña.

Caso de Uso 7: Ingreso de Usuario.

Actores que participan:

Administrador .

Descripción: Administrador crea nuevo usuario.

Notas: Se proporcionan privilegios de usuario de acuerdo al tipo de usuario escogido.

Ilustración 4-37: Caso de Uso 7: Ingreso de Usuario.

Caso de Uso 8: Modificación de Usuario.

Actores que participan:

Administrador .

Descripción: Administrador modifica datos de usuario seleccionado, ya sea su nombre o tipo.

Notas: Se aplican nuevos privilegios al usuario según tipo elegido.

Ilustración 4-38: Caso de Uso 8: Modificación de Usuario.

Caso de Uso 9: Eliminación de Usuario.

Actores que participan:

Administrador .

Descripción: Administrador elimina a un usuario existente.

Ilustración 4-39: Caso de Uso 9: Eliminación de Usuario.

Caso de Uso 10: Ingreso de Asignaciones Especiales.

Actores que participan:

Administrador .

Descripción: Administrador ingresa asignaciones especiales de paralelos.

Notas: Este proceso es previo a la optimización, provoca una reducción en la disponibilidad de horarios de profesor o aula en los objetos involucrados.

Ilustración 4-40: Caso de Uso 10: Ingreso de Asignaciones Especiales.

Caso de Uso 11: Ingreso de Profesor.

Actores que participan:

Administrador , Digitador.

Descripción: Actor ingresa un nuevo profesor.

Notas: Profesor es ingresado a la base de datos, opcionalmente puede ingresar asignaciones de materias en esta ventana.

Ilustración 4-41: Caso de Uso 11: Ingreso de Profesor.

Caso de Uso 12: Modificación de Profesor.

Actores que participan:

Administrador , Digitador.

Descripción: Actor modifica datos de un profesor existente.

Ilustración 4-42: Caso de Uso 12: Modificación de Profesor Exitoso.

Caso de Uso 13: Eliminación de Profesor.

Actores que participan:

Administrador , Digitador.

Descripción: Actor elimina un profesor existente.

Notas: Profesor es eliminado del sistema junto a sus asignaciones si las tuviere.

Ilustración 4-43: Caso de Uso 13: Eliminación de Profesor.

Caso de Uso 14: Ingreso de Materia.
Actores que participan:
 Administrador , Digitador.
Descripción: Actor ingresa una nueva materia.

Ilustración 4-44: Caso de Uso 14: Ingreso de Materia.

Caso de Uso 15.1: Modificación de Materia.
Actores que participan:
 Administrador , Digitador.
Descripción: Actor accede modifica datos de una materia existente.

Ilustración 4-45: Caso de Uso 15: Modificación de Materia.

Caso de Uso 16: Eliminación de Materia.
Actores que participan:
 Administrador , Digitador.
Descripción: Actor elimina una materia existente.
Notas: Materia es eliminada junto a sus asignaciones si las tuviere.

Ilustración 4-46: Caso de Uso 16: Eliminación de Materia.

Caso de Uso 17: Ingreso de Aula.
Actores que participan:
 Administrador , Digitador.
Descripción: Actor ingresar un aula nueva.
Notas: Aula es ingresada a la base de datos, junto a los recursos de que dispone.

Ilustración 4-47: Caso de Uso 17: Ingreso de Aula.

Caso de Uso 18: Modificación de Aula.
Actores que participan:
 Administrador , Digitador.
Descripción: Actor accede modifica datos de un aula existente.

Ilustración 4-48: Caso de Uso 18: Modificación de Aula.

Caso de Uso 19: Eliminación de Aula.

Actores que participan:

Administrador , Digitador.

Descripción: Actor elimina un aula existente.

Ilustración 4-49: Caso de Uso 19: Eliminación de Aula.

Caso de Uso 20: Ingreso de Paralelo.

Actores que participan:

Administrador .

Descripción: Actor agrega un paralelo a la planificación al final de la optimización.

Notas: Este proceso es posterior a la ejecución y puesta en marcha de la planificación optimizada, como resultado de un requerimiento adicional de paralelos para distintas materias. La planificación obtenida no es modificada.

Ilustración 4-50: Caso de Uso 20: Ingreso de Paralelo.

Caso de Uso 21: Asignación de Paralelo.

Actores que participan:

Administrador .

Descripción: Actor crea una nueva asignación profesor-materia-paralelo.

Notas: Creación de nueva asignación profesor-materia-paralelo, también se puede realizar desde la ventana de ingreso de profesores.

Ilustración 4-51: Caso de Uso 21: Asignación de Paralelo.

Caso de Uso 22: Eliminación de Paralelo.

Actores que participan:

Administrador .

Descripción: Actor elimina un paralelo existente.

Notas: Hablamos de eliminar una asignación profesor-materia-paralelo.

Ilustración 4-52: Caso de Uso 22: Eliminación de Paralelo.

Caso de Uso 23: Ingreso de Horario Profesor.

Actores que participan:

Administrador , Digitador.

Descripción: Actor asigna horario de disponibilidad del profesor.

Notas: Un grafico en una celda de horario indica que el profesor esta disponible en dicha hora.

Ilustración 4-53: Caso de Uso 23: Ingreso de Horario Profesor.

Caso de Uso 24: Ingreso de Horario Aula.

Actores que participan:

Administrador , Digitador.

Descripción: Actor asigna horario de disponibilidad del aula.

Notas: La disponibilidad de aulas se determina con una hora de inicio y de cierre.

Ilustración 4-54: Caso de Uso 24: Ingreso de Horario Aula.

Caso de Uso 25: Obtener Lista de Profesores

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor obtiene un reporte de los profesores ingresados

Ilustración 4-55: Caso de Uso 25: Obtener Lista de Profesores

Caso de Uso 26: Obtener Lista de Materias.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor obtiene un reporte de las materias ingresadas.

Ilustración 4-56: Caso de Uso 26: Obtener Lista de Materias

Caso de Uso 27: Obtener Lista de Aulas.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor obtiene un reporte de las aulas ingresadas.

Ilustración 4-57: Caso de Uso 27: Obtener Lista de Aulas.

Caso de Uso 28: Obtener Horarios por Aula Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor obtiene un reporte de la planificación óptima resultado del proceso de optimización, clasificada por Aula.

Ilustración 4-58: Caso de Uso 28: Obtener Horarios por Aula Exitoso

Caso de Uso 29: Obtener Horarios por Materia Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: obtiene un reporte de la planificación óptima resultado del proceso de optimización, clasificada por Materia.

Ilustración 4-59: Caso de Uso 29: Obtener Horarios por Materia Exitoso.

Caso de Uso 30: Obtener Horarios por Profesor Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor a obtiene un reporte de la planificación óptima resultado del proceso de optimización, clasificada por Profesor.

Ilustración 4-60: Caso de Uso 30: Obtener Horarios por Profesor Exitoso.

Caso de Uso 31: Obtener Reporte de Conflictos.

Actores que participan:

Administrador , Digitador.

Descripción: Actor obtiene reporte un reporte de conflictos producto de la optimización.

Ilustración 4-61: Caso de Uso 31: Obtener Reporte de Conflictos.

Caso de Uso 32: Ingreso de Parámetros Iniciales.

Actores que participan:

Administrador.

Descripción: Actor ingresa, modifica o visualiza los parámetros iniciales del sistema.

Notas: Se puede también restaurar los valores por defecto.

Ilustración 4-62: Caso de Uso 32: Ingreso de Parámetros Iniciales.

Caso de Uso 33: Optimización.

Actores que participan:

Administrador, Digitador.

Descripción: Actor ejecuta el proceso de optimización.

Notas: El proceso de optimización involucra previamente el cumplimiento de ciertos parámetros restrictivos como suficiente horario de disponibilidad de aulas y profesores previo al inicio del algoritmo genético que busca encontrar una polución óptima para el problema.

Ilustración 4-63: Caso de Uso 33: Optimización.

4.3.2.3 DIAGRAMA DE CASOS DE USOS

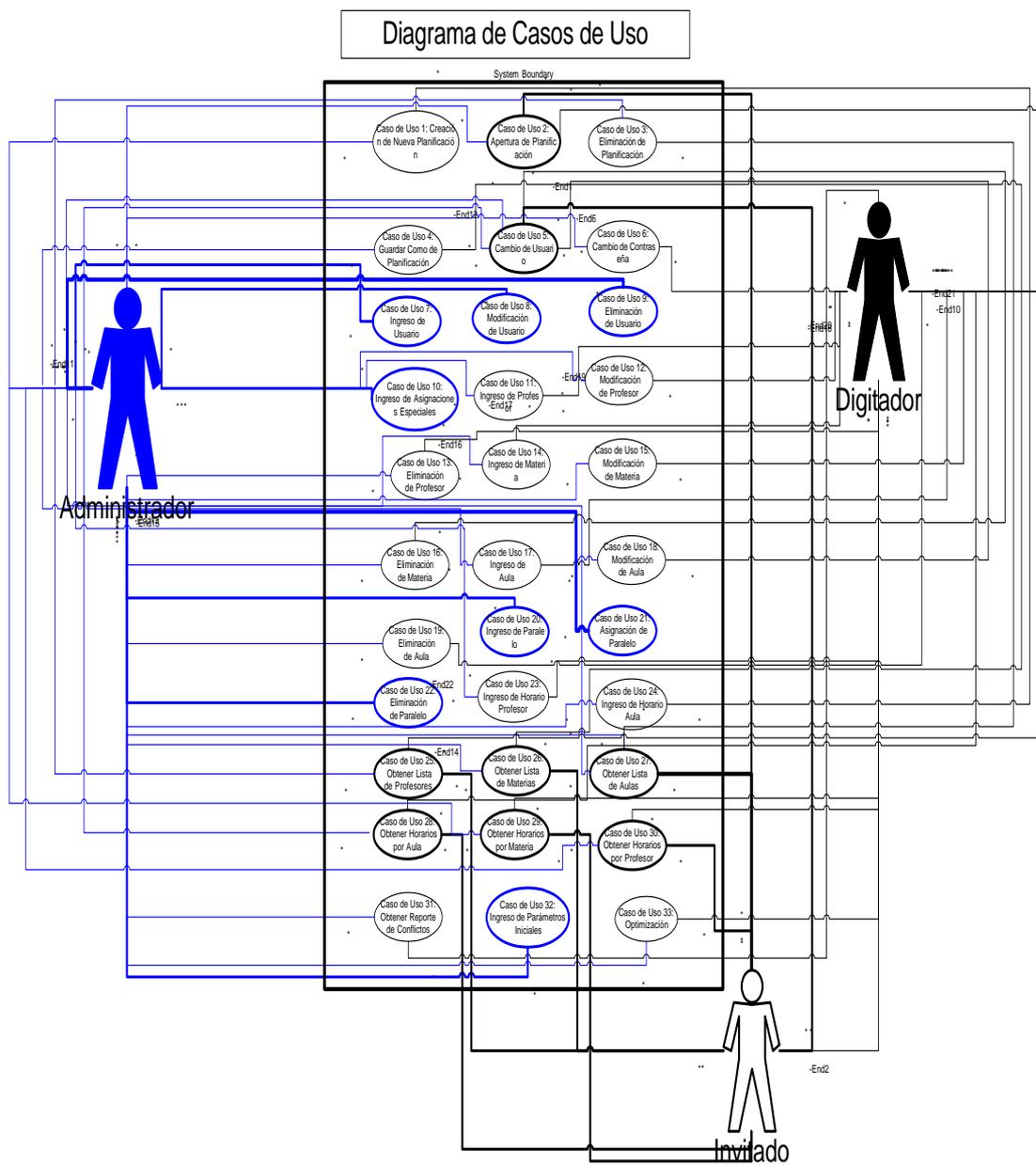


Ilustración 4-64: Diagrama de Casos de Uso

4.3.2.4 ESCENARIOS

Escenario 1.1: Creación de Nueva Planificación Exitosa.

Escenario 2.1: Apertura de Planificación Exitosa.

Escenario 3.1: Eliminación de Planificación Exitosa.

Escenario 4.1: Guardar Como de Planificación Exitosa.

Escenario 5.1: Cambio de Usuario Exitoso.

Escenario 6.1: Cambio de Contraseña Exitosa.

Escenario 7.1: Ingreso de Usuario Exitoso.

Escenario 8.1: Modificación de Usuario Exitoso.

Escenario 9.1: Eliminación de Usuario Exitoso.

Escenario 10.1: Ingreso de Asignaciones Especiales Exitosa.

Escenario 11.1: Ingreso de Profesor Exitoso.

Escenario 11.2: Identificación del profesor invalida.

Escenario 12.1: Modificación de Profesor Exitoso.

Escenario 13.1: Eliminación de Profesor Exitosa.

Escenario 14.1: Ingreso de Materia Exitoso.

Escenario 14.2: Error en el Código de la Materia.

Escenario 15.1: Modificación de Materia Exitoso.

Escenario 15.2: Error en el Nivel de la Materia.

Escenario 16.1: Eliminación de Materia Exitoso.

Escenario 17.1: Ingreso de Aula Exitoso.

Escenario 18.1: Modificación de Aula Exitoso.

Escenario 19.1: Eliminación de Aula Exitoso.

Escenario 20.1: Ingreso de Paralelo Exitoso.

Escenario 21.1: Asignación de Paralelo Exitoso.

Escenario 22.1: Eliminación de Paralelo Exitoso.

Escenario 23.1: Ingreso de Horario Profesor Exitoso.

Escenario 24.1: Ingreso de Horario Aula Exitoso.

Escenario 25.1: Obtener Lista de Profesores Exitosa.

Escenario 26.1: Obtener Lista de Materias Exitosa.

Escenario 27.1: Obtener Lista de Aulas Exitosa.

Escenario 28.1: Obtener Horarios por Aula Exitoso.

Escenario 29.1: Obtener Horarios por Materia Exitoso.

Escenario 30.1: Obtener Horarios por Profesor Exitoso.

Escenario 31.1: Obtener Reporte de Conflictos Exitoso.

Escenario 32.1: Ingreso de Parámetros Iniciales Exitoso.

Escenario 33.1: Optimización Exitosa.

Escenario 33.2: Optimización fallida - Horas Asignadas mayores que horas Disponibles de Aulas.

Escenario 33.3: Optimización fallida - Horas Asignadas mayores que horas Disponibles de laboratorios.

Escenario 33.4: Optimización fallida - Horas disponibles de aulas mayor a las horas de asignación semanal.

Escenario 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad.

Escenario 1.1: Creación de Nueva Planificación Exitosa.

Actores que participan:

Administrador, Digitador.

Descripción: Administrador o Digitador accede al menú Planificación-Nueva para crear una nueva planificación.

Flujo de Eventos:

Selección de la opción Nueva en el menú Planificación.

Ingreso del código de la planificación y descripción de la misma.

Actor presiona botón Aceptar.

Se ejecuta guardaNuevaPlanificacion().

Datos se almacenan en la base.

Salidas: Planificación almacenada con éxito.

Ilustración 4-65: Escenario 1.1: Creación de Nueva Planificación Exitosa.

Escenario 2.1: Apertura de Planificación Exitosa.

Actores que participan:

Administrador, Digitador, Invitado.

Descripción: Actor accede al menú Planificación-Abrir para abrir una planificación existente.

Flujo de Eventos:

Selección de la opción Abrir en el menú Planificación.

Se muestra frmPlanificacion con las planificaciones existentes.

Usuario selecciona la planificación y presiona aceptar.

Se establecen los parámetros iniciales para la planificación seleccionada.

Salidas: Se abre una de las planificaciones almacenadas en la base.

Ilustración 4-66: Escenario 2.1: Apertura de Planificación Exitosa.

Escenario 3.1: Eliminación de Planificación Exitosa.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Planificación-Eliminar para eliminar una planificación existente.

Flujo de Eventos:

Selección de la opción Eliminar en el menú Planificación.

Se muestra frmEliminarPlanificacion con las planificaciones existentes.

Usuario selecciona la planificación a eliminar y presiona eliminar.

Planificación es eliminada de la base de datos.

Salidas: Se elimina una planificación existente.

Ilustración 4-67: Escenario 3.1: Eliminación de Planificación Exitosa.

Escenario 4.1: Guardar Como de Planificación Exitosa.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Planificación-Guardar Como para guardar la planificación existente.

Flujo de Eventos:

Selección de la opción Guardar Como en el menú Planificación.

Se muestra frmGuardarComoPlanificacion.

Usuario ingresa código y descripción de la planificación a almacenar.

Actor escoge entre personalización del almacenaje de los datos o guardar todo

Se presiona botón aceptar.

Se guarda planificación con sus datos de materias, aulas, profesores y asignaciones según selección del actor.

Salidas: Se guarda la planificación actual según selección del actor.

Ilustración 4-68: Escenario 4.1: Guardar Como de Planificación Exitosa.

Escenario 5.1: Cambio de Usuario Exitoso.

Actores que participan:

Administrador, Digitador, Invitado.

Descripción: Actor accede al menú Planificación-Cambiar de Usuario para acceder con un nuevo usuario.

Flujo de Eventos:

Selección de la opción Cambiar de Usuario en el menú Planificación.

Se muestra frmValidacion.

Usuario ingresa usuario y clave para ingresar al sistema.

Presiona aceptar.

Se valida existencia del usuario y el ingreso de su correcta contraseña.

Se da bienvenida al usuario y se asignan permisos de acceso según el tipo del usuario.

Salidas: Se cambia de usuario y se muestra frmMain con restricciones según tipo de usuario.

Ilustración 4-69: Escenario 5.1: Cambio de Usuario Exitoso.

Escenario 6.1: Cambio de Contraseña Exitosa.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Planificación-Cambiar Contraseña para cambiar contraseña actual en el caso de digitador y a cualquier otro usuario en el caso del administrador.

Flujo de Eventos:

Selección de la opción Contraseña en el menú Planificación.

Se muestra frmCambiarContraseña.

Administrador: selecciona usuario, digita nueva contraseña y confirmación de la misma.

Digitador: digita contraseña antigua, nueva y confirmación.

Presiona aceptar para ejecutar el cambio.

Se modifican valores de usuario en la base datos.

Salidas: Se cambia contraseña al usuario.

Ilustración 4-70: Escenario 6.1: Cambio de Contraseña Exitosa.

Escenario 7.1: Ingreso de Usuario Exitoso.

Actores que participan:

Administrador .

Descripción: Actor accede al menú DataBase-Usuario-Ingresa para crear un nuevo usuario.

Flujo de Eventos:

Selección de la opción en Usuario-Ingresa el menú DataBase.

Se muestra frmCrearUsuario.

Se ingresa nombre del usuario, nombre completo, y se define el tipo de usuario, se define una contraseña y se confirma la misma.

Administrados presiona aceptar.

Se verifica existencia del usuario, si no existe se ingresa nueva fila en la base y se actualizan valores en memoria.

Salidas: Se ingresa un nuevo usuario, se aumenta la cantidad de usuarios.

Ilustración 4-71: Escenario 7.1: Ingreso de Usuario Exitoso.

Escenario 8.1: Modificación de Usuario Exitoso.

Actores que participan:

Administrador .

Descripción: Actor accede al menú DataBase-Usuario-Modificar para modificar datos de un usuario existente.

Flujo de Eventos:

Selección de la opción en Usuario-Modificar el menú DataBase.

Se muestra frmModificarUsuario.

Se selecciona el usuario a modificar, se da opción de modificar el nombre o tipo de usuario.

Se presiona botón aceptar.

Se actualizan valores en memoria y en la base.

Salidas: Se ingresa modifican datos del usuario seleccionado.

Ilustración 4-72: Escenario 8.1: Modificación de Usuario Exitoso.

Escenario 9.1: Eliminación de Usuario Exitoso**Actores que participan:**

Administrador .

Descripción: Actor accede al menú DataBase-Usuario-Eliminar para eliminar al usuario seleccionado.

Flujo de Eventos:

Selección de la opción en Usuario-Eliminar el menú DataBase.

Se muestra frmEliminarUsuario.

Se selecciona el usuario a eliminar, se presiona botón eliminar.

Se elimina usuario de la base

Salidas: Usuario seleccionado es eliminado de memoria y de la base.

Ilustración 4-73: Escenario 9.1: Eliminación de Usuario Exitoso.**Escenario 10.1:** Ingreso de Asignaciones Especiales Exitosa.**Actores que participan:**

Administrador .

Descripción: Actor accede al menú DataBase-Asignaciones Especiales para ingresar asignaciones especiales de paralelos antes de la optimización.

Flujo de Eventos:

Selección de la opción Asignaciones Especiales el menú DataBase.

Se muestra frmAsignacionesEspeciales.

Se cargan todas las asignaciones previas.

Se presiona botón nueva, para crear nueva asignación.

Se muestra frmExcepciones.

Selecciona profesor, materia, paralelo, aula, día, hora de inicio, hora de fin

Se presiona botón ingresar.

Se verifica si existe la asignación existeAsignacionManualDb(), existeAsignacionPMDB(codMateria, paralelo) As Boolean. Se ingresa combinación profesor materia a la base.

Presiona aceptar.

Se guarda asignación manual guardaDBAsignacionManual().

Salidas: Datos de la asignación manual de paralelos se ingresa en la base de datos previo a la optimización.

Ilustración 4-74: Escenario 10.1: Ingreso de Asignaciones Especiales Exitosa.

Escenario 11.1: Ingreso de Profesor Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Profesor-Ingresar para ingresar un nuevo profesor.

Flujo de Eventos:

Selección de la opción Profesor-Ingresar en el menú DataBase.

Se muestra frmIngresarBase Tab Profesor.

Se ingresa numero de cedula, nombre y tipo de profesor.

Se presiona aceptar, se verifica existencia del profesor.

Se guarda el profesor y su disponibilidad.

Se verifica existencia de asignaciones manuales.

Se actualizan datos de profesores, materias y aulas.

Salidas: Profesor es ingresado a la base de datos.

Ilustración 4-75: Escenario 11.1: Ingreso de Profesor Exitoso.

Escenario 11.2: Identificación del profesor invalida.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Profesor-Ingresar para ingresar un nuevo profesor.

Flujo de Eventos:

Selección de la opción Profesor-Ingresar en el menú DataBase.

Se muestra frmIngresarBase Tab Profesor.

Se ingresa numero de cedula, nombre y tipo de profesor.

La identificación del profesor es invalida MsgBox "El numero de cedula debe ser de 10 dígitos numéricos".

Salidas: Mensaje de alerta por error en el numero de cedula.

Ilustración 4-76: Escenario 11.2: Identificación del profesor invalida

Escenario 12.1: Modificación de Profesor Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Profesor-Modificar para modificar datos de un profesor existente.

Flujo de Eventos:

Selección de la opción Profesor-Modificar en el menú DataBase.

Se muestra frmModificarBase Tab Profesor.

Usuario selecciona un profesor de frmBuscaPProfesorMateria y se presiona aceptar.

Información del profesor es refrescada en la pantalla.

Al terminar modificaciones se presiona aceptar.

Se verifica que no exista el profesor, datos son actualizados en memoria y en la base.

Se guardan nuevas asignaciones de materia y paralelo.

Se actualizan datos de profesores, materias y aulas en memoria y en la base.

Salidas: Profesor seleccionado es modificado y sus datos son actualizados.

Ilustración 4-77: Escenario 12.1: Modificación de Profesor Exitoso

Escenario 13.1: Eliminación de Profesor Exitosa.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Profesor-Eliminar para eliminar un profesor existente.

Flujo de Eventos:

Selección de la opción Profesor-Eliminar en el menú DataBase.

Se muestra frmEliminarProfesor.

Usuario selecciona un profesor de frmEliminarProfesor y se presiona aceptar.

Verifica existencia de asignaciones existeAsigdePProfPM y existeAsigdePProfManual

Se elimina el profesor de la base y de memoria.

Salidas: Profesor es eliminado exitosamente.

Ilustración 4-78: Escenario 13.1: Eliminación de Profesor Exitosa.

Escenario 14.1: Ingreso de Materia Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Materia-Ingresar para ingresar una nueva materia.

Flujo de Eventos:

Selección de la opción Materia-Ingresar en el menú DataBase.

Se muestra frmIngresarBase Tab Materia.

Se ingresa código , nombre, nivel, paralelos, carga horaria diaria y semanal, el tipo de aula en que se dictara la materia o laboratorio si es el caso, se define además si es materia fundamental o de especialización, de ser el caso reescoge la especialización a la que pertenece la misma.

Se presiona aceptar, se verifica existencia de la materia.

Se guarda materia en memoria y en la base.

Se verifica existencia de asignaciones manuales.

Salidas: Materia es ingresada a la base de datos.

Ilustración 4-79: Escenario 14.1: Ingreso de Materia Exitoso

Escenario 14.2: Error en el Código de la Materia.

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Materia-Ingresar para ingresar una nueva materia.

Flujo de Eventos:

Selección de la opción Materia-Ingresar en el menú DataBase.

Se muestra frmIngresarBase Tab Materia.

Se ingresa código , nombre, nivel, paralelos, carga horaria diaria y semanal, el tipo de aula en que se dictara la materia o laboratorio si es el caso, se define además si es materia fundamental o de especialización, de ser el caso reescoge la especialización a la que pertenece la misma.

Se presiona aceptar.

MsgBox ("El código de materia es una combinación de la facultad y un numero").

Salidas: Materia no es ingresada por error en el código.

Ilustración 4-80: Escenario 14.2: Error en el Código de la Materia

Escenario 15.1: Modificación de Materia Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Materia-Modificar para modificar datos de una materia existente.

Flujo de Eventos:

Selección de la opción Materia-Modificar en el menú DataBase.

Se muestra frmModificarBase Tab Materia.

Usuario selecciona una materia de frmBuscaMateria y se presiona aceptar.

Información de la materia es refrescada en la pantalla.

Al terminar modificaciones se presiona aceptar.

Se verifica que no exista la materia, datos son actualizados en memoria y en la base.

Salidas: Materia seleccionada es modificada y sus datos son actualizados.

Ilustración 4-81: Escenario 15.1: Modificación de Materia Exitoso

Escenario 15.2: Error en el Nivel de la Materia.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Materia-Modificar para modificar datos de una materia existente.

Flujo de Eventos:

Selección de la opción Materia-Modificar en el menú DataBase.

Se muestra frmModificarBase Tab Materia.

Usuario selecciona una materia de frmBuscaMateria y se presiona aceptar.

Información de la materia es refrescada en la pantalla.

Al terminar modificaciones se presiona aceptar.

Se verifica que no exista la materia.

MsgBox ("El nivel es debe ser un valor numérico entre 100 y 500 con pasos de 50").

Salidas: Materia seleccionada no es modificada por valor erróneo en el nivel.

Ilustración 4-82: Escenario 15.2: Error en el Nivel de la Materia

Escenario 16.1: Eliminación de Materia Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Materia-Eliminar para eliminar una materia existente.

Flujo de Eventos:

Selección de la opción Materia-Eliminar en el menú DataBase.

Se muestra frmEliminarMateria.

Usuario selecciona una materia de frmEliminarMateria y se presiona aceptar.

La materia es eliminada de la base de datos y memoria, se verifica existencia de asignaciones.

Salidas: Materia es eliminada exitosamente.

Ilustración 4-83: Escenario 16.1: Eliminación de Materia Exitoso

Escenario 17.1: Ingreso de Aula Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Aula-Ingresar para ingresar un aula nueva.

Flujo de Eventos:

Selección de la opción Aula-Ingresar en el menú DataBase.

Se muestra frmIngresarBase Tab Aula.

Se ingresa código, capacidad, localización, tipo de aula o laboratorio y su nombre si es el caso, recursos disponibles (proyector de diapositivas, aire acondicionado o proyector).

Se presiona aceptar, se verifica existencia del aula.

Se guarda aula en memoria y en la base.

Salidas: Aula es ingresada a la base de datos.

Ilustración 4-84: Escenario 17.1: Ingreso de Aula Exitoso.

Escenario 18.1: Modificación de Aula Exitoso

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Aula-Modificar para modificar datos de un aula existente.

Flujo de Eventos:

Selección de la opción Aula-Modificar en el menú DataBase.

Se muestra frmModificarBase Tab Aula.

Usuario selecciona un aula de frmBuscaAula y se presiona aceptar.

Información del aula es refrescada en la pantalla.

Al terminar modificaciones se presiona aceptar.

Se verifica que no exista el aula, datos son actualizados en memoria y en la base.

Salidas: Aula seleccionada es modificada y sus datos son actualizados.

Ilustración 4-85: Escenario 18.1: Modificación de Aula Exitoso.

Escenario 19.1: Eliminación de Aula Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Aula-Eliminar para eliminar un aula existente.

Flujo de Eventos:

Selección de la opción Aula-Eliminar en el menú DataBase.

Se muestra frmEliminarAula.

Usuario selecciona un aula de frmEliminarAula y presiona aceptar.

El aula es eliminada de la base de datos y memoria.

Salidas: Aula es eliminada exitosamente.

Ilustración 4-86: Escenario 19.1: Eliminación de Aula Exitoso.

Escenario 20.1: Ingreso de Paralelo Exitoso.

Actores que participan:

Administrador .

Descripción: Actor accede al menú DataBase-Paralelo-Ingresar para agregar un paralelo a la planificación al final de la optimización.

Flujo de Eventos:

Selección de la opción Paralelo-Ingresar en el menú DataBase.

Se muestra frmAgregarParalelo.

Se llenan combos de materia, profesor y aula.

Se selecciona profesor, materia a dictar.

Se crea nuevo paralelo para la materia seleccionada.

Selección del aula, día, hora de inicio y hora de fin y se presiona ingresar.

Verifica existencia de horario existeHorario(), existeAsignacionManualDb2(), existeAsignacionPMDB().

Usuario presiona aceptar.

Se guarda nuevo paralelo.

Salidas: Combinación, profesor, materia, aula es almacenada en la base.

Ilustración 4-87: Escenario 20.1: Ingreso de Paralelo Exitoso.

Escenario 21.1: Asignación de Paralelo Exitoso.

Actores que participan:

Administrador .

Descripción: Actor accede al menú DataBase-Paralelo-Asignar para crear una nueva asignación profesor-materia-paralelo.

Flujo de Eventos:

Selección de la opción Paralelo-Ingresar en el menú DataBase.

Se muestra frmAsignarParaleloMateria.

Se llenan combos de materia, profesor.

Se muestran paralelos ya signados, muestraParAsignados(codMat As String)

Se selecciona profesor y materia a dictar.

Se llena combo con paralelos pertenecientes a la materia escogida.

Se presiona boton ingresar.

Validaciones : existeMatParSel(matSel, parSel) As Boolean, existeAsignacionPMDB(codmateria As String, paralelo As Integer) As Boolean, existeAsignacionManualDb(codmateria As String, paralelo As Integer) As Boolean.

Usuario presiona aceptar.

Se guardan paralelos, se actualizan tablas de parámetros, planificación, aula, horario y profesor materia.

Salidas: creación de nueva asignación profesor-materia-paralelo.

Ilustración 4-88: Escenario 21.1: Asignación de Paralelo Exitoso.

Escenario 22.1: Eliminación de Paralelo Exitoso.

Actores que participan:

Administrador .

Descripción: Actor accede al menú DataBase-Paralelo-Eliminar, para eliminar paralelo existente.

Flujo de Eventos:

Selección de la opción Paralelo-Eliminar en el menú DataBase.

Se muestra frmEliminarparalelo.

Se llena combo de materia.

Se llena combo con paralelos pertenecientes a la materia escogida.

Se escoge paralelo a eliminar.

Se presiona botón eliminar.

Paralelo es eliminado de la base eliminarParalelo(codMat As String, paralelo As Integer).

Salidas: Paralelo es eliminado de la base y memoria.

Ilustración 4-89: Escenario 22.1: Eliminación de Paralelo Exitoso.

Escenario 23.1: Ingreso de Horario Profesor Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Horario Profesor, para asignar horario de disponibilidad del profesor.

Flujo de Eventos:

Selección de la opción Horario Profesor en el menú DataBase.

Se muestra frmIngresarBaseHorario Tab Horario por Profesor.

Se presiona botón buscar para seleccionar un profesor.

Muestra frmBuscaProfesor.

Selecciona profesor y presiona aceptar.

Carga disponibilidad del profesor si existe callDisponibilidad().

Modifica disponibilidad del profesor.

Presiona aceptar para almacenarla nueva disponibilidad guardaMatrizProfesor(indiceProfesor As String, dp() As String).

Salidas: Disponibilidad del profesor es almacenada.

Ilustración 4-90: Escenario 23.1: Ingreso de Horario Profesor Exitoso.

Escenario 24.1: Ingreso de Horario Aula Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú DataBase-Horario Aula, para asignar horario de disponibilidad del aula.

Flujo de Eventos:

Selección de la opción Horario Aula en el menú DataBase.

Se muestra frmIngresarBaseHorario Tab Horario por Aula.

Se presiona botón buscar para seleccionar un aula.

Muestra frmBuscaAula.

Selecciona aula y presiona aceptar.

Carga disponibilidad del aula si existe.

Usuario modifica disponibilidad del aula.

Presiona aceptar para guardar nueva disponibilidad guardaMatrizAulas(indiceAula As String, da() As String).

Salidas: Disponibilidad del aula es almacenada.

Ilustración 4-91: Escenario 24.1: Ingreso de Horario Aula Exitoso.

Escenario 25.1: Obtener Lista de Profesores Exitosa.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Lista de Profesores, para obtener un reporte de los profesores ingresados.

Flujo de Eventos:

Selección de la opción Lista de Profesores en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Profesores existentes para la planificación actual.

Ilustración 4-92: Escenario 25.1: Obtener Lista de Profesores Exitosa.

Escenario 26.1: Obtener Lista de Materias Exitosa.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Lista de Materias, para obtener un reporte de las materias ingresadas.

Flujo de Eventos:

Selección de la opción Lista de Materias en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Materias existentes para la planificación actual.

Ilustración 4-93: Escenario 26.1: Obtener Lista de Materias Exitosa.

Escenario 27.1: Obtener Lista de Aulas Exitosa.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Lista de Aulas, para obtener un reporte de las aulas ingresadas.

Flujo de Eventos:

Selección de la opción Lista de Aulas en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Aulas existentes para la planificación actual.

Ilustración 4-94: Escenario 27.1: Obtener Lista de Aulas Exitosa.

Escenario 28.1: Obtener Horarios por Aula Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Horario por Aula, para obtener un reporte de la planificación optima obtenida en el proceso de optimización, clasificada por Aula.

Flujo de Eventos:

Selección de la opción Horario por Aula en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Horarios por Aula para la planificación actual.

Ilustración 4-95: Escenario 28.1: Obtener Horarios por Aula Exitoso

Escenario 29.1: Obtener Horarios por Materia Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Horario por Materia, para obtener un reporte de la planificación optima obtenida en el proceso de optimización, clasificada por Materia.

Flujo de Eventos:

Selección de la opción Horario por Materia en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Horarios por Materia para la planificación actual.

Ilustración 4-96: Escenario 29.1: Obtener Horarios por Materia Exitoso.

Escenario 30.1: Obtener Horarios por Profesor Exitoso.

Actores que participan:

Administrador , Digitador, Invitado.

Descripción: Actor accede al menú Reportes-Horario por Profesor, para obtener un reporte de la planificación optima obtenida en el proceso de optimización, clasificada por Profesor.

Flujo de Eventos:

Selección de la opción Horario por Profesor en el menú Reportes.

Se muestra reporte.

Salidas: Reporte de Horarios por Profesor para la planificación actual.

Ilustración 4-97: Escenario 30.1: Obtener Horarios por Profesor Exitoso.

Escenario 31.1: Obtener Reporte de Conflictos Exitoso.

Actores que participan:

Administrador , Digitador.

Descripción: Actor accede al menú Reportes-Reporte de Conflicto para obtener reporte de conflictos producto de la optimización.

Flujo de Eventos:

Selección de la opción Reporte de Conflictos en el menú Reportes

Se consultan datos de la tabla conflictos

Se muestra reporte

Salidas: Reporte Conflictos para la planificación actual.

Ilustración 4-98: Escenario 31.1: Obtener Reporte de Conflictos Exitoso.

Escenario 32.1: Ingreso de Parámetros Iniciales Exitoso.

Actores que participan:

Administrador.

Descripción: Actor accede al menú Herramientas-Parámetros Iniciales para ingreso, modificación o visualización de parámetros iniciales del sistema.

Flujo de Eventos: Selección de la opción Parámetros Iniciales en el menú Herramientas.

Se despliega frmParametrosIniciales.

Usuario ingresa número máximo de generaciones, cromosomas por generación, porcentaje de mutación, porcentaje de cruzamiento, días por semana en que se asignaran horarios y número máximo de horas diarias que se dictan clases.

Usuario presiona botón aceptar.

Se guardan datos en la base guardaParametrosEnDB.

Se carga en memoria nuevos parámetros cargaEnMemParametros.

Salidas: Ingreso de parámetros iniciales al sistema.

Ilustración 4-99: Escenario 32.1: Ingreso de Parámetros Iniciales Exitoso.

Escenario 33.1: Optimización Exitosa.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Herramientas-Optimizar para ejecutar optimización.

Flujo de Eventos: Selección de la opción Optimizar en el menú Herramientas.

Se despliega frmSolucion.

Usuario presiona botón iniciar.

Verificaciones: verificacionProfesor, disponibilidadPorProfesor, getCargasPorProfesor, verificacionAulas, horasClaseSemanal().

Inicio del algoritmo genético algoritmoGenetico().

Se obtiene generacioninicial generacionInicial(NumCromosomas As Integer) As cCromosoma().

Se generan genes: generarGenes().

Se realiza control de cruce de materias: controlarCruceMaterias() .

Se valida disponibilidad de profesor: validaDispProfesor(codprof As String, horas As Variant).

Se genera un gen: generarGen(idGen As Integer, horaProf As Variant, aula As Integer).

Se actualize disponibilidad del profesor: actualizaDispOptimaProfT(codprof As String).

Calculo del fitness: calcularFitness().

Cruzamiento(ByVal porcCruzamiento As Integer, ByVal genInicial As Variant) As cCromosoma().

Mutacion(porcentajeMutacion As Integer, ByVal cromosCruce As Variant) As cCromosoma().

sumarCromosomas(ByVal cromosomas1 As Variant, ByVal cromosomas2 As Variant, _ numCro1 As Integer, numCro2 As Integer) As cCromosoma().

eliminarCromosoma(NumCromosomas, generacionTemp, numTotCromos).

Obtención de solución optima: SolOptima(cromosoma As cCromosoma).

Se guarda solución optima en la base: guardaEnDBOptimo()

Salidas: Obtención de la planificación optima.

Ilustración 4-100: Escenario 33.1: Optimización Exitosa.

Escenario 33.2: Optimización fallida - Horas Asignadas mayores que horas Disponibles de Aulas.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Herramientas-Optimizar para ejecutar optimización.

Flujo de Eventos: Selección de la opción Optimizar en el menú Herramientas.

Se despliega frmSolucion.

Usuario presiona botón iniciar.

Verificaciones: verificacionProfesor, disponibilidadPorProfesor.

Msgbox("El número total de horas Asignadas para las Aulas es mayor al número de horas Disponibles de las Aulas").

Salidas: Mensaje de alerta indicando que las horas asignadas son mayores que las horas disponibles para las aulas.

Ilustración 4-101: Escenario 33.2: Optimización fallida - Horas Asignadas mayores que horas Disponibles de Aulas.

Escenario 33.3: Optimización fallida - Horas Asignadas mayores que horas Disponibles de laboratorios.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Herramientas-Optimizar para ejecutar optimización.

Flujo de Eventos: Selección de la opción Optimizar en el menú Herramientas.

Se despliega frmSolucion.

Usuario presiona botón iniciar.

Verificaciones: verificacionProfesor, disponibilidadPorProfesor, getCargasPorProfesor, verificacionAulas.

Msgbox("El número total de horas Asignadas para los laboratorios es mayor al número de horas Disponibles de los Laboratorios ").

Salidas: Mensaje de alerta indicando que las horas asignadas son mayores que las horas disponibles para los laboratorios.

Ilustración 4-102: Escenario 33.3: Optimización fallida - Horas Asignadas mayores que horas Disponibles de laboratorios.

Escenario 33.4: Optimización fallida - Horas disponibles de aulas mayor a las horas de asignación semanal.

Actores que participan:

Administrador, Digitador.

Descripción: Actor accede al menú Herramientas-Optimizar para ejecutar optimización.

Flujo de Eventos: Selección de la opción Optimizar en el menú Herramientas.

Se despliega frmSolucion.

Usuario presiona botón iniciar.

Verificaciones: verificacionProfesor, disponibilidadPorProfesor, getCargasPorProfesor, verificacionAulas, horasClaseSemanal().

Msgbox("El número total de horas Asignadas a Aulas es mayor al número de horas Disponibles de las Aulas ").

Salidas: Mensaje de alerta indicando que las horas disponibles de aulas mayor a las horas de asignación semanal.

Ilustración 4-103: Escenario 33.4: Optimización fallida - Horas disponibles de aulas mayor a las horas de asignación semanal.

Escenario 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad.

Actores que participan:
 Administrador, Digitador.

Descripción: Actor accede al menú Herramientas-Optimizar para ejecutar optimización.

Flujo de Eventos: Selección de la opción Optimizar en el menú Herramientas.
 Se despliega frmSolucion.
 Usuario presiona botón iniciar.
 Verificaciones: verificacionProfesor, disponibilidadPorProfesor.
 MsgBox ("El número total de horas Asignadas a Profesores es mayor al número de horas Disponibles de los Profesores").

Salidas: Mensaje de alerta indicando que las horas asignadas a profesores es mayor que sus disponibilidad.

Ilustración 4-104: Escenario 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad

4.3.2.5 DIAGRAMAS DE INTERACCIÓN DE OBJETOS

DIO 1.1: Creación de Nueva Planificación Exitosa.

DIO 2.1: Apertura de Planificación Exitosa.

DIO 3.1: Eliminación de Planificación Exitosa.

DIO 4.1: Guardar Como de Planificación Exitosa.

DIO 5.1: Cambio de Usuario Exitoso.

DIO 6.1: Cambio de Contraseña Exitosa.

DIO 7.1: Ingreso de Usuario Exitoso.

DIO 8.1: Modificación de Usuario Exitoso.

DIO 9.1: Eliminación de Usuario Exitoso.

DIO 10.1: Ingreso de Asignaciones Especiales Exitosa.

DIO 11.1: Ingreso de Profesor Exitoso.

DIO 11.2: Identificación del profesor invalida.

DIO 12.1: Modificación de Profesor Exitoso.

DIO 13.1: Eliminación de Profesor Exitosa.

DIO 14.1: Ingreso de Materia Exitoso.

DIO 14.2: Error en el Código de la Materia.

DIO 15.1: Modificación de Materia Exitoso.

DIO 15.2: Error en el Nivel de la Materia.

DIO 16.1: Eliminación de Materia Exitoso.

DIO 17.1: Ingreso de Aula Exitoso.

DIO 18.1: Modificación de Aula Exitoso.

DIO 19.1: Eliminación de Aula Exitoso.

DIO 20.1: Ingreso de Paralelo Exitoso.

DIO 21.1: Asignación de Paralelo Exitoso.

DIO 22.1: Eliminación de Paralelo Exitoso.

DIO 23.1: Ingreso de Horario Profesor Exitoso.

DIO 24.1: Ingreso de Horario Aula Exitoso.

DIO 25.1: Obtener Lista de Profesores Exitosa.

DIO 26.1: Obtener Lista de Materias Exitosa.

DIO 27.1: Obtener Lista de Aulas Exitosa.

DIO 28.1: Obtener Horarios por Aula Exitoso.

DIO 29.1: Obtener Horarios por Materia Exitoso.

DIO 30.1: Obtener Horarios por Profesor Exitoso.

DIO 31.1: Obtener Reporte de Conflictos Exitoso.

DIO 32.1: Ingreso de Parámetros Iniciales Exitoso.

DIO 33.1: Optimización Exitosa.

DIO 33.2: Optimización fallida - Horas Asignadas mayores que horas Disponibles de Aulas.

DIO 33.3: Optimización fallida - Horas Asignadas mayores que horas Disponibles de laboratorios.

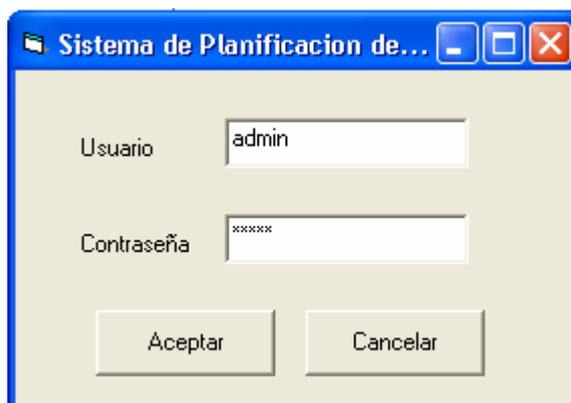
DIO 33.4: Optimización fallida - Horas disponibles de aulas mayor a las horas de asignación semanal.

DIO 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad.

4.3.2.6 FLUJO DE VENTANA Y LAYOUT

Proceso de creación de una Planificación de Horarios

1. Escriba su nombre de usuario y contraseña en los cuadros correspondientes:



The image shows a standard Windows-style dialog box with a blue title bar. The title bar text is partially visible as 'Sistema de Planificación de...'. The dialog has a light beige background. It contains two text input fields. The first field is labeled 'Usuario' and contains the text 'admin'. The second field is labeled 'Contraseña' and contains six asterisks '*****'. Below these fields are two buttons: 'Aceptar' on the left and 'Cancelar' on the right. The dialog box has standard minimize, maximize, and close buttons in the top right corner.

Ilustración 4-105: Pantalla de Validación de Usuarios

Elija la Planificación actual con la que va a trabajar o cree una nueva

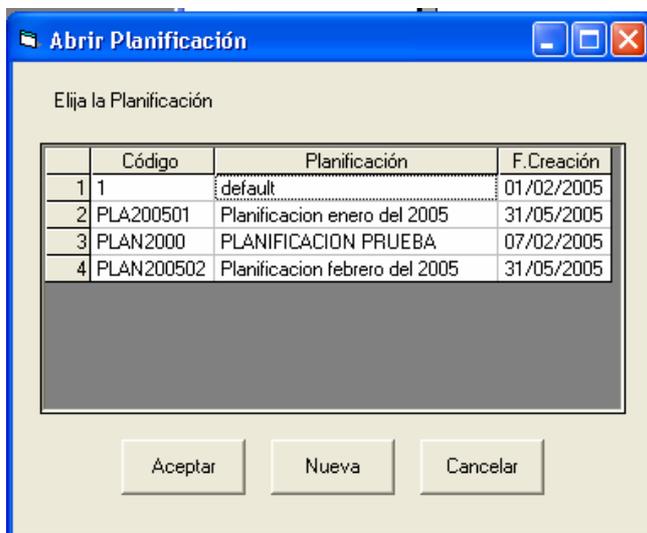


Ilustración 4-106: Selección o Apertura de Nuevas Planificaciones

Si elige crear una nueva planificación aparecerá la siguiente pantalla para ingresar código y descripción de la planificación



Ilustración 4-107: Ingreso de los Datos para la Nueva Planificación

- Configure los Parámetros del Sistema desde el menú Herramientas:

Parámetros Iniciales del Sistema

Parametros Recomendados

Condiciones de Parada

Numero Maximo de Generaciones: 1

Cromosomas por Generacion: 10

Condiciones para Modas Generacionales

Porcentaje de Mutacion: 50 %

Porcentaje de Cruzamiento: 50 %

Parámetros Organizacionales

Dias por semana: 6

Horas clase diarias: 14

Aceptar Salir

Ilustración 4-108: Parámetros Organizacionales y del Algoritmo Genético

- Realice los ingresos en la base de datos

Paralelos Especiales

Paralelos Especiales Ingresados

	Materia	Paralelo	Profesor
1	ALTA TENSIÓN	1	ALVARADO M. OTTO ING.

Nuevo Eliminar Cancelar

Ilustración 4-109: Vista de Paralelos Especiales

Ingreso de paralelo especiales

Ingreso de paralelos especiales

Profesor: ALVARADO M. OTTO ING.

Materia:

Paralelo: Aula:

Día:

Hora Inicio: Hora Fin:

Ingresar

Horario Asignado

	Materia	Paralelo	Aula	Día	Hora Ini

Borrar Horario Eliminar Ultimo

Aceptar Salir

Ilustración 4-110: Ingreso de Paralelos Especiales

Ingresos a la Base de Datos

Profesor Materia Aula

Datos Personales

Cedula de Identidad:

Nombre:

Tipo:

Asignacion de Materias

Materia:

Paralelo: Ingresar

Materias que Dicta

	Materia	Paralelo

Borrar Ingresos Eliminar Ultimo

Aceptar Salir

Ilustración 4-111: Ingresar Profesor

Ingresos a la Base de Datos

Profesor | **Materia** | Aula

Codigo:

Nombre:

Nivel: Paralelos:

Carga Horaria

Diaria: Semanal:

Tipo de Aula

Aula Laboratorio

Tipo de Materia

Fundamental Especializacion

Aceptar Salir

Ilustración 4-112: Ingresar Materia

Ingresos a la Base de Datos

Profesor | Materia | **Aula**

Codigo: Capacidad:

Localizacion:

Tipo de Aula

Aula Laboratorio

Recursos

Aire Acondicionado Proyector de Slides

Infocus

Aceptar Salir

Ilustración 4-113: Ingresar Aula

Ingresos a la Base de Datos

Horario por Profesor | Horario por Aula

Datos Personales

Identificación: P00007

Nombre: AVILES C. JUAN C.ING.

Horarios Disponibles

	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
7:30-8:30	
8:30-9:30	
9:30-10:30	
10:30-11:30						
11:30-12:30						
12:30-13:30	Horario de disponibilidad, si se muestra el grafico el profesor esta d					
13:30-14:30	

Ilustración 4-114: Ingresar Horario Profesor

Ingresos a la Base de Datos

Horario por Profesor | **Horario por Aula**

Datos del Aula

Codigo: A112

Ubicacion: Bloque A FIEC

Horario Disponible

Hora de Apertura: 07:30 Hora de Cierre: 21:30

Ilustración 4-115: Ingresar Horario Aula

Crear Usuario...

Nombre de usuario: jaragundi

Nombre Completo: Jorge Aragundi

Tipo: Administrador

Contraseña: *****

Confirmar contraseña: *****

Aceptar Cancelar

Ilustración 4-116: Ingresar Usuario

4. Se pueden realizar modificaciones a los datos desde el menú DataBase seleccionando los datos a modificar ya sean profesores, materias o aulas:

Modificaciones a la Base de Datos

Profesor | Materia | Aula

Datos Personales:

Cedula de Identidad: P00003

Nombre: ALVARADO M. OTTO ING.

Tipo: Sin Nombramiento-Antiguo

Asignacion de Materias:

Materia:

Paralelo:

Materias que Dicta:

	Materia	Paralelo	R
1	LABORATORIO DE CONTROLES	5	
2	LABORATORIO DE CONTROLES	6	
3	LABORATORIO DE CONTROLES	7	
4	LABORATORIO DE CONTROLES	8	
5	ANÁLISIS DE REDES I	4	

Ilustración 4-117: Modificar Profesor

Modificaciones a la Base de Datos

Profesor | **Materia** | Aula

Codigo : FIEC02048

Nombre : ANÁLISIS Y DISEÑO ORIENTADO A
Nombre de la materia

Nivel : 350 Paralelos: 1

Carga Horaria

Diaria : 2 Semanal : 4

Tipo de Aula

Aula Laboratorio

Tipo de Materia

Fundamental Especialización COMPUTACION

Ilustración 4-118: Modificar Materia

Modificaciones a la Base de Datos

Profesor | Materia | **Aula**

Codigo LSDI

Capacidad: 30

Localizacion Lab. Sist. Digitales

Tipo de Aula

Aula Laboratorio LAB. LSDI

Recursos

Proyector de Slides Infocus

Aire Acondicionado

Ilustración 4-119: Modificar Aula

Crear Usuario...

Nombre de usuario: csalazar

Nombre Completo: Christian Salazar

Tipo: Administrador

Aceptar Cancelar

Ilustración 4-120: Modificar Usuario

5. Eliminaciones

Eliminar Profesor

Elija el profesor a eliminar

	Código	Profesor
1	P00001	ABAD CRISTINA AS MSC
2	P00002	ALEJANDRO MOLINA OTILIA ING.
3	P00003	ALVARADO M. OTTO ING.
4	P00004	ANDRADE DAVID ING.
5	P00005	ARAGUNDI RODRIGUEZ JORGE ING.
6	P00006	ASTUDILLO B. KARINA ING.
7	P00007	AVILES C. JUAN C.ING.
8	P00008	BERMUDEZ FLORES GUSTAVO ING.

Eliminar Cancelar

Ilustración 4-121: Eliminar Profesor

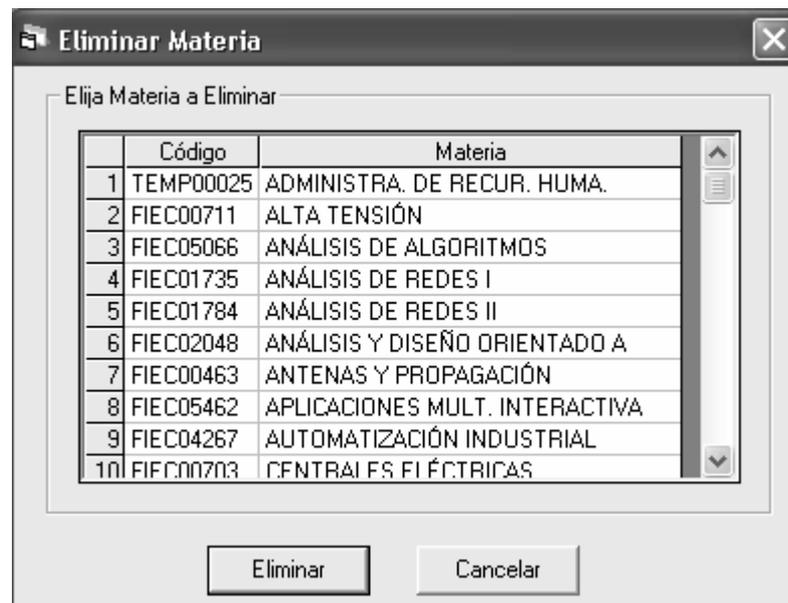


Ilustración 4-122: Eliminar Materia

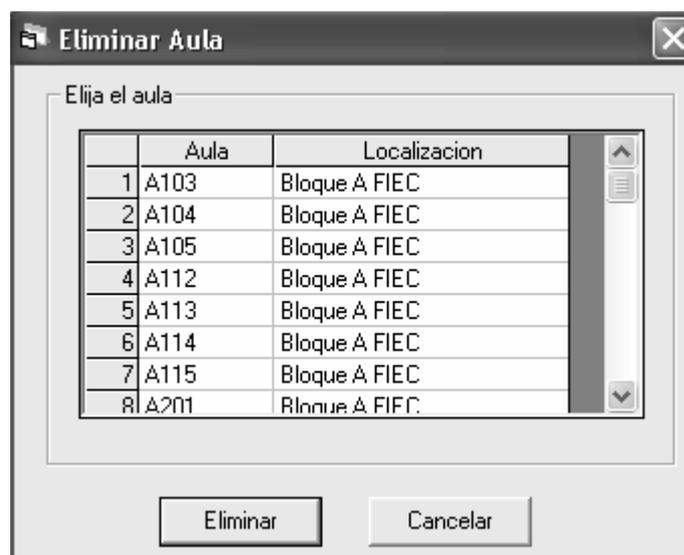


Ilustración 4-123: Eliminar Aula



Ilustración 4-124: Eliminar Usuario



Ilustración 4-125: Eliminación de un Paralelo.

6. Creación de asignaciones materia-profesor-paralelo.

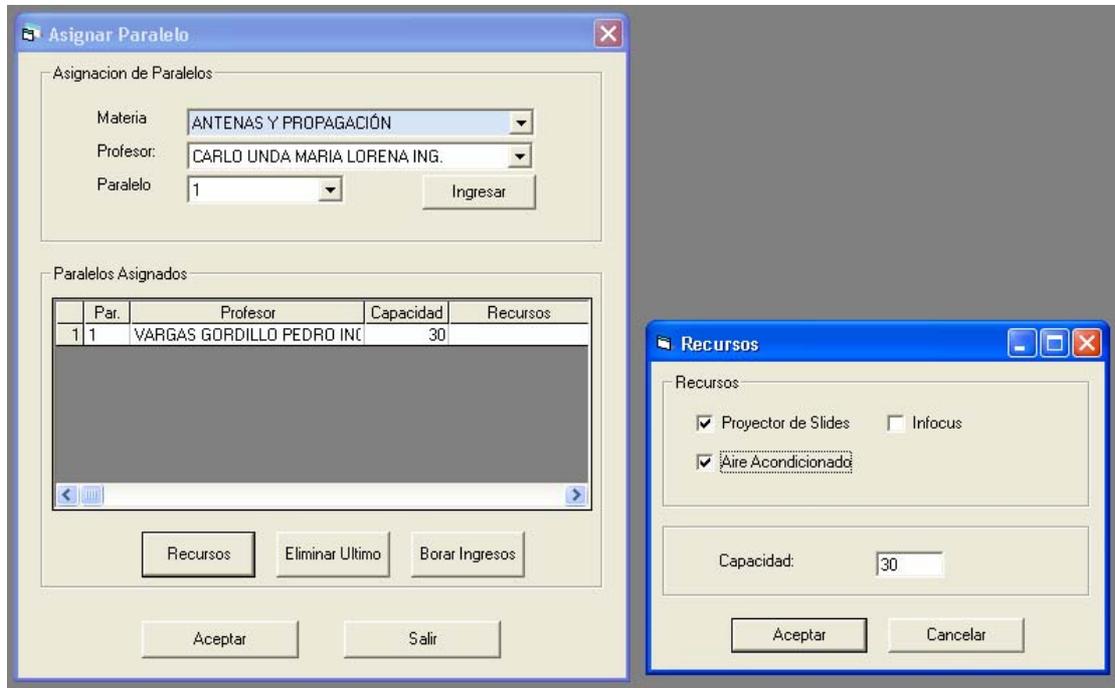


Ilustración 4-126: Asignación de paralelos

7. Iniciar el proceso de optimización

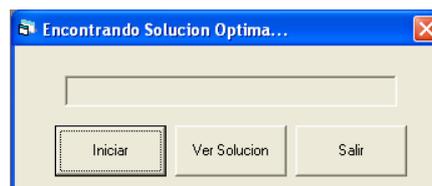


Ilustración 4-127: Optimizar

8. Creación de nuevos paralelos.

Agregar Paralelo a la Planificación

Ingreso de nuevo paralelo

Profesor: ALEJANDRO MOLINA OTILIA ING.

Materia: ANÁLISIS DE ALGORITMOS

Paralelo: 3 Aula: A104

Día: Lunes

Hora Inicio: 07:00 Hora Fin: 09:00

Ingresar

Horario Asignado

	Materia	Paralelo	Aula	Día	Hora Ini	f
1	ANÁLISIS DE ALGORITMOS	3	A104	Lunes	07:00	0

Borrar Horario Eliminar Ultimo

Aceptar Salir

Ilustración 4-128: Creación de nuevo paralelo, posterior a la optimización

9. Obtención de reportes:

Listado de Profesores

21/05/2005

Contratado/Planta	Profesor	Identificación
C		
C	ABAD CRISTINA Msc	P00001
C	ALEJANDRO MOLINA OTILIA ING.	P00002
C	ALVARADO M. OTTO ING.	P00003
C	BERMUDEZ FLORES GUSTAVO ING.	P00008
C	CAICEDO R. GUIDO ING.	P00009
C	CARLO UNDA MARIA LORENA ING.	P00010
C	DEL POZO C. CARLOS ING.	P00015
C	DEL POZO L. JUAN ING.	P00016
C	ECHVERRIA B. PEDRO ING.	P00017
C	GALLO GALARZA JUAN ING.	P00022
C	GARCÉS MENDOZA MARCIA ING.	P00023
C	GARCÍA AGUILAR SIXTO DR.	P00024
C	HERRERA EFREN ING.	P00029
C	JORDAN VILLAMAR CARLOS ING.	P00030
C	LARCO GOMEZ ALBERTO ING.	P00031
C	LOPEZ DIEGO ING.	P00036
C	LUCERO GUILLEN JAIME ING.	P00037
C	MACIAS MENDOZA MARIA ING.	P00038
C	MEDINA M. WASHINGTON ING.	P00043
C	MERA GENCON CRISTOBAL DR.	P00044
C	MOLINEROS M. ERNESTO ING.	P00045
C	NUÑEZ BORJA MANUEL ING.	P00050
C	OCHOA CH. XAVIER ING.	P00051
C	PAILLACHO DENNYS ING.	P00052
C	PROFESOR INGLES A1	P00057
C	PROFESOR INGLES A2	P00058
C	PROFESOR INGLES B1	P00059
C	RUBIO ROLDAN GOMER ING.	P00064
C	SALAZAR LOPEZ CARLOS ING.	P00065
C	SALCEDO G. ADOLFO ING.	P00066
C	TAPIA ROSERO ANA TERESA ING.	P00071
C	UQUILLAS GÓMEZ VERÓNICA ING.	P00072
C	URQUIZO C. JAVIER ING.	P00073
C	VÁSQUEZ VERA LUIS ING.	P00078
C	VELOZ JESSICA ECON.	P00079

Ilustración 4-129: Reporte de Profesores

Listado de Materias

21/05/2005 Especializaci

Nivel	Código	Materia	Aula		Tip	um Paralelos
			arga Horaria	Aula/Laboratorio		
Comput						
400	FIEC01545	INTERACCIÓN HOMBRE MÁQUINA	4 A		E	1
400	FIEC01552	LENGUAJES DE PROGRAMACIÓN	4 A		E	2
400	FIEC01578	COMUNICACIÓN DE DATOS	4 A		E	2
400	FIEC02048	ANÁLISIS Y DISEÑO ORIENTADO A	4 A		E	1
400	FIEC02097	SISTEMAS OPERATIVOS	4 A		E	1
400	FIEC02105	SISTEMAS DE INFORMACIÓN	4 A		E	1
400	FIEC02121	SISTEMAS DE BASE DE DATOS	4 A		E	2
400	FIEC03012	ESTRUCTURA DE DATOS	4 A		E	4
400	FIEC03046	INGENIERÍA DE SOFTWARE I	4 A		E	2
400	FIEC03053	INGENIERÍA DE SOFTWARE II	4 A		E	1
400	FIEC03079	REDES DE COMPUTADORES I	4 A		E	1
400	FIEC03087	REDES DE COMPUTADORES II	4 A		E	1
400	FIEC03111	COMPUTACIÓN APLICADA A LA INGE	4 A		E	1
400	FIEC03319	ORGANIZACIÓN DE COMPUTADORES	4 A		E	2
400	FIEC03343	SISTEMAS EXPERTOS	4 A		E	1
400	FIEC03459	INTELIGENCIA ARTIFICIAL	4 A		E	1
400	FIEC04275	SISTEMAS DE MICROPROCESADORES	4 A		E	1
400	FIEC04341	FUNDAMENTOS DE PROGRAMACIÓN	4 A		E	5
400	FIEC04358	INTRODUCCIÓN A LA INFORMÁTICA	4 A		E	6
400	FIEC04374	PARADIGMA ORIENTADO A OBJETOS	4 A		E	3
400	FIEC04382	SEÑALES Y SISTEMAS	4 A		E	2
400	FIEC04390	CHARLAS PROFESIONALES	4 A		E	1
400	FIEC04861	SISTEMAS DE INFORMACIÓN APLICA	4 A		E	1
400	FIEC05066	ANÁLISIS DE ALGORITMOS	4 A		E	1
400	FIEC05314	INTRODUCCIÓN A LAS REDES DE CO	4 A		E	1
400	FIEC05405	GRÁFICOS POR COMPUTADORA I	4 A		E	1
400	FIEC05413	PROCESAMIENTO DE AUDIO Y VIDEO	4 A		E	1
400	FIEC05421	GRÁFICOS POR COMPUTADORA II	4 A		E	1
400	FIEC05439	PROCESAMIENTO DIGITAL DE IMÁGE	4 A		E	1
400	FIEC05447	SISTEMAS DE MULTIMEDIA	4 A		E	1
400	FIEC05454	INTERFACES POR VOZ	4 A		E	1
400	FIEC05462	APLICACIONES MULT. INTERACTIVA	4 A		E	1

Ilustración 4-130: Reporte de Materias

Listado de Aulas

21/05/2005

Tipo	Aula	Laboratorio	Recursos
A			
A	A103		
A	A104		Infocus , A/C
A	A105		Infocus , A/C
A	A112		Infocus , A/C
A	A113		Infocus , A/C
A	A114		
A	A115		
A	A201		
A	A202		Infocus , A/C
A	A203		Infocus , A/C
A	A204		
A	A205		
A	A206		
A	A207		Infocus , A/C
A	A208		Infocus , A/C
A	A209		
A	A210		
A	ASAT		Infocus , A/C
A	AUTO		Infocus , A/C
A	COM1		Infocus , A/C
A	COM2		Infocus , A/C
A	COM3		Infocus , A/C
A	COM4		Infocus , A/C
L			
L	LABT	LAB. LABT	
L	LCIE	LAB. LCIE	
L	LELA	LAB. LELA	
L	LELB	LAB. LELB	
L	LELM	LAB. LELM	
L	LLEP	LAB. LLEP	
L	LMA	LAB. LMAQ	

Ilustración 4-131: Reporte de Aulas

Planificador de Horarios - PLANIFICACION PRUEBA AL 25-09-05 - Administrador - [Viewer]

Ver

100% 32 de 50+

Vista previa

- PLANPRUEBA
 - A103
 - A104
 - A105
 - A112
 - A113
 - A114
 - A115
 - A201
 - A202
 - A203
 - A204
 - A205
 - A206
 - A207
 - A208
 - A209
 - A210
 - ASAT
 - AUTO
 - LABT
 - LCIE
 - LELA
 - LELB
 - LLEP
 - LMAQ
 - LRED
 - LSDI
 - LSDP

Horario de clases clasificados por aula

23/10/2005

Aula	Día	Horario	Paralelo	Codigo	Materia	Profesor
A104						
Lunes						
		08:30-09:3	1	TEMP00	INGLES B	PROFESOR INGLES B1
		09:30-10:3	1	TEMP00	INGLES B	PROFESOR INGLES B1
		11:30-12:3	7	FIEC002	SISTEMAS DIGITALES I	MARTIN MORENO CESAR I
		12:30-13:3	7	FIEC002	SISTEMAS DIGITALES I	MARTIN MORENO CESAR I
		17:30-18:3	1	FIEC053	SISTEMAS PARA LA TOMA D	VALVERDE L. GALO ING.
		18:30-19:3	1	FIEC053	SISTEMAS PARA LA TOMA D	VALVERDE L. GALO ING.
		19:30-20:3	4	FIEC043	FUNDAMENTOS DE PROGR	VACA RUIZ CARMEN ING.
		20:30-21:3	4	FIEC043	FUNDAMENTOS DE PROGR	VACA RUIZ CARMEN ING.
Martes						
		08:30-09:3	1	TEMP00	INGLES B	PROFESOR INGLES B1
		09:30-10:3	1	TEMP00	INGLES B	PROFESOR INGLES B1
		11:30-12:3	1	FIEC021	SISTEMAS DE INFORMACIÓ	ALEJANDRO MOLINA OTILI
		12:30-13:3	1	FIEC021	SISTEMAS DE INFORMACIÓ	ALEJANDRO MOLINA OTILI
		13:30-14:3	2	TEMP00	MICROCONTROLADORES	VILLAVICENCIO V. HUGO I
		14:30-15:3	2	TEMP00	MICROCONTROLADORES	VILLAVICENCIO V. HUGO I
		15:30-16:3	1	FIEC001	MAQUINARIA ELÉCTRICA II	BERMUDEZ FLORES GUSTA
		16:30-17:3	1	FIEC001	MAQUINARIA ELÉCTRICA II	BERMUDEZ FLORES GUSTA
		19:30-20:3	1	FIEC001	MAQUINARIA ELÉCTRICA I	BERMUDEZ FLORES GUSTA
		20:30-21:3	1	FIEC001	MAQUINARIA ELÉCTRICA I	BERMUDEZ FLORES GUSTA

Ilustración 4-132: Reporte de Horarios por Aula

Planificador de Horarios - default - Administrador - [Viewer]

Ver

100%

1 de 52+

Vista previa

ABAD CRISTINA

Horario de Clases Clasificado por Profesor

23/10/2005

Aula	CodigoMateria	Paralelo	Materia
ABAD CRISTINA Msc			
Lunes			
09:30-10:3	A208	FIEC053	1 INTRODUCCIÓN A LAS REDE
10:30-11:3	A208	FIEC053	1 INTRODUCCIÓN A LAS REDE
11:30-12:3	A113	FIEC055	9 VISION POR COMPUTADOR
12:30-13:3	A113	FIEC055	9 VISION POR COMPUTADOR
Martes			
07:30-08:3	A114	FIEC015	1 COMUNICACIÓN DE DATOS
08:30-09:3	A114	FIEC015	1 COMUNICACIÓN DE DATOS
13:30-14:3	A202	TEMP00	1 SISTEMAS DISTRIBUIDOS
14:30-15:3	A202	TEMP00	1 SISTEMAS DISTRIBUIDOS
Miércoles			
09:30-10:3	A208	FIEC053	1 INTRODUCCIÓN A LAS REDE
10:30-11:3	A208	FIEC053	1 INTRODUCCIÓN A LAS REDE
11:30-12:3	A113	FIEC055	9 VISION POR COMPUTADOR
12:30-13:3	A113	FIEC055	9 VISION POR COMPUTADOR

Ilustración 4-133: Reporte de Horarios por Profesor

Planificador de Horarios - default - Administrador - [Viewer]

Ver

100%

54 de 54+

Vista previa

PLANPRUEBA

AUTOMATIZACION INDUSTRIAL

Horario de Clases Clasificado por Materia

23/10/2005

CodigoMateria	Materia	Horario	Profesor
APLICACIONES MULT. INTERACTIVA			
Jueves			
FIEC054	APLICACIONES MULT. INTE	17:30-18:3	LOPEZ DIEGO ING.
FIEC054	APLICACIONES MULT. INTE	18:30-19:3	LOPEZ DIEGO ING.
Viernes			
FIEC054	APLICACIONES MULT. INTE	13:30-14:3	LOPEZ DIEGO ING.
FIEC054	APLICACIONES MULT. INTE	14:30-15:3	LOPEZ DIEGO ING.
AUTOMATIZACIÓN INDUSTRIAL			
Martes			
FIEC042	AUTOMATIZACIÓN INDUS	15:30-16:3	MANZUR.HANNA ALBEF
FIEC042	AUTOMATIZACIÓN INDUS	16:30-17:3	MANZUR.HANNA ALBEF

Ilustración 4-134: Reporte de Horarios por Materia

Reporte de Conflictos						
24/10/2005						
Planificación	Aul	Día	Horario	Materia	Paralelo	Profesor
PLANTOT2						
A104						
Lunes						
			07:00-08:00	VISION POR COMPUTADORA	1	ABAD CRISTINA MSC
			07:00-08:00	VISION POR COMPUTADORA	2	ABAD CRISTINA MSC
			08:00-09:00	VISION POR COMPUTADORA	1	ABAD CRISTINA MSC
			08:00-09:00	VISION POR COMPUTADORA	2	ABAD CRISTINA MSC
		Lunes			4	
Miércoles						
			07:00-08:00	VISION POR COMPUTADORA	1	ABAD CRISTINA MSC
			07:00-08:00	VISION POR COMPUTADORA	2	ABAD CRISTINA MSC
			08:00-09:00	VISION POR COMPUTADORA	1	ABAD CRISTINA MSC
			08:00-09:00	VISION POR COMPUTADORA	2	ABAD CRISTINA MSC
		Miércoles			4	
		A104			8	
		PLANTOT2			8	
		Total de Horas Cruzadas			8	

Ilustración 4-135: Reporte de Conflictos

4.4 MODELOS DE DISEÑOS

4.4.1 MODELOS ESTÁTICOS

4.4.1.1 DISEÑO DE LA BASE DE DATOS

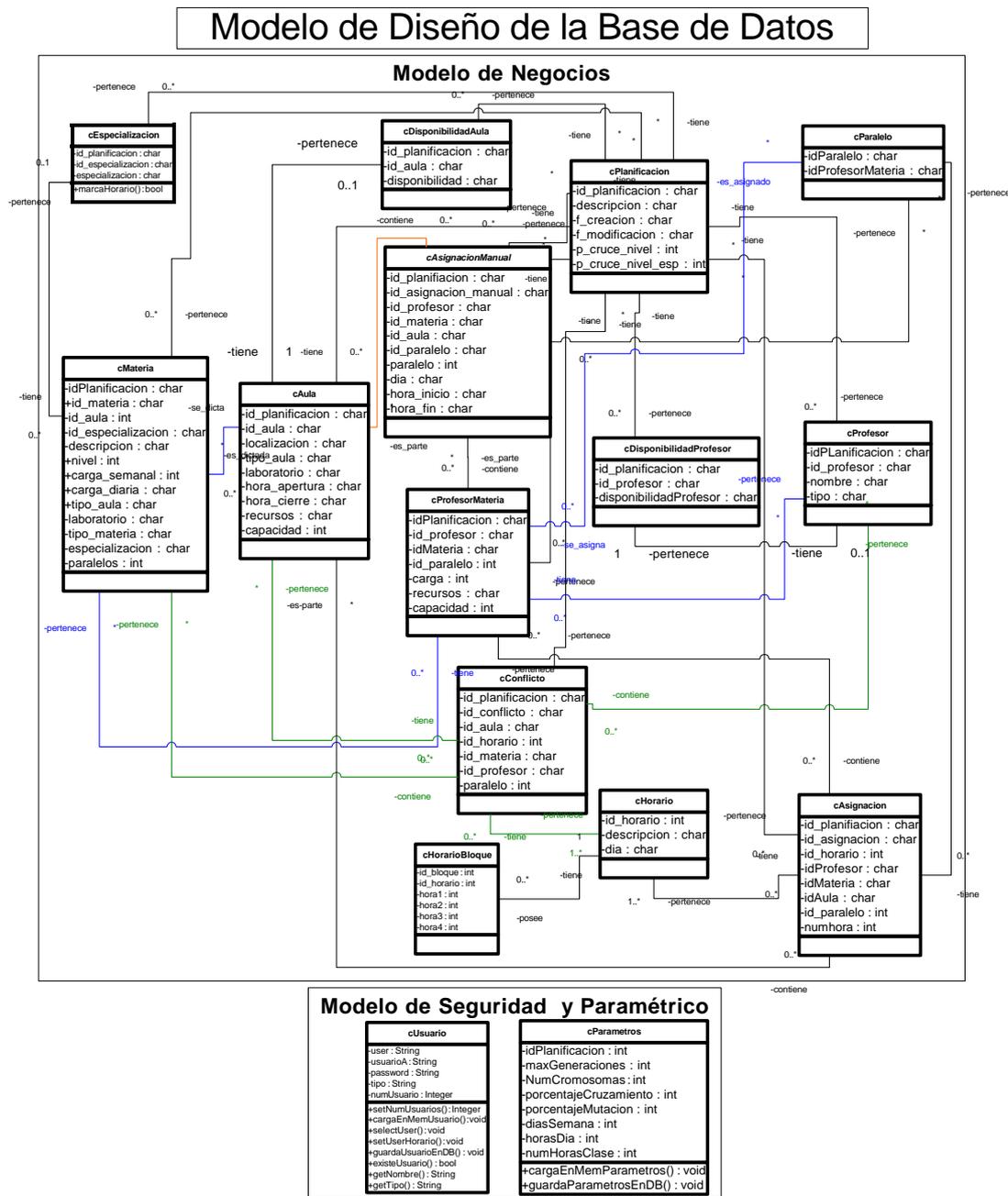


Ilustración 4-136: Modelo Lógico de la Base de Datos

4.4.2 MODELOS DINÁMICOS

4.4.2.1 DIAGRAMAS DE INTERACCIÓN DE OBJETOS

Escenario 1.1: Creación de Nueva Planificación Exitosa

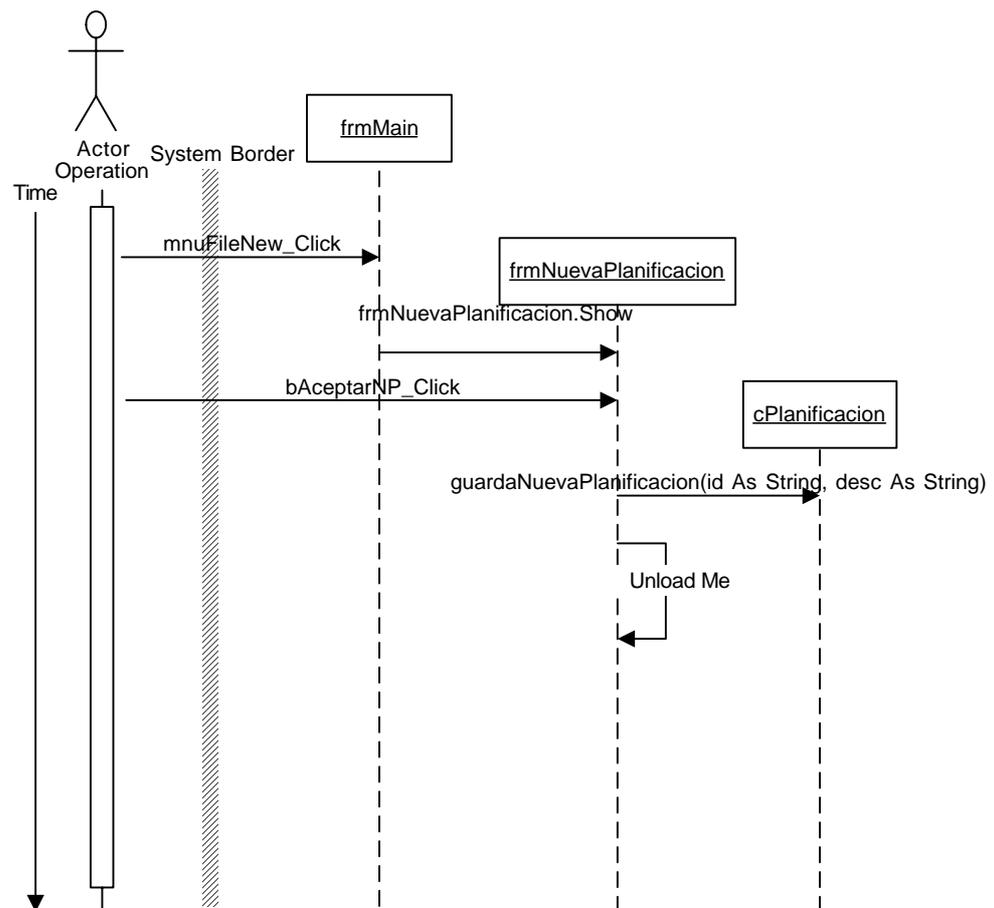


Ilustración 4-138: DIO 1.1: Creación de Nueva Planificación Exitosa.

Escenario 2.1: Apertura de Planificación Exitosa

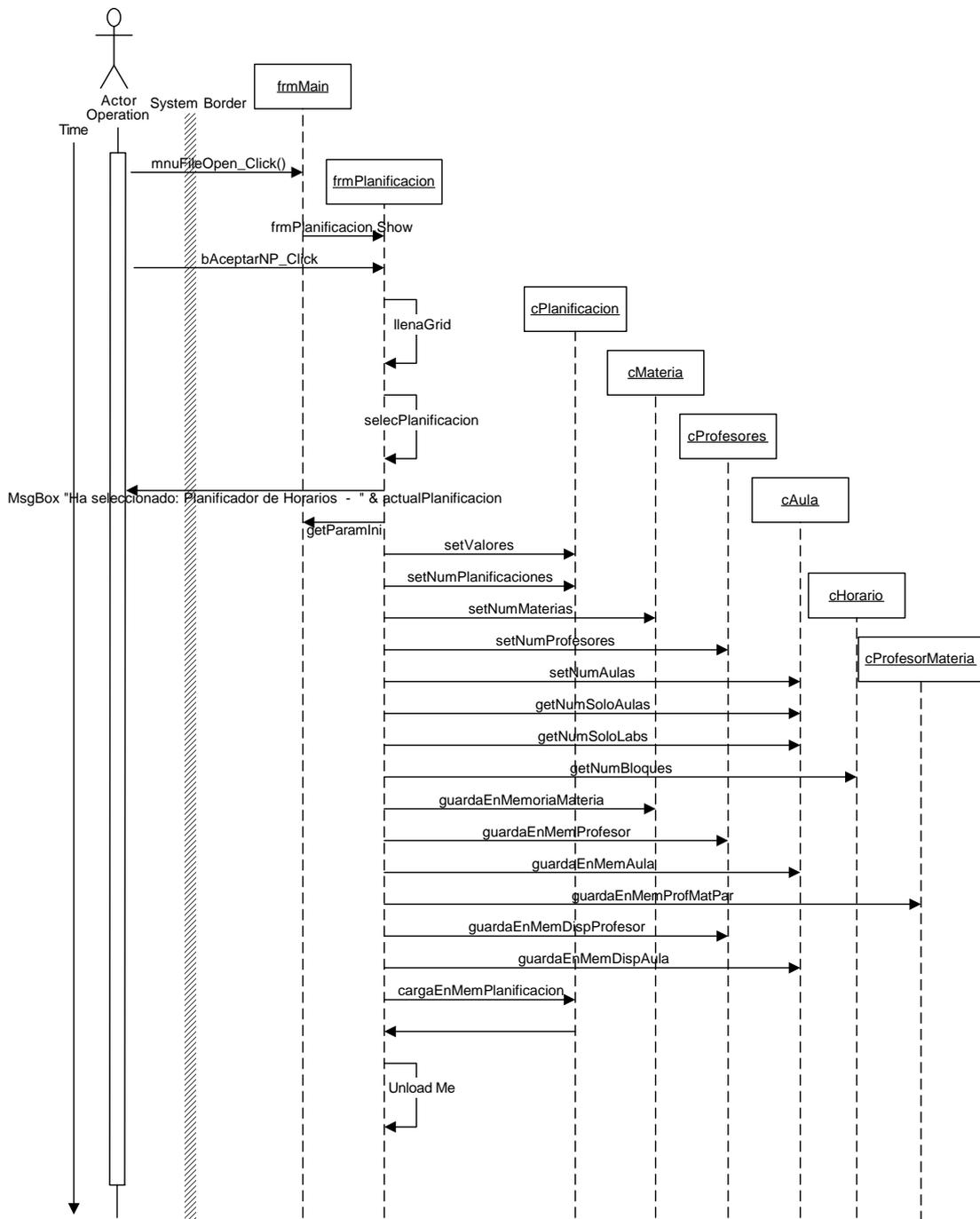


Ilustración 4-139: DIO 2.1: Apertura de Planificación Exitosa.

Escenario 3.1: Eliminación de Planificación Exitosa

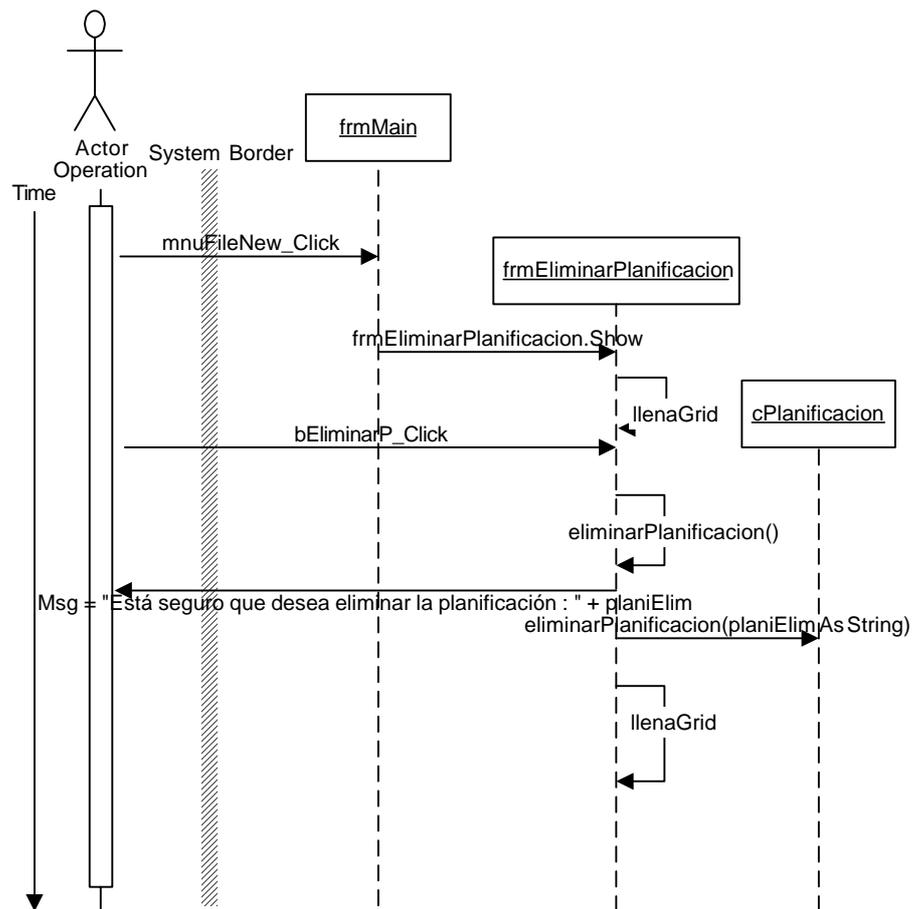


Ilustración 4-140: DIO 3.1: Eliminación de Planificación Exitosa.

Escenario 4.1: Guardar Como de Planificación Exitosa

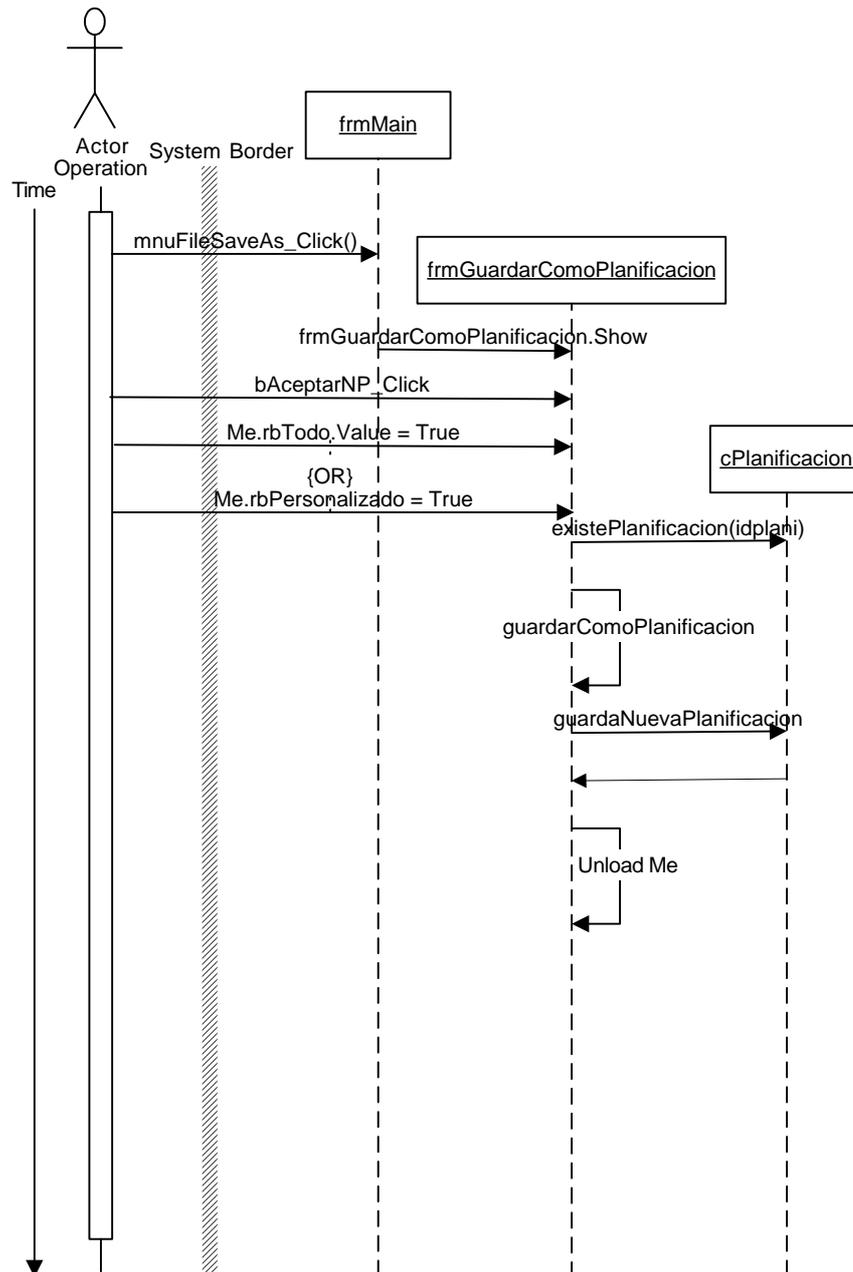


Ilustración 4-141: DIO 4.1: Guardar Como de Planificación Exitosa.

Escenario 5.1: Cambio de Usuario Exitoso

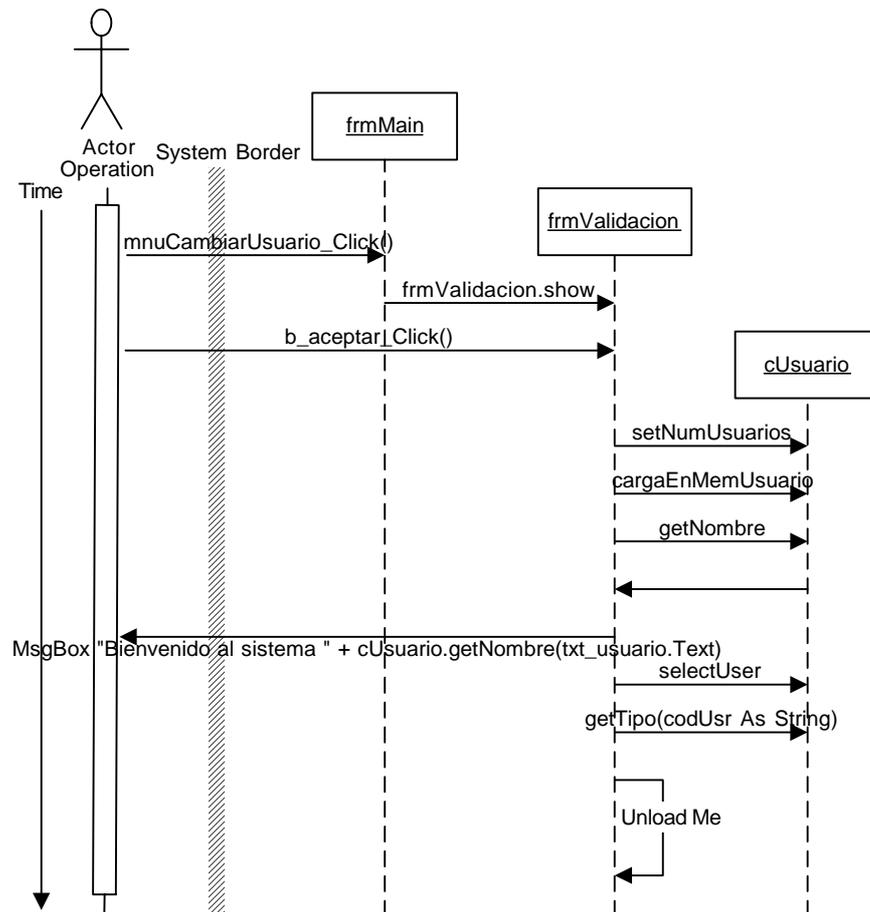


Ilustración 4-142: DIO 5.1: Cambio de Usuario Exitoso.

Escenario 6.1: Cambio de Contraseña Exitosa

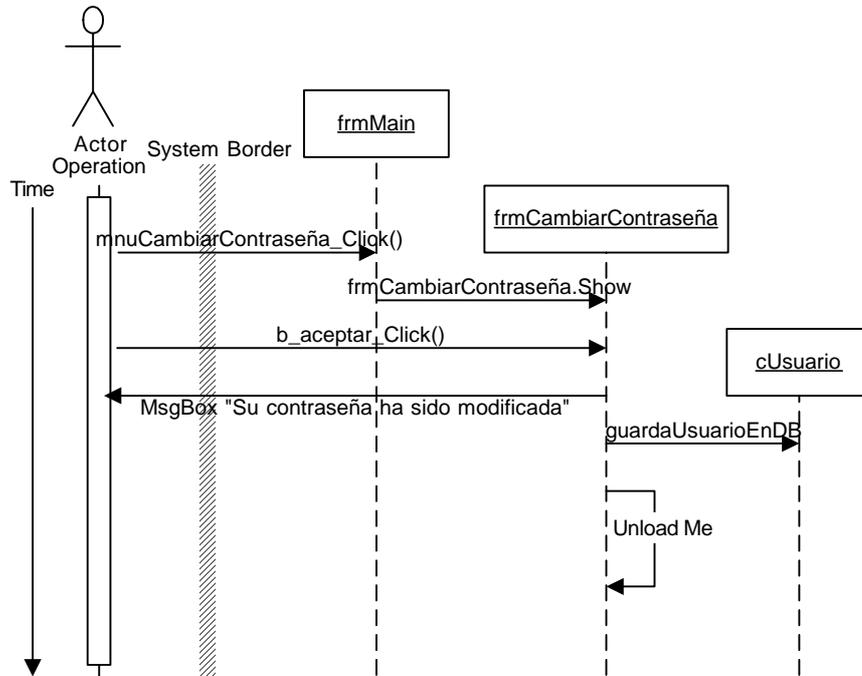


Ilustración 4-143: DIO 6.1: Cambio de Contraseña Exitosa.

Escenario 7.1: Ingreso de Usuario Exitoso

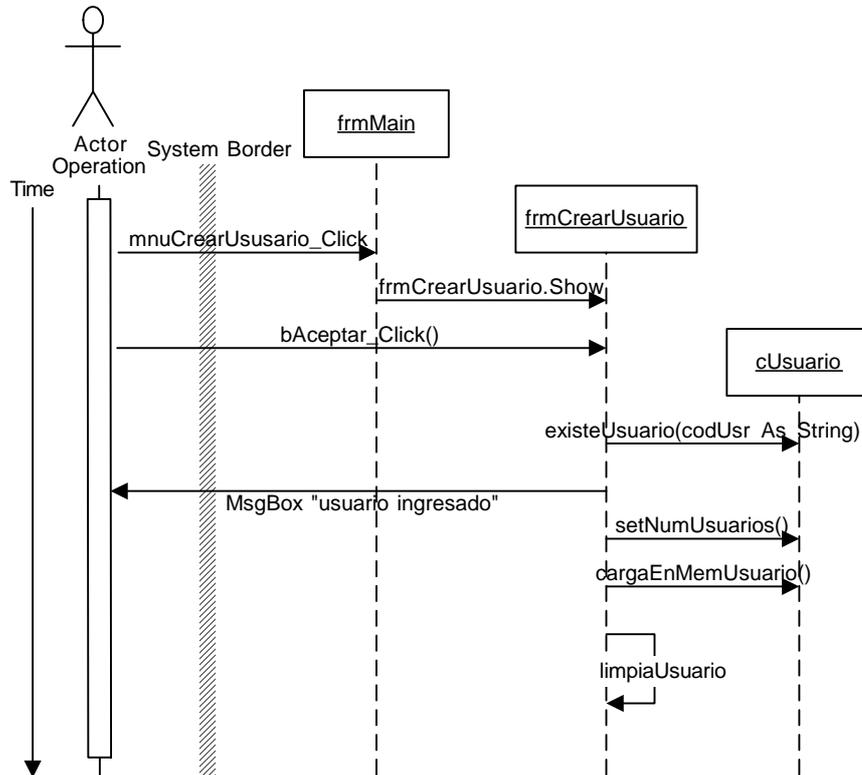


Ilustración 4-144: DIO 7.1: Ingreso de Usuario Exitoso.

Escenario 8.1: Modificación de Usuario Exitoso

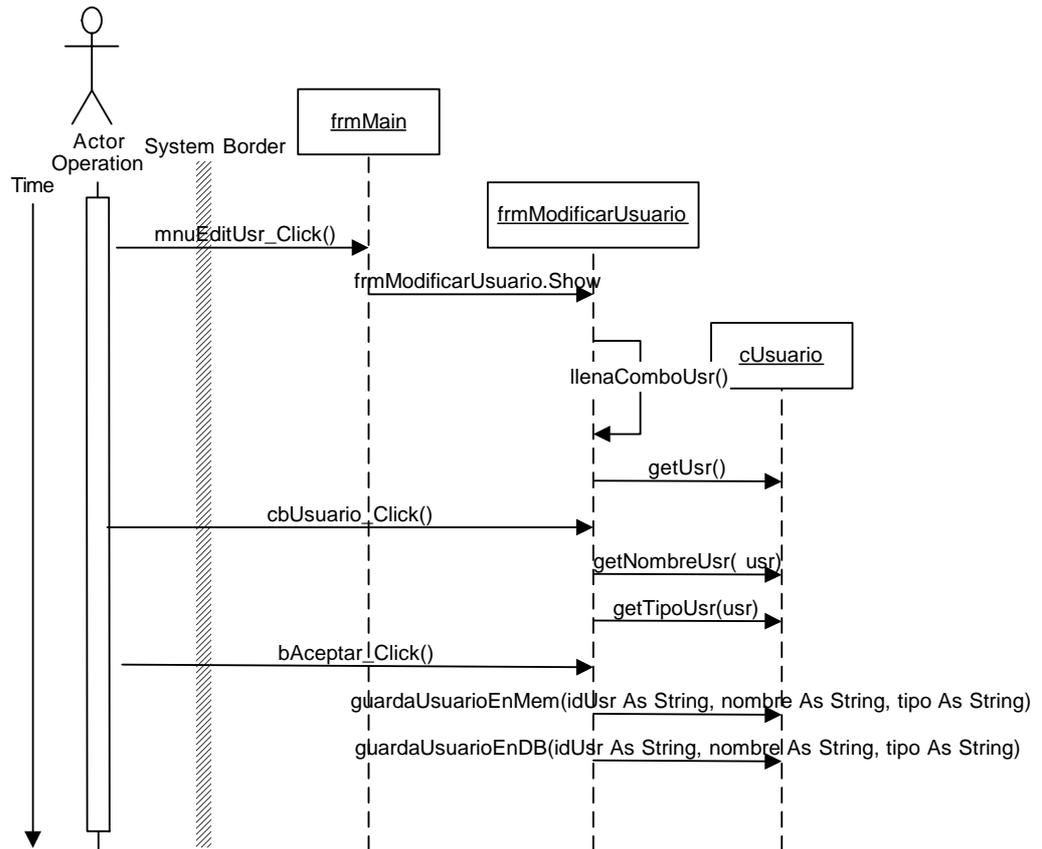


Ilustración 4-145: DIO 8.1: Modificación de Usuario Exitoso.

Escenario 9.1: Eliminación de Usuario Exitoso

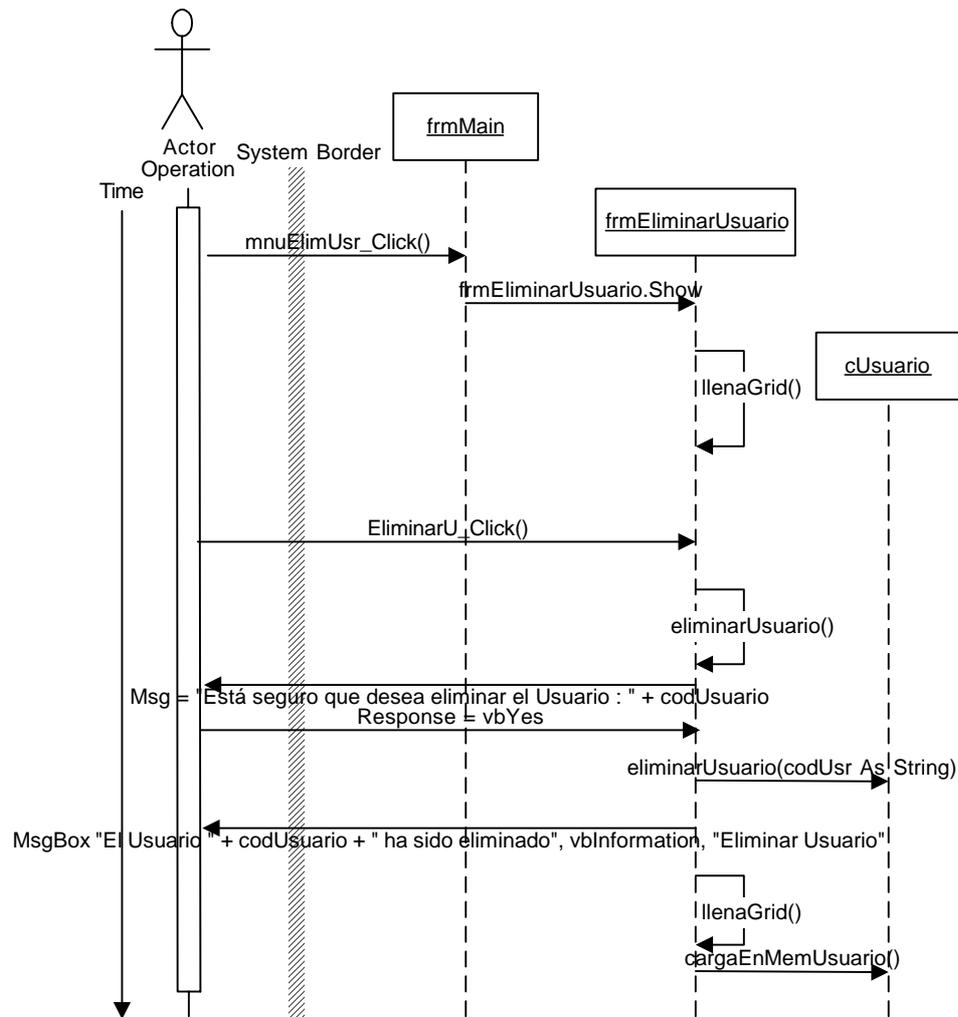


Ilustración 4-146: DIO 9.1: Eliminación de Usuario Exitoso.

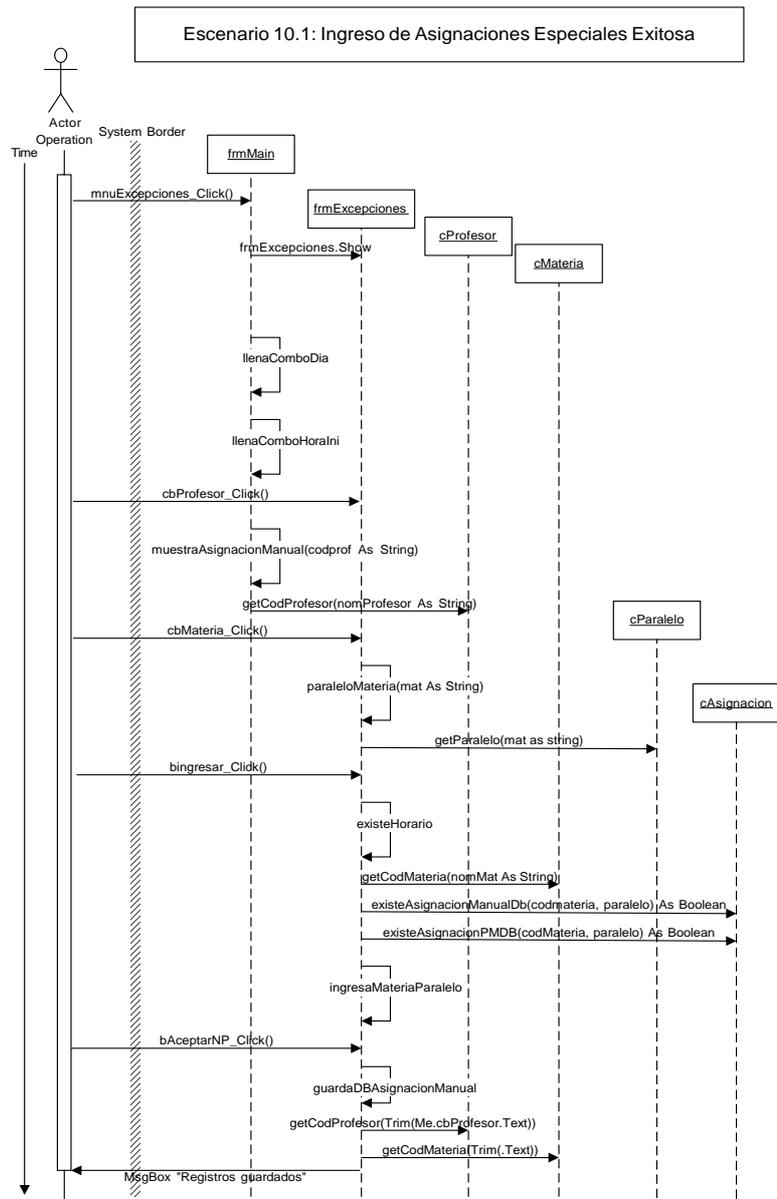


Ilustración 4-147: DIO 10.1: Ingreso de Asignaciones Especiales Exitosa.

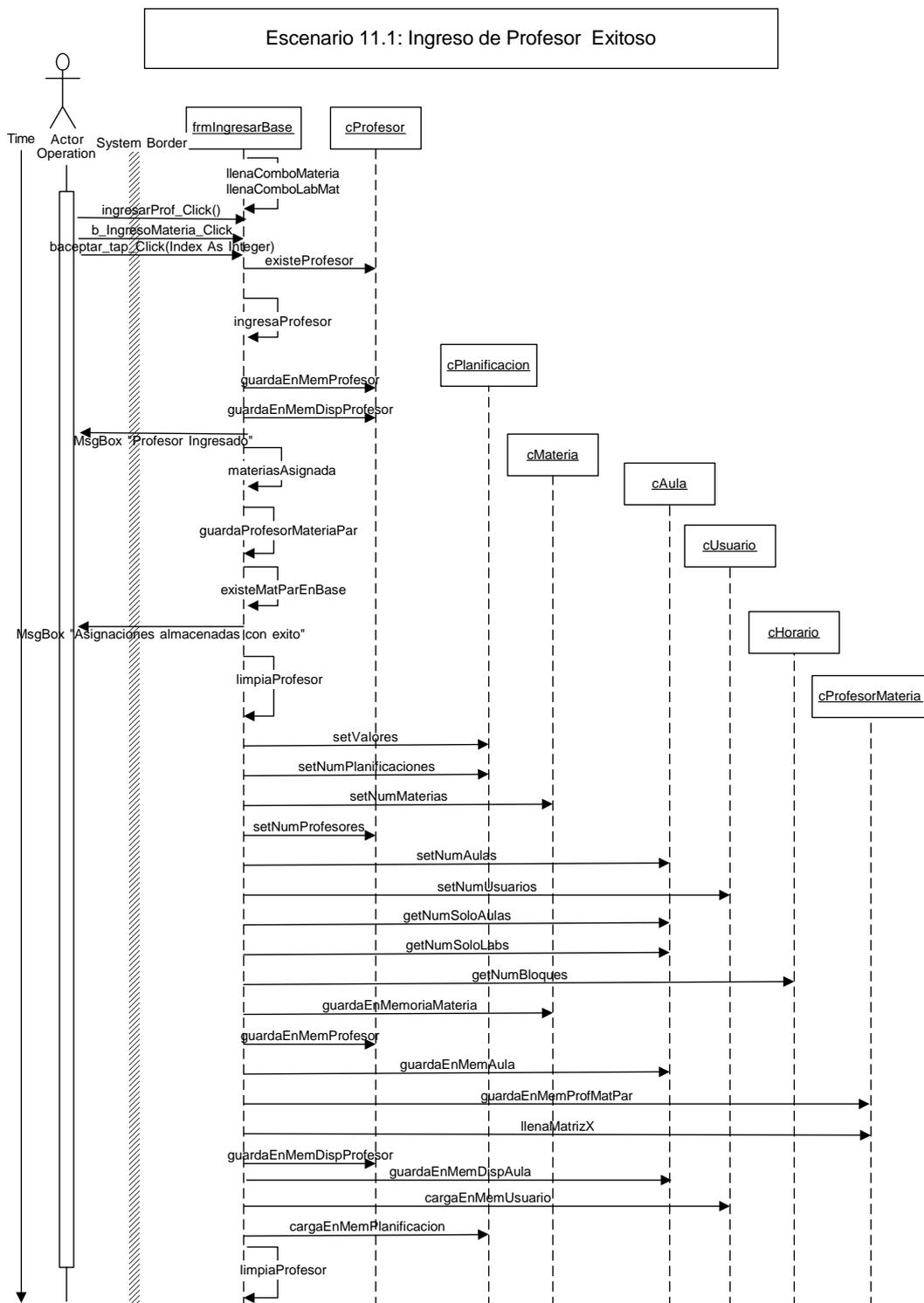


Ilustración 4-148: DIO 11.1: Ingreso de Profesor Exitoso.

Escenario 11.2: Identificación del profesor invalida

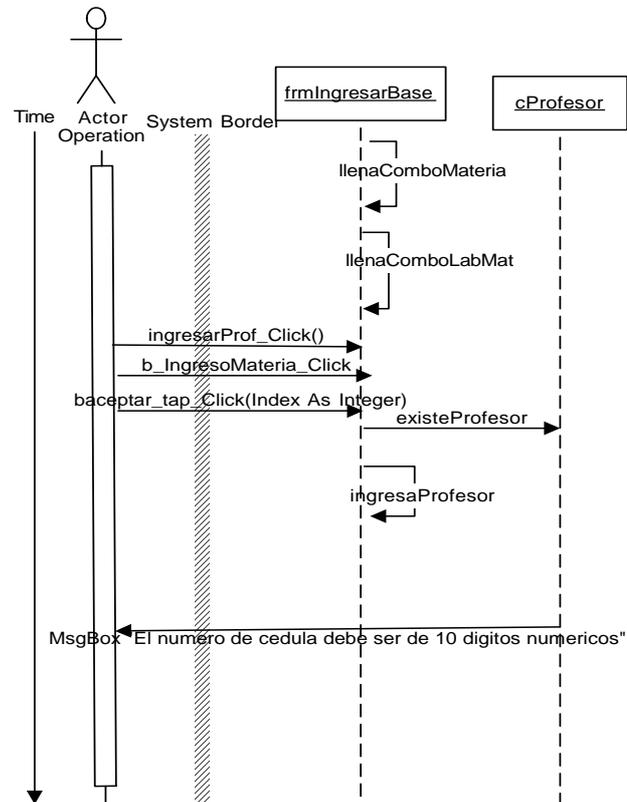


Ilustración 4-149: DIO 11.2: Identificación del profesor invalida.

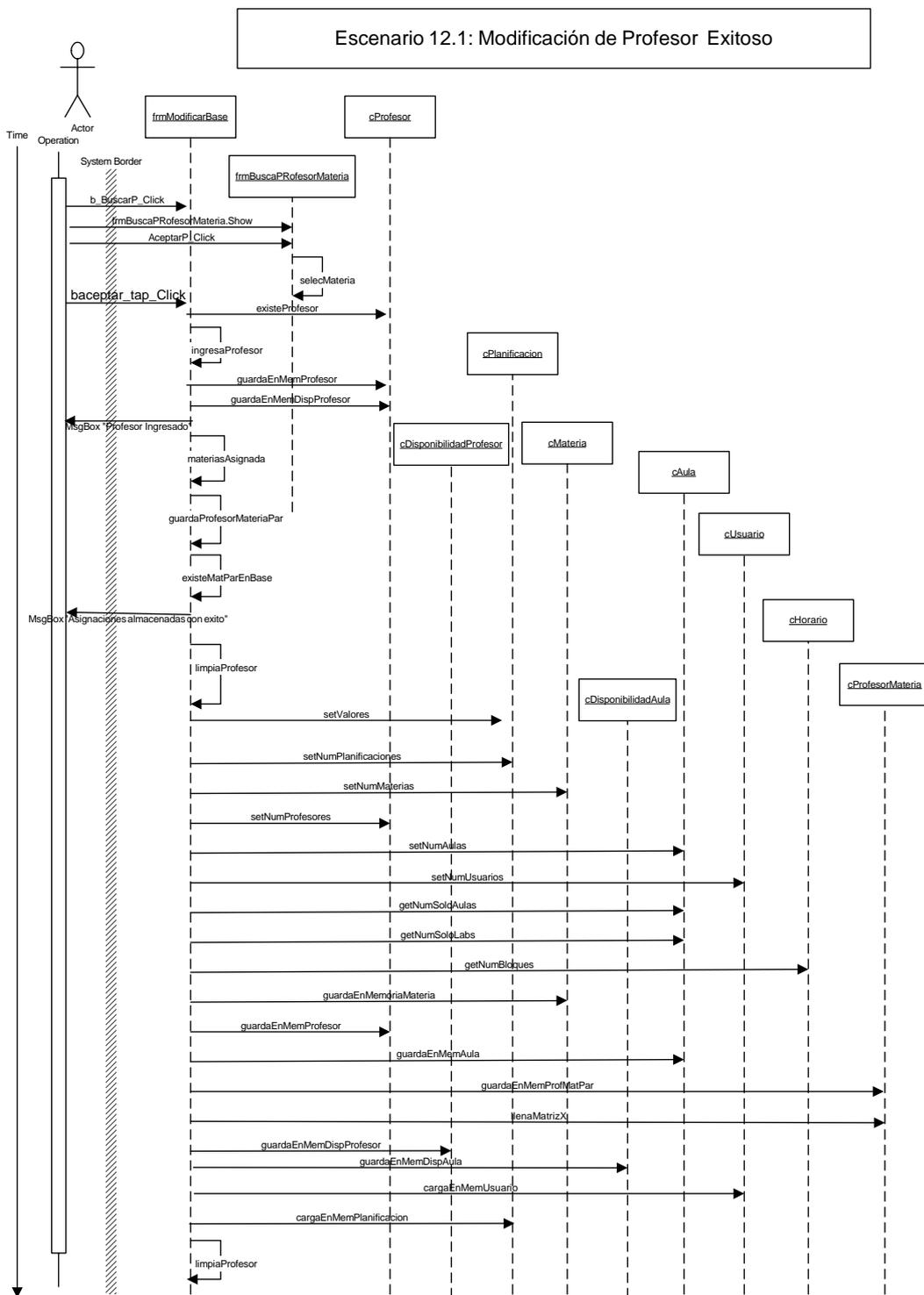


Ilustración 4-150: DIO 12.1: Modificación de Profesor Exitoso.

Escenario 13.1: Eliminación de Profesor Exitosa

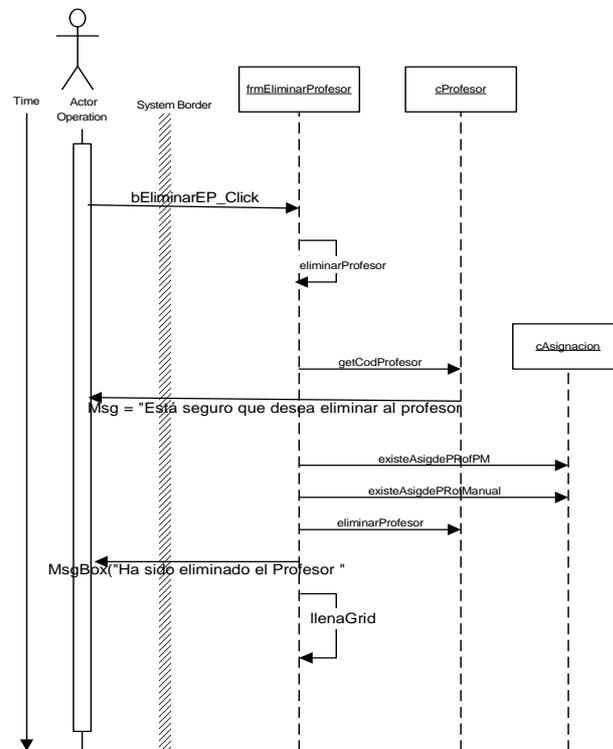


Ilustración 4-151: DIO 13.1: Eliminación de Profesor Exitosa.

Escenario 14.1: Ingreso de Materia Exitoso

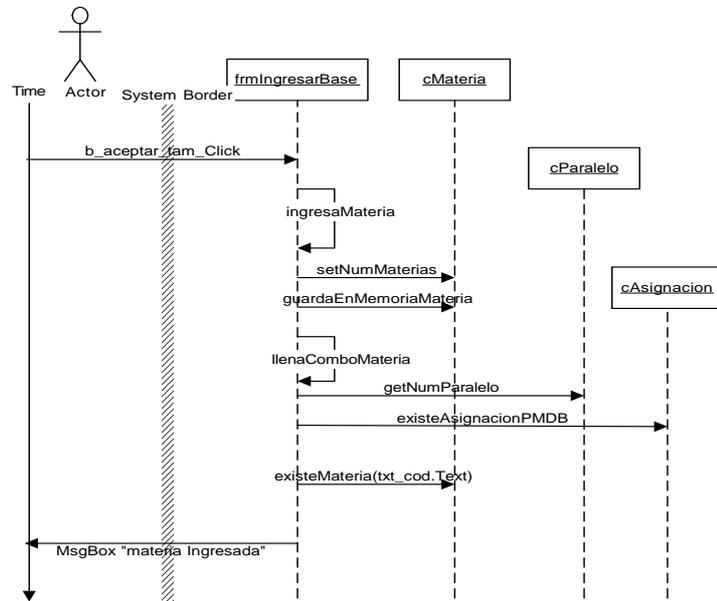


Ilustración 4-152: DIO 14.1: Ingreso de Materia Exitoso.

Escenario 14.2: Error en el Código de la Materia

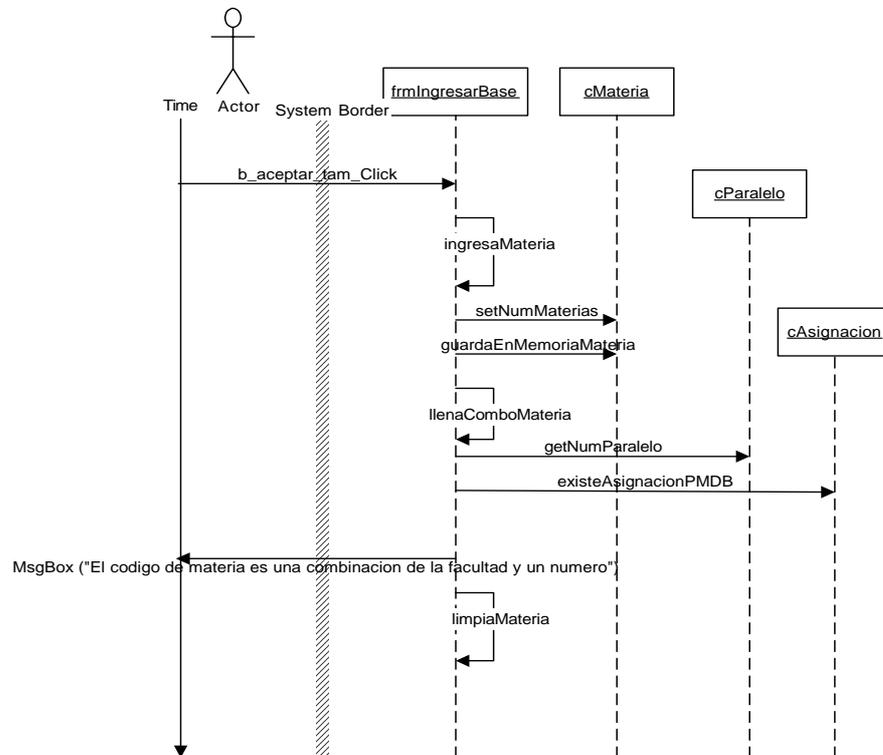


Ilustración 4-153: DIO 14.2: Error en el Código de la Materia

Escenario 15.1: Modificación de Materia Exitoso

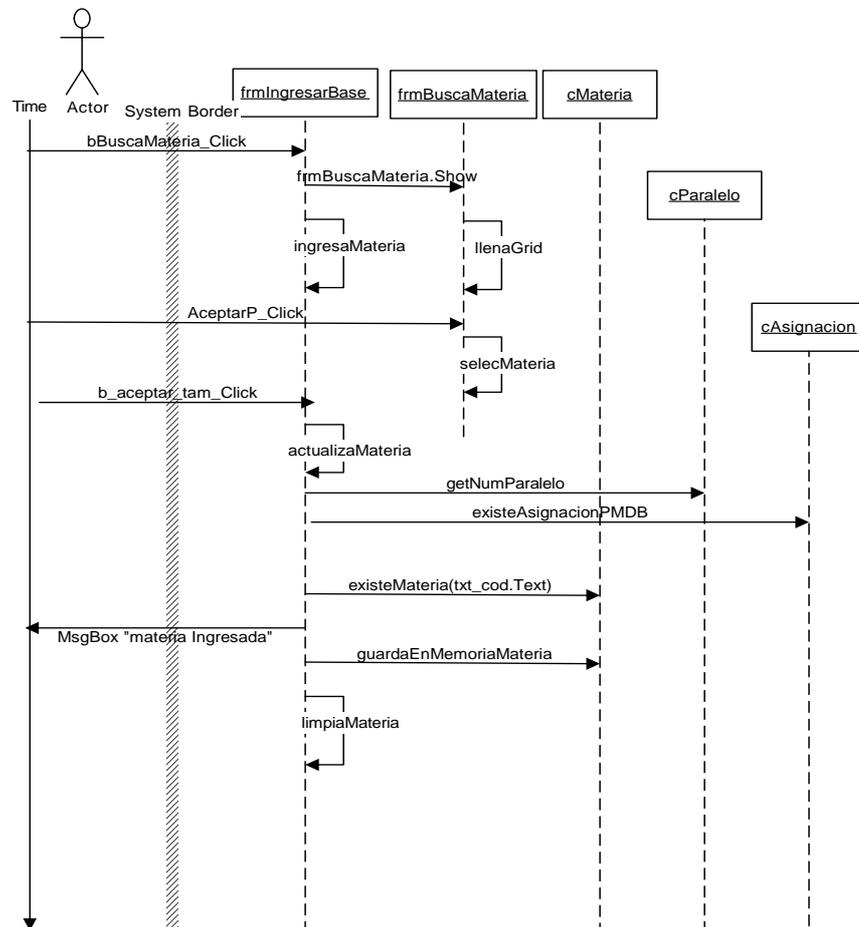


Ilustración 4-154: DIO 15.1: Modificación de Materia Exitoso.

Escenario 15.2: Error en el Nivel de la Materia

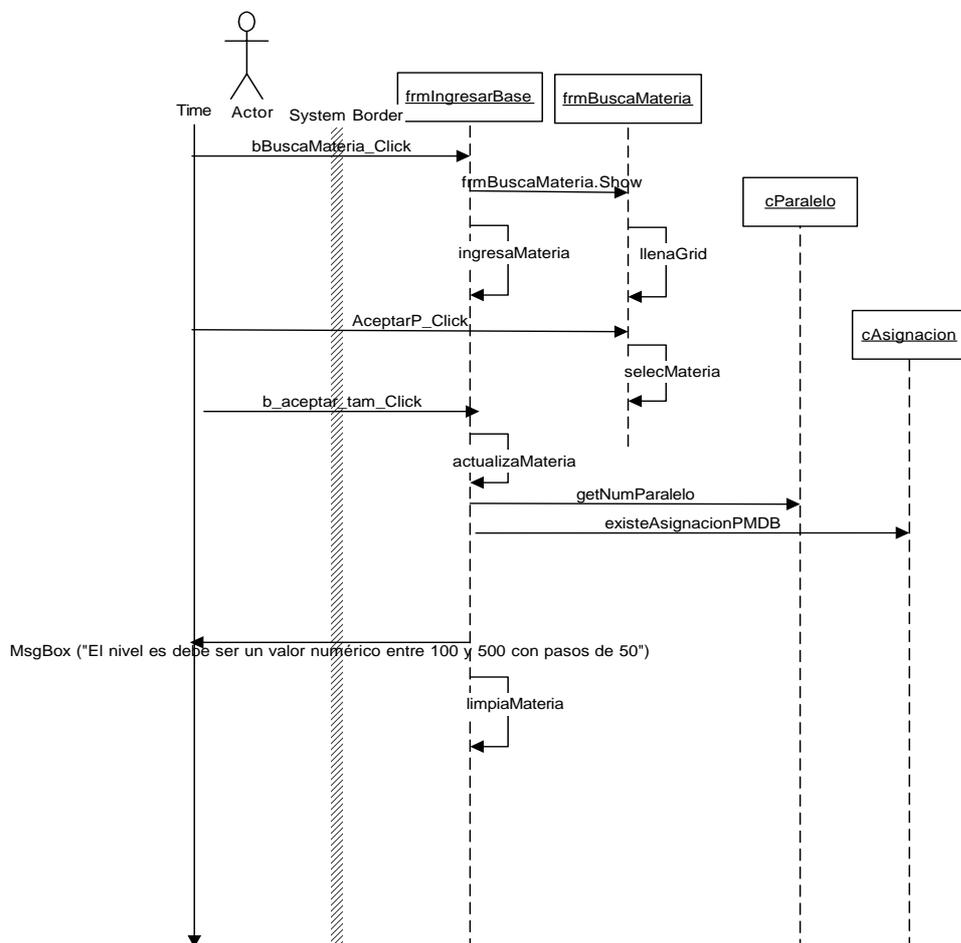


Ilustración 4-155: DIO 15.2: Error en el Nivel de la Materia.

Escenario 16.1: Eliminación de Materia Exitoso

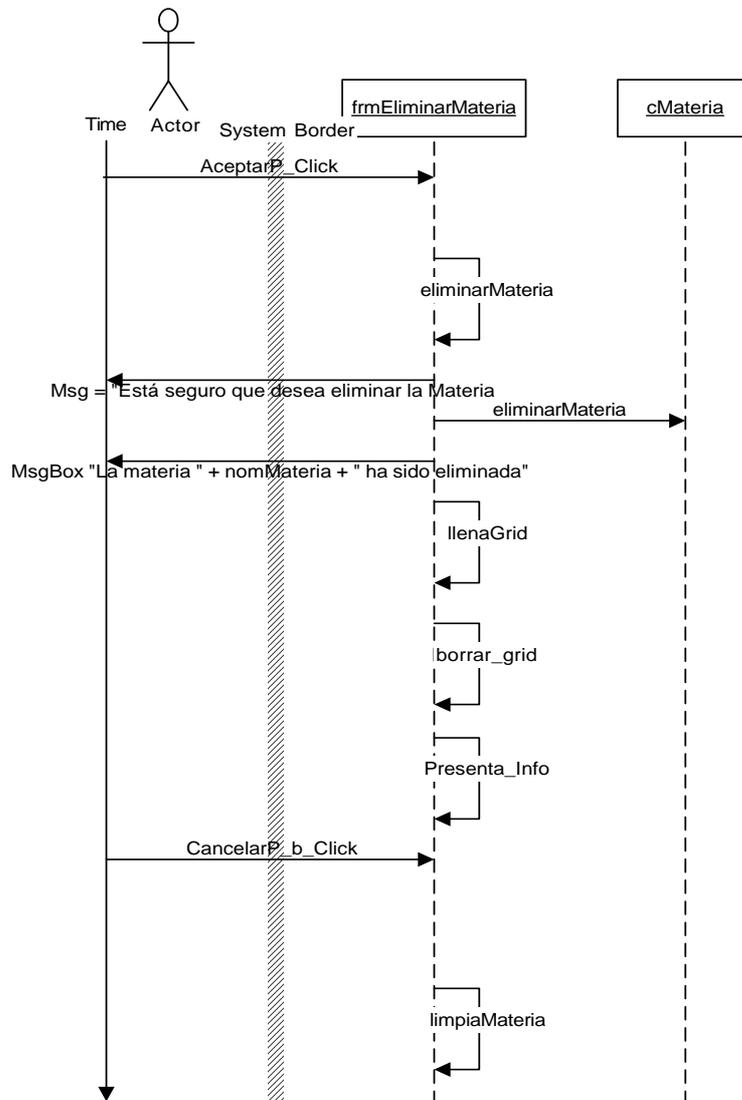


Ilustración 4-156: DIO 16.1: Eliminación de Materia Exitoso.

Escenario 17.1: Ingreso de Aula Exitoso

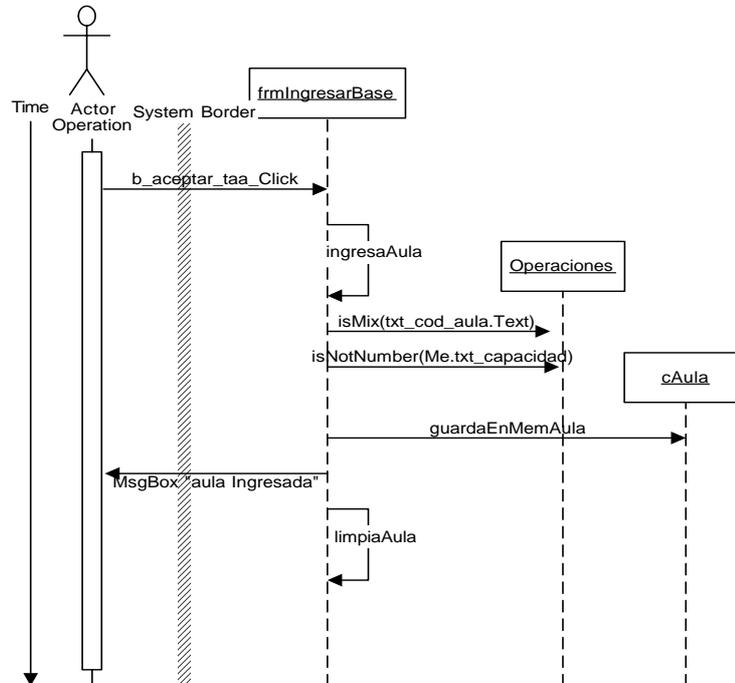


Ilustración 4-157: DIO 17.1: Ingreso de Aula Exitoso.

Escenario 18.1: Modificación de Aula Exitoso

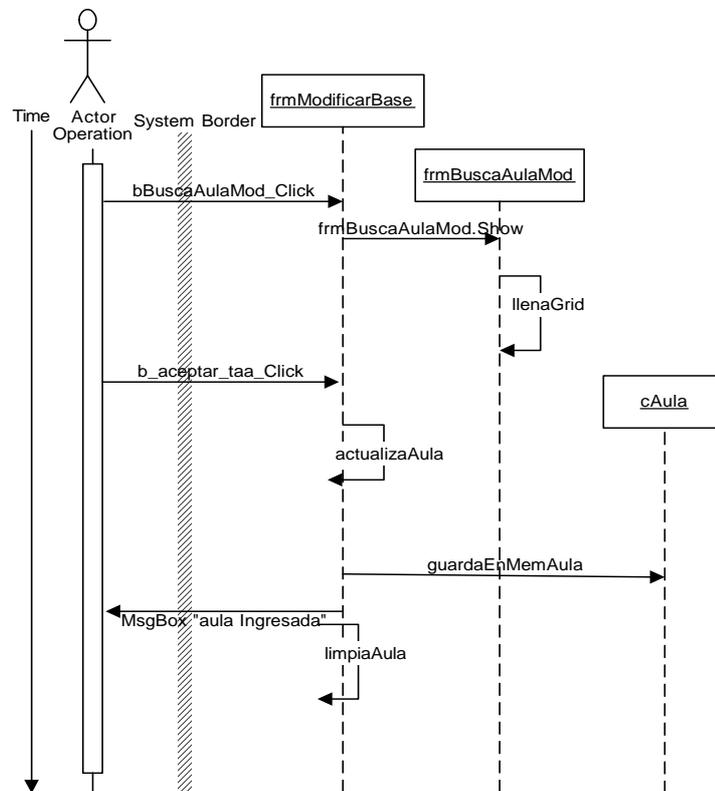


Ilustración 4-158: DIO 18.1: Modificación de Aula Exitoso.

Escenario 19.1: Eliminación de Aula Exitoso

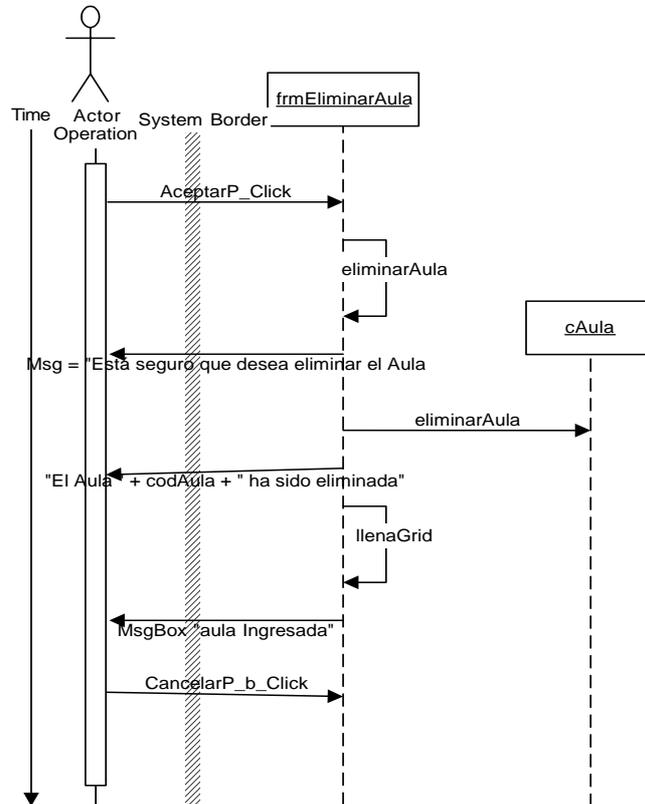


Ilustración 4-159: DIO 19.1: Eliminación de Aula Exitoso.

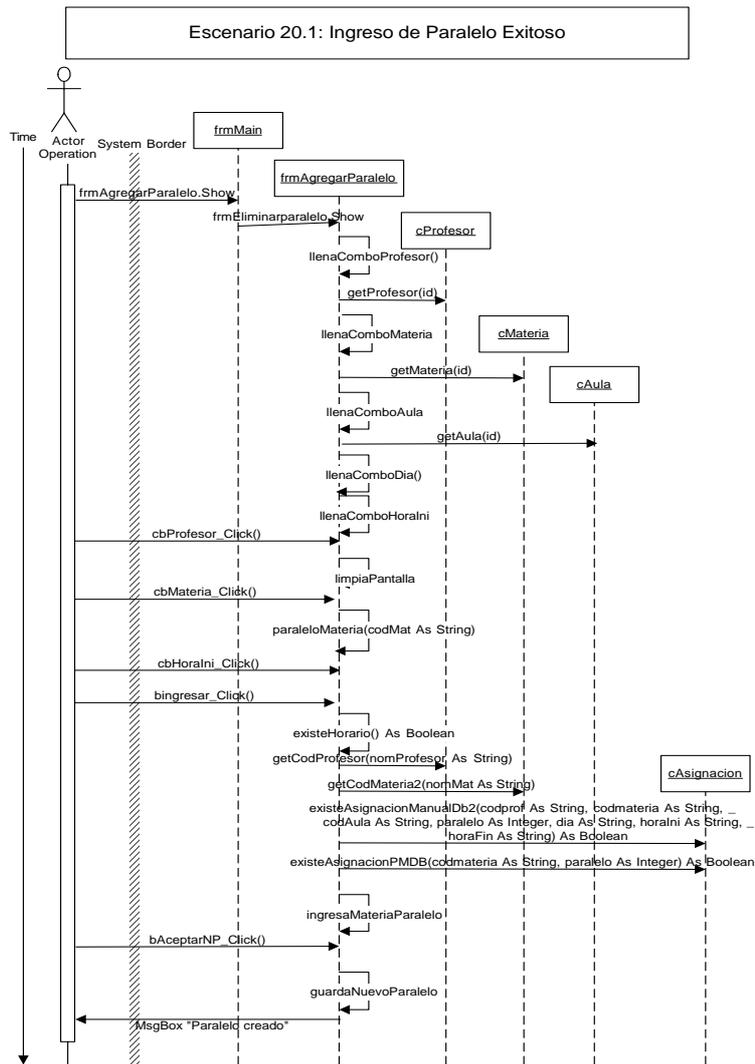


Ilustración 4-160: DIO 20.1: Ingreso de Paralelo Exitoso.

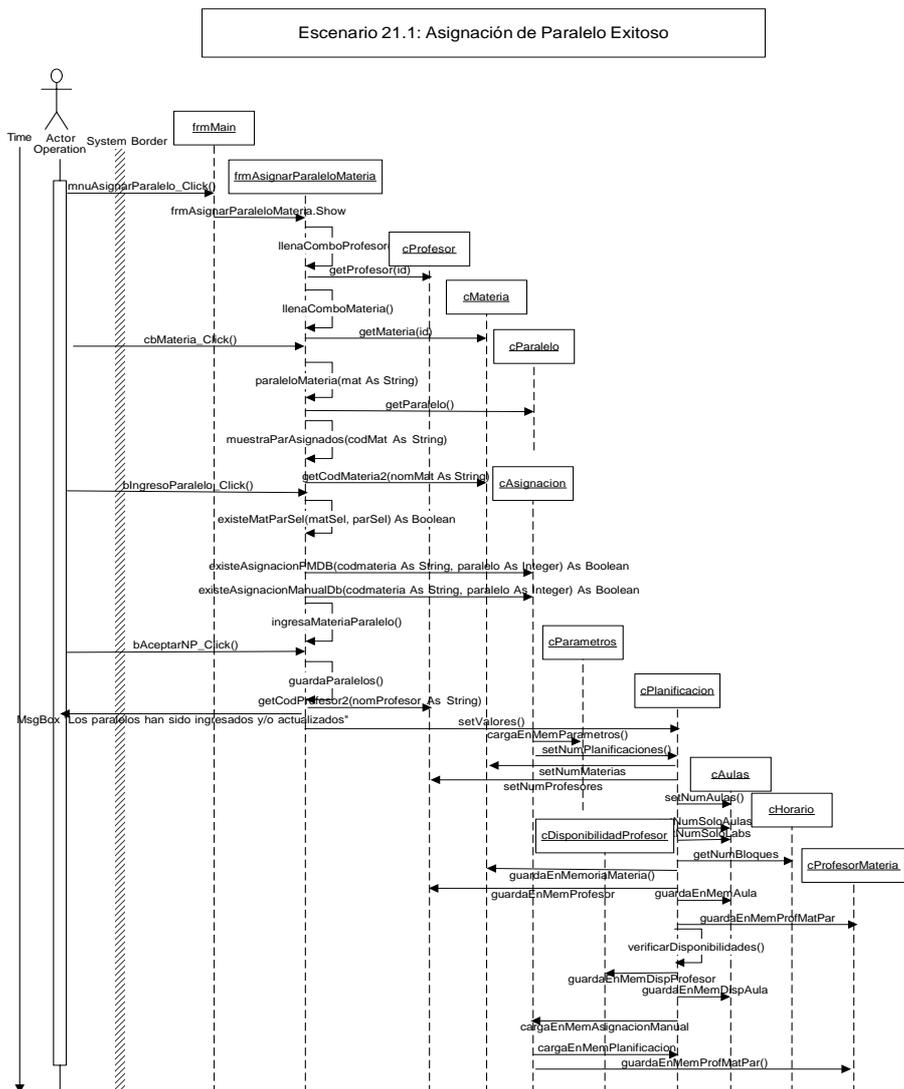


Ilustración 4-161: DIO 21.1: Asignación de Paralelo Exitoso.

Escenario 22.1: Eliminación de Paralelo Exitoso

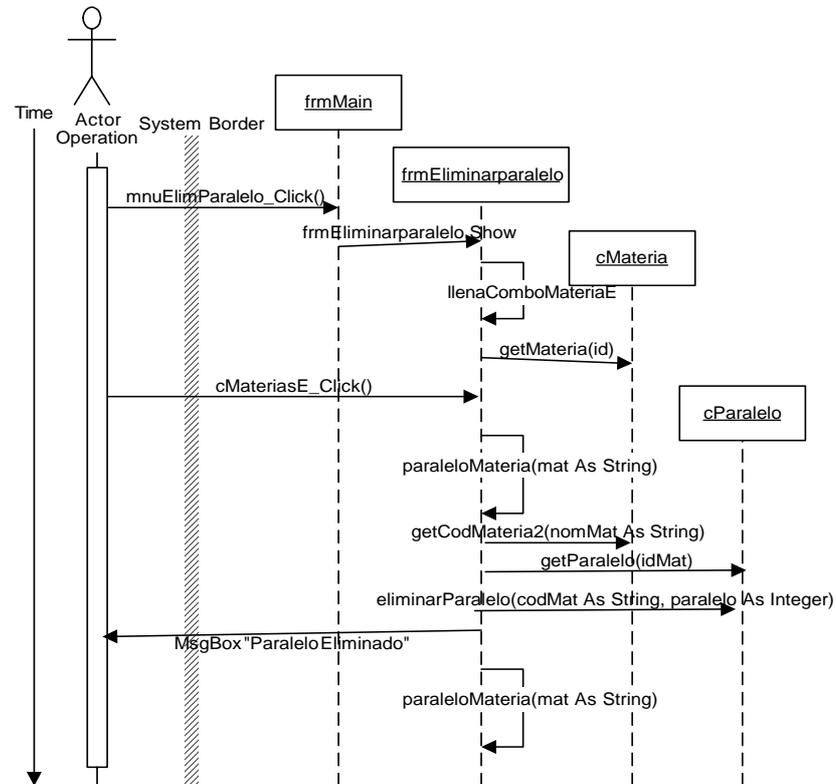


Ilustración 4-162: DIO 22.1: Eliminación de Paralelo Exitoso.

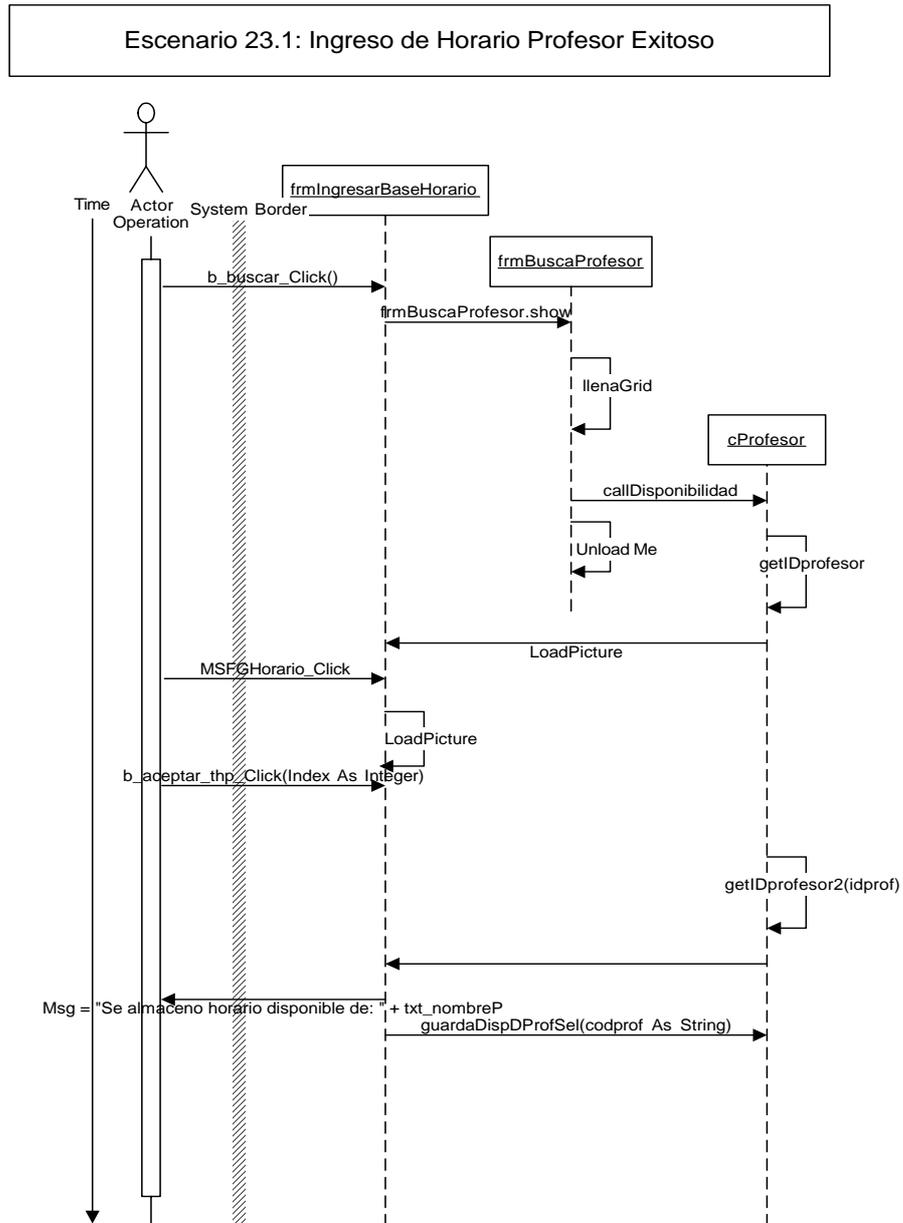


Ilustración 4-163: DIO 23.1: Ingreso de Horario Profesor Exitoso.

Escenario 24.1: Ingreso de Horario Aula Exitoso

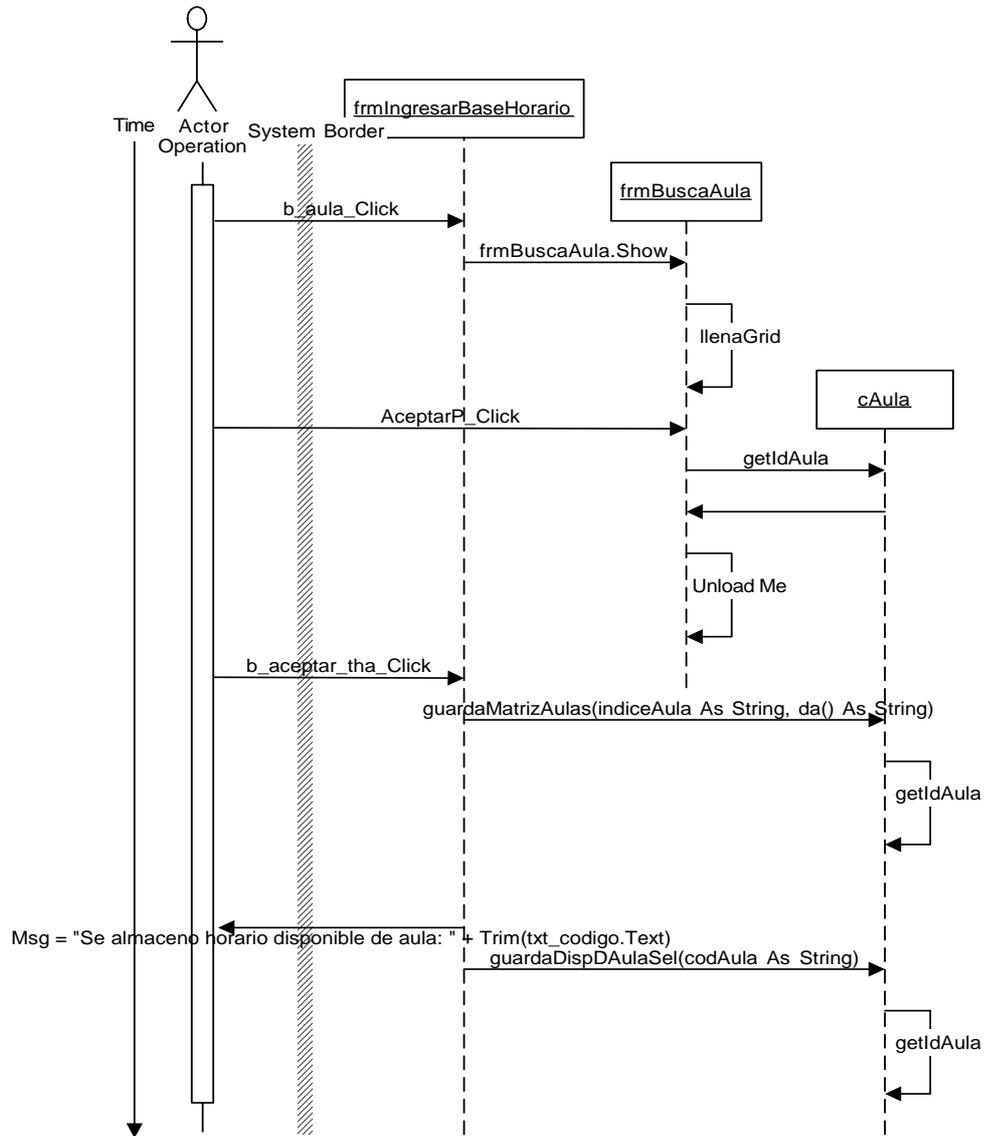


Ilustración 4-164: DIO 24.1: Ingreso de Horario Aula Exitoso.

Escenario 25.1: Obtener Lista de Profesores Exitosa

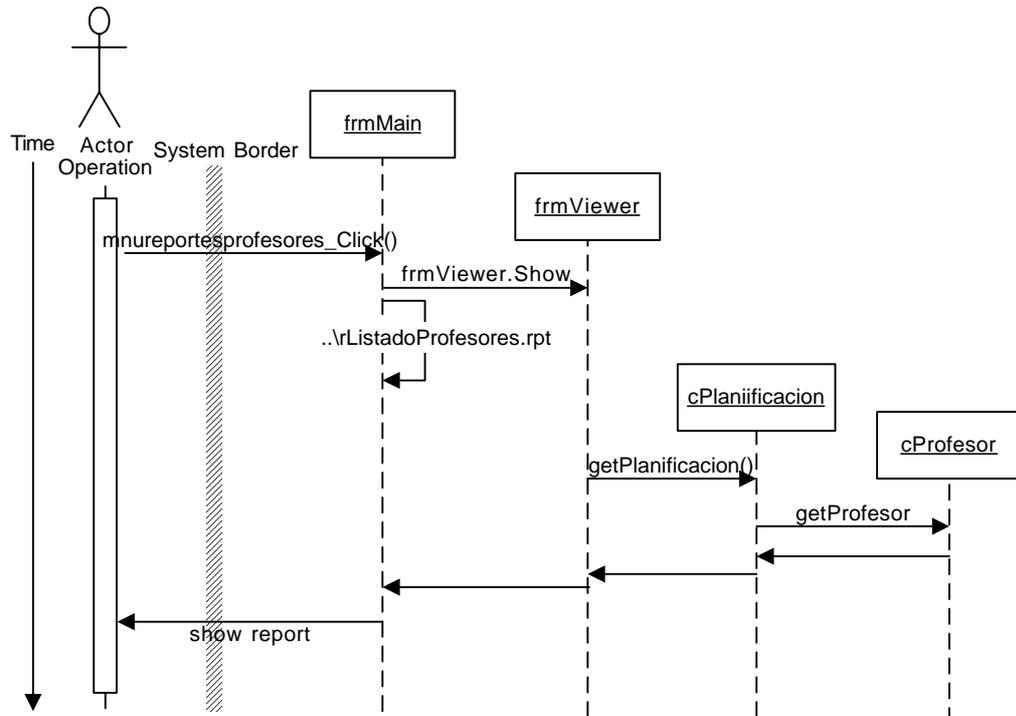


Ilustración 4-165: DIO 25.1: Obtener Lista de Profesores Exitosa.

Escenario 26.1: Obtener Lista de Materias Exitosa

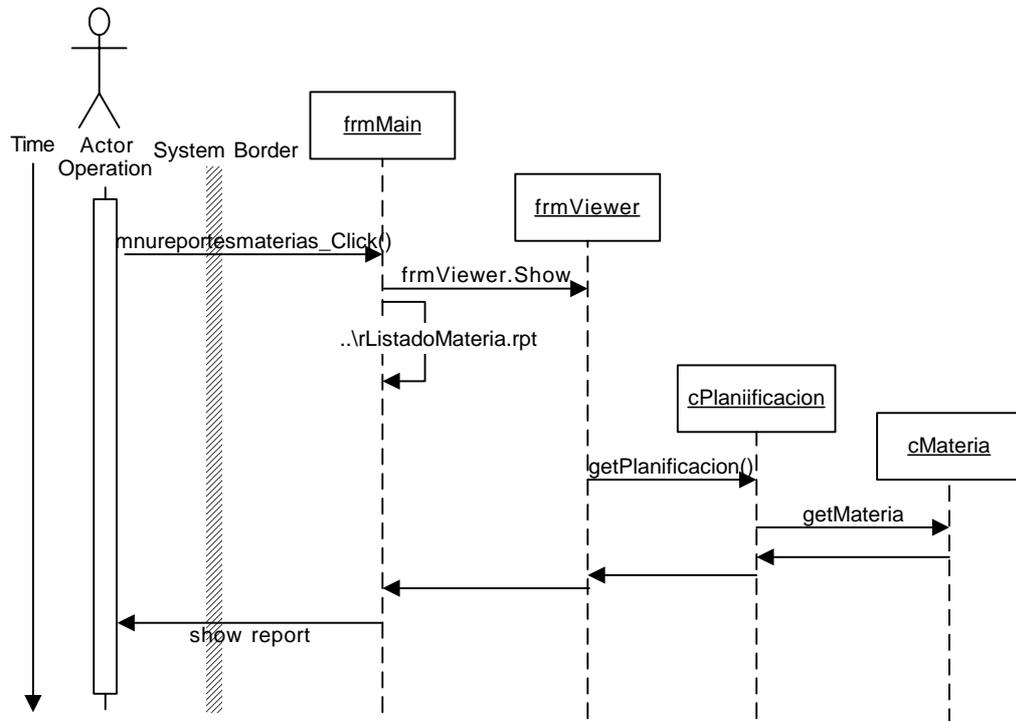


Ilustración 4-166: DIO 26.1: Obtener Lista de Materias Exitosa.

Escenario 27.1: Obtener Lista de Aulas Exitosa

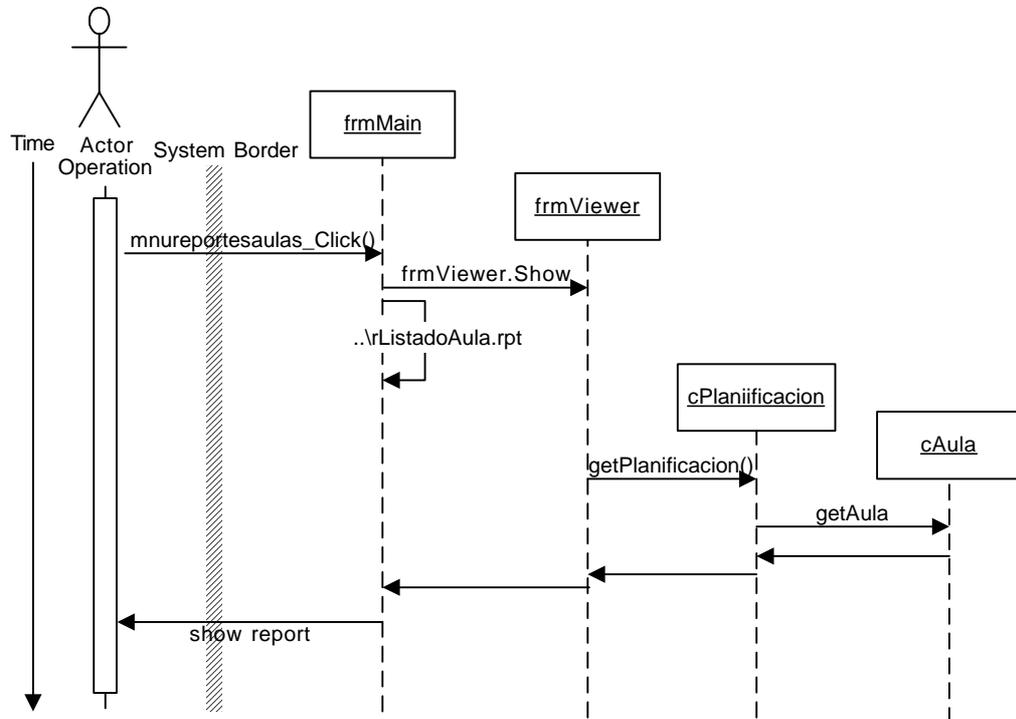


Ilustración 4-167: DIO 27.1: Obtener Lista de Aulas Exitosa.

Escenario 28.1: Obtener Horarios por Aula Exitoso

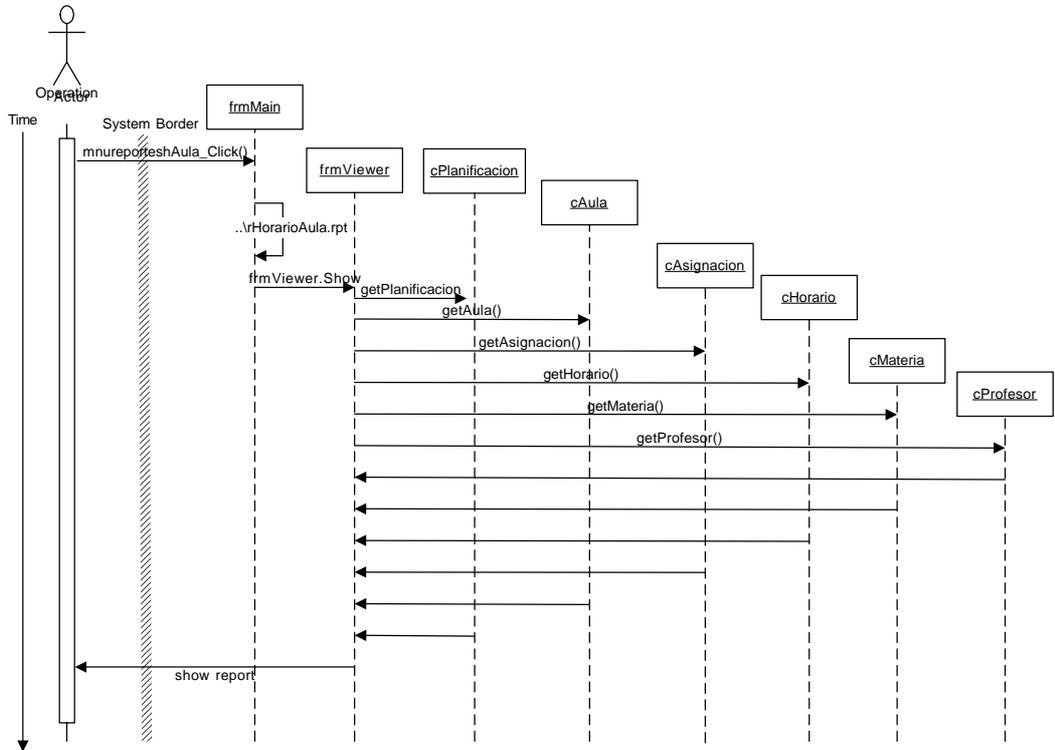


Ilustración 4-168: DIO 28.1: Obtener Horarios por Aula Exitoso.

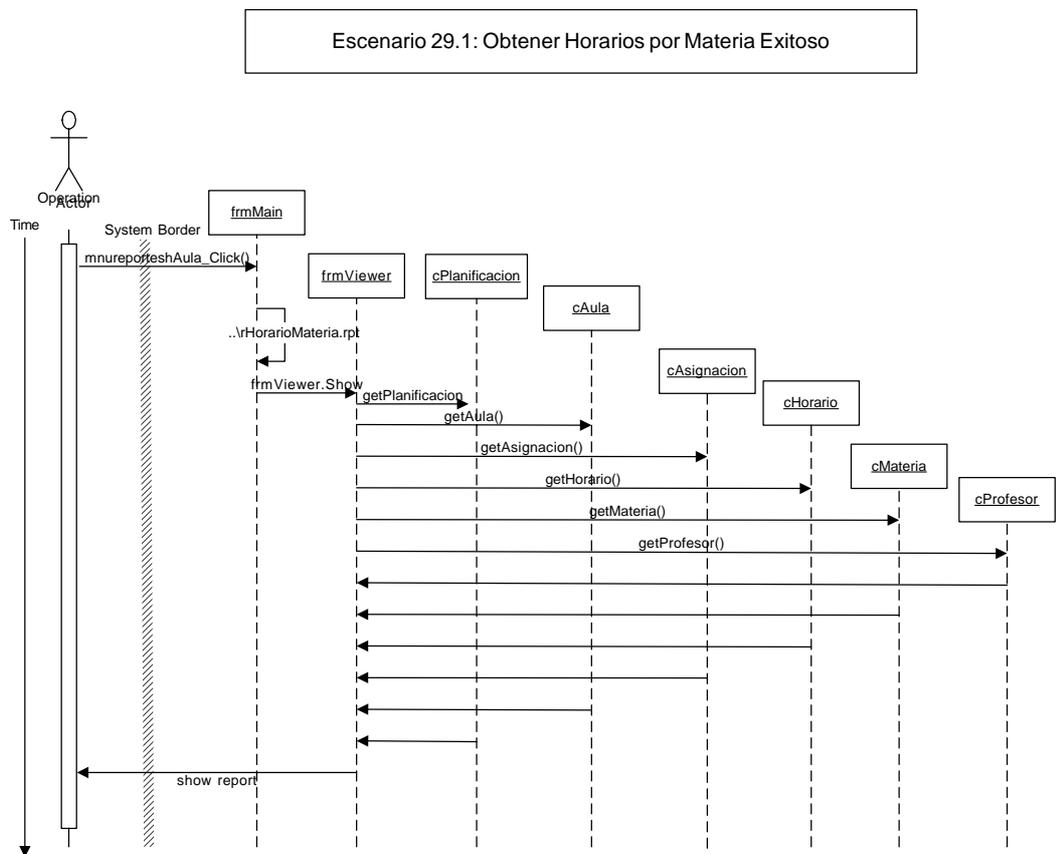


Ilustración 4-169: DIO 29.1: Obtener Horarios por Materia Exitoso.

Escenario 30.1: Obtener Horarios por Profesor Exitoso

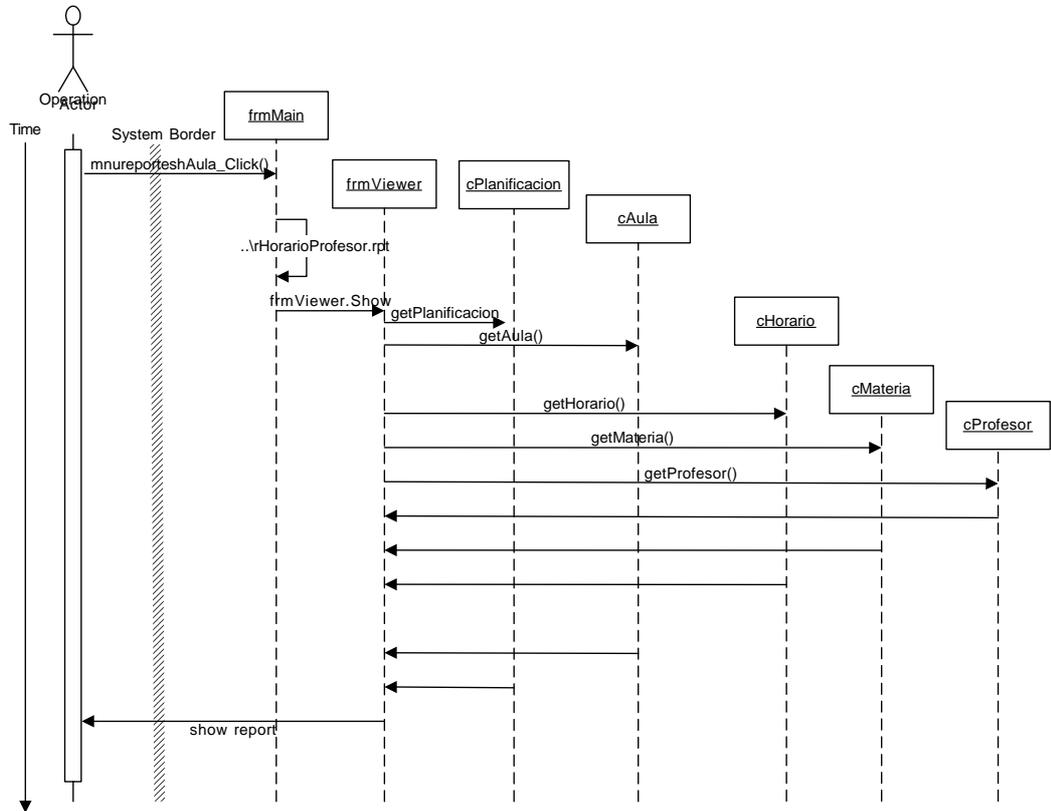


Ilustración 4-170: DIO 30.1: Obtener Horarios por Profesor Exitoso.

Escenario 31.1: Obtener Reporte de Conflictos Exitoso

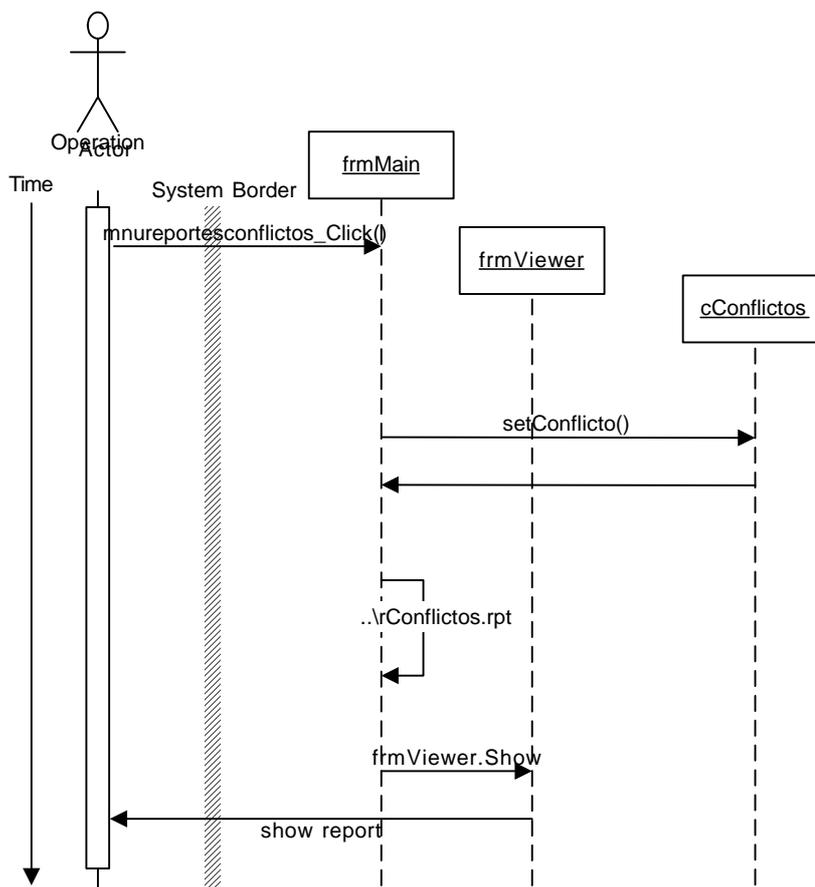


Ilustración 4-171: DIO 31.1: Reporte de Conflicto Exitoso

Escenario 32.1: Ingreso de Parámetros Iniciales Exitoso

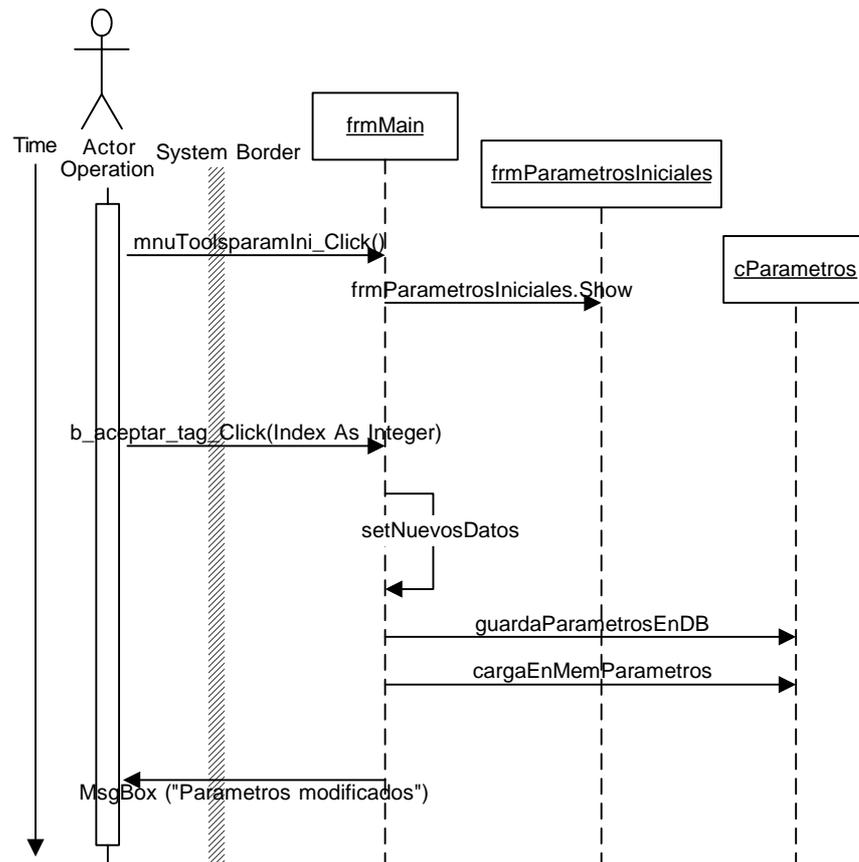


Ilustración 4-172: DIO 32.1: Ingreso de Parámetros Iniciales Exitoso.

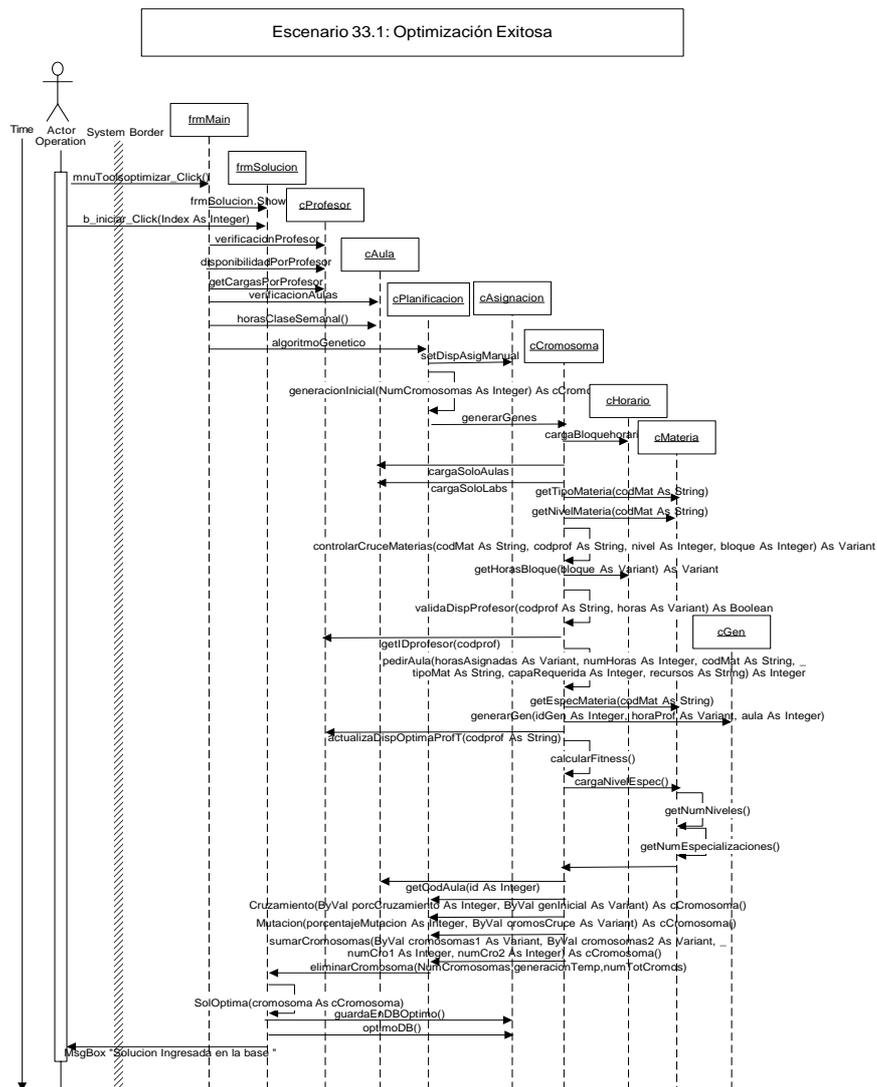


Ilustración 4-173: DIO 33.1: Optimización Exitosa.

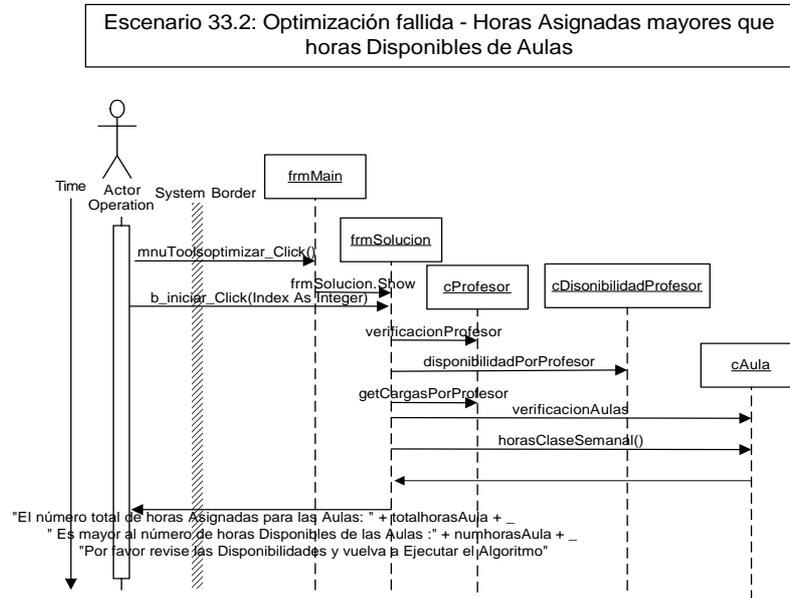


Ilustración 4-174: DIO 33.2: Optimización fallida - Horas Asignadas mayores que horas Disponibles de Aulas.

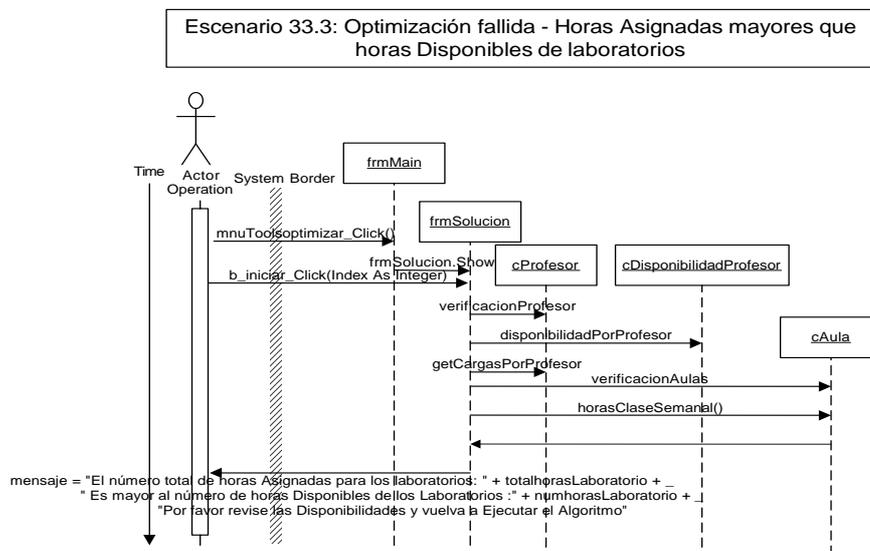


Ilustración 4-175: DIO 33.3: Optimización fallida - Horas Asignadas mayores que horas Disponibles de laboratorios.

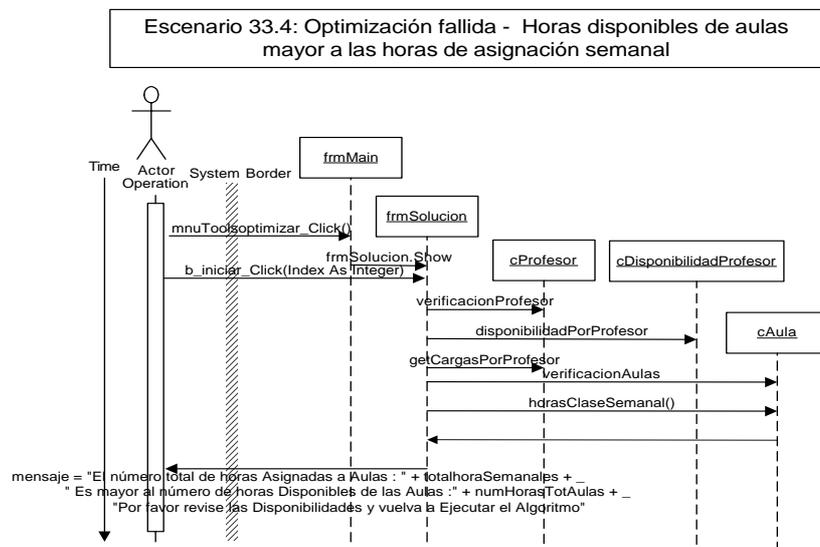


Ilustración 4-176: DIO 33.4: Optimización fallida - Horas disponibles de aulas mayor a las horas de asignación semanal.

Escenario 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad

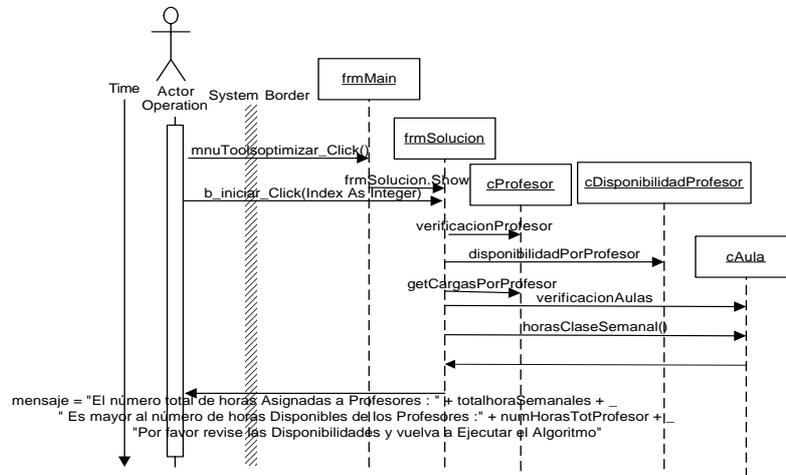


Ilustración 4-177: DIO 33.5: Optimización fallida - Horas asignadas a profesores es mayor que sus disponibilidad.

CAPÍTULO 5

5. DEFINICIÓN DEL PRODUCTO

5.1 ANÁLISIS FODA

El término FODA es una sigla conformada por las primeras letras de las palabras Fortalezas, Oportunidades, Debilidades y Amenazas (en ingles SWOT: Strenghts, Weaknesses, Oportunities, Threats). ^[XXXIII] Siendo las fortalezas y debilidades variables internas de la organización, por ende manipulables, en cambio las oportunidades y amenazas variables externas de difícil transformación.

Las fortalezas u oportunidades especiales con las que contaría el sistema que lo pondría en una situación privilegiada frente a otros serían:

- Beneficio de poder obtener múltiples soluciones
- Al final de una corrida exhaustiva se puede obtener una solución con el menor costo posible.
- Ahorro del costo operativo producto de las operaciones manuales y repetitivas.
- Eliminación de cruces de horarios.
- Reducción del tiempo de ocio de los recursos y mejor distribución de los recursos limitados.
- Manejo de información histórica de las planificaciones para términos posteriores.
- Interfaz intuitiva y amigable.
- Valores óptimos en los porcentajes de mutación y cruzamiento, considerados como valores por defecto como entradas iniciales del algoritmo genético, resultado de evaluaciones en el comportamiento de los datos (refiérase a los resultados obtenidos en el capítulo sexto).
- Creación de usuarios y privilegios administrativos.
- Visualización de reportes personalizados para profesores, alumnos y aulas.
- Reporte de conflictos.

Las oportunidades se determinan para los siguientes aspectos:

- Una interesante ventaja competitiva es el uso de algoritmos genéticos para la resolución del problema, encontrando múltiples soluciones cada una ponderada con un costo asociado que muestra el nivel de confiabilidad y la perfecta clasificación y asignación de la planificación de horarios.
- Al ser una implementación innovadora en muchas instituciones puede dar como resultado una predisposición positiva para quienes lo utilicen, asociado con una mejora en su rendimiento laboral y la efectividad de los resultados obtenidos.

Algunas debilidades aún latentes que se muestran en el sistema son las siguientes:

- El tiempo de respuesta es un resultado proporcional a la cantidad de datos a evaluar y procesar, las combinaciones profesor materia y número de horas por materia serán quienes determinen el tamaño de los cromosomas que implicara a su vez mayor tiempo de proceso.

- Accesos desde el sitio web de la FIEC de la interfaz de planificación, así como de los reportes, pueden ser considerados para nuevas versiones.
- Se simula un sistema orientado a objetos, siendo una propiedad de la que carece el lenguaje de programación utilizado, sin embargo se presenta una perfecta encapsulación de los datos, de sus propiedades y funciones.

Entre las amenazas del entorno que pueden acotarse se encuentran:

- Negativa a su uso o un uso infrecuente.
- Poco conocimiento del manejo del sistema y de las bondades que ofrece por parte del coordinador académico.
- Ingreso de datos basura o equivocaciones en el ingreso de aulas, laboratorios u horarios de disponibilidad de profesores que provoquen asignaciones erróneas.

Una conclusión importante a este análisis es convertir las amenazas en oportunidades y las debilidades en fortalezas, para asegurar la existencia y el uso efectivo de la herramienta.

En lo que respecta a las amenazas es de manera imprescindible que se realice una capacitación previa a las personas que utilizaran el sistema, organizar sesiones de evaluación y analizar los niveles de satisfacción obtenidos, esto asegurará el definitivo uso de la herramienta y la explotación de todas sus capacidades. Las depuraciones a la base de datos son también de manera fundamentales para eliminar los datos basura, inspecciones en los datos de entrada y resultados para asegurar correctas asignaciones.

Siendo esta una herramienta de tipo administrativo se ha considerado que esta es una aplicación de escritorio, sin embargo una buena alternativa sería presentar los reportes en un sitio web a manera de información para estudiantes y profesores, dejando a su vez plena libertad y movilidad al coordinador de la planificación para lograr accesos usando servicios HTTP (Hyper Text Transport Protocol).

Por el lado del tiempo de respuesta se ha considerado realizar optimizaciones en los procesos de búsquedas heurísticas, creación de generaciones iniciales, mutaciones y cruzamiento con la finalidad de optimizar los tiempos de respuesta del sistema, que en un proceso inicial llegó a tomar siete horas ha logrado optimizarse hasta

obtener un tiempo de procesamiento promedio de 1.68 minutos con un cromosoma de dimensión 912, una sola generación y un total de 10 cromosomas por generación, con valores aleatorios en los porcentajes de cruce y mutación, según se muestra en la tabla 8-1.

Run	% Cruce	% Mutación	Tiempo de Ejecución
Average:	54,95313891	55,39743792	1,68
Std.Dev:	18,33594498	19,28160735	0,41
Std.Error:	1,296547104	1,363415531	0,03

Tabla 5-1: Simulación considerando una generación con diez cromosomas

5.2 SEGMENTACIÓN DE MERCADO

En lo que respecta a la delimitación del mercado y los tipos de consumidores a los que esta enfocado el sistema podemos identificar claramente los siguientes nichos: unidades educativas (a manera conjunta jardín, escuela, colegio y universidad), escuelas, colegios, universidades, academias (idiomas, computación, carreras cortas), institutos técnicos.

El sistema ha sido planteado desde el marco más elaborado o de mayor detalle, requerimientos, y variedad de recursos que son las entidades educativas de tercer nivel, lo que permite crear fácilmente una variedad distinta del mismo producto para cada mercado.

Una alternativa para las entidades educativas de tercer nivel será implementación de un sistema central que permita hacer planificaciones inter-facultades que almacenen la información en un repositorio común o tener la libertad de realizar planificaciones independientes.

5.3 COSTO DE LA INVERSIÓN

5.3.1 COSTO DEL LICENCIAMIENTO

5.3.1.1 LICENCIA DEL SISTEMA

En términos de costeo, se calcula el número de horas hombre invertidas en el proyecto que en promedio sería un total de 900 horas (valor que hace referencia a las 25 horas semanales o 100 horas mensuales invertidas por tres programadores en tres meses de implementación) sin incluir el tiempo que toma la selección heurística, la representación del algoritmo y la recopilación de datos.

Se definen el número de horas por programador senior (con experiencia en la herramienta y el área de aplicación) y el número de horas por programador junior (analista programador) multiplicado por el costo de hora laboral para cada uno de ellos.²

² Cortesía del Ing. Guido Caicedo, catedrático y coordinador de la carrera Ingeniería en Computación de la ESPOL

Se adhieren también otros costos como consumo de energía eléctrica, costo por uso de los equipos y software, tiempo requerido para la puesta a punto, instalación y entrenamiento del personal que utilizara el sistema que para nuestro caso de ejemplo será el administrador y un asistente. La sumatoria de estos valores obtenidos representa el costo de licenciamiento del sistema.

	NÚMERO DE HORAS	COSTO POR HORA	TOTAL
Horas hombre programador senior	300	\$ 5,62	\$ 1.686,00
Horas hombre programador junior	600	\$ 2,80	\$ 1.680,00
Consumo de Energía Eléctrica	900	\$ 0,90	\$ 810,00
Costo de uso de equipos	900	\$ 0,70	\$ 630,00
Costo por uso de software	900	\$ 0,50	\$ 450,00
Instalación y Entrenamiento del Personal que hará uso de la herramienta	2	\$ 50,00	\$ 100,00
Factor de Calidad del Software- Algoritmo Genético Representación, Selección Heurística, Recopilación de Datos	100	\$ 10,00	\$ 1.000,00
Precio de Venta			\$ 6.356,00

Tabla 5-2: Costo de Licenciamiento del Sistema

5.3.1.2 LICENCIAMIENTO DE LA BASE DE DATOS

En lo que respecta al licenciamiento de la base de datos a utilizar, en este caso SQL Server, los costos en el esquema de licenciamiento abierto educativo y de sus accesos por usuario serian los siguientes:

Part number	Descripción	Costo Mercado
228-00883	SQL Server 2000 Standard Edition Spanish Open License Program Academic Edition	\$390.0
359-01211	SQL Server CAL 2000 Spanish OLP NL AE Device or User CAL	\$ 42.5
	Medio de Instalación	\$ 25.0
	Total	\$500*

Tabla 5-3 : Costos de Licenciamiento Base de Datos Programa de Licenciamiento Académico Abierto

***Considerando un total de dos accesos**

5.3.1.3 LICENCIAMIENTO SISTEMA OPERATIVO DE SERVIDOR

Las alternativas para licenciar el sistema operativo incluyen las siguientes:

Las primera alternativa es utilizar el paquete de licenciamiento SBS que incluye todos los componentes incluido el de la base de datos tales como: Share Point Server, ISA Server, Windows 2003 Server, Herramientas de Configuración, SQL Server, 5 CAL. Los costos incurridos serian los siguientes:

Part number	Descripción	Costo Mercado
T75-00231	Windows Small Business Server Premium 2003 Spanish Open License Program NL 5 Clients	\$1587
	Medio de Instalación	\$25
	Total	\$1612

Tabla 5-4: Costos de Licenciamiento SBS Server Programa de Licenciamiento Comercial³

Sin embargo, debido a la no existencia del paquete Small Business Server en licenciamiento académico, debe utilizarse esta alternativa pero en licenciamiento abierto para negocios.

La segunda alternativa, mucho más económica que debe ir sumada al costo de la base de datos sería:

Part number	Descripción	Costo Mercado
P73-00312	Windows Svr Std 2003 Spanish OLP NL AE	\$147.5
R18-00592	Windows Server CAL Spanish SA OLP NL AE Device CAL	\$5
	Medio de Instalación	\$25
	Total	\$167.5*

Tabla 5-5: Costos de Licenciamiento Windows Server 2003 Programa de licenciamiento Académico. *Considerando un total de dos accesos⁴

³ Fuente: Precios al 1ero de agosto del 2005 Compumarket S.A. Mayorista Autorizado de Microsoft para el Ecuador.

⁴ Fuente: Precios al 1ero de agosto del 2005 Compumarket S.A. Mayorista Autorizado de Microsoft para el Ecuador.

Una tercera alternativa para nada despreciable sería simplemente el uso de un sistema operativo para estaciones de trabajo, lo que involucraría los siguientes costos, igualmente sumado al costo de la base de datos:

Part number	Descripción	Costo Mercado
E85-00539	Windows XP Professional Spanish UPG OLP NL AE	\$83.75
	Medio de Instalación	\$25
	Total	\$108.75

Tabla 5-6: Costos de Licenciamiento Windows XP Professional Programa de Licenciamiento Académico⁵

5.3.2 REQUERIMIENTOS MÍNIMOS DEL SISTEMA

COSTO DE EQUIPOS

La configuración recomendada del sistema para asegurar un tiempo mínimo de procesamiento requeriría de las características siguientes respecto al hardware:

⁵ Fuente: Precios al 1ero de agosto del 2005 Compumarket S.A. Mayorista Autorizado de Microsoft para el Ecuador

Cpu Intel Pentium IV 3.0Ghz LGA775 1Mb 800Mhz BOX Mb 775S61 P4 LGA 775 (vid/son/lan/mdm) Memoria 512Mb PC-3200 Disco Duro 160Gb 7200 RPM Cd-Rom 52X Interno Drive Disketera 1.44Mb 3.5Pulg. Mon 15" 0.28mm 1024x768	\$ 780,00
---	------------------

Tabla 5-7: Requerimientos Mínicos de Hardware⁶

5.4 COSTOS RELACIONADOS

5.4.1 RETORNO SOBRE LA INVERSIÓN

El cálculo del ROI (Retoro sobre la inversión) se basa en dividir el VNP de los beneficios entre el valor presente de la inversión y tomar el valor absoluto.

Licenciamiento Base de Datos	\$ 500,00
Licenciamiento SO Usuarios x2	\$ 217,44
Licenciamiento Sistema	\$ 6.356,00
Equipos x2	\$ 1.560,00
Total	\$ 7.073,44

Tabla 5-8: Total de Costos Incurridos o TCO (Total Cost Ownership)

⁶ Fuente: Precios al 1ero de agosto del 2005 Compumarket S.A. Mayorista Autorizado de Hardware y Software

Costo Mano de Obra con el uso de la Herramienta

	Horas de Labor por semestre	Costo por Hora	Total Semestre	Total Anual (3 planificaciones)
Planificador de Horario	5	30	\$ 150,00	\$ 450,00
Asistente	7	17,5	\$ 122,50	\$ 367,50
Total			\$ 272,50	\$ 817,50

Tabla 5-9: Costo Mano de Obra con el uso de la Herramienta

ROI = Dinero Ahorrado /Dinero Invertido

	Primer Año	Segundo Año
Dinero Ahorrado	\$ 3.509,06	\$ 10.582,50
Dinero Invertido	\$ 7.073,44	\$ 7.073,44
ROI	0,496089597	1,496089597

Tabla 5-10: ROI: Retorno Sobre la Inversión

Costo Actual de la Mano de Obra

	Sueldo Mensual	Sueldo Anual
Planificador de Horario	\$ 600,00	\$ 7.200,00
Asistente	\$ 350,00	\$ 4.200,00
Total		\$ 11.400,00

Tabla 5-11: Costo Actual de la Mano de Obra

El periodo de recuperación óptimo para proyectos pequeños de informática es comúnmente aceptado como menor a 12 meses. Sin embargo, un periodo de recuperación menor a 24 meses es aceptable para un proyecto grande de informática, dado el ciclo de tiempo necesario para actualizar los proyectos de alta tecnología ^[XXXIV]

CAPÍTULO 6

6. ANÁLISIS DE RESULTADOS

6.1. PLAN DE PRUEBAS

Las pruebas realizadas con el planificador de horarios se hicieron durante tres sesiones que permitieron ajustar la aplicación a diversos puntos clave que pudieron ser obviados en la recolección de información.

El primer plan de pruebas fue realizado con un prototipo, permitiendo definir de una forma muy general la interfaz de interacción con el usuario y la forma en que se presentarían los reportes.

En una siguiente etapa de evaluación con el planificador y con una aplicación operativa en un 75%, se lograron determinar

modificaciones, ajustes y adicionales que debía manejar la aplicación, entre ellos:

- Reubicación de pantallas
- Creación de pantallas de verificación de acceso.
- Permisos de acceso a las tablas de la base de datos.
- Creación de diferentes planificaciones y respaldo de las mismas.
- Modificación de la carga horaria diaria mínima asignada por paralelos.
- Permitir el ingreso de asignaciones de paralelos previo al proceso de optimización para casos excepcionales.
- Forzar a que para un mismo paralelo, la materia sea dictada en la misma aula y en el mismo horario preferiblemente pasando un día.
- Ingreso de nuevos paralelos posterior a la optimización, sin modificar la planificación actual.

En una tercera sesión con una aplicación operativa al 100% se explicó la completa funcionalidad del sistema, los reportes obtenidos, las diferentes alternativas para la colocación de parámetros para el

correcto desempeño del algoritmo y la administración del sistema y la base de datos.

6.2. ANÁLISIS DE RESULTADOS OBTENIDOS

Se logró que la mayor parte de las materias de un mismo nivel se dicten en horarios diferentes de manera que los estudiantes pueden tomar las materias del mismo nivel con un bajo porcentaje de cruces.

La probabilidad que todas las materias de un mismo nivel no se crucen no es del 100%, esto debido a que existe más de un paralelo por materia.

Se logro mantener el modelo tradicional de asignación de horarios en el que una paralelo independientemente de la materia que se este dictando tenga asociado un mismo horario de clases y una misma aula en días diferentes.

6.3 MEJORAS AL DESEMPEÑO

El algoritmo genético obtenido puede ganar velocidad con más procesadores o implementando hilos, puesto que todo el trabajo está

equitativamente repartido en las operaciones de cruzamiento, mutación y evaluación de fitness.

En las pruebas realizadas se practicó un aumento del ritmo de mutación, los resultados obtenidos muestran que es preferible evitar en el problema la pérdida de diversidad genética en una población de cromosomas, o bien introducir mecanismos en los cuales el fitness de un individuo se divida por el número de individuos similares a él.

6.4 CONVERGENCIA DE RESULTADOS.

Según los resultados de las pruebas, para obtener mejores soluciones se debe aumentar los porcentajes de mutación y cruzamiento, y aumentar el número de generaciones, pero debemos tener en cuenta que al realizar estos cambios, al algoritmo le tomará más tiempo encontrar una solución factible

Al generar una población de soluciones y elegir al mejor cromosoma (trátase de una mejor planificación), si bien es cierto no se llega a una planificación óptima, pero se obtiene una planificación con asignaciones eficiente.

Para la realización de las pruebas se deben ingresar los siguientes parámetros:

- Número de generaciones
- Cromosomas por Generación
- Porcentaje de Cruzamiento
- Porcentaje de Mutación

Y para la medición de la calidad del cromosoma se van a medir los siguientes parámetros:

Fitness.- valor que se le da al cromosoma que depende del número de violaciones a los requerimientos, el mejor cromosoma es el que tenga menor fitness

Tiempo de ejecución.- es el tiempo que tarda en correr la aplicación en búsqueda del mejor cromosoma.

Porcentaje de Cruzamiento por Nivel.- Es el número de horas que se cruzan dividido para el número total de horas por nivel que se dictan de clases.

Porcentaje de Cruzamiento por Nivel-Especialización Nivel.- Es el número de horas que se cruzan dividido para el número total de horas por especialización y luego el promedio por nivel-especialización de horas de clases que se dictan.

Las pruebas se realizaron en un computador con las siguientes características:

- Procesador Pentium 4 de 16. GHz
- 512 MB en memoria RAM
- Disco duro de 40 GB
- Windows 2000 Profesional

Para realizar las pruebas del algoritmo, se asume que todos los profesores están disponibles a toda hora, así como también las aulas, en el horario establecido de clases, de lunes a viernes de 7:30 a 21:30 y los sábados de 7:30 a 13:30.

Observemos los resultados de las pruebas:

Teniendo los siguientes ingresos:

Número de Generaciones	:	1
Cromosomas por generación	:	1
Porcentaje de Cruzamiento	:	0 %
Porcentaje de Mutación	:	0 %

Se generaron los siguientes resultados:

Fitness	:	393.6
Tiempo de ejecución	:	0:00:35 segundos

Porcentaje de Cruzamiento por Nivel	:	34 %
Porc. de Cruzamiento por Nivel-Especialización	:	5 %

Es importante recalcar la diferencia entre el porcentaje de cruzamiento, el porcentaje de cruzamiento por Nivel y el porcentaje de cruzamiento por Nivel-Especialización. El primero es una variable de ingreso que indica cual es porcentaje de cromosomas que se van a cruzar en base al número de cromosomas por generación, los dos siguientes son resultados del algoritmos, los cuales reflejan las violaciones a los requerimientos de que materias del mismo nivel no se dicten en el mismo horario.

Analizando los resultados podemos observar que para para la obtener una planificación de horarios necesitamos 35 segundos, esta planificación tiene un fitness de 393.6, que es la acumulación de puntos que al algoritmo le asigna a la planificación por cada violación a los requerimientos, esta planificación tiene un porcentaje de cruzamiento del 34% por nivel, es decir, la división entre todas las horas que se cruzan en determinadas aulas y la suma de todas las horas que se dictaran en ese nivel. El porcentaje de cruzamiento por Nivel-Especialización obtenido es del 5%, el cual se obtiene de igual forma que el porcentaje de Cruzamiento por Nivel, pero se realiza un

doble promedio, el primero por la especialización de cada nivel y el segundo por todos los niveles existentes en la planificación.

En la siguiente tabla se muestra la variación de los porcentajes de Cruzamiento y de mutación, manteniendo el número de generaciones y el número de cromosomas por generación del resultado anterior.

	Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
1	1	1	0	0	393,6	0:00:35	34	5
2	1	1	50	50	373,6	0:00:26	31	6
3	1	1	50	100	393,6	0:00:33	34	5
4	1	1	100	50	403,6	0:00:25	33	7
5	1	1	100	100	383,6	0:00:27	32	6

Tabla 6-1: Resultado del algoritmo con una generación y un cromosoma

La tabla 6-1 indica las variaciones de los porcentajes de cruzamiento y de mutación (0, 50 y 100) se obtiene que el mejor fitness, el menor, 373.6 se obtiene con un porcentaje de cruzamiento y de mutación de 50%.

Si nos guiamos por el Porcentaje de Cruzamiento por nivel la mejor planificación se obtiene también con un porcentaje de cruzamiento y de mutación de 50%. Si observamos el porcentaje de cruzamiento Nivel-Especialización la mejor planificación es la que tiene un porcentaje de cruzamiento y de mutación de 0%.

Al incrementar los porcentajes de cruzamiento y de mutación la carga para el algoritmo aumenta, debido a que tendrá que realizar cruzamientos y mutaciones de cromosomas, además de evaluarlos para determinar el fitness de cada cromosoma (planificación), por lo cual deberá tomarse un mayor tiempo de ejecución, pero como observamos en la tabla 6-1 en la segunda planificación se obtiene un tiempo de 26 segundos. Esto tiene una explicación, el tiempo de respuesta depende también de las condenaciones que se apliquen al cromosoma, si una planificación tiene un fitness mayor al de otra planificación es por que la de mayor fitness ha ingresado a más funciones de condenación debido a que existen más violaciones a los requerimientos en ella, esto sucede cuando el número de generaciones y de cromosomas es pequeño, ya que cuando tenemos una mayor cantidad de generaciones y de cromosomas, el ingreso a funciones de condenación es despreciable comparado con el procesamiento de cruces y mutaciones, esto se apreciará en resultados posteriores.

	Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
1	1	10	0	0	373,6	0:01:39	31	6
2	1	10	50	50	373,6	0:02:50	32	5
3	1	10	50	100	373,6	0:02:47	32	5
4	1	10	100	50	363,6	0:03:47	30	6
5	1	10	100	100	353,6	0:05:16	30	5

Tabla 6-2: Resultado del algoritmo con una generación y diez cromosomas

En la Tabla 6-2 observamos que a medida que aumentamos los porcentajes de cruzamiento y de mutación los tiempos de ejecución van aumentando como es de esperarse y el tiempo de ingreso por las condenaciones se vuelve despreciable.

A partir de los resultados obtenidos en la tabla 6-2, podemos eliminar el tiempo de ejecución como una variable para determinar el mejor cromosoma, debido a que el tiempo se incrementa cuando aumentamos en porcentaje de cruzamiento y de mutación.

También observamos en la tabla 6-2 que la solución que presenta el mejor fitness también presenta el mejor porcentaje de cruzamiento por Nivel y menor porcentaje de cruzamiento por Nivel-especialización. Hasta ahora el mejor porcentaje de cruzamiento por Nivel es del 30 %.

	Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
1	1	20	0	0	353,6	0:03:46	30	5
2	1	20	50	50	353,6	0:05:54	30	5
3	1	20	50	100	363,6	0:06:02	31	5
4	1	20	100	50	333,6	0:07:33	28	5
5	1	20	100	100	363,6	0:09:12	31	5

Tabla 6-3: Resultado del algoritmo con una generación y veinte cromosomas

Para obtener los resultados de la tabla 6-3 se ingresó 20 en el número de cromosomas por generación, dando como resultado un cromosoma con un porcentaje de cruzamiento por nivel de 28, el mismo cromosoma (plañificación o solución), presenta un fitness de 333.6 el menor de la tabla 6-3 y el menor de todos resultados obtenidos, observamos también que el porcentaje de cruzamiento por Nivel-Especialización es de 5 para todas las combinaciones de la tabla 6-3.

Nótese que la mejor solución no es la que tiene el 100% en los porcentajes de cruzamiento y de mutación, se obtuvo el mejor resultado con un 100% de cruzamiento y 50% de mutación, a diferencia de las tablas anteriormente analizadas.

Hasta ahora el parámetro que no hemos modificado es el número de generaciones, por cada generación se van a generar el número de

cromosomas por generación ingresados, es decir, si ingresamos 4 generaciones con 3 cromosomas por generación, se generarán 12 cromosomas, debido que por cada generación se generará 3 cromosomas, teniendo en cuenta también los porcentajes de cruzamiento y de mutación. Cabe recordar que los cromosomas mutados en realidad son cromosomas nuevos, debido a que para cumplir con los requerimientos se tiene que hacer una reestructuración del cromosoma, por lo cual la mutación es una mutación a la generación.

	Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
1	5	4	0	0	363,6	0:02:04	31	5
2	5	4	50	50	383,6	0:02:51	32	6
3	5	4	50	100	363,6	0:03:59	31	5
4	5	4	100	50	343,6	0:05:04	29	5
5	5	4	100	100	323,6	0:06:23	28	4

Tabla 6-4: Resultado del algoritmo con cinco generaciones y cuatro cromosomas

En la tabla 6-4 se presenta una solución con un 28 % de porcentaje de cruzamiento por nivel y un 4% de porcentaje de cruzamiento por Nivel-Especialización, además tiene un fitness de 323.6, esto con 100% de cruzamiento y de mutación, todo esto en un tiempo menor al presentado en la tabla 6-3 cuando se tenían 20 cromosomas.

	Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
1	5	20	0	0	333,6	0:05:30	29	4
2	5	20	50	50	353,6	0:13:25	30	5
3	5	20	50	100	333,6	0:18:03	28	5
4	5	20	100	50	353,6	0:27:47	31	4
5	5	20	100	100	340,6	0:32:38	29	4

Tabla 6-5: Resultado del algoritmo con cinco generaciones y veinte cromosomas

En los resultados de la tabla 6-5, observamos que la mejor solución es la que presenta un 28% de cruzamiento por Nivel, con un porcentaje de cruzamiento de 50% y un porcentaje de mutación de 100%. El fitness de 333.6 es menor de la tabla 6-5 pero con un porcentaje de cruzamiento de 31%, esto se debe a que este cromosoma cumple mejor con los requerimientos de recursos de requeridos de aulas con preferencia a determinadas carreras, pero presenta un mayor porcentaje de cruzamiento por nivel en comparación con el cromosoma que obtuvo un 28% que es la tercera combinación de la tabla 6-5.

Pero estos valores no son mejores que los de la tabla 6-4, la cual presenta un cromosoma con un fitness de 323.6, un porcentaje de cruzamiento por Nivel de 28% y un porcentaje de cruzamiento por Nivel-especialización de 4%.

A pesar que para obtener los resultados de la tabla 6-5 aumentamos el número de cromosomas de 4 a 20, no se generó una planificación mejor que la de la tabla 6-4. Esto se debe a que la mejora de la generación de una planificación no depende totalmente del número de generaciones, del número de cromosomas por generación, ni de los porcentajes de cruzamiento y mutación, debido a que el algoritmo se basa en la exploración, es decir la elección aleatoria de una solución dentro del espacio de soluciones.

Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
20	20	100	100	333,6	1:32:06	27	6

Tabla 6-6: Resultado del algoritmo con veinte generaciones y veinte cromosomas

En la tabla 6-6, observamos que obtenemos una planificación con un porcentaje de cruzamiento por nivel del 27%, a pesar que el fitness no es el mejor de todas las planificaciones generadas, esto debido a que el número de generaciones aumentó a 20, con 20 cromosomas por generación, con un porcentaje de cruzamiento y mutación de 100%, esto le permitió al algoritmo explorar mas planificaciones en el espacio de solución. La solución obtenida en la tabla 6-6 se realizó en 1 hora, 32 minutos y 6 segundos.

Para obtener los resultado de la tabla 6-7 se utilizó un computador con un mejor procesador.

Num. de Generaciones	Cromosomas por Generación	Porcentaje de Cruzamiento	Porcentaje de Mutación	Fitness	Tiempo	% de cruz. Nivel	% de cruz. Nivel-Esp
10	100	100	100	323,6	1:27:49	27	5

Tabla 6-7: Resultado del algoritmo con diez generaciones y cien cromosomas

La tabla 6-7 presenta una solución con un 27 % de cruzamiento por nivel y unos 5% de cruzamiento por Nivel-Especialización, los menores porcentajes de cruzamiento obtenidos, esto debido a que el algoritmo ha podido realizar una exploración mayor con lo cual encontró una mejor planificación.

CONCLUSIONES Y RECOMENDACIONES

Los Algoritmos Genéticos no nos podrán siempre ofrecer la mejor solución, pero si nos proporcionan una buena solución aunque no sea la óptima, es posible que nunca encontremos la mejor solución, pero los AG nos proporcionan una solución válida, dentro del porcentaje de efectividad aceptado.

Dentro del grupo de algoritmos analizados como Recocido Simulado y Redes Neuronales, el algoritmo que más se ajustó a la representación del problema fue el Algoritmo Genético.

Los algoritmos genéticos son relativamente fáciles de entender e implementar, y su principal ventaja y desventaja a la vez es su robustez, si se posee una mala implementación, el AG seguirá corriendo y tarde o temprano encontrará

una solución que no represente al problema planteado o encontrará un óptimo local. Pero si se realiza una buena representación que proporcione datos útiles para el algoritmo, tendremos muy buenos resultados con la aplicación de algoritmos genéticos.

Para obtener una planificación automatizada solo se utiliza un recurso, que puede tratarse coordinador de la planificación con privilegios de administrador, inclusive la opción de un digitador puede ser innecesaria luego de una primera ejecución del sistema debido a que en este punto los datos ya fueron ingresados al sistema y de ahora en adelante solo se requiere hacer asignaciones de profesores materias y aulas, labor que puede ser realizada por el administrador de la planificación, disminuyendo en un 70% la cantidad de recursos que previamente se usaban para obtener una planificación manual.

Para la obtención de una solución se requieren en promedio 18.73 segundos en promedio, este resultado promediando la generación del algoritmo con 1, 10 y 20 cromosomas, con una generación, un porcentaje de cruzamiento y de mutación de 0%.

Al incrementar el número de generaciones y el número de cromosomas por generación, incrementa la posibilidad de obtener una mejor solución, esto manteniendo los porcentajes de cruzamiento y de mutación.

Incrementando el porcentaje de cruzamiento y de mutación, aumenta el porcentaje de encontrar una mejor solución, pero no en todos los casos, ya que como observamos en la tabla 6-3, no fue necesario ingresar un porcentaje de cruzamiento y de mutación de 100%, para obtener la mejor solución.

En realidad al incrementar algunos de los parámetros de entrada, lo que estamos haciendo es incrementar el número de búsquedas de soluciones válidas. Debido a que el algoritmo realiza búsquedas aleatorias dentro del espacio de soluciones puede que con determinados parámetros encuentre mejores soluciones que con parámetros mayores, esto se debe a la exploración y explotación del algoritmo en búsqueda de planificaciones en el espacio de soluciones.

APÉNDICES

APÉNDICE A. DICCIONARIO DE DATOS

Variable	Tipo	Clase/Módulo	Descripción
asignacion_manual()	string	cAula	Matriz que almacena las Asignaciones manuales
aulaMem()	integer	cAula	matriz que contiene la información de las aulas para calculos
aulas()	integer	cAula	almacena los codigos de las aulas
capacidad	string	cAula	Capacidad de las aulas
controlaCruce()	string	cAula	Matriz que contiene y lleva el control de cruce de las materias asignadas
DispAula()	integer	cAula	matriz que contiene disponibilidad del aula para los calculos del algoritmo
disponibilidadAulas()	string	cAula	matriz que contiene la disponibilidad semanal de un aula
disponibilidadAulasT()	string	cAula	matriz temporal de la disponibilidad del aula
mSoloAulas()	string	cAula	Matriz que contiene las aulas de la Planificación
mSoloLabs()	string	cAula	Matriz que contiene los laboratorios de la Planificación

numAsignacionManual	integer	cAula	Número de asignaciones manuales
numAulas	integer	cAula	número de aulas de la planificación
numhorasAula	integer	cAula	Número de horas de las aulas
numhorasLaboratorio	integer	cAula	Número de horas de los laboratorios
numHorasTotAulas	integer	cAula	Número total de disponibilidad de todas las aulas
numSoloAulas	integer	cAula	número de aulas de la planificación
numSoloLabs	integer	cAula	Número de laboratorios
tipoAula	integer	cAula	tipo de aula en la que se dictará la materia
totalhorasAula	integer	cAula	Número total de horas asignadas a las aulas
totalhorasLaboratorio	integer	cAula	Número total de horas asignadas a los laboratorios
Conexion	Connection	cConexion	contiene la conexión a la base de datos
Registro	Recordset	cConexion	recordset para consultas a la base de datos
C_status	string	cCromosoma	estatus del cromosoma
fitness	double	cCromosoma	valor que se le asigna a una solución , este es el valor de evaluación de cada solución
gen()	cGen	cCromosoma	contiene el gen de la solución
porcCruceTotNivel	integer	cCromosoma	Porcentaje de cruce por Nivel
porcentajeCruce	integer	cCromosoma	Porcentaje de cruce por Nivel-Especializacion
x	integer	cGen	parte del gen que representa la relación profesor-materia
y	integer	cGen	parte del gen que representa la relación el aula
z	integer	cGen	parte del gen que representa la relación el horario
bloqueHorarios()	string	cHorario	Matriz que contiene los bloques de horarios
horalnicio	string	cHorario	Hora de Inicio

horario()	integer	cHorario	matriz que contiene los horarios
horarioOptimo()	integer	cHorario	matriz que contiene la planificación generada
codigo	integer	cMateria	codigo de la materia
especializacion()	string	cMateria	matriz que almacena las especializaciones
horasDiarias	integer	cMateria	horas máximas por día que se dictará la una materia
horasSemanales	integer	cMateria	total de horas a la semana
materia()	string	cMateria	matriz que contiene las materias
Materias()	integer	cMateria	almacena los codigos de las materias, utilizadas en el algoritmo
nivel	integer	cMateria	nivel de la materia
nombre	string	cMateria	nombre de la materia
numEspec	integer	cMateria	Número de Especializaciones
numEspecializaciones	integer	cMateria	Número de Especializaciones en la base
numMaterias	integer	cMateria	número de materias de la planificación
tipoMateria	integer	cMateria	tipo de materia
numparalelo	integer	cParalelo	Número de paralelo
maxGeneraciones	Integer	cParametros	número máximo de generaciones
NumCromosomas	integer	cParametros	Número de cromosomas
numDiasSemana	integer	cParametros	numero de días de clases a la semana
numHorasClase	integer	cParametros	numero de horas semanales de clase
porcentajeCruzamiento	integer	cParametros	Porcentaje de cruzamiento de la planificación
porcentajeMutacion	integer	cParametros	Porcentaje de mutación de la planificación
tmaxEjecucion	integer	cParametros	Tiempo máximo de ejecución de la aplicación
actualPlanificacion	string	cPlanificacion	variable que contiene el id de la planificacion actual
descPlani	string	cPlanificacion	Descripción de la planificación
idPlani	string	cPlanificacion	Identificador de la planificación

planificacion()	string	cPlanificacion	matriz que contiene las planificaciones
porcentajeCruceNivel	integer	cPlanificacion	Porcentaje de cruce por nivel (total)
porcentajeCruceNivelEsp	integer	cPlanificacion	Porcentaje de cruce por nivel-Especialización
numProfesorMateria	integer	cProfesorMateria	número de relaciones profesor-materia
numProfMatPar	integer	cProfesorMateria	numero profesor materia paralelo (maximo del gen)
profesorMateria	string	cProfesorMateria	Matriz que contiene la relacion profesor-materia que representa en x del gen
profesorMatPar()	string	cProfesorMateria	matriz que contiene las materias
cargaPorProfesor()	string	cProfesor	matriz que contiene la carga por profesor
disponibilidadProfesor()	string	cProfesor	matriz que contiene la disponibilidad semanal de un profesor
disponibilidadProfesorT()	string	cProfesor	matriz temporal de la disponibilidad del profesor
dispPorProfesor()	string	cProfesor	matriz que contiene las horas disponibilidad por Profesor
DispProfesor()	string	cProfesor	matriz que contiene disponibilidad deñ profesor para los calculos del algoritmo
dispProfesoroptima()	string	cProfesor	Matriz que contiene el número de horas disponibles del profesor
dispProfesoroptimaT()	string	cProfesor	Matriz temp que contiene el número de horas disponibilidad del profesor, matriz temporal
numHorasTotProfesor	integer	cProfesor	numero total de disponibilidad de todos los profesores
numProfesores	integer	cProfesor	número de profesores de la planificación
profesores()	integer	cProfesor	contiene Id de los profesores utilizada en la formacion de los genes
profesores2()	string	cProfesor	matriz que contiene los profesores
tipo_profesor	string	cProfesor	Tipo de profesor

idUsuarioA	string	cUsuario	Id del usuario acrual
nombreUsuarioA	string	cUsuario	Nombre del usuario actual
numUsuario	integer	cUsuario	Número de usuarios existentes
tipoA	string	cUsuario	tipo del usuario actual
Usuario()	string	cUsuario	Matriz que almacena la informacion de los usuarios
banCruzamiento	boolean	m	bandera de cruzamiento
banGeneral	integer	m	Bandera del gen
finAG	boolean	m	Matriz que indica la finalización del algoritmo
generacionNew()	cCromosoma	m	Objeto que almacena la nueva generacion del algoritmo
historial()	single	m	matriz historial
intentos	integer	m	variable que lleva registro de los intentos de ingreso al sistema
matprob()	integer	m	matriz de probabilidad
matprobTP()	integer	m	matriz de probabilidad temporal
matrizCalculaCruce	string	m	matriz que almacena el número de horas de clase por nivel y especializacion
MejCro	cCromosoma	m	Almacena el mejor cromosoma
modaAsexual	integer	m	contiene el porcentaje de moda Asexual
modaSexual	integer	m	contiene el porcentaje de moda sexual
numBloques	integer	m	Número de Bloques de la planificación
numGeneraciones	integer	m	numero de generaciones procesadas en la corrida del algoritmo
NumHorario	integer	m	número de horas en la semana que se pueden dictar clases
numhorasDia	integer	m	numero de horas por dia
numHorasTotAulas	integer	m	numero total de disponibilidad de todas las aulas
numNiveles	integer	m	Número de niveles de la planificación

numPerturbaciones	integer	m	numero de permutaciones
numPlanificaciones	string	m	número de planificaciones
password	string	m	Matriz que almacena los passwords en memoria
Recurso1	string	m	Recurso correspondiente al Proyector
Recurso2	string	m	Recurso correspondiente al Infocus
Recurso3	string	m	Recurso correspondiente al Aire Acondicionado
totalhoraSemanales	integer	m	numero total de carga de los profesores
user	string	m	matriz en la cual se almacena los usuarios en memoria
usuarioA	string	m	Almacena el usuario Administrador
matSolucion()	string	m	matriz que contiene la solución

APÉNDICE B. CODIFICACIÓN

```
Public Sub algoritmoGenetico()  
'encontrar generacion inicial de cromosomas, cruzar cromosomas en base al porcentaje  
de cruzamiento  
'mutar un porcentaje de los nuevos individuos y encontrar la nueva generacion  
'iterar hasta que el fitness promedio de la poblacion sea constante  
'de la ultima generacion encontrada seleccionar al individuo con mejor fitness  
(fitness mas bajo)  
  
    Dim giAnt() As cCromosoma  
    Dim GenerTemp() As cCromosoma  
    Dim gi() As cCromosoma  
    Dim gc() As cCromosoma  
    Dim gm() As cCromosoma  
    Dim generacionCM() As cCromosoma  
    Dim generacionTemp() As cCromosoma  
    Dim generacionNew() As cCromosoma  
    Dim CromosomasVacio() As cCromosoma  
    Dim CromoTemp As New cCromosoma  
  
    Dim i, j, k As Integer  
    Dim cromosoma As Integer  
    Dim numCromoCruce As Integer           'Numero de cromosomas a cruzar  
    Dim numCromoMuta As Integer           'Numero de cromosomas a mutar  
    Dim numCromoCM As Integer             'Numero de cromosomas cruzados y mutados  
    Dim numTotCromos As Integer           'Numero total de cromosomas ini+cru+mut  
    Dim indMejCromo As Integer             ' indice del mejor cromosoma  
  
    Dim dispAulaBackup() As String         'matriz backup de disponibilidad aulas  
    Dim dispProfBackup() As String         'matriz backup de disponibilidad  
profesores  
  
    dispAulaBackup = cAula.disponibilidadAulas  
    dispProfBackup = cProfesor.disponibilidadProfesor  
  
    'seteamos las disponibilidades de aulas y profesores de los ingresos manuales  
    cAsignacion.setDispAsigManual
```



```

tipoMat = getTipoMateria(codmateria)
nivel = getNivelMateria(codmateria)
Erase horasAsig
ReDim horasAsig(cargaMat)
Dim resultCruce() As Variant
Erase resultCruce
ReDim resultCruce(2)
aula = 0

r = 0
Do 'pide id del profesor en Profesormateria, sacar a futuro
    r = r + 1
    If dispProfesorOptimaT(r, 0) = codProfesor Then IdProfesor = r
Loop While dispProfesorOptimaT(r, 0) <> codProfesor And r < numProfesores

' CONTROL DE CRUCE DE HORARIOS

If cargaMat = 4 And tipoMat = "A" Then
    Dim horasProfOK As Boolean
    horasProfOK = False
    'horasAulaOK = False
    k = 0
    Do
        k = k + 1
        Randomize
        bloque = Int((Rnd * numBloques) + 1)

        resultCruce = controlarCruceMaterias(codmateria, codProfesor,
nivel, bloque)

        If resultCruce(2) <> "" Then 'resultCruce(1) <> "C" And
            horasAsig = getHorasBloque(resultCruce(2))
            horasProfOK = Me.validaDispProfesor(codProfesor, horasAsig)
            If horasProfOK Then
                s = 0
                Do
                    s = s + 1
                    aula = pedirAula(horasAsig, cargaMat, codmateria,
tipoMat, capacidad, recursos)
                Loop While aula = 0 And s < numAulas * 2
            End If
        End If
        'Loop While aula = 0 And horasProfOK = False
        Loop While resultCruce(2) <> "" And k < numBloques * 2 And aula = 0
'resultCruce(1) <> "C" And

    End If
    ' AGREGAR CODIGO DE GRABADO EN MATRIZ CONTROLACRUCE, CON VALIDACION
    ' optimizar grabando sin recorrer arrglo controlacruce (Pendiente)
    If aula <> 0 And (resultCruce(1) = "D" Or resultCruce(1) = "C") Then
        Dim t As Integer
        Dim grabo As Boolean
        grabo = False
        'redim controlaCruce(numMaterias,4)
        t = 0
        Do
            t = t + 1
            If controlaCruce(t, 1) = "" Then
                grabo = True
                controlaCruce(t, 1) = codmateria
                controlaCruce(t, 2) = codProfesor
                controlaCruce(t, 3) = nivel
                controlaCruce(t, 4) = bloque

            End If
        Loop While grabo = False And controlaCruce(t, 1) <> ""

```



```

Private Function pedirAula(horasAsignadas As Variant, numHoras As Integer, codMat As
String, _
    tipoMat As String, capaRequerida As Integer, recursos As String) As Integer
'Solicita un aula de acuerdo a un conjunto de horas y asigna el respectivo
'laboratorio de acuerdo a la recaición con la materia y al tipo de materia, constituye
el alelo Y de cada gen

    Dim asignado As Boolean
    Dim aula As Integer
    Dim labDesc As String
    Dim lab As Integer
    Dim codLab As String
    Dim especializacion As String
    Dim i, j, k As Integer
    Dim intRecursos As Integer          ' numero de intentos en busqueda de recursos

    asignado = False
    especializacion = Trim(getEspecMateria(codMat))

'Inicio Asignacion de aulas
If tipoMat = "A" Then
    t = 0
    intRecursos = 0
    Do

        t = t + 1

otra:
        Randomize
        aula = Int((Rnd * numSoloAulas) + 1)

        If (aulaMem(aula, 1) = "COM1" Or aulaMem(aula, 1) = "COM2" Or _
            aulaMem(aula, 1) = "COM3" Or aulaMem(aula, 1) = "COM4") And _
            especializacion <> "Computacion" Then
            GoTo otra

        End If

        ' Verifica si la capacidad del aula satisface la capacidad requerida
        If capaRequerida > aulaMem(aula, 8) Then GoTo otra

        ' Verificacion de los recursos para el paralelo

        intRecursos = intRecursos + 1
        If intRecursos < cAula.numAulas Then

            'Verificar Recursos requeridos
            If InStr(1, recursos, "Proy") <> 0 Then
                If InStr(1, aulaMem(aula, 7), "Proy") <> 0 Then
                    'fitness = fitness - 1
                Else
                    GoTo otra
                End If
            End If

            If InStr(1, recursos, "Inf") <> 0 Then
                If InStr(1, aulaMem(aula, 7), "Inf") <> 0 Then
                    'fitness = fitness - 1
                Else
                    GoTo otra
                End If
            End If

            If InStr(1, recursos, "A/C") <> 0 Then
                If InStr(1, aulaMem(aula, 7), "A/C") <> 0 Then

```

```

        'fitness = fitness - 1
        Else
        GoTo otra
        End If
    End If

    End If

'Fin de Verificacion de los recursos para el paralelo

If numHoras = 4 Then
    If disponibilidadAulasT(aula, horasAsignadas(1)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(2)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(3)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(4)) = 0 Then
        asignado = True

    End If
End If

If numHoras = 5 Then
    If disponibilidadAulasT(aula, horasAsignadas(1)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(2)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(3)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(4)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(5)) = 0 Then

        asignado = True

    End If
End If

If numHoras = 6 Then
    If disponibilidadAulasT(aula, horasAsignadas(1)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(2)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(3)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(4)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(5)) = 0 _
        And disponibilidadAulasT(aula, horasAsignadas(6)) = 0 Then

        asignado = True

    End If
End If

    Loop While asignado = False And t < numAulas * 2 'And capaRequerida >=
aulaMem(Aula, 8)

    If t > numAulas * 2 - 1 Then aula = 0

End If
'Fin Asignacion de aulas

'inicio Asignacion laboratorios

If tipoMat = "L" Then
    ' asignacion del lab que le corresponde al lab

    i = 0
    labDesc = ""
    lab = 0
Do

```



```

sal = 0

Do
  i = i + 1
  If dispProfesoroptimaT(i, 0) = codProfesor Then IdProfesor = i
Loop While dispProfesoroptimaT(i, 0) <> codProfesor And i < numProfesores

Do
  ' VALIDAR QUE LA HORA PIBOTE NO SE PARTA EN DOS DIAS
  Randomize
  horaPibote = Int(Rnd * dispProfesoroptimaT(IdProfesor, numHorasClase + 1)) + 1
  horaPibote = dispProfesoroptimaT(IdProfesor, horaPibote)

  ' INICIO 4 HORAS
  If numHoras = 4 Then
    ' Inicio bloque hacia adelante
    If horaPibote + numhorasDia * 2 + 1 <= numHorasClase Then

      If disponibilidadProfesorT(IdProfesor, horaPibote + 1) = 0 Then
        If horaPibote <> numhorasDia And _
          horaPibote <> numhorasDia * 2 And horaPibote <> numhorasDia
* 3 _
          And horaPibote <> numhorasDia * 4 And horaPibote <>
numhorasDia * 5 Then

          If disponibilidadProfesorT(IdProfesor, horaPibote +
numhorasDia * 2) = 0 _
          And disponibilidadProfesorT(IdProfesor, horaPibote
+ numhorasDia * 2 + 1) = 0 Then
            'Parametrizar
            asignado = True
            horasAsignadas(1) = horaPibote
            horasAsignadas(2) = horaPibote + 1
            horasAsignadas(3) = horaPibote + numhorasDia * 2
'numhorasDia=14
            horasAsignadas(4) = horaPibote + numhorasDia * 2
+ 1
          End If
        End If
      Else
        If horaPibote <> 1 And horaPibote <> numhorasDia + 1 And _
          horaPibote <> numhorasDia * 2 + 1 And horaPibote <>
numhorasDia * 3 + 1 _
          And horaPibote <> numhorasDia * 4 + 1 And horaPibote <>
numhorasDia * 5 + 1 Then
          If disponibilidadProfesorT(IdProfesor, horaPibote - 1)
= 0 Then
            If disponibilidadProfesorT(IdProfesor, horaPibote
+ numhorasDia * 2 - 1) = 0 _
            And disponibilidadProfesorT(IdProfesor,
horaPibote + numhorasDia * 2) = 0 Then
              'Parametrizar
              asignado = True
              horasAsignadas(1) = horaPibote - 1
              horasAsignadas(2) = horaPibote
              horasAsignadas(3) = horaPibote + numhorasDia
* 2 - 1 'numhorasDia=14
              horasAsignadas(4) = horaPibote + numhorasDia
* 2
            End If
          End If
        End If
      End If
    End If
  End If
End If

```

```

'Fin bloque hacia adelante

'Inicio Bloque hacia atras
If horaPibote + numhorasDia * 2 + 1 > numHorasClase Then
  If horaPibote <> 1 And horaPibote <> numhorasDia + 1 And _
    horaPibote <> numhorasDia * 2 + 1 And horaPibote <>
numhorasDia * 3 + 1 And _
numhorasDia * 5 + 1 Then
  If disponibilidadProfesorT(IdProfesor, horaPibote - 1) = 0
Then
  If disponibilidadProfesorT(IdProfesor, horaPibote -
numhorasDia * 2) = 0 _
- numhorasDia * 2 - 1) = 0 Then
    'Parametrizar
    asignado = True
    horasAsignadas(1) = horaPibote - numhorasDia * 2
- 1
    horasAsignadas(2) = horaPibote - numhorasDia * 2
'numhorasDia=14
    horasAsignadas(3) = horaPibote - 1
    horasAsignadas(4) = horaPibote
    End If
  End If
End If
End If
' Fin bloque hacia atras

End If

'FIN 4 HORAS

'INICIO 3 HORAS
If numHoras = 3 Then
  'Inicio bloque hacia adelante
  If horaPibote + 2 <= numHorasClase Then
    If disponibilidadProfesorT(IdProfesor, horaPibote + 1) = 0 _
      And disponibilidadProfesorT(IdProfesor, horaPibote + 2) = 0
Then
    If horaPibote <> numhorasDia - 1 And horaPibote <>
numhorasDia And _
numhorasDia * 2 And _
numhorasDia * 3 And _
numhorasDia * 4 And _
numhorasDia * 5 Then
      horaPibote <> numhorasDia * 2 - 1 And horaPibote <>
      horaPibote <> numhorasDia * 3 - 1 And horaPibote <>
      horaPibote <> numhorasDia * 4 - 1 And horaPibote <>
      horaPibote <> numhorasDia * 5 - 1 And horaPibote <>
      asignado = True
      horasAsignadas(1) = horaPibote
      horasAsignadas(2) = horaPibote + 1
      horasAsignadas(3) = horaPibote + 2
    End If
  Else
    If horaPibote <> 1 And horaPibote <> numhorasDia + 1 And _
      horaPibote <> numhorasDia * 2 + 1 And horaPibote <>
numhorasDia * 3 + 1 And _
numhorasDia * 5 + 1 Then
      horaPibote <> numhorasDia * 4 + 1 And horaPibote <>
      If disponibilidadProfesorT(IdProfesor, horaPibote + 1)
= 0 _

```



```

horasAsignadas(2) = horaPibote + 1
horasAsignadas(3) = horaPibote + numhorasDia
'numhorasDia=14
1
horasAsignadas(4) = horaPibote + numhorasDia +
horasAsignadas(5) = horaPibote + numhorasDia *
2
'numhorasDia=14
horasAsignadas(6) = horaPibote + numhorasDia *
2 + 1
End If
End If
Else
If horaPibote <> 1 And horaPibote <> numhorasDia + 1 And _
horaPibote <> numhorasDia * 2 + 1 And horaPibote <>
numhorasDia * 3 + 1 _
And horaPibote <> numhorasDia * 4 + 1 And horaPibote <>
numhorasDia * 5 + 1 Then
If disponibilidadProfesorT(IdProfesor, horaPibote - 1)
= 0 Then
If disponibilidadProfesorT(IdProfesor, horaPibote
+ numhorasDia * 2 - 1) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote + numhorasDia * 2) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote + numhorasDia - 1) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote + numhorasDia) = 0 Then
'Parametrizar
asignado = True
horasAsignadas(1) = horaPibote - 1
horasAsignadas(2) = horaPibote
horasAsignadas(3) = horaPibote + numhorasDia
- 1 'numhorasDia=14
horasAsignadas(4) = horaPibote + numhorasDia
horasAsignadas(5) = horaPibote + numhorasDia
* 2 - 1 'numhorasDia=14
horasAsignadas(6) = horaPibote + numhorasDia
* 2
End If
End If
End If
End If
'Fin bloque hacia adelante
'Inicio Bloque hacia atras
If horaPibote + numhorasDia * 2 + 1 > numHorasClase Then
If horaPibote <> 1 And horaPibote <> numhorasDia + 1 And _
horaPibote <> numhorasDia * 2 + 1 And horaPibote <>
numhorasDia * 3 + 1 And _
horaPibote <> numhorasDia * 4 + 1 And horaPibote <>
numhorasDia * 5 + 1 Then
If disponibilidadProfesorT(IdProfesor, horaPibote - 1)
= 0 Then
If disponibilidadProfesorT(IdProfesor, horaPibote
- numhorasDia * 2) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote - numhorasDia * 2 - 1) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote - numhorasDia) = 0 _
And disponibilidadProfesorT(IdProfesor,
horaPibote - numhorasDia - 1) = 0 Then
'Parametrizar
asignado = True
horasAsignadas(1) = horaPibote - numhorasDia
* 2 - 1 'numhorasDia=14

```



```

Erase disponibilidadProfesorTemp

ReDim disponibilidadAulasTemp(numAulas, NumHorario)
ReDim disponibilidadProfesorTemp(numProfesores, NumHorario)

disponibilidadAulasTemp = disponibilidadAulas
disponibilidadProfesorTemp = disponibilidadProfesor

Dim codProfesor As String
Dim codAula As String
Dim codmateria As String
Dim IdProfesor As String
Dim idAula As String
Dim IdMateria As String
Dim hora As Integer          ' Hora asignada de clases del gen
Dim recursos As String
Dim capacidad As Integer
Dim nivel As String
Dim especializacion As String

cargaNivelEspec
'encerrar los valores de las horas de la matriz matrizCalculaCruce va fuera del for
  Dim p, q, r As Integer
  For p = 1 To numNiveles
    For q = 0 To numEspec
      For r = 1 To numHorasClase + 2
        matrizCalculaCruce(p, q, r) = 0
      Next r
    Next q
  Next p

fitness = 0
For i = 1 To numProfMatPar
  ' cambiar la matriz profesorMatPar esta matriz solo contiene los paralelos y
no esta por horas
  ' aqui se necesita una matriz de paralelos por horas , ed de 912
  'cargaMat = CInt(profesorMatPar(i, 4))
  codProfesor = ProfesorMateria(gen(i).X, 1)
  codmateria = ProfesorMateria(gen(i).X, 2)
  IdProfesor = getIDprofesor(codProfesor)
  idAula = gen(i).y
  hora = gen(i).z
  recursos = ProfesorMateria(gen(i).X, 5)
  If ProfesorMateria(gen(i).X, 6) = "" Then ProfesorMateria(gen(i).X, 6) = 0
  capacidad = ProfesorMateria(gen(i).X, 6)
  tipoMat = getTipoMateria(codmateria)
  nivel = getNivelMateria(codmateria)
  especializacion = getEspecMateria(codmateria)
  If idAula <> 0 Then codAula = getCodAula(CInt(idAula))

  'Condenar la asignacion de aula
  If idAula = 0 Then
    fitness = fitness + 100
    GoTo cruce
  End If

  'CONDENAR EL CRUCE DE HORAS
  If disponibilidadAulasTemp(idAula, hora) = 0 Then
    disponibilidadAulasTemp(idAula, hora) = 1
  Else
    fitness = fitness + 1
  End If

  If disponibilidadProfesorTemp(IdProfesor, hora) = 0 Then

```

```

        disponibilidadProfesorTemp(IdProfesor, hora) = 1
    Else
        fitness = fitness + 1
End If

'Verificar la capacidad
If aulaMem(idAula, 8) = "" Then aulaMem(i, 8) = 0
If capacidad > CInt(aulaMem(idAula, 8)) Then fitness = fitness + 10

'Verificar Recursos requeridos
If InStr(1, recursos, "Proy") <> 0 Then
    If InStr(1, aulaMem(idAula, 7), "Proy") <> 0 Then
        fitness = fitness - 0.1
    Else
        fitness = fitness + 1
    End If
End If

If InStr(1, recursos, "Inf") <> 0 Then
    If InStr(1, aulaMem(idAula, 7), "Inf") <> 0 Then
        fitness = fitness - 0.1
    Else
        fitness = fitness + 1
    End If
End If

If InStr(1, recursos, "A/C") <> 0 Then
    If InStr(1, aulaMem(idAula, 7), "A/C") <> 0 Then
        fitness = fitness - 0.1
    Else
        fitness = fitness + 1
    End If
End If

'Condenar un gen si se asigna a un COM y no es de computacion
If (aulaMem(idAula, 1) = "COM1" Or aulaMem(idAula, 1) = "COM2" Or _
    aulaMem(idAula, 1) = "COM3" Or aulaMem(idAula, 1) = "COM4") And _
    especializacion <> "Computacion" Then
    fitness = fitness + 10

End If

labs:
'Condenar si las materias no se dan en su respectivo laboratorio
If tipoMat = "L" Then
    Dim X, y As Integer
    Dim labDesc, codLab As String
    X = 0
    labDesc = ""

    Do
        X = X + 1
        If materia(X, 1) = codmateria Then labDesc = materia(X, 7)
    Loop While materia(X, 1) <> codmateria And X < numMaterias

    If labDesc <> "" Then
        y = 0
        Do
            y = y + 1
            If mSoloLabs(y, 4) = labDesc Then codLab = mSoloLabs(y, 1)
        Loop While mSoloLabs(y, 4) <> labDesc And y < numSoloLabs
    End If

    If codAula <> codLab Then
        fitness = fitness + 50
    End If

```

```

End If

cruce:

' Calcular el porcentaje de cruce de horas de materias del mismo nivel

' incrementar el valor de las horas por nivel y especializacion
Dim t, u, v As Integer
t = 0
Do
    t = t + 1
    If nivel = matrizCalculaCruce(t, 0, 0) Then
        u = 0
        Do
            u = u + 1
            If especializacion = matrizCalculaCruce(t, u, 0) Then
                matrizCalculaCruce(t, u, hora) = matrizCalculaCruce(t, u,
hora) + 1
                matrizCalculaCruce(t, 0, hora) = matrizCalculaCruce(t, 0,
hora) + 1
            End If
            'u = u + 1
        Loop While numNiveles > u And especializacion <>
matrizCalculaCruce(t, u, 0)

    End If
    't = t + 1
    Loop While numNiveles > t And nivel <> matrizCalculaCruce(t, 0, 0)

' verificar que todas las horas se den en la misma aula -> YA ESTA VALIADADO
'MATERIA DEL MISMO NIVEL no estan en el mismo horario
'La misma MATERIA para todos los paralelos se dicta a la misma hora
'Que las materias de laboratorios se dicten en los laboratorios respectivos-> YA
ESTA VALIADADO

Next i

'Calcula el porcentaje de cruce
Dim totHoras, horasCruce, porcentajeNE, porcentajeNivel As Integer
Dim porcentajeN As Integer
totHoras = 0          ' total de horas por nivel- especializacion
horasCruce = 0       ' horas de cruce por nivel- especializacion
porcentajeNE = 0     ' porcentaje de cruce por nivel- especializacion
porcentajeN = 0      ' porcentaje de cruce por nivel
porcentajeCruce = 0  ' porcentaje de cruce total
porcentajeNivel = 0  ' porcentaje de cruce por nivel total
p = 0
q = 0
r = 0

For p = 1 To numNiveles
    porcentajeN = 0
    porcentajeNivel = 0
    For q = 0 To numEspe
        totHoras = 0          ' total de horas por nivel- especializacion
        horasCruce = 0
        porcentajeNE = 0
    
```



```

    Dim cromocruzados() As cCromosoma          ' generacion de cromosomas
cruzados
    Dim cromol, cromol2 As Integer             ' indices de los cromomas a cruzar
    Dim puntoCruce, puntoCruce1, puntoCruce2 As Integer
    Dim numGenACruzar As Integer              ' numero de Genes a cruzar
    Dim tempCromosoma As New cCromosoma
    Dim tipoCruce As Integer                  ' tipo de cruce 1= one point,
2=two point crossover
    Dim i, j, k As Integer

numGenACruzar = Int(porcCruzamiento * cParametros.NumCromosomas / 100)
ReDim cromocruzados(1 To numGenACruzar)
'Set tempCromosoma = New cCromosoma

' Aleatoriamente sed elige el tipo de Cruzamiento
' tipoCruce= 1   One point cross over
' tipoCruce= 2   Two point cross over
Randomize
tipoCruce = Int((2 * Rnd) + 1)

'One point cross over
If tipoCruce = 1 Then

    For i = 1 To numGenACruzar

        'Elegimos dos cromosomas de forma aleatoria para cruzarlos
        Randomize
        cromol = Int((NumCromosomas * Rnd) + 1)
        Randomize
        cromol2 = Int((NumCromosomas * Rnd) + 1)

        Randomize
        puntoCruce = Int((numProfMatPar * Rnd) + 1)

        For j = 1 To puntoCruce
            tempCromosoma.SetGen j, genInicial(cromol).GetGen(j)
        Next j

        For j = puntoCruce + 1 To numProfMatPar
            tempCromosoma.SetGen j, genInicial(cromol2).GetGen(j)
        Next j

        Set cromocruzados(i) = tempCromosoma
        cromocruzados(i).C_status = "C1"
        cromocruzados(i).calcularFitness
        Set tempCromosoma = Nothing
    Next i
End If

'Two point cross over
If tipoCruce = 2 Then

    For i = 1 To numGenACruzar
        'Generamos dos ids de cromosomas de forma aleatoria para cruzarlos
        Randomize
        cromol = Int((NumCromosomas * Rnd) + 1)
        Randomize
        cromol2 = Int((NumCromosomas * Rnd) + 1)

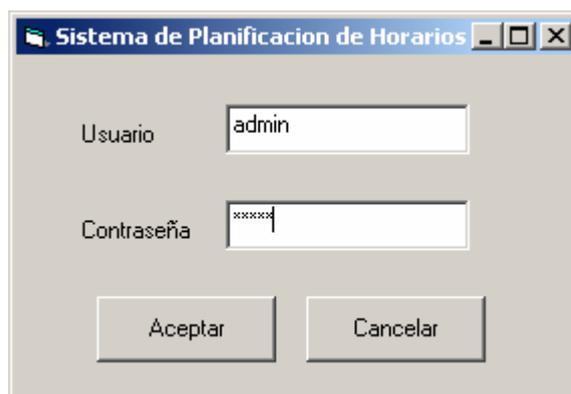
        'generamos los puntos de cruce
        Randomize
        puntoCruce1 = Int((numProfMatPar * Rnd) + 1)
        Do
            Randomize

```


APÉNDICE C. MANUAL DE USUARIO

¿CÓMO INGRESAR AL SISTEMA?

Para ingresar al sistema, debemos ingresar el usuario, la clave y presionar el botón aceptar.



The image shows a screenshot of a software window titled "Sistema de Planificación de Horarios". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains two text input fields. The first field is labeled "Usuario" and contains the text "admin". The second field is labeled "Contraseña" and contains five asterisks "xxxxx". Below these fields are two buttons: "Aceptar" on the left and "Cancelar" on the right.

Ilustración C-1: Pantalla de Validación de Usuarios

Existen tres tipos de usuarios: administradores, digitadores e invitados, sus diferencias se dan en los privilegios en las actividades del sistema con los que cuenta cada uno.

Los administradores pueden crear, modificar y eliminar usuarios, pueden también modificar la contraseña de cualquier usuario.

Crear Usuario...

Nombre de usuario: jaragundi

Nombre Completo: Jorge Aragundi

Tipo: Administrador

Contraseña: *****

Confirmar contraseña: *****

Aceptar Cancelar

Ilustración C-2: Ingresar Usuario

Crear Usuario...

Nombre de usuario: csalazar

Nombre Completo: Christian Salazar

Tipo: Administrador

Aceptar Cancelar

Ilustración C-3: Modificar Usuario

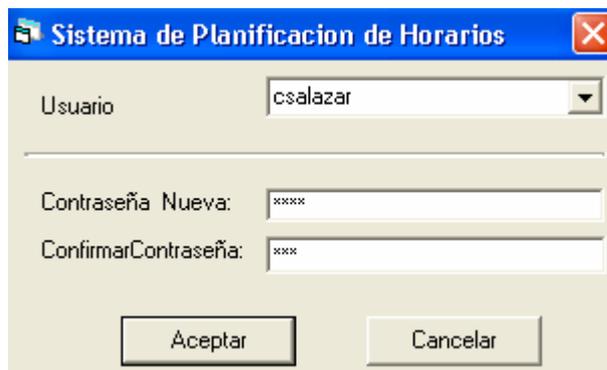
Eliminar Usuario

Elija el usuario a eliminar

	Usuario	Nombre	Tipo
1	admin	Administrador	Administrador
2	csalazar	Christian Salazar	Administrador
3	digitador1	Secretaria	Digitador
4	invitado	Alumno1	Invitado
5	jaragundi	Jorge Aragundi	Administrador
6	kviteri	Katherine Viteri	Administrador

Eliminar Cancelar

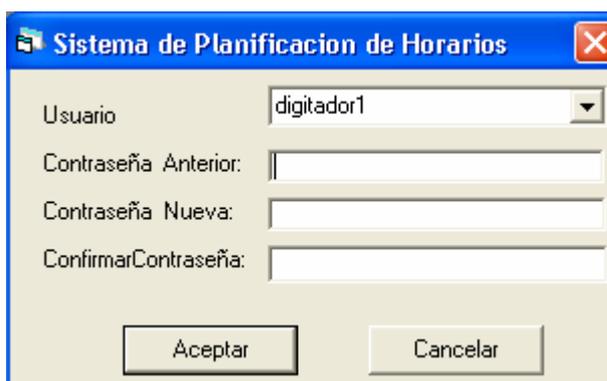
Ilustración C-4: Eliminar Usuario



The screenshot shows a dialog box titled "Sistema de Planificación de Horarios". It contains a dropdown menu for "Usuario" with "csalazar" selected. Below it are two text input fields: "Contraseña Nueva:" containing "****" and "ConfirmarContraseña:" containing "****". At the bottom are two buttons: "Aceptar" and "Cancelar".

Ilustración C-5: Cambio de contraseña desde usuario administrador.

Los usuarios con privilegios de digitadores son capaces de modificar únicamente sus claves propias.



The screenshot shows a dialog box titled "Sistema de Planificación de Horarios". It contains a dropdown menu for "Usuario" with "digitador1" selected. Below it are three text input fields: "Contraseña Anterior:" (empty), "Contraseña Nueva:" (empty), and "ConfirmarContraseña:" (empty). At the bottom are two buttons: "Aceptar" and "Cancelar".

Ilustración C-6: Cambio de contraseña desde usuario digitador.

Los invitados no pueden ejecutar ninguna de las acciones anteriores.

¿CÓMO CREAR UNA NUEVA PLANIFICACIÓN?

Existen dos formas, la primera es presionando en botón nueva en la pantalla inicial al abrir la aplicación y la segunda es estando ya dentro de una planificación seleccionando la opción nuevo del menú Archivo, los dos

caminos conducen a la ventana siguiente en la que debemos ingresar el código, la descripción y presionar el botón aceptar.



Nueva Planificación

Código: PLAN2005II

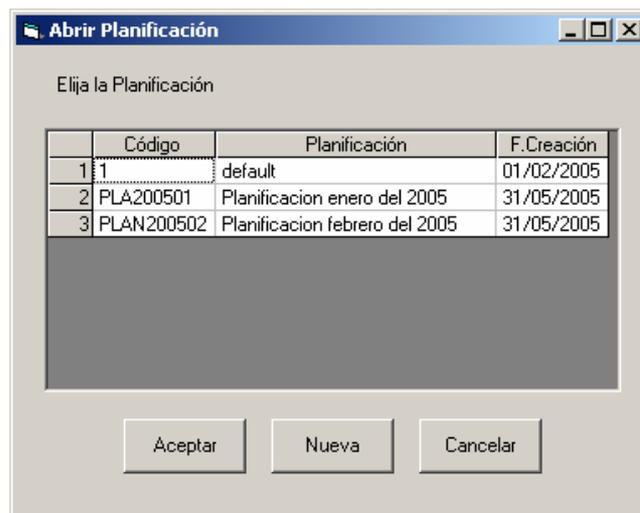
Descripción: Planificación 2005 término II

Aceptar Cancelar

Ilustración C-7: Pantalla de Creación de una nueva Planificación

¿CÓMO ABRIR UNA NUEVA PLANIFICACIÓN?

Se selecciona la aplicación que deseamos abrir y presionamos el botón aceptar.



Abrir Planificación

Elija la Planificación

	Código	Planificación	F.Creación
1	1	default	01/02/2005
2	PLA200501	Planificacion enero del 2005	31/05/2005
3	PLAN200502	Planificacion febrero del 2005	31/05/2005

Aceptar Nueva Cancelar

Ilustración C-8: Pantalla de Selección de Planificación

¿CÓMO INGRESAR LOS PARÁMETROS DE LA SOLUCIÓN?

Para poder editar los parámetros del algoritmo seleccionamos la opción Parámetros Iniciales del menú Herramientas y se presentará la siguiente pantalla:

Parametros Iniciales del Sistema

Parametros Recomendados

Condiciones de Parada

Numero Maximo de Generaciones: 1

Cromosomas por Generacion: 10

Condiciones para Modas Generacionales

Porcentaje de Mutacion: 50 %

Porcentaje de Cruzamiento: 50 %

Parametros Organizacionales

Dias por semana: 6

Horas clase diarias: 14

Aceptar Salir

Ilustración C-9: Parámetros del Sistema y del Algoritmo Genético

En la cual podemos editar el número de cromosomas de la solución, el número de generaciones y los porcentajes de mutación y cruzamiento, así como los días de clase por semana y el número máximo de horas clase diaria.

¿CÓMO EJECUTAR LA OPTIMIZACIÓN?

Una vez ingresados todos los parámetros, materias, profesores, aulas, asignaciones y disponibilidades, para inicial la búsqueda de la solución elegimos la opción Optimizar del menú Herramientas, este proceso puede demorar un tiempo considerable dependiendo de la complejidad de la planificación y de los parámetros de la misma.

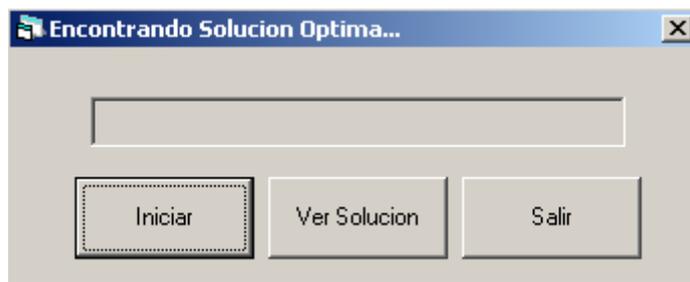


Ilustración C-10: Optimización usando Algoritmos Genéticos

¿CÓMO INGRESAR UN PROFESOR?

En la Opción DataBase del menú elegimos Profesores y luego ingresar, se nos mostrará la siguiente pantalla en la cual ingresaremos los datos del nuevo profesor.

The screenshot shows a window titled "Ingresos a la Base de Datos" with a tabbed interface. The active tab is "Profesor". The window contains three main sections:

- Datos Personales:** Includes text boxes for "Identificacion" and "Nombre", and radio buttons for "Contratado" and "De Planta".
- Asignacion de Materias:** Includes a dropdown menu for "Materia", a dropdown menu for "Paralelo", and an "Ingresar" button.
- Materias que Dicta:** Includes a table with columns for subject and class, and buttons for "Borar Ingresos" and "Eliminar Ultimo".

At the bottom of the window are three buttons: "Aceptar", "Ingresar Nuevo", and "Salir".

Ilustración C-11: Ingreso de Profesores

¿CÓMO EDITAR LA INFORMACIÓN DE UN PROFESOR?

En la Opción DataBase del menú elegimos Profesores y luego Modificar, se presentará una pantalla sin datos, damos un clic en el botón buscar y se nos presentará la siguiente pantalla.

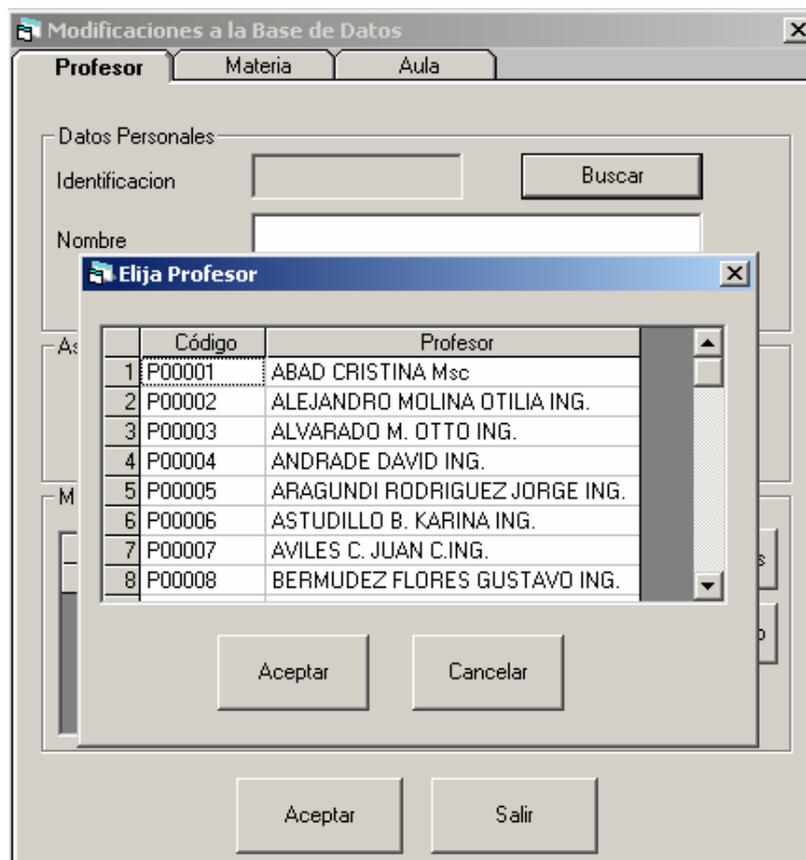


Ilustración C-12: Selección de profesores para modificación.

Elegimos al profesor deseado y presionamos el botón aceptar.

Modificaciones a la Base de Datos

Profesor | Materia | Aula

Datos Personales:

Identificación: P00001 [Buscar]

Nombre: ABAD CRISTINA Msc

Contratado De Planta

Asignación de Materias:

Materia: [] Paralelo: [] [Ingresar]

Materias que Dicta:

	Materia	Paralelo
1	COMUNICACIÓN DE DATOS	1
2	INTRODUCCIÓN A LAS REDES DE CI	1
3	SISTEMAS DISTRIBUIDOS	1

[Borar Ingresos] [Eliminar Ultimo]

[Aceptar] [Salir]

Ilustración C-13: Modificación de Profesores

En esta pantalla modificaremos los datos del profesor y también podemos agregar y quitar asignaciones de materias.

¿CÓMO ASIGNAR LA DISPONIBILIDAD DE UN PROFESOR?

Escogemos la opción Disponibilidad profesor del menú Database, se mostrará la pantalla a continuación, para escoger un profesor presionamos el botón Buscar.

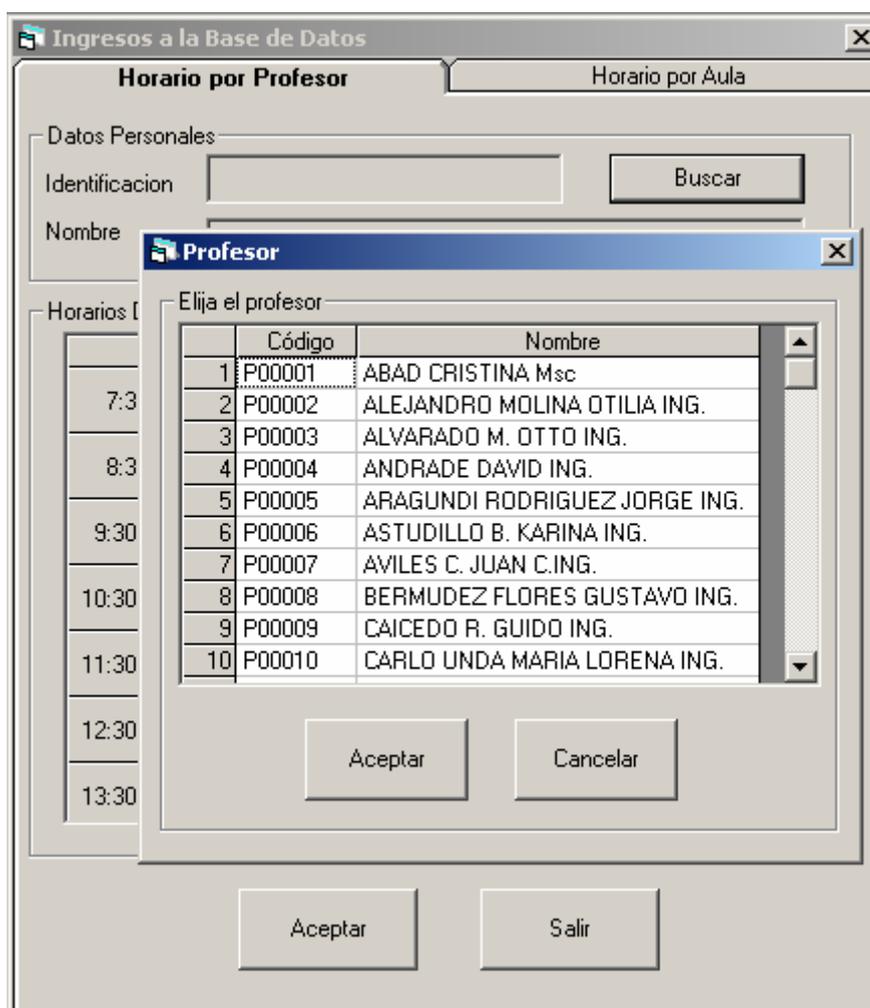


Ilustración C-14: Selección de profesores para modificar su horario de disponibilidad.

Seleccionamos el profesor deseado y presionamos el botón Aceptar.

Ingresos a la Base de Datos

Horario por Profesor | Horario por Aula

Datos Personales

Identificacion: P00001 [Buscar]

Nombre: ABAD CRISTINA Msc

Horarios Disponibles

	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado		
7:30-8:30	
8:30-9:30			
9:30-10:30			
10:30-11:30			
11:30-12:30			
12:30-13:30			
13:30-14:30			

Aceptar | Salir

Ilustración C-15: Horario de disponibilidad por profesor

En esta pantalla seleccionaremos las horas disponibles del profesor presionando la celda correspondiente a la hora en la cual el profesor esta disponible.

¿CÓMO ELIMINAR UN PROFESOR?

En el menú DataBase seleccionamos la opción profesor y luego eliminar, seleccionamos al profesor y procedemos a su eliminación.

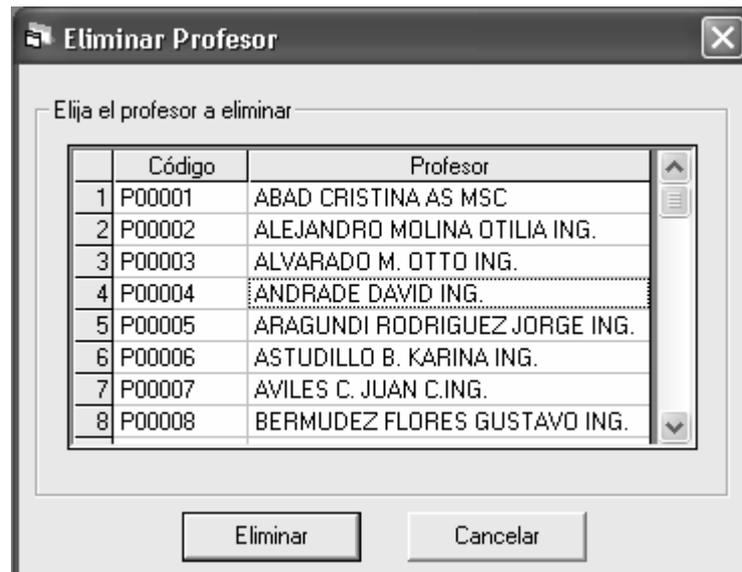


Ilustración C-16: Eliminar Profesor

¿CÓMO CREAR UNA MATERIA?

En la Opción DataBase del menú elegimos Materias y luego ingresar, se presentará la siguiente pantalla en la cual ingresaremos los datos de la nueva materia.

The image shows a software window titled "Ingresos a la Base de Datos" with three tabs: "Profesor", "Materia", and "Aula". The "Materia" tab is selected. The form contains the following fields and controls:

- Codigo:** A text input field.
- Nombre:** A text input field.
- Nivel:** A text input field.
- Paralelos:** A text input field.
- Carga Horaria:** Two radio buttons labeled "Diaria" and "Semanal", each followed by a small square checkbox.
- Tipo de Aula:** Two radio buttons labeled "Aula" and "Laboratorio", followed by a dropdown menu.
- Tipo de Materia:** Two radio buttons labeled "Fundamental" and "Especializacion", followed by a dropdown menu.
- Buttons:** Three buttons at the bottom: "Aceptar", "Ingresar Nuevo", and "Salir".

Ilustración C-17: Ingreso de Materias

Luego del ingreso de los datos presionamos el botón aceptar.

¿CÓMO EDITAR LA INFORMACIÓN DE UNA MATERIA?

En la Opción DataBase del menú elegimos Materias y luego Modificar, se presentara una pantalla sin datos, damos un clic en el botón buscar y se nos presentará la siguiente pantalla.

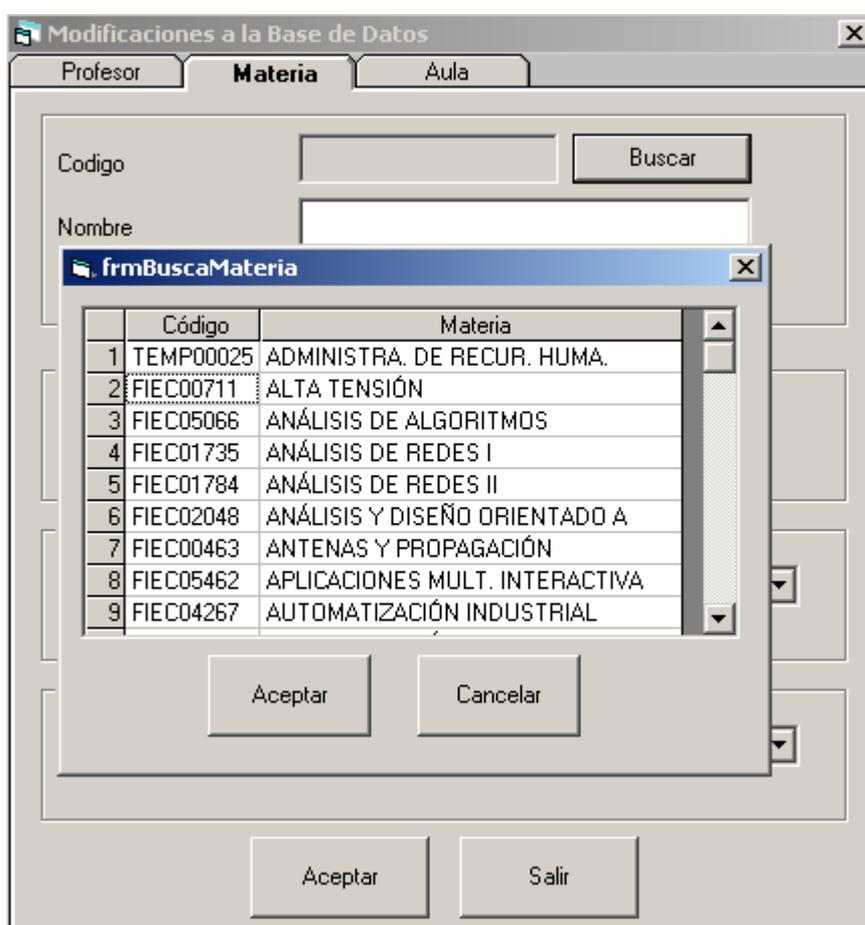


Ilustración C-18: Selección de materia para su modificación

Elegimos la materia deseado y presionamos el botón aceptar.

The screenshot shows a software window titled "Modificaciones a la Base de Datos" with three tabs: "Profesor", "Materia", and "Aula". The "Materia" tab is active. The window contains several input fields and buttons:

- Codigo:** FIEC01735 (with a "Buscar" button next to it)
- Nombre:** ANÁLISIS DE REDES I
- Nivel:** 200
- Paralelos:** 7
- Carga Horaria:** Diaria: 2, Semanal: 4
- Tipo de Aula:** Radio buttons for "Aula" (selected) and "Laboratorio", followed by a dropdown menu.
- Tipo de Materia:** Radio buttons for "Fundamental" and "Especializacion" (selected), followed by a dropdown menu showing "Electronica".
- Buttons:** "Aceptar" and "Salir" at the bottom.

Ilustración C-19: Modificación de materia.

En esta pantalla modificaremos los datos de la materia, luego que hayamos realizado las modificaciones presionamos el botón aceptar.

¿CÓMO ELIMINAR UNA MATERIA?

En el menú DataBase seleccionamos la opción materia y luego eliminar, seleccionamos a la materia y procedemos a su eliminación.



Ilustración C-20: Eliminar Materia

¿CÓMO INGRESAR UN AULA?

En la Opción DataBase del menú elegimos Aulas y luego ingresar, se presentará la siguiente pantalla en la cual ingresaremos los datos del aula que deseamos ingresar.

The image shows a software window titled "Ingresos a la Base de Datos" with a tabbed interface. The active tab is "Aula". The window contains the following elements:

- Two tabs: "Profesor" and "Materia".
- Input fields: "Codigo" and "Localizacion".
- Radio buttons: "Aula" (selected) and "Laboratorio".
- A dropdown menu.
- Checkboxes under the heading "Recursos":
 - Proyector de Slides
 - Aire Acondicionado
 - Infocus
- Buttons: "Aceptar", "Ingresar Nuevo", and "Salir".

Ilustración C-21: Ingreso de Aula.

¿CÓMO EDITAR LA INFORMACIÓN DE UN AULA?

En la Opción DataBase del menú elegimos Aulas y luego Modificar, se presentará una pantalla sin datos, damos un clic en el botón buscar y se nos presentará la siguiente pantalla.

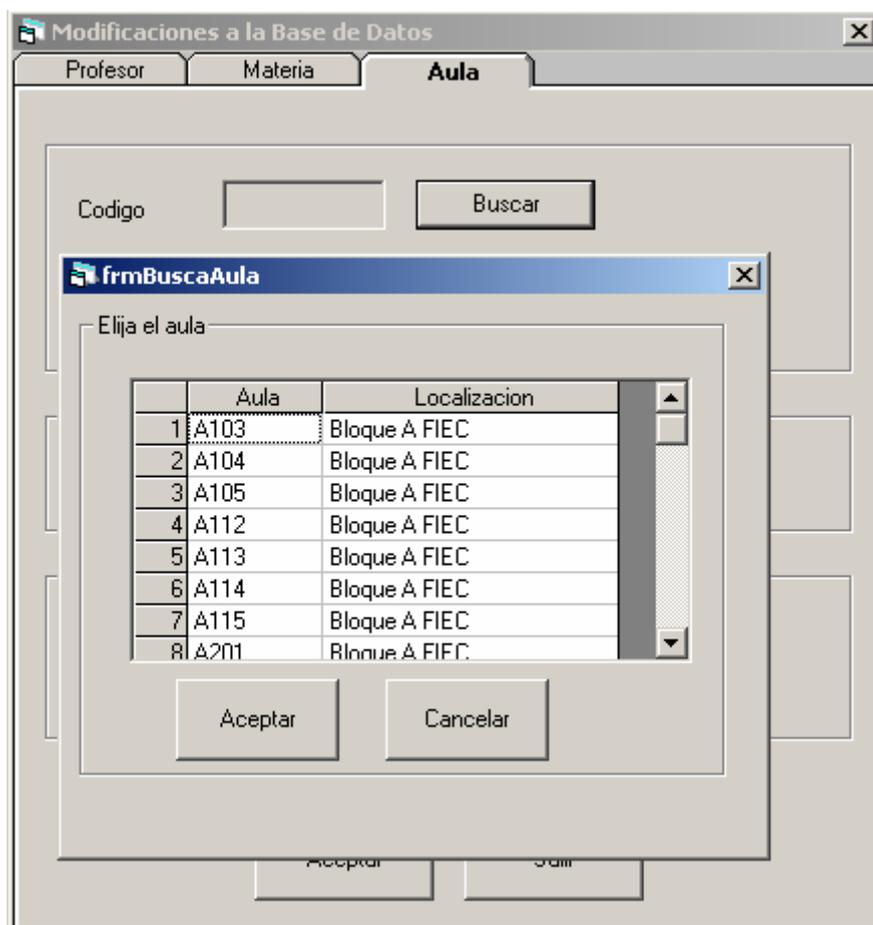


Ilustración C-22: Selección de Aula para su modificación

Elegimos al aula deseada y presionamos el botón aceptar.

Modificaciones a la Base de Datos

Profesor Materia **Aula**

Codigo: A103 Buscar

Localizacion: Bloque A FIEC

Tipo de Aula:

Aula Laboratorio [Dropdown]

Recursos:

Proyector de Slides Infocus

Aire Acondicionado

Aceptar Salir

Ilustración C-23: Modificación de Aula

En esta pantalla modificaremos los datos del aula, luego de modificar presionamos el botón aceptar.

¿CÓMO ASIGNAR LA DISPONIBILIDAD DE UN AULA?

Escogemos la opción Disponibilidad Aula del menú Database, se presentará la pantalla a continuación, para escoger un aula presionamos el botón buscar.

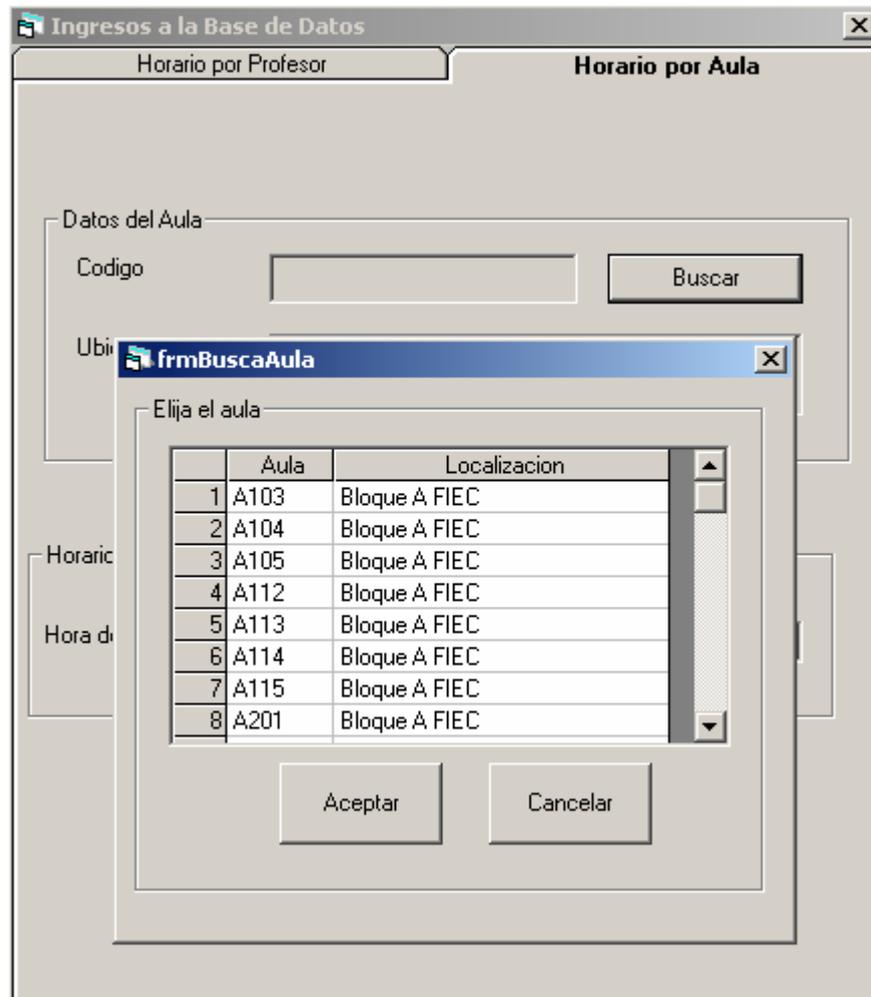


Ilustración C-24: Selección de aula para modificar su horario de disponibilidad

Seleccionamos el aula deseada y presionamos el botón Aceptar.

The screenshot shows a software window titled "Ingresos a la Base de Datos" with a sub-tab "Horario por Aula". The window is divided into two main sections. The top section, "Datos del Aula", contains a "Codigo" field with the value "A103", a "Ubicacion" field with the value "Bloque A FIEC", and a "Buscar" button. The bottom section, "Horario Disponible", contains two dropdown menus: "Hora de Apertura" set to "07:30" and "Hora de Cierre" set to "20:30". At the bottom of the window, there are two buttons: "Aceptar" and "Salir".

Ilustración C-25: Modificación de horario de disponibilidad de aula

En esta pantalla seleccionaremos las horas en la que el aula está disponible y presionamos el botón aceptar.

¿CÓMO ELIMINAR UN AULA?

En el menú DataBase seleccionamos la opción aula y luego eliminar, seleccionamos el aula y procedemos a su eliminación.

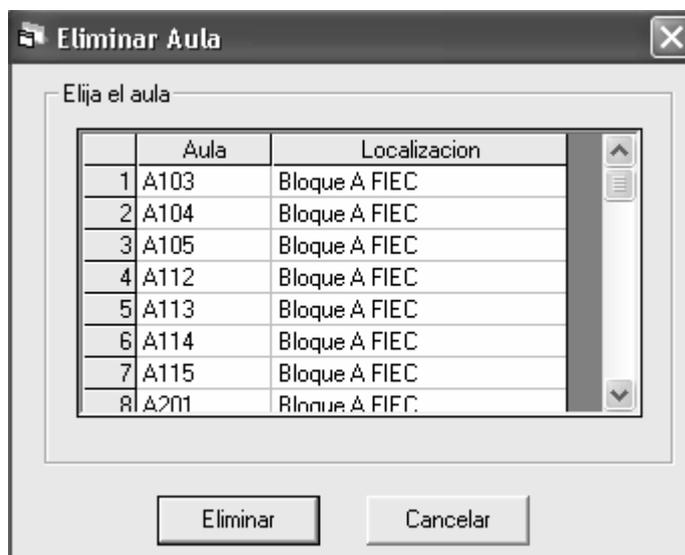


Ilustración C-26: Eliminar Aula

¿CÓMO ASIGNAR MATERIAS A UN PROFESOR?

En la Opción DataBase del menú elegimos Profesores y luego Modificar, se presentará una pantalla sin datos, damos un clic en el botón buscar y se nos presentará la siguiente pantalla.

Modificaciones a la Base de Datos

Profesor | Materia | Aula

Datos Personales

Identificación: P00005 [Buscar]

Nombre: ARAGUNDI RODRIGUEZ JORGE ING.

Contratado De Planta

Asignación de Materias

Materia: [] Paralelo: [] [Ingresar]

Materias que Dicta

	Materia	Paralelo
2	LABORATORIO DE CONTROLES IND	2
3	LABORATORIO DE CONTROLES IND	3
4	LABORATORIO DE CONTROLES IND	4
5	LABORATORIO DE REDES ELÉCTRIC	13
6	TEORÍA ELECTROMAGNÉTICA I	7

[Borrar Ingresos] [Eliminar Ultimo]

[Aceptar] [Salir]

Ilustración C-27: Asignación de materias por profesor

En esta pantalla podemos asignar o eliminar materias asignadas al profesor elegido, para asignar elegimos la materia en el combo box materia, elegimos el paralelo y presionamos el botón ingresar, esto en la etiqueta Asignación de Materias, esta asignación se verá reflejada en el gris de la etiqueta Materias que Dicta, una vez que hayamos asignado las materias deseadas al profesor elegido presionamos el botón aceptar.

¿CÓMO ELIMINAR UN PARALELO?

En la Opción DataBase del menú elegimos paralelo y luego eliminar, seleccionamos la materia y el paralelo a eliminar.



Ilustración C-28: Eliminación de un Paralelo.

¿CÓMO REALIZAR LAS ASIGNACIONES DE PROFESOR-MATERIA-PARALELO?

Las asignaciones especiales de profesores se realizan en el menú DataBase-paralelo seleccionando la opción asignar, escogemos la materia, el profesor y el paralelo y lo ingresamos como una nueva asignación, posteriormente se seleccionan los recursos con que debe contar el aula como aire acondicionado, proyector de diapositivas o proyector.

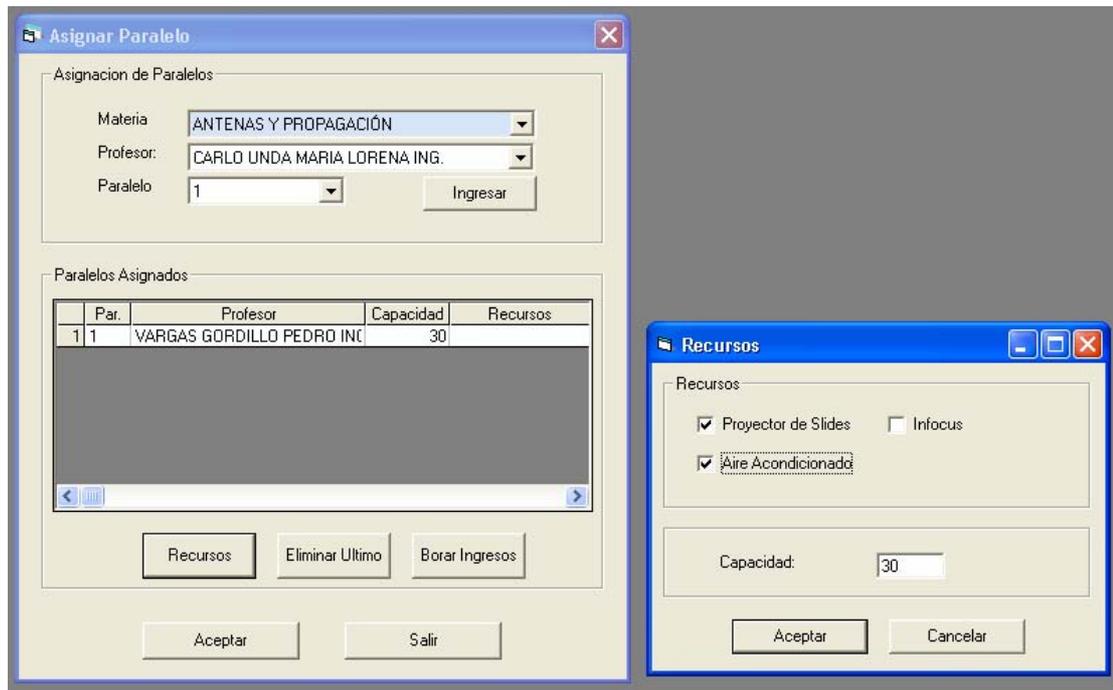
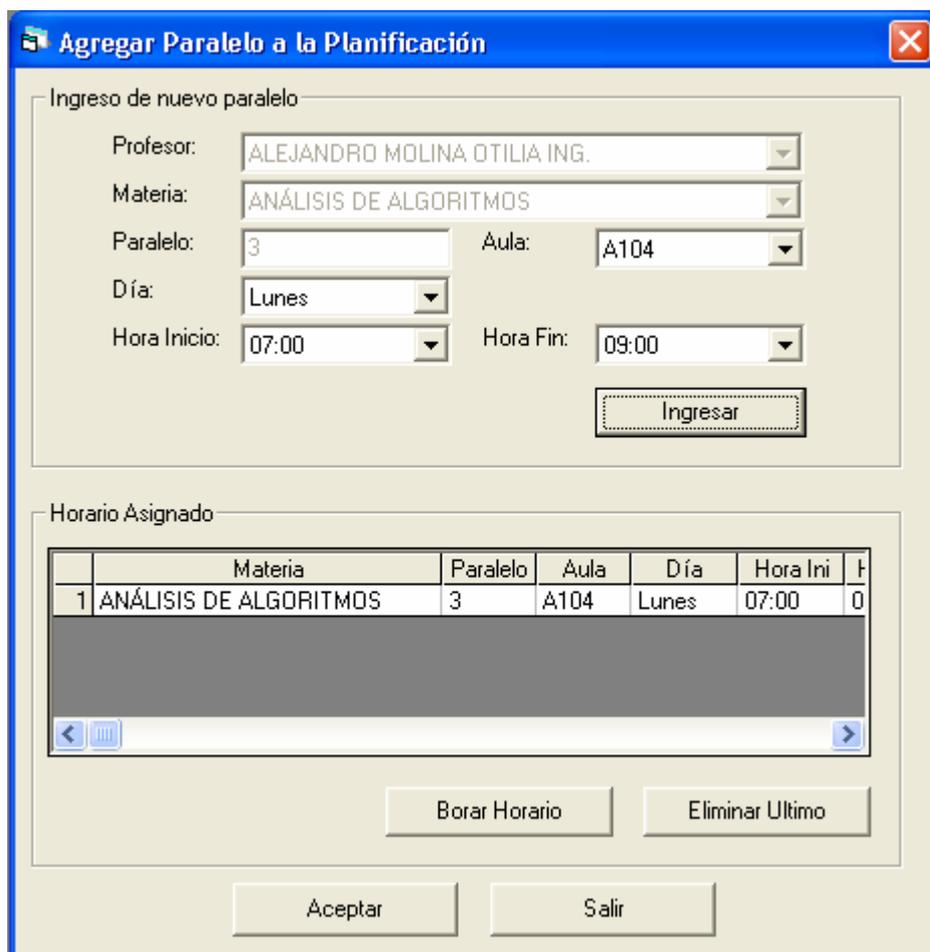


Ilustración C-29: Asignación de paralelos

¿CÓMO AGREGAR PARALELOS DESPUES DE LA OPTIMIZACIÓN?

Seleccionamos la opción DataBase-paralelo, seleccionando la opción crear, variante que esta disponible únicamente después de la ejecución del proceso de optimización para crear nuevos paralelos sin modificar la plantificación obtenida del proceso de optimización.

El valor del paralelo es creado automáticamente, aumentando el uno el número de paralelos disponibles para la materia seleccionada.



The screenshot shows a software window titled "Agregar Paralelo a la Planificación" with a close button in the top right corner. The window is divided into two main sections: "Ingreso de nuevo paralelo" and "Horario Asignado".

Ingreso de nuevo paralelo

This section contains several input fields:

- Profesor: ALEJANDRO MOLINA OTILIA ING.
- Materia: ANÁLISIS DE ALGORITMOS
- Paralelo: 3
- Aula: A104
- Día: Lunes
- Hora Inicio: 07:00
- Hora Fin: 09:00

There is an "Ingresar" button at the bottom right of this section.

Horario Asignado

This section contains a table with the following data:

	Materia	Paralelo	Aula	Día	Hora Ini	F
1	ANÁLISIS DE ALGORITMOS	3	A104	Lunes	07:00	0

Below the table are two buttons: "Borar Horario" and "Eliminar Ultimo".

At the bottom of the window are two buttons: "Aceptar" and "Salir".

Ilustración C-30: Creación de nuevo paralelo, posterior a la optimización

APÉNDICE D. MANUAL DE INSTALACIÓN DEL SISTEMA

Cierre todos los programas que tenga abierto e inserte CD de instalación

Se ejecutará automáticamente el autorun, de no ser así, dar doble clic en el icono setup.

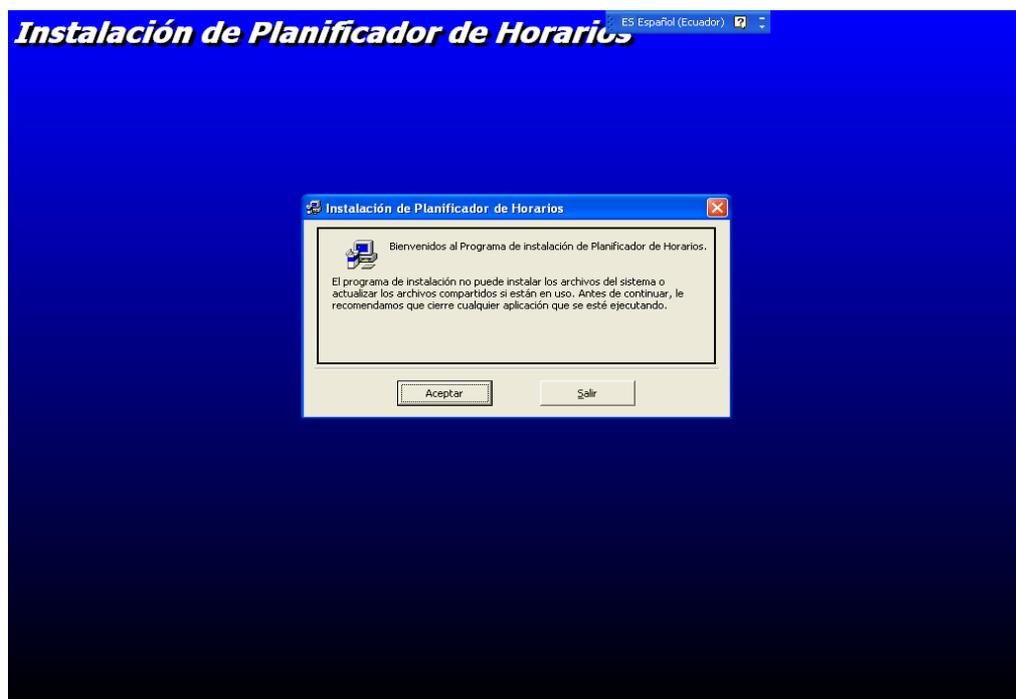


Ilustración D-1: Bienvenida al programa de instalación

Dar clic en el botón aceptar.

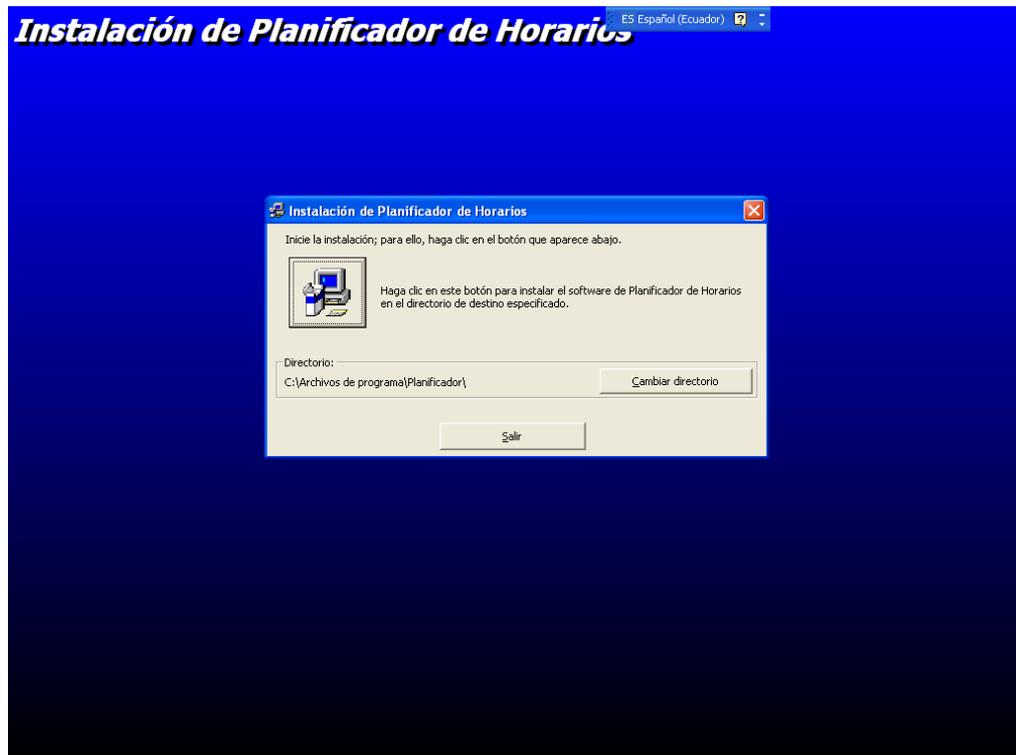


Ilustración D-2: Inicio del proceso de instalación

Se presentará la siguiente pantalla en la que se podrá seleccionar la ruta de la instalación.



Ilustración D-3: Selección de directorio de destino de la instalación

Para continuar con la instalación presionamos el botón iniciar para continuar la instalación.

Seleccionar el Grupo de programas.



Ilustración D-4: Selección de grupo de programas

Inicio de la instalación.

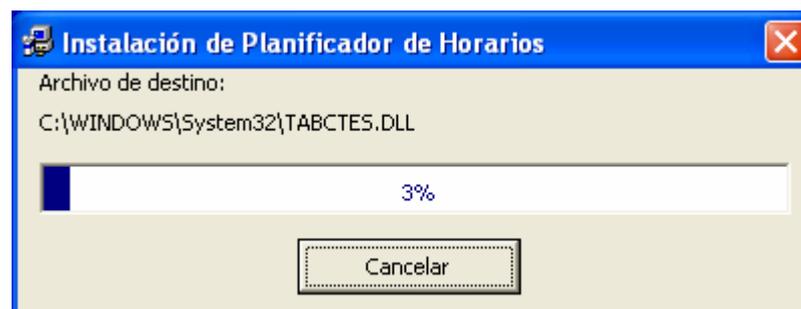


Ilustración D-5: Progreso de la instalación

Al finalizar completase la barra de desplazamiento se presentará una pantalla indicando una instalación exitosa.

BIBLIOGRAFÍA

- [I] Miebach Logística: Consultores e Ingenieros especializados en brindar soluciones integrales factibles para mejorar la eficiencia de las operaciones e inversiones en logística, alineando la cadena de suministro con la estrategia de negocio de las organizaciones. Tomado en septiembre 3 del 2005.
<<http://www.miebach.com/>>.

- [II] Best Practices of Highly Successful Manufacturers for Leveraging Advanced Planning and Scheduling, RSS Solutions Inc., 2004.

- [III] Soft Select Inc. Software empresarial de misión crítica. Tomado en agosto 25 del 2005.
<<http://www.softselect.com>>.

- [IV] When ERP is not enough – A White Paper Advanced Planning & Scheduling RSS Solutions, Inc. 2003.
- [V] Como implantar en Occidente los métodos japoneses de Gerencia. Kazuo Murata y Alan Harrison Legis. 1991.
- [VI] Kewill Systems USA 2003. Tomado en septiembre 3 del 2005 <<http://www.kewill.com/> >
- [VII] Grant Norris. E-Bussines and ERP Transforming the Enterprise. 2000.
- [VIII] Grave Cisterna, Héctor Daniel: ¿Por Que Fallan Las Implementaciones de ERP? .2000.
- [IX] Interwave. Lider en soluciones de negocios y sistemas de planeación. <www.interwavetech.com>.
- [X] Puerto Luis Alfredo. Impacto de las soluciones de Planeación de Recursos Empresariales (E.R.P.), en empresas de Santa Fe de Bogotá. Universidad Javeriana, Santa Fe de Bogotá Colombia. Junio del 2000.
- [XI] Camacho Salvador. Planeación de Recursos Empresariales. 1997.

- [XII] Software Infor SyteLine 7. ERP Extendido. Tomado en septiembre 3 del 2005.
<<http://www.e-cis.com.mx/erp.htm> >
- [XIII] Manejo de la Cadena de Suministros, Universidad Mayor de Santiago, Septiembre de 2002.
- [XIV] Espaciopyme, SA. SCM – Gestión De La Cadena De Suministro, Junio 2003.
- [XV] RSS Solutions Inc. The Roadmap to Advanced Planning and Scheduling - A Practical, Step-by-Step Guide to Whether An Advanced Planning and Scheduling (APS) Solution is Right For Your Organization. Tomado en agosto 20 del 2005
<<http://erp.knowledgestorm.com/>>.
- [XVI] Universidad de Oviedo – Informática de Gijón. Tomado el 29 de junio del 2005.
<<http://www.inforg.uniovi.es/ia/Genetico-TSP/Introduccion.htm>>
- [XVII] Diaz Adenso, Glover Fred, Gonzales JL, Laguna Manuel, Moscato Pablo y Tseng Fan. Optimización Heurística y Redes Neuronales. Editorial Paraninfo. Madrid. 1996.

- [XVIII] David Santo Orcero. Noviembre 2002. Tomado el 30 de junio del 2005.
< <http://www.orcero.org/irbis/disertacion/node207.htm>.>
- [XIX] Genotipo. Tomado en Septiembre 3 del 2005.
<www.inforg.uniovi.es/ia/Genetico-TSP/Genotipo.htm>
- [XX] Obitko, Marek and Slavík, Pavel. Visualization of Genetic Algorithms in a Learning Environment. In: Spring Conference on Computer Graphics, SCCG'99. Bratislava: Comenius University, 1999, p. 101-106. ISBN 80-223-1357-2. Tomado el 20 de agosto del 2005.
<<http://cs.felk.cvut.cz/~xobitko/ga/gaintro.html>, Tomado el 2 de Julio del 2005>.
- [XXI] Marczyk Adam. Algoritmos genéticos y computación evolutiva, 2004. Tomado el 3 de Septiembre del 2004.
<<http://the-geek.org/docs/algen/>>.
- [XXII] Yolis E., Britos P., Sicre J, Servetto A., García-Martínez R, Perichinsky. G. Algoritmos genéticos aplicados a la categorización Automática de documentos Tomado el 25 de Julio del 2005.
<http://www.itba.edu.ar/capis/webcapis/RGMITBA/comunicacionesrgm/cacic2003-categorizaciondedocumentosusandoag.pdf>.

- [XXIII] Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. 2000, 2001, 2002. Tomado en agosto 20 del 2005.
<<http://www.orcero.org/irbis/disertacion/node208.html>>.
- [XXIV] Data Mining Institute, S.L. Tomado el 3 de julio del 2005.
<<http://estadistico.com/dic.html?p=4169&PHPSESSID=2fc2d2b3b0d00fd6ed2e001389cd5f74>>
- [XXV] HEUR NET, Sitio de Investigación y desarrollo en optimización. Tomado en agosto 20 del 2005.
< <http://heur.uv.es/ventana/descripcion.php>>.
- [XXVI] Redes Neuronales. Tomado en agosto 15 del 2005.
<www.inteligenciaartificial.cl/iabot/ciencia/software/ia/sistemase_y_re desn.doc>
- [XXVII] GRASP: Greedy Randomized Adaptive Search Procedures ,Mauricio G.C. Resende (1), José Luis González Velarde ,Tecnológico de Monterrey, Garza Sada 2501 Monterrey, NL 64849 México.
<<http://tornado.dia.fi.upm.es/caepia/numeros/19/resende.pdf>>
- [XXVIII] HEUR NET, Sitio de Investigación y desarrollo en optimización. Tomado en agosto 20 del 2005:
<<http://heur.uv.es/ventana/metodos.php>>

- [XXIX] Rumbaugh James, Blaha Michael, Premerlani William, Hedí Frederick, Lorensen William. Modelado y Diseño Orientado a Objetos. Prentice Hall. Nueva York.1999
- [XXX] Terryense W. Pratt. Lenguajes de Programación: diseño e implementación. Prentice Hall. Mexico-Nueva York. 1998.
- [XXXI] Lee Richard C., Terfernhart William. Practical Object-Oriented Development with UML and Java. Prentice Hall. United States of America. 2002.
- [XXXII] Adenso Díaz, Fred Glover, JL Gonzales, Manuel Laguna, Pablo Moscazo y Fan Tseng. Optimización Heurística y Redes Neuronales. Capítulo 3.1996.
- [XXXIII] How To Perform a SWOT Analysis Tomado el 20 de Julio del 2005.
< www.bplans.com.>
- [XXXIV] Mayores rendimientos: Los beneficios financieros de Windows XP Professional. Tomado el 1 de Septiembre del 2005.
<<http://www.microsoft.com/latam/windowsxp/pro/evaluacion/actualizar/bizval/default.asp>>.

- [XXXV] Ángel Martínez Sánchez (1996: p 85) ESIC Market, Abril-junio.
- [XXXVI] Manejo de la Cadena de Suministros, Universidad Mayor de Santiago, Septiembre del 2002.
- [XXXVII] Adenso Díaz, Fred Glover, JL Gonzáles, Manuel Laguna, Pablo Moscazo y FanTseng. Optimización Heurística y Redes Neuronales. 1996. Capitulo 3. Pagina 72