UCAB Universidad Católica ANDRES BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

DESARROLLAR UN SISTEMA COMPUTACIONAL PARA LA AUTOMATIZACIÓN DE GENERACIÓN DE HORARIOS DE PROFESORES DE LA UNIVERSIDAD CATOLICA ANDRES BELLO.

TRABAJO ESPECIAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

Realizado por: Clara Emir Magin Romero
Daniel Antonio Zamora Bustamante

Tutor: Rosaura Paladino

Fecha: 25 de octubre de 2013

UCAB Universidad Católica ANDRES BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INFORMÁTICA

DESARROLLAR UN SISTEMA COMPUTACIONAL PARA LA AUTOMATIZACIÓN DE GENERACIÓN DE HORARIOS DE PROFESORES DE LA UNIVERSIDAD CATOLICA ANDRES BELLO.

Este jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 19

JURADO EXAMINADOR

Firma:	Firma:	Firma:
Nombre: Susana García	Nombre: Rosaura Paladino	Nombre: Rafael Lara Campos

Realizado por: Clara Emir Magin Romero
Daniel Antonio Zamora Bustamante

Tutor: Rosaura Paladino

Fecha: 25 de octubre de 2013

DEDICATORIA

El dedicar este Trabajo Especial de Grado, nos gustaría ofrecer la obtención de esta grandiosa meta a Nuestros familiares y amigos que disfrutaron, sufrieron y vivieron cada una de los retos que día a día se presentaron y que con mucho esfuerzo y trasnochos fueron superados con éxito en el transcurso de la carrera, sin ustedes no lo hubiésemos logrado.

Con este trabajo nos dimos cuenta que el aprendizaje obtenido nos hizo crecer como personas y que hoy en día como nuevo profesionales con las mejores herramientas y conocimientos.

A todos los queremos y agradecemos el acompañarnos en este largo camino y que aquí compartimos con ustedes nuestro éxito.

Clara Magin y Daniel Zamora

AGRADECIMIENTOS Y/O RECONOCIMIENTOS

A la profesora Rosaura Paladino, por su excelencia como tutora, gracias por su paciencia, atención, disposición, confianza, amistad, apoyo y darnos la oportunidad con el tema de la tesis, confió en nosotros.

Al Profesor Wilmer Pereira, por abrirnos los ojos en el área de Algoritmo Genético, por su invaluable ayuda gracias por la paciencia, inducción y conocimientos aportados, estamos sumamente agradecidos.

A familiares y amigos que siempre estuvieron dispuestos a colaborar y por el simple hecho de estar ahí.

A Luis Borjas, una gran persona durante poco tiempo en la carrera pero que consideramos un amigo y hermano, gracias por cada día hacernos reír con tus ocurrencias y hacer más placentera los últimos semestres de la carrera, Lo logramos.

Muchas personas contribuyeron en el desarrollo de este proyecto, La profesora Susana García por permitirnos realizar este tema y el apoyo dado. Ana Karina Fernández por las asesorías, sumamente agradecidos.

Clara Magin y Daniel Zamora

INDICE DE CONTENIDO

DEDICATORIA 1

AGRADECIMIENTOS Y/O RECONOCIMIENTOS 2

INDICE DE CONTENIDO 3

INDICE DE FIGURAS 7

INDICE DE TABLAS 7

SINOPSIS 8

INTRODUCCIÓN 9

CAPITULO I. PLANTEAMIENTO DEL PROBLEMA 10

1.1 Identificación del Problema 10

1.2 Objetivos 13

1.2.1 Objetivo General 13

1.2.2 Objetivos Específicos 13

1.3 Justificación 13

1.4 Alcance y Limitaciones 14

1.4.1 Alcance 14

1.4.2 Limitaciones 15

CAPITULO II. MARCO DE REFERENCIA / TEORICO 16

2.1 Restricciones 16

2.2 Proceso actual de la Generación de Horarios 16

2.3 Problemas 17

2.7.1 Tipos de Problemas 17

2.4 Técnica Computacional 19

2.5 Clases de Complejidad 20

2.6 Arquitectura del Software 20

2.10.1 Tipos de arquitecturas 21

2.10.2 Arquitectura Cliente – Servidor 21
2.7 Patrones de diseño. 22
2.12.1 Patrón de diseño Modelo-Vista-Presentador 22
2.12.2 Patrón de diseño DAO. 23
2.8 Algoritmo Genético 23
2.11.1 Algoritmo Genético Simple 25
2.11.2 Ventajas y Desventajas 26
2.11.3 Cromosoma 26
2.11.4 Función Evaluación o Aptitud 27
2.11.5 Proceso de Selección 27
2.11.6 Tipos de Selección 28
2.11.7 Cruce 30
2.11.8 Mutación 32
CAPITULO III. METODOLOGIA 33
3.1 Características de la Metodología 33
3.2 Descripción de las fases 34
3.2.1 Definición de especificaciones 34
3.2.2 Diseño Conceptual 34
3.2.3 Desarrollo del Prototipo 35
3.2.4 Pruebas del usuario 35
3.2.5 Implantación 35
3.2.6 Auditoria y seguimiento 36
3.3 Representación gráfica del modelo de prototipo rápido, según sus fases. 36
3.4 Justificación de la metodología implantada 38
CAPITULO IV. DESARROLLO 39

4.1 Fase #1: Definición de Especificaciones 39
4.2 Fase #2: Diseño Conceptual 43
4.3 Fase #3: Desarrollo del Prototipo 45
4.3.1 Representación Cromosómica 46
4.3.2 Codificación del Cromosoma 49
4.3.3 Definición de la población inicial 50
4.3.4 Función aptitud (Cromosoma). 52
4.3.5 Función aptitud (Población) 53
4.3.6 Proceso de selección 53
4.3.7 Cruce 54
4.3.8 Mutación. 55
4.3.9 Función Comprobadora 55
4.4 Fase #4: Pruebas 58
4.5 Fase #5. Implantación. 59
CAPITULO V. RESULTADOS 59
5.1 Fase #1: Definición de especificaciones 59
5.1.1 Herramientas utilizadas 59
5.1.2 Lenguaje C# 60
5.1.3 MySql 60
5.1.4 Arquitectura del software. 61
5.1.7 Entradas al Sistema 62
5.2 Fase #2: Diseño Conceptual 63
5.3 Fase # 3: Desarrollo del Prototipo 64
5.4 Tabla de Resultados Vs Objetivos 81
CAPITULO VI. CONCLUSIONES 83
RECOMENDACIONES 85

GLOSARIO DE TERMINOS. 86

2.9 Horario 86

2.10 Pensum 86

2.11 Prelaciones 86

2.12 Profesor Tiempo Completo 86

2.13 Profesor Tiempo Parcial 86

BIBLIOGRAFIA 87

APENDICES 89

Apendice A. Definicion de Especificaciones. 89

Apéndice A.1. Diccionario de datos 89

Apéndice A.2. Interfaces de Usuario 95

Apendice B. 99

Apéndice B.1. Descripción de los Módulos del Sistema 99

B.2.1 Profesor 99

B.2.2 Materia 99

B.2.3. Coordinación 99

B.2.4 Pensum 99

B.2.5 Disponibilidad Profesor 99

B.2.6 Planificación Coordinación 100

B.2.7 Horario 100

INDICE DE FIGURAS

Figura # 1. Espacio de solución de un problema de optimización de dos parámetros (Garcia, 2010) 18

Figura # 2. Ejemplo de cruzamiento a un punto (Cruce monopunto) (Rafael Lara, 2004) 31

Figura # 3. Ejemplo de Cruzamiento Multipunto(Rafael Lara, 2004) 31

Figura # 4 Generación Horaria Manual. Fuente: Elaboración Propia. 40

Figura # 5 Diagrama de Casos de Uso. Fuente: Elaboración Propia. 44

Figura # 6 Diagrama Entidad-Relación. Fuente: Elaboración Propia 44

Figura # 7 Algoritmo Genético. Fuente: Elaboración Propia. 46

Figura # 8 Primer Cromosoma. Fuente: Elaboración Propia. 48

Figura # 9. Resultados AGs Simple. Fuente: Elaboración Propia. 56

Figura # 10. Resultados AGs aumento de la población. Fuente: Elaboración Propia. 57

Figura # 11 Representación Gráfica de Interacciones entre fases de Desarrollo y Prueba para el modelado del AGs. Fuente: Elaboración Propia. 59

Figura # 12 Representación Cromosómica. Fuente: Elaboración Propia 65

Figura # 13 Función Aptitud (Cromosoma). Fuente: Elaboración Propia. 66

Figura # 14 Calculo de la Función Aptitud. Fuente: Elaboración Propia..... 66

Figura # 15. Función Aptitud (Población). Fuente: Elaboración Propia 67

Figura # 16 Resultados de Ejecución del algoritmo genético con la función comprobadora 80

INDICE DE TABLAS

Tabla # 1 Bloques de Horario. Fuente: Elaboración Propia. 47

Tabla # 2 . Numeración Binaria. Fuente: Elaboración Propia. 49

Tabla # 3. Modelo de Construcción de Ruleta. Fuente: Elaboración Propia. 54

SINOPSIS

Actualmente la elaboración de horarios de los profesores de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello, se encuentra ante la necesidad de mejorar los procesos manuales sin pérdida de funcionalidad y así reemplazar los papeles por sistemas automatizados que puedan ahorrar tiempo y recursos.

Para la generación horaria al inicio de cada periodo académico es necesario asignar y coordinar los recursos, así como también organizar y distribuir los horarios de los profesores teniendo en cuenta condiciones particulares, originadas por el avance natural de los alumnos y otros factores como la disponibilidad de los profesores.

Según lo expuesto anteriormente, se busca implementar un sistema automatizado para la generación de horarios de los profesores de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello, sede Caracas basado en Algoritmo Genético.

Los Algoritmos Genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización, son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacía valores óptimos del problema depende de una buena medida y de una adecuada codificación de las mismas.

La implementación del modelo a desarrollar se basa en el uso de la Metodología Modelo de Prototipo Rápido, ya que moldea el producto final y permite efectuar pruebas para satisfacer las necesidades planteadas actualmente.

Para la implementación del Algoritmo Genético en el problema planteado, se deben seguir ciertos pasos para llegar al producto final. En el Proceso se comienza con la cantidad de secciones a abrir por materias, disponibilidad de los profesores y asignar los bloques de horario para así generar una serie de soluciones que satisfagan las restricciones que impone la Escuela en el proceso de Generación de horarios.

INTRODUCCIÓN

La generación de horarios académicos es un problema complejo que se presenta en una gran cantidad de instituciones, dado al número de restricciones que deben ser tomadas en cuenta para la resolución del mismo. Es por esto que la obtención de una solución factible al problema no es una tarea sencilla y que representa un alto nivel de análisis para los ejecutantes.

Muchas investigaciones han sido realizadas en el campo de la Informática utilizando diversas técnicas entre las cuales se encuentra la utilización del Algoritmo Genético.

Los Algoritmos Genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización, se implementan principalmente en problemas NP-Completo ybásicamente es un algoritmo que mejora las soluciones hasta encontrar una que posiblemente esté cerca del óptimo y cumpla las restricciones impuestas por el problema.

El proceso actual para la Generación de Horarios de los Profesores de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello es manual y sus dificultades cada vez son más grandes, ya queeste tiene un gran número de restricciones que deben ser cumplidas a cabalidad para que la solución obtenida pueda ser tomada en cuenta. La generación de horarios es un trabajo que se realiza semestralmente y para su ejecución se debe tener en cuenta la planificación de las coordinaciones, disponibilidad de profesores, pensum, entre otros.

El trabajo que se presenta a continuación pretende automatizar la generación horaria para la Escuela de Ingeniería Informática mediante la utilización del Algoritmo Genético con la finalidad de obtener una solución que satisfaga todas las restricciones.

CAPITULO I. PLANTEAMIENTO DEL PROBLEMA

1.1 Identificación del Problema

Actualmente en Venezuela existe un gran número de instituciones académicas que presentan dificultades para resolver problemas de asignación horaria a diferentes variables, como por ejemplo: profesores, alumnos, asignaturas, aulas, entre otros. La causa principal de este problema es el constante cambio en las restricciones que permiten la generación de horarios, impidiendo que los proyectos ambientados a este problema sean utilizados luego de su realización.

La mayoría de estas instituciones logran un resultado implementando procesos manuales, para los cuales se necesita un personal especializado y un tiempo notable, obteniendo resultados que en su mayoría pueden ser poco efectivos y que requieren de modificaciones futuras, lo cual puede generar una sensación de tedio, desagrado y desmotivación, en el personal encargado de dicha labor y molestias en los usuarios finales de estos procesos.

El problema de la construcción de horarios radica en proponer una planificación que cumpla con las restricciones impuestas y obtenga una solución de calidad aceptable en un tiempo razonable. Conseguir este propósito no es fácil, puesto que a veces las restricciones en cada generación pueden ser distintas y difíciles de resolver por lo que suelen ser extremadamente consumidoras de tiempo.

Este caso se enfocará en la generación de horarios de los profesores y se tomará como base el Trabajo de Grado realizado por el Profesor Rafael Lara y Astrid Sanabria el cual pretendía solucionar el problema con el desarrollo de un modelo para la asignación de horarios de clases a través del Algoritmo Genético.En Escuela de Ingeniería Informática el proceso de generación horaria es realizado semestralmente tomando en cuenta las restricciones en la planificación y asignación de los horarios de clases; entre las cuales cabe destacar:

- Disponibilidad de horarios de los profesores: en la escuela de Ingeniería Informática, cada semestre los profesores deberán llenar una disponibilidad dependiendo de las materias que dicte y posibilidad de horas a nivel personal. Se debe tener en cuenta que existen 2 tipos de profesores con respecto a su dedicación, los profesores de tiempo completo y los de tiempo convencional. Los profesores de tiempo completo cumplen una jornada laboral en la universidad, por lo tanto no generan mayores inconvenientes ya que su disponibilidad es bastante amplia lo que facilita la ubicación de sus bloques de horario, mientras que los profesores de tiempo convencional son aquellos que solo asisten para cubrir las horas asignadas y su disponibilidad depende en gran parte del resto de sus ocupaciones personales o profesionales, por lo tanto el tiempo a dedicar para impartir clases en las instituciones es menor, generalmente en horarios extremos – principios de la mañana o finales de la tarde – siendo esto un factor determinante.
- Planificación de coordinadores de cátedra: esta restricción se relaciona directamente con la cantidad requerida de profesores para dictar una materia, cantidad de alumnos y secciones a abrir, esta planificación es realizada por los coordinadores de cátedra de ciclo básico (Facultad de Ingeniería) y del ciclo profesional (Escuela de Ingeniería Informática).
- Prelaciones por materia: son de gran importancia para la ubicación de bloques de materias, ya que condicionan aún más el proceso debido a que un alumno podrá ver materias del semestre próximo y del semestre anterior, impidiendo así la colocación de horarios de 3 materias de semestres consecutivos en un mismo bloque al menos que su prelación lo permita. Por ejemplo, al observar el pensum actual de la Escuela de Ingeniería Informática, los horarios de Base de Datos I, Base de Datos II y Desarrollo del Software pueden coincidir, pero Computación Gráfica y Contabilidad General no deberían coincidir con ninguna de las tres materias anteriormente mencionadas.
- Distribución de las horas de clases: esta restricción indica cuantas horas debe dictarse por materia, además del tiempo por clase.

Por las razones anteriormente expuestas, la búsqueda de la mejor solución no es una tarea sencilla; al iniciar cada periodo académico es necesario organizar los horarios de los profesores ya que semestre a semestre se originan condiciones y restricciones propias, lo cual hace este cálculo particular para cada periodo académico.

Actualmente la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello en su sede de Caracas presenta este problema; particularmente partiendo de la experiencia de generación de horarios para el periodo 201221 (octubre 2011 – febrero 2012), el mecanismo utilizado para su ejecución tomó 2 semanas y 3 recursos humanos; siendo el proceso el siguiente, inicialmente se ubican las materias electivas, luego se prosigue con los semestres, comenzando por el décimo hasta llegar al primero, teniendo en cuenta que para las materias pertenecientes al Ciclo Básico se debe esperar que la carpeta de planificación proporcionada por la Facultad de Ingeniería esté disponible. Para la generación del mismo, se utilizaron como principales entradas, las planificaciones de los Coordinadores de cátedra, planificación Facultad de Ingeniería, pensum y disponibilidad de los profesores, teniendo en cuenta que para la asignación de un bloque de horario es necesario que se cumplan ciertas restricciones, siendo estas expuestas, bien sea por las entradas o según el reglamento actual de la Escuela.

Teniendo como base las restricciones y el proceso anteriormente descrito, lo que se espera obtener es la automatización de la generación de horarios para los profesores ofreciendo soluciones aceptables que faciliten la búsqueda y generación, que satisfagan las necesidades de la Escuela de Ingeniería Informática, logrando como ventaja una disminución notable de tiempo empleado para su ejecución, una mínima utilización de recursos y distintas opciones permitiendo así la elección más conveniente para la escuela en ese periodo, sin llegar a esperar la solución óptima ya que se está en presencia de un problema NP completo.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema computacional para la automatización de generación de horarios de profesores de la Universidad Católica Andrés Bello.

1.2.2 Objetivos Específicos

- Estudiar y elegir los módulos para la administración del sistema.
- Estudiar y seleccionar la tecnología a utilizar basada en los estándares del DTI.
- Evaluar los parámetros de entrada y sus posibles variables y determinar cuáles deben ser considerados como factores determinantes en la generación de horarios.
- Implementar los módulos para el manejo de entradas y administración del sistema que satisfagan las necesidades y peticiones.
- Diseñar un modelo de Base de Datos que se ajuste a los requerimientos del sistema.
- Modelar y construir el Algoritmo Genético a implementar.
- Realizar las pruebas del modelo necesarias para monitorear desde la factibilidad de los parámetros hasta la obtención de una solución aceptable siendo esta aquella que ofrece una o varias respuestas que cumpla con todas las restricciones impuestas para la generación de un horario.

1.3 Justificación

El sistema propuesto busca ofrecer una solución tecnológica mediante la automatización del proceso de generación de horarios de profesores, teniendo como objetivo principal reducir tiempo en la obtención de un resultado final, ocurrencias de errores y utilización excesiva de recursos humanos y de oficina, para obtener una solución de calidad aceptable en un tiempo razonable, satisfaciendo las necesidades de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello.

La importancia de esto se debe a que la complejidad del problema es NP completo, lo que significa que no se conoce un método para hallar una

solución óptima en un tiempo razonable, además se debe tener en cuenta que en la actualidad se han realizado una cantidad significativa de investigaciones acerca de este dilema, pero la obtención de un resultado que satisfaga las necesidades actuales y que permita la parametrización no han sido ejecutadas con éxito.

En vista de que el proceso actual es totalmente manual, la presente solución de automatización reduciría la cantidad de recursos necesarios para la ejecución de esta tarea, pudiendo así ser utilizados en otras instancias; reduciendo costos y obteniendo una mejor calidad de trabajo.

En esta solución se dispondrá de las funcionalidades necesarias para la gestión de los factores relacionados a la generación de horarios de los profesores, aportando flexibilidad y adaptabilidad a los cambios de diversos elementos, eficiencia, confiabilidad y desempeño, utilizando arquitectura de vanguardia que facilite la mantenibilidad del sistema.

1.4 Alcance y Limitaciones

1.4.1 Alcance

- Tras la experiencia de generación de horarios para el periodo 201221 (octubre 2011 – febrero 2012), se recopilaron una serie de entradas que podrían ser tomadas en cuenta, aunque luego del estudio de éstas pueden surgir modificaciones. Dichas entradas para el sistema son: horario del semestre anterior, disponibilidad de profesores, planificación de los coordinadores, pensum, prelaciones, artículo 7 del "Reglamento sobre Régimen de Estudios de la Facultad de Ingeniería" y lista de electivas a abrir con su disponibilidad.
- Desarrollar un sistema parametrizable que permita la gestión para el manejo de entradas, con el fin de obtener la solución de un horario de clase para los profesores.
- El módulo para el manejo de entradas será implementado para ser utilizado como aplicación de escritorio, y este permitirá cargar toda la información básica del sistema.

- Se diseñará un modelo de base de datos que se ajuste a los requerimientos que el problema impone, con la finalidad de guardar la data de manera permanente y de forma consistente en el tiempo.
- Durante el desarrollo de esta aplicación, se realizarán pruebas constantes que guiarán y ayudarán a conseguir la mejor solución a este problema, siempre teniendo en cuenta el tiempo de ejecución.
- El sistema se limitara a la generación de horarios de la escuela de Ingeniería Informática de la Universidad Católica Andrés Bello - Sede Caracas.

1.4.2 Limitaciones

- El cambio de las restricciones del problema puede que requiera una modificación importante en la ejecución del Algoritmo Genético lo que puede que conlleve a un replanteamiento del modelo a implementar.

CAPITULO II. MARCO DE REFERENCIA / TEORICO

2.1 Restricciones

Las restricciones son definidas como todas aquellas limitantes que deben ser cumplidas para que la solución al problema pueda ser tomada en cuenta. Las restricciones se clasifican de la siguiente manera:

- Restricciones Fuertes: son aquellas que deben ser cumplidas a cabalidad para que la solución pueda ser tomada en cuenta.
- Restricciones débiles: son aquellas que pueden ser o no cumplidas en la solución del problema, sin embargo, el cumplimiento de esta ofrece una mejor calidad al resultado obtenido.

2.2 Proceso actual de la Generación de Horarios

Actualmente la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello en su sede de Caracas siguen un proceso manual para la generación de horarios de los profesores, dicho proceso requirió para su ejecución dos semanas aproximadamente y tres recursos humanos. El proceso es el siguiente, inicialmente se ubican las materias electivas, luego se prosigue con los semestres, comenzando por el décimo hasta llegar al primero, teniendo en cuenta que para las materias pertenecientes al Ciclo Básico se debe esperar que la carpeta de planificación proporcionada por la Facultad de Ingeniería esté disponible. Para la generación del mismo, se utilizan como principales entradas, las planificaciones de los Coordinadores de cátedra, planificación Facultad de Ingeniería, pensum y disponibilidad de los profesores, teniendo en cuenta que para la asignación de un bloque de horario es necesario que se cumplan ciertas restricciones, siendo estas expuestas, bien sea por las entradas o según el reglamento actual de la Escuela.

Las restricciones consideradas para la resolución del problema son las siguientes:

- Dos o más clases dictadas por un profesor no pueden ser programadas a la misma hora de un mismo día.

- Una clase teórica no puede ser programada a la misma hora que su laboratorio.
- No se puede asignar a un profesor a una materia que no dicte.
- Una materia no puede ser asignada en un mismo bloque horario de otra materia del semestre superior a ella, al menos que exista una prelación entre ellas.
- En un mismo semestre, los bloques asignados a secciones impares no deben chocar con los bloques de secciones impares y de igual forma con los bloques de secciones pares.
- Se deben abrir tantas secciones como hayan sido planificadas en la Planificación de coordinación.
- Un profesor debe dictar todas las clases de una sección de una misma materia.
- En cumplimiento con el artículo 7 del "Reglamento sobre Régimen de Estudios de la Facultad de Ingeniería" los alumnos que tuviesen que cursar su último semestre adicional con una asignatura, la cual tuviere una prelación perteneciente a cualquier de los últimos tres periodos del plan de estudios de la carrera, dicha prelación será exonerada.
- Ninguna materia práctica debe exceder de horario las 6pm.

2.3 Problemas

Se considera importante la investigación realizada por el autor Guillermo García para la definición de un problema, así como también los tipos de problemas(Garcia, 2010).

Un problema es un determinado asunto o una cuestión que requiere de una solución. A nivel social, se trata de alguna situación en concreto que, en el momento en que se logra solucionar, aporta beneficios a la sociedad.

2.7.1 Tipos de Problemas

Un detalle importante a tener en cuenta al momento de plantear problemas es saber determinar la naturaleza del mismo, las ciencias de la computación clasifican a los problemas computables según su naturaleza en los siguientes tipos:

Problemas de Búsqueda: Cuando el problema planteado consiste en buscar en un conjunto de entrada aquellos elementos que cumplan con una determinada condición (o condiciones) se está frente a un Problema de Búsqueda. El espacio de solución de este tipo de problemas son tuplas de elementos, pudiendo ser estos números o cadenas de caracteres (pertenecientes a un lenguaje). Para este caso, el espacio de solución no ayuda a la definición una técnica de resolución, ya que aquí el fono está en estudiar la entrada y no la salida.

Problemas de Optimización: Cuando se debe maximizar o minimizar el resultado arrojado por una función o procedimiento, se dice que es un Problema de Optimización. Su forma usual es una función, llamada función objetivo, que a partir de un grupo de parámetros de entrada, arroja una única salida. El objetivo es encontrar los valores de los parámetros de entrada que arrojen la mejor salida (máximo o mínimo).

El espacio de solución de este tipo de problemas está conformado por tuplas de tamaño igual al número de parámetros que recibe la función objetivo, donde cada posición de la tupla representa un parámetro. A continuación se muestra un gráfico con un Problema de Optimización de dos parámetros.

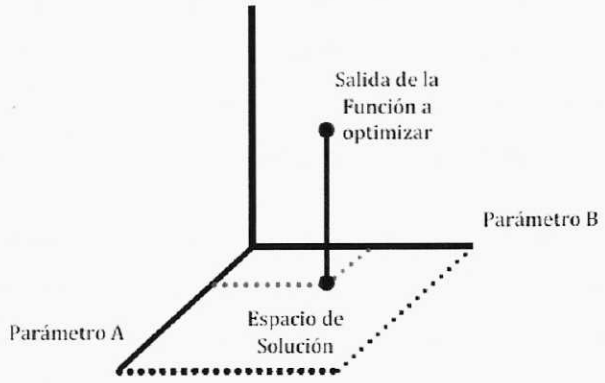


Figura # 1. Espacio de solución de un problema de optimización de dos parámetros (Garcia, 2010)

Problemas de Decisión: Son problemas que buscan determinar la validez de una sentencia. Nótese que una de las características fundamentales que definen a este tipo de problemas es que su espacio de solución es binario: SI o NO, VERDADERO o FALSO, UNO o CERO. Esto influye en la técnica de resolución a aplicar, ya que usualmente se asume que la sentencia es verdadera hasta que se demuestre lo contrario, o viceversa.

Problemas de Conteo: Cuando se necesita saber la cantidad de elementos que cumplen con determinada condición, se está frente a un Problema de Conteo. Usualmente se define de la siguiente manera: dado el conjunto de entrada E, determinar el tamaño de S, donde S es un subconjunto de E formado por los elementos que cumplen con la condición dada. Esta es la forma de los problemas de conteo más estudiada por las Ciencias de la Computación porque son las que requieren un cómputo particular para su resolución.

2.4 Técnica Computacional

Son Técnicas que se buscan dentro de todas las posibles soluciones de un problema, seleccionando la mejor encontrada como resultado. Es oportuno considerar que la búsqueda puede ser realizada de las siguientes formas (Garcia, 2010):

Determinística: Es Determinística cuando para una misma entrada E siempre arroja el mismo resultado R. Adicionalmente, este tipo de técnicas dan como resultado la solución óptima e inmejorable ya que garantizan de una forma u otra que todas las soluciones factibles fueron consideradas.

Heurística: Es heurística cuando cada vez que se aplica, el resultado R puede ser distinto para la misma entrada E. La solución encontrada por este tipo de técnicas es considerada buena, pero jamás se puede decir que es óptima, ya que solo considera una parte de las soluciones factibles, usando un criterio considerado correcto, mas no perfecto para descartar soluciones no óptimas.

2.5 Clases de Complejidad

Es la dificultad implícita en la resolución de un problema que suele medirse en función del tiempo de demora en la ejecución del algoritmo que resuelve el problema, o en función de otros factores tales como la cantidad de memoria ocupada, o la complejidad del código fuente. (Frittelli, 2012)

Es muy común realizar los análisis e interpretaciones en función del tiempo de ejecución. Según el autor Augusto Cortéz los problemas se clasifican en(Cortéz, 2004):

Clase P: Son aquellos que tienen como máximo una complejidad o costo computacional polinómico, es decir, la relación entre el tamaño del problema y su tiempo de ejecución es polinómica

Clase NP: Los problemas con costo no polinomial están agrupados en la clase NP. Estos problemas no tienen una solución algorítmica, es decir, una máquina no puede resolverlos en un tiempo razonable.

Clase NP-Completo: Es el subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas de NP-completo son los problemas más difíciles de NP y muy probablemente no formen parte de la clase de complejidad P. La razón es que de tenerse una solución polinómica para un problema NP-completo, todos los problemas de NP tendrían también una solución en tiempo polinómico.

2.6 Arquitectura del Software

La Arquitectura de Software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software. (Casanovas, 2004)

La arquitectura de software se compone por:

- Clientes y servidores.
- Bases de datos.
- Niveles en sistemas jerárquico.
- Interacciones

Entre los componentes de la arquitectura de software existe un conjunto de interacciones entre las que sobresalen:

- Llamadas a procedimientos.
- Comportamiento de variables.
- Protocolos cliente servidor.
- Transmisión asíncrona de eventos.

2.10.1 Tipos de arquitecturas

Para utilizar la arquitectura de software se sigue un conjunto de patrones arquitectónicos, entre los cuales podemos encontrar:

- Cliente-Servidor
- Blackboard.
- Modelo entre capas.
- Intérprete.
- Orientado a servicios.

2.10.2 Arquitectura Cliente – Servidor

Según la investigación de Tania Sequeira realizada en el 2010 una arquitectura Cliente – Servidor es definida como cualquier combinación de sistemas que pueden colaborar entre sí para dar a los usuarios toda la información que ellos necesiten sin que tengan que saber dónde está ubicada.

Es una arquitectura de procesamientos cooperativa donde uno de los componentes pide servicios a otro.

Es un procesamiento de datos de índole colaborativo entre dos o más computadoras conectadas a una red.

El término cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante.

"Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información". (Sequeira, 2010)

2.7 Patrones de diseño.

Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

2.12.1Patrón de diseño Modelo-Vista-Presentador.

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, el patrón aplicado lo cual sirve para ayudar a realizar pruebas automáticas de la interfaz gráfica es el Patrón Modelo-Vista-Presentador (MVP), la idea es codificar la interfaz de usuario lo más simple posible, teniendo el menor código, toda la lógica de la interfaz de usuario se hace en una clase separada (que se conoce como presentador), que no depende de los componentes de la interfaz gráfica y que, por tanto, es más fácil realizar pruebas. Se puede decir que el patrón MVP es una mejora del patrón Modelo-Vista-Controlador (MVC) basado en tres características:

- La vista no conoce el modelo.
- El presentador es independiente de la tecnología de interfaz de usuario.
- La vista y el presentador son testeables puesto que está basada en un contrato.

2.12.2 Patrón de diseño DAO.

Estepatrón se encargada de gestionar los datos del sistema y se tiene en cuenta que dentro de las aplicaciones el principal objetivo es almacenar información de alguna manera y en algún lugar, este lugar es una base de datos relacional, pero se tiene que tener en cuenta que para cada mecanismo de almacenamiento, existen múltiples formas de acceder a la fuente física de los datos.

El patrón de diseño DAO (Data Access Object), muestra una forma de envolver los datos y evitar que la capa de negocio se involucre, la consulta de datos se hará mediante objetos DAO y la información obtenida es encapsulada y así el negocio solo recibe por medio de objetos la información que necesita y no le interesa realmente que mecanismo utilizo para obtener la información.

2.8 Algoritmo Genético

El Algoritmo Genético (AG) es un método adaptativo que puede usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas. (Intelligent Systems Group).

Los principios básicos de los Algoritmos Genéticos fueron establecidos por Holland (1975), y se encuentran bien descritos en varios textos - Goldberg (1989), Davis (1991), Michalewicz (1992), Reeves (1993).

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer

compañeros tienen mayor probabilidad de generar mayor número de descendientes. Por el contrario, individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "súper individuos", cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los Algoritmos Genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor sería la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos - descendientes de los anteriores - los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor sería la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

El poder de los Algoritmos Genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con respecto al resto de algoritmos de optimización combinatoria. En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al Algoritmo Genético, tanto en rapidez como en eficacia. El gran campo de aplicación de los Algoritmos Genéticos se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas hibridándolas con los Algoritmos Genéticos.

2.11.1 Algoritmo Genético Simple

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

```
    END
    IF la población ha convergido THEN
        Terminado:= TRUE
    END
END
```

2.11.2 Ventajas y Desventajas

2.11.2.1 Ventajas

- No necesitan conocimientos específicos sobre el problema que intentan resolver.(Alfaro)
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se usan para problemas de optimización -maximizar una función objetivo- resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

2.11.2.2 Desventajas

- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, número de generaciones, etc.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole. Por ejemplo, tras el proceso de selección al inicio de la ejecución del Algoritmo Genético puede que se obtenga una solución sin que se haya cumplido un proceso natural del Algoritmo.

2.11.3 Cromosoma

Los cromosomas son secuencias de ADN y sirven como modelo para organismos enteros. Uncromosoma está formado por genes: bloques de ADN.

mejor adaptados (con mayor valor de Aptitud), pero dar probabilidad de ser elegidos a todos los individuos.

El proceso de Selección puede verse desde dos puntos de vista, y en cada caso, puede separarse en dos procesos distintos:

Población

- Determinación de la probabilidad de elección de cada individuo (basada en el valor de la función de Fitness para cada individuo de la Población)
- Selección de los individuos a reproducir, basada en la probabilidad ya definida

Individuo

- Determinación de la probabilidad de ser elegido o no (basada en el valor de la función de Fitness para cada individuo de la Población)
- Determinación de la probabilidad de ser elegido un número determinado de veces (número de veces a ser reproducido)
- Selección efectiva, basada en la probabilidad ya definida

En general, la probabilidad un individuo en particular de ser elegido, no coincide con el número de veces que el individuo es efectivamente elegido. O sea, individuos con alto valor de Fitness son elegidos pocas veces, e individuos con valores menores de Fitness son elegidos una mayor cantidad de veces.

Esto es debido a que los generadores aleatorios no responden exactamente a la distribución propuesta, especialmente en números bajos (cuando el tamaño de la muestra aumenta, esta diferencia disminuye)(Will, 2002).

2.11.6Tipos de Selección

Los tipos de selección son definidos de la siguiente manera: (Parra Truyol, 1997)

Selección por Rueda de Ruleta: Se crea para esta selección una ruleta con los individuos presentes en una generación. Cada individuo tendrá una parte de esa ruleta mayor o menor en función a la puntuación que tenga cada uno. Se hace girar la ruleta y se selecciona el cromosoma en el que se para la ruleta. Obviamente el cromosoma con mayor puntuación saldrá con mayor probabilidad. En caso de que las probabilidades difieran mucho, este método de selección dará problemas puesto que si un cromosoma tiene un 90% de posibilidades de ser seleccionado, el resto apenas saldrá lo que reduciría la diversidad genética.

Selección por Rango: En este método a cada cromosoma se le asigna un rango numérico basado en su aptitud y la selección se realiza en base a este ranking. El problema de esta selección es que la convergencia puede ser más lenta ya que no existe tanta diferencia entre el mejor cromosoma y el resto como ocurría antes.

Selección Elitista: En ciertas ocasiones puede suceder que tras el cruce y la mutación, perdamos el cromosoma con mejor adaptación. Este método de selección copia el mejor cromosoma o alguno de los mejores en la nueva población. El resto se realiza de la misma forma que hemos visto anteriormente. El elitismo puede mejorar el funcionamiento de los algoritmos genéticos al evitar que se pierda la mejor solución. Una variación del elitismo es que el mejor cromosoma solo se copie a la siguiente generación en caso de que tras una reproducción/mutación no se haya generado un cromosoma mejor.

Selección por Estado Estacionario: La descendencia de los individuos seleccionados en cada generación vuelve a la población genética preexistente, reemplazando a algunos de los miembros menos aptos de la anterior generación. Se conservan algunos individuos entre generaciones.

Selección por Torneo: Se escogen de forma aleatoria un número de individuos de la población, y el que tiene puntuación mayor se reproduce, sustituyendo su descendencia al que tiene menor puntuación.

Selección Escalada: Al incrementarse la aptitud media de la población, la fuerza de la presión selectiva también aumenta y la función de aptitud se hace más discriminadora. Este método puede ser útil para seleccionar más tarde, cuando todos los individuos tengan una aptitud relativamente alta y sólo les distingan pequeñas diferencias en la aptitud.

Selección Jerárquica: En esta selección, los individuos atraviesan múltiples rondas de selección en cada generación. Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son evaluados más rigurosamente. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran pocos o nada prometedores, y sometiendo a una evaluación de aptitud más rigurosa y computacionalmente más costosa sólo a los que sobreviven a esta prueba inicial.

2.11.7Cruce

Una vez se realiza la selección de los cromosomas se procede a realizar la reproducción o cruce entre dos de estos cromosomas Más concretamente, el cruce consiste en el intercambio de material genético entre dos cromosomas. El objetivo del cruce es conseguir que el descendiente mejore la aptitud de sus padres.

Para aplicar el cruce habrá que seleccionar con anterioridad dos individuos de la población con una de las diversas técnicas de selección que hemos mencionado en el punto anterior. Además esta selección puede elegir el mismo padre para un descendiente. Esto no es ningún problema pues se asegura la perpetuación del cromosoma más dominante pero si este cruce se produjese con mucha frecuencia podría acarrear consecuencias adversas en caso de que ese cromosoma dominante presente algunos genes no deseados.(Parra Truyol, 1997)

Tipos de Cruzamiento

Cruzamiento a un punto (monopunto): En este tipo de *cruzamiento* se selecciona al azar un punto de cruce del *cromosoma*, y se copia desde el principio de éste hasta el punto de cruce del primer pariente y el resto es copiado desde el punto de cruce hasta el otro pariente.Ver Figura #2.

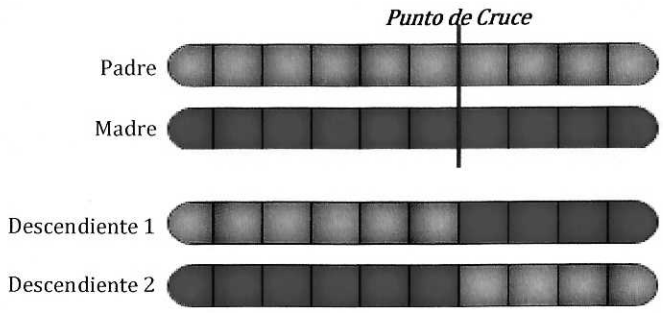


Figura # 2. Ejemplo de cruzamiento a un punto (Cruce monopunto) (Rafael Lara, 2004)

Cruzamiento Multipunto: Dos o más puntos de cruce son escogidos aleatoriamente. El proceso consiste en seleccionar desde el inicio del *cromosoma* hasta el primer punto de cruce del primer pariente. Luego seleccionar la información que empieza desde el primer punto de cruce y termina en el segundo punto de cruce del segundo pariente. Y así sucesivamente hasta llegar al último punto de cruce, como sigue: Ver Figura # 3.

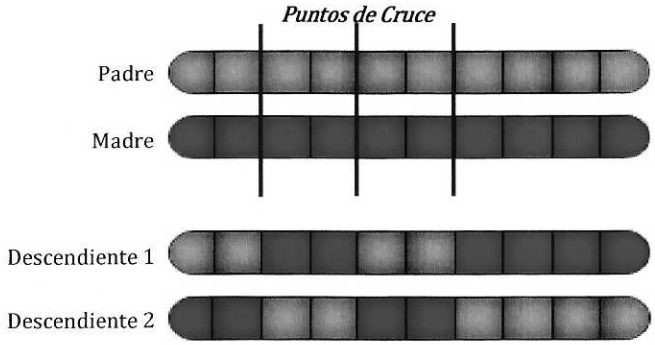


Figura # 3. Ejemplo de Cruzamiento Multipunto(Rafael Lara, 2004)

en modificar el valor en unos decimales. Por último, en una codificación en árbol, la mutación podría radicar en el cambio de operador, de un número o incluso en la mutación de una rama entera.(Parra Truyol, 1997)

CAPITULO III. METODOLOGIA

La metodología seleccionada para el desarrollo del sistema es el modelo de prototipo rápido, la elección de este modelo se debe principalmente a que moldea el producto final y permite efectuar pruebas para satisfacer las necesidades a la Escuela de Ingeniería Informática, esto permite que el desarrollador entienda mejor lo que se debe hacer y se puedan visualizar resultados a corto plazo.

El Modelo de Prototipo rápido comprende la facilidad y aclara ciertos aspectos en los que se aseguran que el desarrollador y el usuario final estén de acuerdo con lo que se necesite, así como también la solución que se propone para la necesidad de minimizar el riesgo y la incertidumbre en el desarrollo del mismo, de igual forma ayuda al Ingeniero y a la Escuela a entender de la mejor manera cual será el resultados del sistema cumpliendo todos los requisitos del mismo.

A continuación se mencionan las características de la Metodología del Prototipo Rápido y se describen las fases de la metodología según Sanchez Jiménez,

3.1 Características de la Metodología

LaMetodología del Prototipo Rápido está orientada al desarrollo de prototipos y fuertemente apoyada en tecnologías de bases de datos y herramientas visuales para desarrollo orientados a objetos; se concentra un gran esfuerzo en la involucración del usuario en 2 fases fundamentales: la definición del problema que se va a abordar y en la ejecución de las pruebas, donde además se potencia el uso de lenguajes de cuarta generación utilizados como lenguajes de consulta para verificar la estructura y funcionalidad del prototipo desarrollado, asegurándose de que su diseño responde a las definiciones especificadas. Como se ha expuesto

anteriormente, la idea fundamental de la metodología es el desarrollo de prototipos.

Un prototipo es un modelo inicial de lo que al final se corresponderá con la base de datos definitiva y sus procedimientos asociados. Este prototipo se procederá a pruebas para comprobar su funcionalidad, de las que surgirán modificaciones que darán origen a un segundo prototipo, versión mejorada y posiblemente ampliada del primero, el cual podrá en algunos casos aportar mejoras en la solución, repitiéndose sucesivamente el proceso hasta alcanzar el prototipo definitivo.

3.2 Descripción de las fases

3.2.1 Definición de especificaciones

En esta primera fase, tiene por objeto revisar la información del problema con el fin de obtener todos los datos necesarios para la resolución y culminación del problema.

En esta fase se debe lograr conseguir la mayor cantidad de información posible sobre el problema.

En esta etapa se definen los siguientes puntos:

- Levantamiento de información.
- Objetivos.
- Limitaciones.
- Alcance.
- Restricciones.
- Formatos de entrada.
- Herramientas a utilizar.

3.2.2 Diseño Conceptual

El Objetivo de esta fase es construir un modelo de información que refleje el esquema conceptual del prototipo. Es muy importante que este modelo este lo más ajustado a la realidad para que el diseño posterior de la base de datos pueda cumplir con sus objetivos.

En esta fase se encuentran los siguientes puntos:

- Ámbito del proyecto.

- Diagrama Entidad-Relación (ER).
- Diccionario de Datos.
- Diagrama de Casos de Uso.
- Interfaces de Usuarios.

3.2.3 Desarrollo del Prototipo

Esta etapa es básicamente técnica, consiste en crear el primer prototipo operativo de la aplicación.

Esta fase consta de dos actividades principales, una para desarrollar la documentación asociada a los prototipos, y la otra parte es el desarrollo técnico del prototipo a partir de los requerimientos y necesidades obtenidos en la fase anteriormente descrita.

Es esta fase se realizan los siguientes puntos:

- Breve descripción de módulos a desarrollar.
- Desarrollo del Prototipo.
- Diseño e Implementación del Algoritmo Genético.

3.2.4 Pruebas del usuario

En esta fase se realizan todas las pruebas necesarias para la validación del prototipo desarrollado en la fase anterior. Si en estas pruebas se detectara la necesidad de modificar el prototipo, para corregir defectos o para añadirle funcionalidad, se volverá a la fase anterior y se realizaran todas las iteraciones necesarias hasta que el usuario se sienta satisfecho con el sistema.

3.2.5 Implantación

En esta fase se ejecutara el plan de formación de usuarios y se llevara a cabo el proceso de migración al entorno de ejecución real de la aplicación. Una vez completada la migración, se realizaran las pruebas finales y se llevara a cabo las actividades correctoras finales, revisando de paso toda la documentación del proyecto.

Como final de esta fase de deberá obtener la aceptación del usuario.

3.2.6 Auditoria y seguimiento

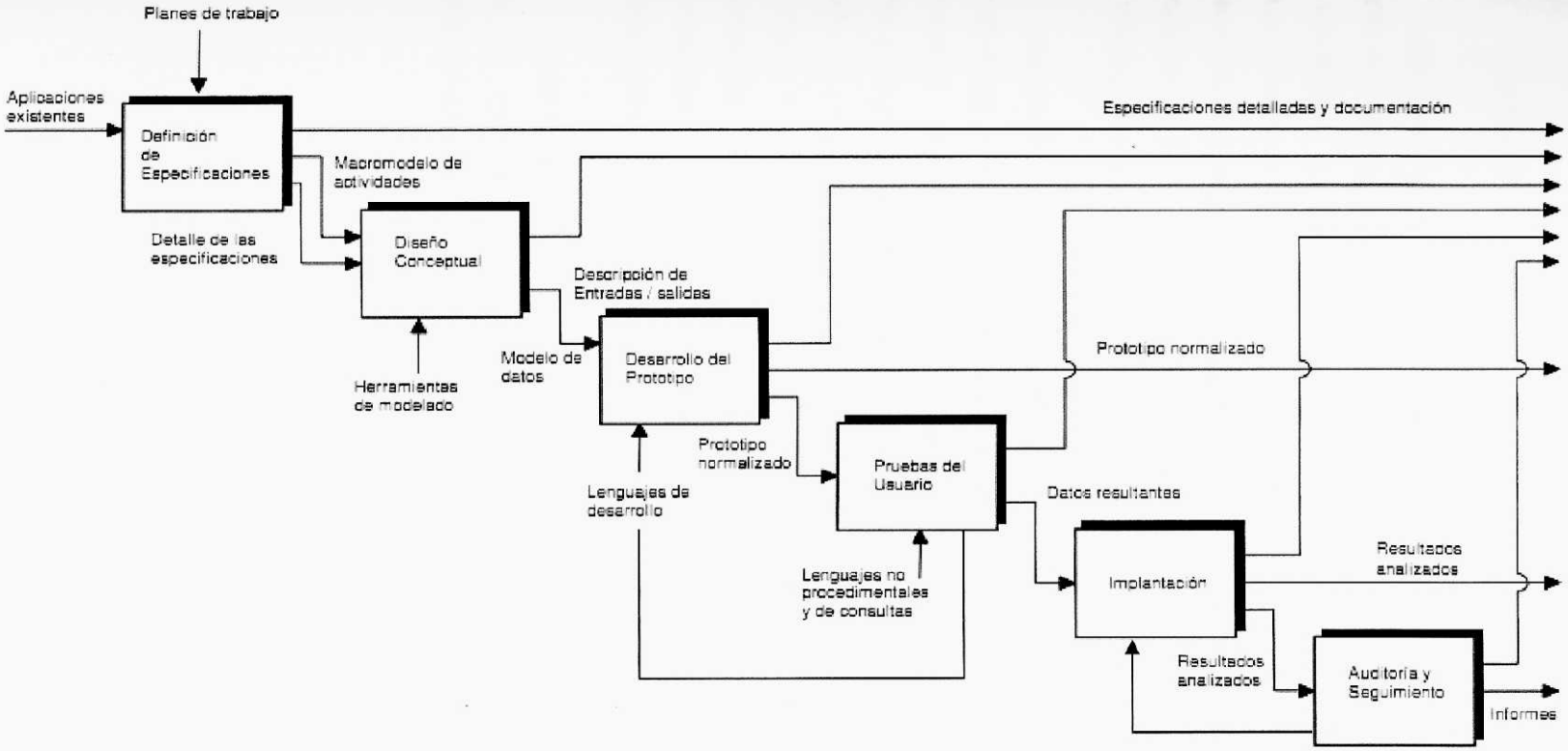
La última fase de la Metodología de Prototipo Rápido consiste en realizar una auditoría al sistema para verificar la calidad del sistema y si es necesario realizarle modificaciones en caso de que ocurran fallas o porque necesita ampliarse de alguna manera.

3.3 Representación gráfica del modelo de prototipo rápido, según sus fases.

Para representar Modelo de Prototipo Rápido en su mayor grado de abstracción, se hace uso de un diagrama en el que se muestran las seis fases de las que se compone.

La simbología empleada en el diagrama es la siguiente: cada una de las cajas representa "algo que hay que hacer", en este caso una fase de la Metodología. A una caja pueden llegar flechas por su lado izquierdo (entradas necesarias para ejecutar la fase), por su parte superior (causas por las que se realiza una fase) y por su parte inferior (herramientas y técnicas con las que se hace lo indicado en la fase). Asimismo, por su parte derecha salen flechas que muestran algo que se ha obtenido en la fase y que pasa normalmente a la fase siguiente o sale directamente fuera del sistema, indicando un producto ya terminado y que no necesita más procesos.

Desde cada fase de la Metodología se puede volver a cualquiera de las fases anteriores. En el diagrama SADT se han omitido voluntariamente estas conexiones para facilitar su estudio.



Scaridad. (s.f). obtenido de http://www.scaridad.com/files/MPR.pdf. Adaptado al problema planteado, colocando un retorno desde las pruebas del usuario al desarrollo de prototipo, siendo esto necesario para el caso donde el prototipo obtenido no proporcione una solución factible y se tenga la posibilidad de modificarse en la etapa de desarrollo.

3.4 Justificación de la metodología implantada

La escogencia de este modelo se debe a que es muy útil cuando el cliente conoce los objetivos generales para el sistema a implementar, pero no se identifican los requisitos detallados de las entradas, procesamiento o salida. Ofrece un mejor enfoque cuando el responsable del desarrollo está indeciso con la eficacia del algoritmo a implementar (en este caso, Algoritmo Genético), de la adaptabilidad de un sistema o de la forma que debería tomar la interacción humano - maquina.

La metodología a usar facilita el uso y la implementación del Algoritmo Genético ya que son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización, requiriendo de un análisis de los datos y de cómo se va a poner en funcionamiento el algoritmo, por lo que el diseño de un prototipo rápido y la ejecución de pruebas es fundamental, para así lograr lo que se quiere del sistema final, en este caso la generación de horarios siendo un problema complejo con una gran cantidad de variables y restricciones hace que la metodología a implementar sea la mejor siguiendo el enfoque sistemático de las etapas sugeridas por el ciclo de vida del software.

Para el presente trabajo se va a implementar la metodología hasta la fase de Pruebas al usuario basándose en el alcance del proyecto anteriormente descrito.

CAPITULO IV. DESARROLLO

El problema planteado lleva a proponer un proceso para el desarrollo del sistema y una planificación con el objetivo de obtener una solución de calidad aceptable en un tiempo razonable, a través de una producción eficaz y eficiente de un producto de software.

Para lograr un resultado exitoso ante un problema computacional complejo que requiere de soluciones factibles, se cuenta con conocimientos en desarrollo de software, herramientas y lenguajes de programación, que serán definidos en este capítulo y los aspectos más importantes del Algoritmo Genético, ya que es el adecuado para la solución del problema, que es la generación de horarios de manera automatizada.

Para el desarrollo de la solución, se utilizó como guía las fases de la metodología anteriormente descrita, a continuación se listan las fases explicando los procesos que ocurrieron en cada una de ellas para la obtención del producto final.

4.1 Fase #1: Definición de Especificaciones

En la ejecución de esta fase se realizaron reuniones para la definición del proceso actual en la generación de horarios de los profesores de la Escuela de Ingeniería Informática que es mostrado gráficamente en el grafico a continuación.

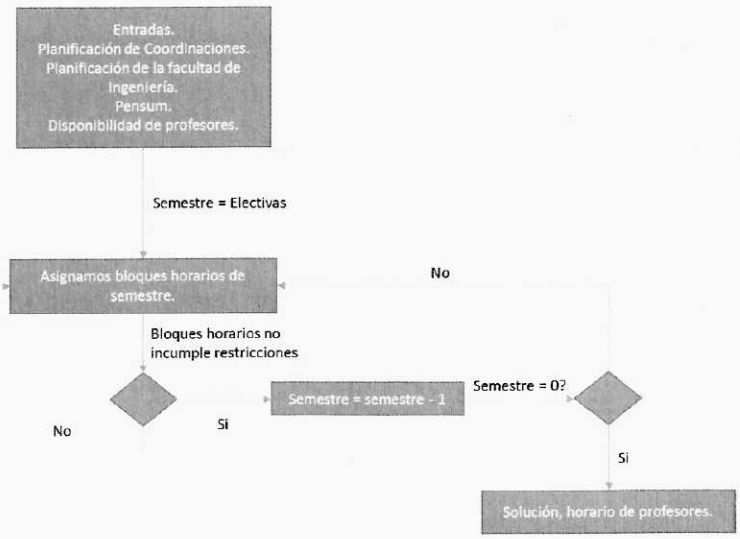


Figura # 4 Generación Horaria Manual. Fuente: Elaboración Propia.

Inicialmente se realizó el proceso de la generación de horario en la Escuela de Ingeniería Informática, junto la beca trabajo Juhilmar Salvatierra, persona encargada en ese momento (julio 2011);Dicho proceso se realiza de forma manual, lo cual llevódos semanas de trabajo con un promedio tres horas diarias para la obtención de un producto final que posteriormente pasa a manos de la Directora de la Escuela para su verificación.

Una vez culminada dicha tarea se obtuvo toda la información necesaria para la descripción del problema, así como también la obtención del alcance, objetivos, limitaciones, restricciones y las entradas al sistema a utilizar en este trabajo investigativo.

A continuación se listan las restricciones del problema así como también las entradas que se tomarán como base para la resolución del problema.

4.1.1 Restricciones.

Para la generación de horarios de los Profesores de la Escuela de Ingeniería Informática, se puede decir que hay varios tipos de restricciones, nombradas y explicadas a continuación:

Las restricciones fuertes.

- Dos o más clases dictadas por un profesor no pueden ser programadas a la misma hora de un mismo día.
- Una clase teórica no puede ser programada a la misma hora que su laboratorio.
- No se puede asignar a un profesor a una materia que no dicte.
- Una materia no puede ser asignada en un mismo bloque horario de otra materia del semestre superior a ella, al menos que exista una prelación entre ellas.

- En un mismo semestre, los bloques asignados a secciones impares no deben chocar con los bloques de secciones impares y de igual forma con los bloques de secciones pares.
- Se deben abrir tantas secciones como hayan sido planificadas en la Planificación de coordinación.
- Un profesor debe dictar todas las clases de una sección de una misma materia.
- En cumplimiento con el artículo 7 del "Reglamento sobre Régimen de Estudios de la Facultad de Ingeniería" los alumnos que tuviesen que cursar su último semestre adicional con una asignatura, la cual tuviere una prelación perteneciente a cualquier de los últimos tres periodos del plan de estudios de la carrera, dicha prelación será exonerada.
- Ninguna materia práctica debe exceder de horario las 6pm.

Las restricciones débiles:

- Se debe respetar la disponibilidad de horario de un profesor siempre y cuando sea posible.
- Deben existir la menor cantidad de horas libres entre bloques asignados.

4.1.1 Entradas al sistema.

- Pensum.
- Prelaciones.
- Planificación de las coordinaciones.
- Disponibilidad de los profesores.
- Lista de electivas a abrir con su disponibilidad.

Tras la definición del problema y aprobación del trabajo por parte de la Escuela, se procedió a evaluar las herramientas a utilizar para la implementación del presente trabajo, tomando en cuenta aquellas que permitan y cumplan con las siguientes características; facilidad para la creación del software, eficacia, portabilidad, rendimiento del sistema y a su

vez se adapten con los componentes del software, siendo estas; lenguaje de programación, IDE y manejador de la base de datos. Se basó la elección de estos en los conocimientos obtenidos durante los estudios de pregrado, adaptabilidad con el algoritmo genético y el alto desempeño necesario para la resolución de problemas NP completos, ya que estos necesitan un alto volumen de iteraciones y altos consumos de recursos computacionales.

Para la elección del lenguaje de programación se tomó como base los lenguajes de programación C# y PHP, siendo el ultimo el estándar utilizado por el Departamento de Tecnología de la Universidad Católica Andrés Bello. Entre ambos lenguaje se seleccionó C# utilizando Visual Studio en su Framework 4.0 y su elección fue basada en los siguientes criterios.

- Es un lenguaje de alto nivel el cual maneja la memoria del computador de forma eficiente, lo que era necesario para la resolución del problema.
- Tras definir el ámbito del problema y determinar que se realizaría una aplicación de escritorio, este IDE en su Framework 4.0 provee de herramientas que permiten el desarrollo de una Interfaz Gráfica amigable de forma rápida.
- El desarrollo guiado por pruebas es más fácil de implementar, ya que existe un Framework de pruebas unitarias (N-Unit) que permite probar funcionalidades a funciones.
- El mantenimiento de la aplicación y escalabilidad del mismo, es más fácil ya que es un lenguaje que la mayoría de los alumnos de la escuela de Ingeniería Informática domina desde el tercer semestre de la carrera.

Para el manejador de base de datos, se seleccionó My-Sql y su elección se basó a que este es gratuito y no tiene limitante de espacio, además, dicho manejador es de fácil integración con el lenguaje seleccionado y es fácil de obtener documentación en la red en caso de dudas.

4.2 Fase #2: Diseño Conceptual

Con los resultados obtenidos en la fase anterior se procede a realizar el diseño conceptual del sistema, dado que el mismo debe ser utilizado por la Escuela de Ingeniería Informática.

Se estudiaron las posibles soluciones que se tomaron en cuenta para la resolución del problema, se procedió a realizar una reunión con el Departamento de Tecnología de Información de la Universidad Católica Andrés Bello (DTI) y se evaluaron los estándares de programación utilizados en dicho departamento. Según el estudio realizado los estándares del DTI no fueron considerados para el desarrollo ya que exigen como lenguaje de programación la utilización de PHP, siendo este un lenguaje poco utilizado para la implementación del Algoritmo Genético y el desarrollo está orientado a una aplicación de escritorio.

Una vez definido el ámbito del sistema a implementar y las herramientas a utilizar se realizaron los componentes y la documentación que permiten llevar la trazabilidad durante el desarrollo del software, siendo estos; diagrama Entidad - Relación, diagramas de Casos de Uso, diccionario de datos e interfaces de usuarios. A continuación se muestran los diagramas que serán utilizados para mantener la trazabilidad durante el desarrollo del presente trabajo:

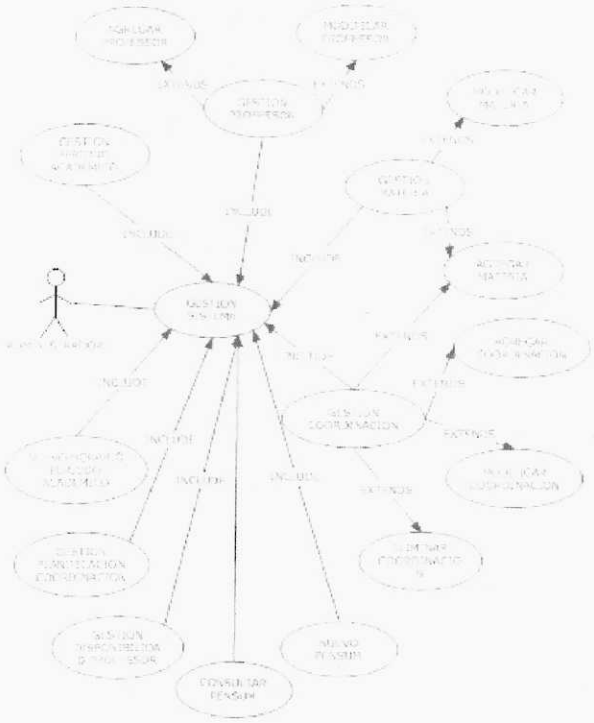


Figura # 5 Diagrama de Casos de Uso. Fuente: Elaboración Propia.

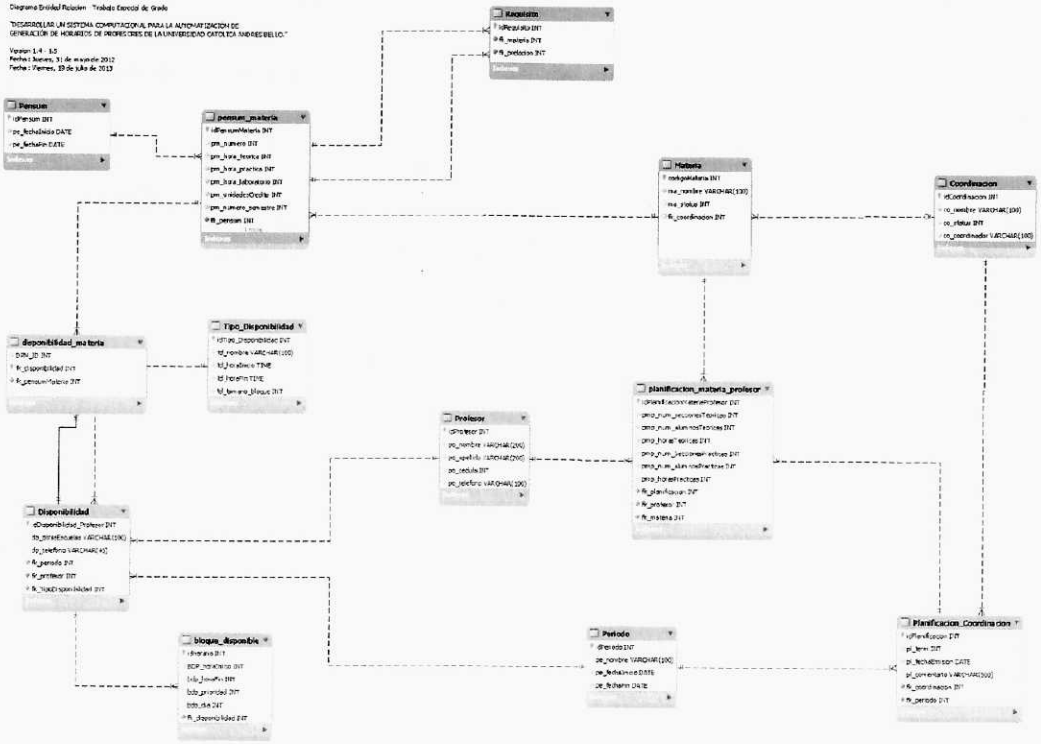


Figura # 6 Diagrama Entidad-Relación. Fuente: Elaboración Propia

4.3 Fase #3: Desarrollo del Prototipo

Basándose en la documentación obtenida en el diseño conceptual, se procedió a realizar el prototipo de la aplicación, dividiendo el desarrollo en dos fases:

1. Desarrollo para la gestión de entradas al sistema.
2. Diseño e Implementación del Algoritmo Genético para la resolución del problema.

Para el desarrollo del sistema para la gestión de entradas, se tomaron en cuenta los casos de uso que debían ser desarrollados y se llevó a cabo un desarrollo guiado por pruebas, el cual se adapta perfectamente con la metodología ya que todo modulo o función que fuese desarrollada era probada. Si la prueba era exitosa se procedía al desarrollo de siguiente modulo, para el caso contrario se corregían los errores y comenzaba el proceso de prueba nuevamente. Dicha interacción entre las fases del desarrollo del prototipo y pruebas fue realizada tantas veces fuera necesario para que un módulo o función implementada cumpliera con las necesidades.

Una vez desarrollado el sistema de gestión de entradas, fue probado completamente validando que este tuviese una interfaz agradable, fácil uso y ofreciera una guía al usuario para su utilización.

Tras culminar el sistema de gestión de entradas se diseñó e implemento el Algoritmo Genético a utilizar, para ello se llevó un proceso de investigación en el cual se evaluaron los trabajos realizados en el área y considerando la opinión de expertos en el tema.

El algoritmo genético desarrollado en el trabajo se resume en el siguiente diagrama:

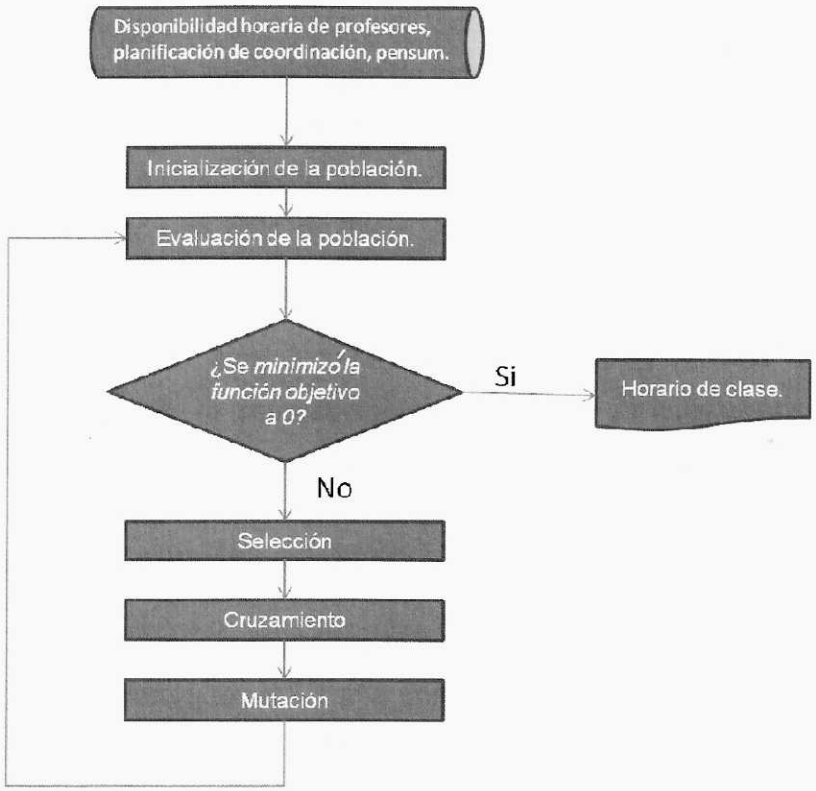


Figura # 7 Algoritmo Genético. Fuente:Elaboración Propia.

Para la implementación del algoritmo genético es necesario definir y tomar en cuenta ciertas consideraciones: representación cromosómica, definición de la población inicial, codificación del cromosoma, función aptitud para la población y el cromosoma, proceso de selección, cruce, mutación y función comprobadora.

Las consideraciones nombradas anteriormente se explicaran a continuación.

4.3.1 Representación Cromosómica

Como se discutió anteriormente la codificación genética es la base del algoritmo genético, la construcción del cromosoma se realizó basándose en el resultado a obtener, siendo este uno o varios horarios.

La representación horaria o solución del problema está compuesta por bloques de materias, una materia es impartida a un conjunto de estudiantes

que son agrupados en una sección y dictada por un profesor. Dicha materia a su vez está compuesta por un grupo de horas que pueden ser o no dictadas por separado. A este grupo de horas se les conoce como clase, siendo esta una combinación de una materia, una sección, un profesor y la duración de la misma, donde tiene una hora de inicio en la que comienza dicha clase y su duración.

Los bloques de horario tienen una duración de 60 min, para este problema se escogió la representación por hora, ya que esta facilita la comparación entre cromosomas y la evaluación de la función aptitud del problema. Un horario de clase está compuesto por un total de 13 bloques por día y 5 días por semana.

Para la numeración de dichos bloques y días, se tomó como base que los días serán enumerados del 1 al 5 comenzando del día lunes hasta el día viernes y los bloques horarios serán enumerados del 1 al 13 comenzando del primer bloque que es de 7:00 am a 8:00 am hasta el último bloque que es de 7:00 pm a 8:00 pm, como es mostrado en la tabla presentada a continuación:

Tabla # 1 Bloques de Horario. Fuente: Elaboración Propia.

Bloque Horario	**Lunes**	**Martes**	**Miércoles**	**Jueves**	**Viernes**
7 – 8	0	0	0	0	0
8 – 9	1	1	1	1	1
9 – 10	2	2	2	2	2
10 – 11	3	3	3	3	3
11 – 12	4	4	4	4	4
12 – 13	5	5	5	5	5
13 – 14	6	6	6	6	6
14 – 15	7	7	7	7	7
15 – 16	8	8	8	8	8
16 – 17	9	9	9	9	9
17 – 18	10	10	10	10	10

18 – 19	11	11	11	11	11
19 – 20	12	12	12	12	12

Inicialmente se tomó en cuenta un cromosoma con muy poca información genética ya que fue construido basándose solo en las entradas al sistema, dicho cromosoma fue representado como se muestra a continuación:

M	P	S	Se	DP	Lunes	Martes	Miércoles	Jueves	Viernes

M – Materia | P – Profesor | S – Sección | Se – Semestre | DP – Disponibilidad profesor

Figura # 8 Primer Cromosoma. Fuente: Elaboración Propia.

Tras reuniones realizadas con los profesores Wilmer Pereira y Rosaura Paladino e investigaciones posteriores se consideró que el cromosoma presentaba muchos inconvenientes que afectarían de forma negativa la evolución de una población en el Algoritmo Genético, ya que la información genética contenida en él era muy pobre y los cálculos para la convergencia del algoritmo aumentaría el número de iteraciones y duraría mucho más tiempo, esto llevó a la reevaluación del cromosoma y a la construcción de uno nuevo que involucrara mayor cantidad de restricciones en su información genética.

Hablando en términos de algoritmo genético, se construyó un cromosoma compuesto por una parte estática y otra variable donde, la información estática es compuesta por todos los datos que no varían a lo largo del proceso evolutivo y que solo cambia si y solo si son modificadas las entradas (cambio de pensum, cambio en la planificación de coordinación, distribución de carga académica, etc.). Dicha información es la cabecera de cromosoma y contiene la siguiente información.

- Materia
- Semestre
- Sección

- Materia a las que prela.
- Formato en el que se dicta la materia (Teórica o Práctica).
- Profesor.
- Disponibilidad del profesor.

Mientras tanto, la información que varía constantemente en el proceso evolutivo es la que constituye la cola del cromosoma y que posee el material genético real está compuesto por la siguiente información.

- Días de la semana.
- Hora de inicio.
- Duración del bloque.

Como medida de simplificación, en este trabajo se utiliza un esquema de cromosoma de longitud fija. Esto es, cada cromosoma mantendrá el mismo número de genes de manera invariable a lo largo de las generaciones.

4.3.2 Codificación del Cromosoma

Para la codificación del cromosoma, se tomó como base una codificación binaria de 4 bits, es importante destacar que el cromosoma solo es codificado en su parte estática, es decir, los bloques horarios.

A continuación se muestra como se encuentra codificados los bloques por cada día de clase.

Tabla # 2 . Numeración Binaria. Fuente: Elaboración Propia.

Bloque Horario	**Lunes**	**Martes**	**Miércoles**	**Jueves**	**Viernes**
7 – 8	0001	0001	0001	0001	0001
8 – 9	0010	0010	0010	0010	0010
9 – 10	0011	0011	0011	0011	0011
10 – 11	0100	0100	0100	0100	0100
11 – 12	0101	0101	0101	0101	0101
12 – 13	0110	0110	0110	0110	0110

13 – 14	0111	0111	0111	0111	0111
14 – 15	1000	1000	1000	1000	1000
15 – 16	1001	1001	1001	1001	1001
16 – 17	1010	1010	1010	1010	1010
17 – 18	1011	1011	1011	1011	1011
18 – 19	1100	1100	1100	1100	1100
19 – 20	1101	1101	1101	1101	1101

4.3.3Definición de la población inicial

Para que el proceso evolutivo se lleve a cabo es necesaria la creación de una población inicial. La población inicial es un proceso que no tiene una única solución, por lo que se presta a la experimentación constante.

Para el presente trabajo la creación de la población inicial se llevó a cabo tomando en cuenta todos los procesos e información que constituye el cromosoma y es importante tener en cuenta que el tamaño de esta permanecerá constante en el transcurso del proceso evolutivo. Para la construcción del cromosoma se comienza por su parte estática para luego completar su parte variable.

Adicionalmente, se crean copias locales de la información referente a pensum, planificación de la coordinación y profesores. Con esta información se procede a crear la parte estática de los cromosomas.

Comenzando por las materias electivas y el décimo semestre hasta llegar al primer semestre, son tomadas las materias para crear tantos cromosomas como sean necesarios, dichos cromosomas corresponden a su vez a una sección. Al seleccionar la materia, se asigna una sección correspondiente disponible en la planificación de la coordinación, el semestre dependiendo de su ubicación en el pensum y las materias las cuales prela. Una vez completado este proceso para continuar con la creación de la parte estática del cromosoma, se debe asignar el profesor que va a dictar la

sección, su disponibilidad y si este cromosoma va a ser intocable en el proceso evolutivo.

Para seleccionar el profesor que va a dictar dicha materia, se toma como primera opción el profesor que es designado por la coordinación, en el caso de que este profesor no disponga de horas suficientes es tomado cualquier otro profesor que dentro de su disponibilidad haya incluido la materia en cuestión, sin embargo, este criterio no se puede aplicar indefinidamente ya que puede darse el caso que sólo exista un profesor disponible o que los horarios de todos aquellos que estén disponibles hayan sido copados por otra materia, para este caso se selecciona el profesor indicado por la coordinación y se viola su disponibilidad.

Una vez asignado el profesor a la parte estática de cromosoma, se completa este con su disponibilidad.

Para finalizar la construcción de la parte estática de los cromosomas pertenecientes a la población inicial, se debe determinar si el profesor es "Variable" o "Estática". Dicho concepto hace referencia y permite saber si un cromosoma puede ser tomado en cuenta para la evolución de la población. Un cromosoma es Estático, si se cumplen las siguientes condiciones:

- En la planificación de la coordinación se especifica que una materia M debe ser dictada por un profesor P en una hora especifica H.
- Si al profesor se le debe asegurar carga académica ya que lleva más de 2 semestres dictando la misma materia, y su disponibilidad es igual al total de horas de la materia.

Para el resto de los casos los cromosomas son variables, es decir, pueden formar parte del proceso evolutivo.

El proceso anteriormente descrito debe repetirse por cada una de las materias del pensum y cada vez que sea construida la parte estática del cromosoma deben colocarse como No Disponible cada recurso utilizado (Horas del profesor en su disponibilidad y sección asignada).

Para la construcción de la parte variable del cromosoma, se debe asignar tantos bloques de clases sean necesarios para cubrir el total de las horas requeridas por una materia. Para ello, se divide el total de horas de la materia en la cantidad de bloques horarios necesarios, luego de esto se selecciona un día aleatorio de la semana hasta encontrar uno donde no haya sido ubicada una clase y partiendo de esto se selecciona una hora de inicio para la materia, generando un numero aleatorio del 1 al 13 que representa los bloques horarios en los cuales está permitido dictar clases dentro de la UCAB, este bloque es codificado e incluido al cromosoma.

Dicha operación, anteriormente descrita es realizada hasta completar el total de la población.

4.3.4 Función aptitud (Cromosoma).

Durante el proceso evolutivo es necesario determinar cuáles de los individuos son los más aptos para formar parte de una nueva población, para ello se necesita de una función que permite la evaluación individual, tomando en cuenta las restricciones que cumplen e incumplen cada uno de ellos.

En el presente trabajo se implementó un sistema de banderas que permite marcar a los cromosomas como tocables o intocables dentro del proceso evolutivo,permitiendo optimizar el proceso de selección y cruzamiento dentro de la población. Los criterios utilizados para determinar si un cromosoma es tocable o intocable son los siguientes:

- En la planificación de la coordinación se especifica que una materia M debe ser dictada por un profesor P en una hora especifica H.
- Si al profesor se le debe asegurar carga académica ya que lleva más de 2 semestres dictando la misma materia, y su disponibilidad es igual al total de horas de la materia.
- Si un semestre evaluado no incumple ninguna de las restricciones fuertes del problema, todos los cromosomas pertenecientes a este semestre son colocados como intocables.

4.3.5 Función aptitud (Población).

Esta función permite la evaluación completa de la población determinando si la solución puede ser tomada en cuenta o no.

Para el cálculo de la función Fitness se evalúan cada una de las restricciones fuertespara cada uno de los cromosomas de la población y tras realizar la sumatoria de las infracciones de las restricciones se determina el valor de la función para la población y se comprueba si la solución es factible o no.

El objetivo de esta función es minimizar la cantidad de errores e incumplimientos de las restricciones que pueda tener una población.

4.3.6 Proceso de selección

El proceso de selección implementado para este trabajo después de haber comparado los métodos de selección descritos en el marco teórico fue el de Ruleta, ya que es el que mejor se adapta porque tras el cálculo de la función aptitud de los cromosomas el puntaje obtenido o calificaciones por cada uno de ellos era muy parecido, por lo que el beneficio de selección probabilístico que ofrece este método complementa la selección de los individuos más aptos, sin embargo existe la posibilidad de que individuos poco aptos o con baja calificación sean seleccionados en el proceso evolutivo.

La implementación de ruleta se realizó bajo los siguientes lineamientos:

- Por cada cromosoma se evalúa la función Aptitud.
- Se calcula el total obtenido por todos los cromosomas de la población.
- Se calcula el espacio porcentual que ocupa dicho cromosoma y su calificación dentro de la ruleta.

El número de segmentos de la ruleta es igual al tamaño de la población a evaluar y este disminuye a medida que se seleccionan los cromosomas para los cruces.

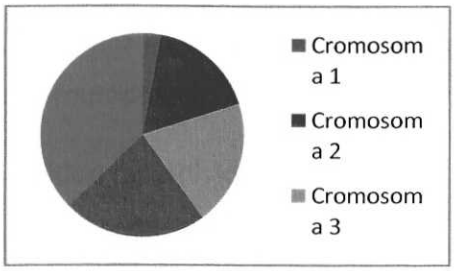
Una vez asignado un segmento de la ruleta a cada cromosoma la ruleta es girada n veces hasta encontrar dos individuos que serán

seleccionados para ser cruzados entre ellos, esto da origen a dos nuevos individuos y a su vez la muerte de sus padres, ya que el tamaño de la población no es variable desde el inicio del algoritmo genético.

En el siguiente cuadro se muestra de forma gráfica como es calculado los rangos porcentuales de cada individuo dentro de la ruleta.

Tabla # 3. Modelo de Construcción de Ruleta. Fuente: Elaboración Propia.

Cromosoma	Aptitud	%Total	Rango Inicial	Rango Final
Cromosoma 1	1	0,08	-	0,08
Cromosoma 2	5	0,38	0,08	0,46
Cromosoma 3	1	0,08	0,46	0,54
Cromosoma 4	1	0,08	0,54	0,62
Cromosoma 5	5	0,38	0,62	1,00
Total	13	1,00		



4.3.7 Cruce

El proceso de cruzamiento consiste en seleccionar genes aleatorios de dos cromosomas y alternarlos entre ellos. En el presente trabajo se utilizó un sistema de cruce monopunto.

En este proceso se toma como base los cromosomas seleccionados por el modelo de ruleta y de forma aleatoria se escoge para cada uno de ellos un punto de cruce, luego la información genética es intercambiada entre ellos para dar origen a dos nuevos cromosomas. El proceso se repite hasta haber seleccionado todos los cromosomas de una población A hasta generar una población B compuesta por los sucesores de los cromosomas padres cruzados.

Se eligió el sistema de cruce monopunto ya que el comportamiento evolutivo de la población guarda coherencia al crear una nueva, es decir al solo intercambiar una sola información genética se respeta la planificación de la materia como viene dada en el pensum, a diferencia de que el sistema multipunto puede crear sucesores cuyo total de horas por materia puede superar la planificación que tiene en el pensum.

4.3.8 Mutación.

El operador genético de mutación pretende modificar aleatoriamente genes de un individuo seleccionado; de tal forma que permita aumentar la variedad genética dentro de la población y a su vez promover la creación de nuevos individuos que pueden o no mejorar el proceso evolutivo.

Para este trabajo el proceso de mutación se basa en la elección de una cantidad de individuos de la población que serán tomadas para mutar de manera aleatoria, dicha cantidad es especificada como un valor constante al inicio del proceso evolutivo y por cada nueva generación son seleccionados los individuos que van a ser mutados siguiendo los siguientes criterios:

- Se selecciona la cantidad de individuos a mutar.
- Por cada uno de los individuos seleccionados se toma un gen donde uno de sus bits será modificado.

4.3.9 Función Comprobadora

Durante el desarrollo del algoritmo genético fueron probadas varias técnicas que ayudan a la convergencia del proceso evolutivo, es decir, minimizar la función objetivo del problema al mínimo requerido.

Inicialmente se implementó un algoritmo genético simple donde se toma el proceso de selección por ruleta, operadores de cruce y mutación, este mecanismo utilizado no dio resultados factibles para el problema, ya que al comenzar el proceso evolutivo la función objetivo de la población se mantenía en un rango de valores sin mejoras a medida que pasaba el tiempo. Se observó que no mejoraba dado que la diversidad de la población era limitada.

A continuación se presenta un gráfico que representa los valores obtenidos durante la ejecución del proceso anteriormente descrito en 500 iteraciones, donde el eje X representa el número de iteraciones y el eje Y representa los valores de convergencia:

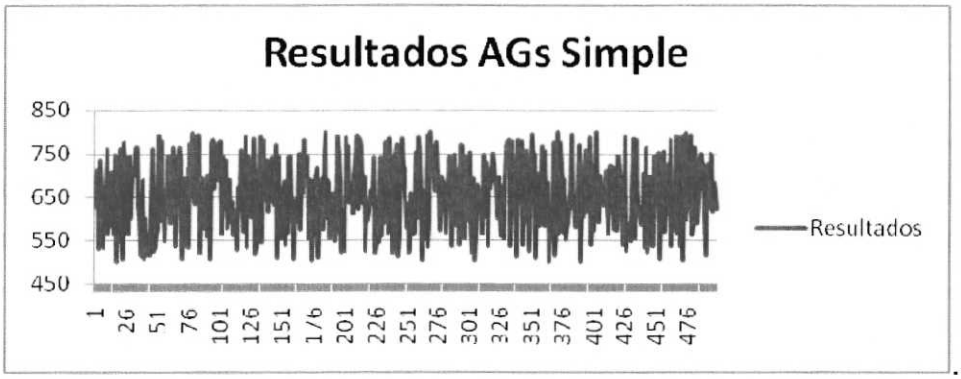


Figura # 9. Resultados AGs Simple. Fuente: Elaboración Propia.

Dado que los resultados obtenidos no eran favorables para el presente trabajo, se optó por buscar alternativas que permitan al algoritmo converger en su evolución, para ello se pensó en aumentar el tamaño de la población y así aumentar la diversidad de individuos para que los procesos de cruce y mutación seleccionen individuos más aptos. Esta solución no funcionó ya que al aumentar la diversidad el cálculo de la función aptitud para la población tomaba mayor tiempo y a su vez por cada proceso evolutivo se incumplían mayor cantidad de restricciones.

A continuación se muestra un gráfico que representa los valores obtenidos durante la ejecución del proceso anteriormente descrito en 500

iteraciones, donde el eje X representa el número de iteraciones y el eje Y representa los valores de convergencia:

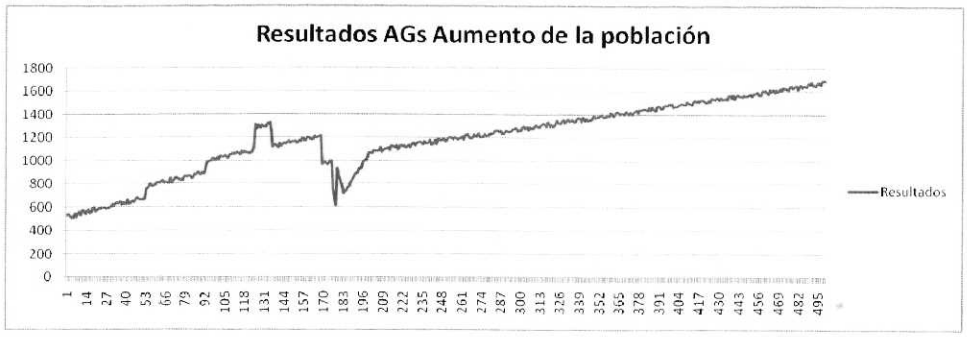


Figura # 10. Resultados AGs aumento de la población. Fuente: Elaboración Propia.

En los procesos descritos anteriormente no se encontró ninguna solución factible para la generación horaria, para ello se intentó simular el proceso actual que lleva la Escuela utilizando Algoritmo Genético, es decir, primero se genera la población inicial luego se calcula la función Fitness de la población, se aplican los operadores de cruzamiento y mutación y se genera una nueva población, este proceso se repite una y otra vez hasta encontrar un resultado al problema aunque luego de cada nueva generación se aplica una función comprobadora que funciona de la siguiente manera.

La función comprobadora comienza por las materias electivas hasta llegar al primer semestre, evalúa las restricciones en cada una de las materias y si no hay incumplimiento de las restricciones los cromosomas son colocados como estáticos, una vez colocados los cromosomas como estáticos se guarda la población como estaba en ese momento, dicha operación se repite por cada generación de población y hasta no colocar como estático el semestre estudiado no se procede a la evaluación del siguiente semestre inferior, dicho criterio no puede ser repetido infinitas veces porque puede existir el caso en que sean colocados los cromosomas de un semestre como estáticos y los cromosomas del siguiente semestre inferior no tenga un bloque factible disponible. Para este caso se regresa la población al momento en donde fue

guardada y se procede a evaluar el semestre que fue colocado como intocable en su momento.

4.4 Fase #4: Pruebas

La fase de pruebas constituyo un punto clave dentro del desarrollo del presente Trabajo Especial de Grado.

La ejecución de esta fase fue llevada en conjunto con la etapa de desarrollo del proyecto. Dicha fase de la metodología se realizó haciendo uso del modelo de desarrollo guiado por pruebas el cual promueve que cada módulo o función implementada debía ser probado inmediatamente con la finalidad de mitigar fallas en el sistema durante el desarrollo del mismo.

Durante el desarrollo del proyecto las interacciones entre la fase de Desarrollo y Pruebas fueron realizadas de forma constantes, tanto para la implementación del sistema de gestión de entradas como para el modelado del Algoritmo Genético.

Para diseñar e implementar un modelo de Algoritmo Genético que permitiera encontrar una solución factible al problema se realizaron tres interacciones que permitieron la ejecución de cada uno de los modelos propuestos y que fueron explicados en la fase anterior de la metodología. A continuación se muestra de forma gráfica las interacciones realizadas entre la fase de Desarrollo y Prueba para la ejecución de cada uno.

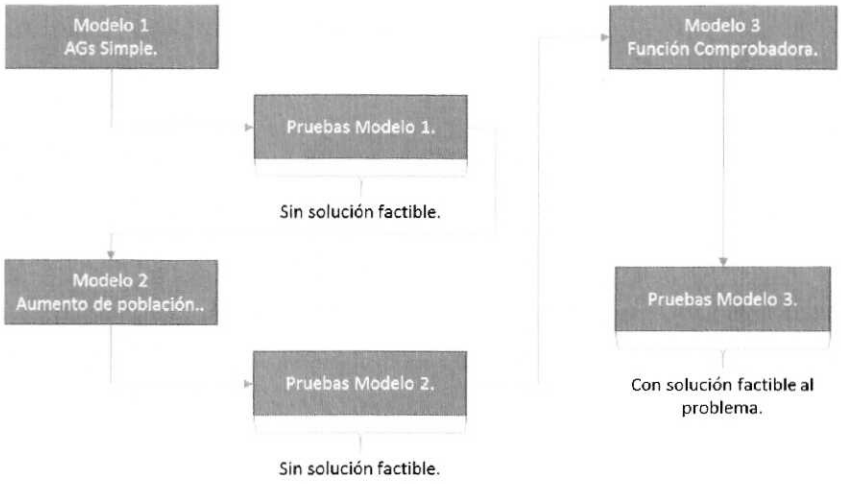


Figura # 11 Representación Gráfica de Interacciones entre fases de Desarrollo y Prueba para el modelado del AGs. Fuente: Elaboración Propia.

4.5 Fase #5. Implantación.

Dicha fase no fue ejecutada, sin embargo, esta será llevada a cabo para el próximo proceso de generación de horario de la Escuela de Ingeniería Informática para el periodo académico que tendrá inicio el año 2014 en su primera etapa.

CAPITULO V. RESULTADOS

En este capítulo, se explican los diversos resultados obtenidos en cada una de las fases que se representan en la metodología aplicada, basándose en un análisis, implementación y desarrollo de las mismas. En cada una de las fases se aprecian diferentes resultados que ayudaron para el desarrollo exitoso del proyecto.

A continuación se expondrán los resultados obtenidos en cada una de las fases de la metodología, y la explicación de los productos finales.

5.1 Fase #1: Definición de especificaciones

En la ejecución de esta fase se realizó un estudio completo del problema en el cual se evaluó el proceso que se lleva a cabo en la escuela de Ingeniería Informática, lo cual ayudó a la obtención de los objetivos, limitaciones, alcance, restricciones, entradas al sistema y herramientas a utilizar, que serán descritas a continuación.

5.1.1 Herramientas utilizadas

El sistema informático a implementar se basa en las diferentes fases de su ciclo de vida, desde la captura de requisitos hasta su mantenimiento. En la actualidad se cuenta con múltiples herramientas que facilitan la creación, eficacia, portabilidad y rendimiento de un sistema. A continuación se describen las herramientas utilizadas en el presente trabajo y la justificación de su uso.

5.1.2 Lenguaje C#

Tomando como base que el ámbito del proyecto fue definido como una aplicación de escritorio y que los estándares del DTI exigían como lenguaje de desarrollo PHP, el lenguaje seleccionado fue C#, este es un lenguaje de propósito general diseñado por Microsoft para su plataforma .NET de desarrollo.

A diferencia de PHP el lenguaje C# es un lenguaje de alto nivel que ofrece la posibilidad de construir código robusto gracias a la seguridad de tipos y a la gestión automática de memoria.

Por otro lado, el lenguaje C# facilitó a el desarrollo guiado por pruebas mediante el uso del Framework N-Unit el cual permite probar funcionalidades especificas del sistema, ayudando a encontrar más fácilmente posibles errores en la aplicación.

El desarrollo del presente Trabajo Especial de Grado está orientado a una aplicación de escritorio, dado a esta situación, la elección del lenguaje se hizo utilizando como criterio base la utilización de un lenguaje e IDE (Entorno de desarrollo integrado) que permitirá la construcción de las interfaces de usuarios de forma rápida y sencilla, además de ser un lenguaje orientado a objetos considerando que la arquitectura del sistema será implementado en capas, se hará más fácil la escalabilidad, mantenibilidad del sistema y que para futuras modificaciones y soporte del mismo sea un lenguaje dominado por un porcentaje alto de personas dentro de la Escuela de Ingeniería Informática.

5.1.3 MySQL

En el proyecto el uso de un manejador de base de datos permite definir de manera más ordenada las diferentes entradas y salidas, así como también relacionar la información y la implementación del Algoritmo Genético en el momento de búsqueda y comparación de datos que se requiere para el prototipo desarrollado en este proyecto. La mejor opción es el uso de una base de datos ya que es rápida su administración, menos laborioso, es actual

y cómodo para los usuarios del sistema. La base de datos se manejara a través de MySQL.

El motivo por el cual se eligió MySQL como manejador de base de datos para el proyecto en primer lugar fue por lo económico, es OpenSource, en segundo lugar es un producto con reconocido prestigio, fiabilidad, velocidad, rendimiento, facilidad de administración y conexión con otros productos y su documentación es buena con una evolución y soporte.

5.1.4 Arquitectura del software.

Para el desarrollo del sistema se utilizó una arquitectura de tres niveles o también conocida como arquitectura de tres capas, dicha arquitectura define como organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores a ella. Este patrón es importante porque simplifica la compresión y organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores y estás pueden ser modificadas o sustituidas por otro componente sin afectar el funcionamiento.

Tras ser definida la arquitectura del sistema, se adoptaron patrones de diseño que facilitarían la definición de estructuras con las que se va a construir sistemas de software. Los patrones de diseño describen un problema que ocurre más de una vez y son utilizados para facilitar la reutilización del código, mantenibilidad del sistema y proveer un mayor grado de escalabilidad al sistema.

Al seleccionar un patrón de diseño se pretende:

- Proporcionar catálogo de elementos reusables en el diseño de sistemas de software.
- Formalizar un vocabulario común.
- Estandarizar el modo en que se implementa la solución.
- Facilitar la agregación de nuevas funciones al sistema.

- Facilitar el entendimiento del código fuente.

Luego de conocer la arquitectura del sistema y las necesidades para el correcto diseño de la aplicación, se adoptaron patrones de diseños dependiendo de la capa en la que se trabajaría, dichos patrones de diseño son los siguientes:

- Capa de presentación: Patrón de diseño MVP.
- Capa de acceso a datos: DAO.

5.1.7Entradas al Sistema

Para la generación de horario en la Escuela de Ingeniería Informática influyen los siguientes factores:

1. Pensum: para la generación de horarios es muy importante la utilización del pensum actual, este nos provee de información que es utilizada para la construcción de Horarios para los Profesores, como por ejemplo, prelaciones, número de créditos y número de horas teóricas y prácticas por materia.
2. Prelaciones: las prelaciones es de las restricciones con más peso dentro de la generación y es importante cumplirlas al 100% para que el resultado sea el buscado y el más cercano al optimo, ya que es posible que tres asignaturas consecutivas dentro de una cadena coincidan en las horas de clases, es decir, los horarios de las materias que no forman parte de una cadena dentro del pensum no deberían coincidir para tres semestres consecutivos. Por ejemplo, los horarios de Estructura del Computador, Sistemas de Operación y Redes de Computación I pudieran coincidir, pero Circuitos Electrónicos y Calculo IV, no deberían coincidir con ninguna de las tres materias anteriormente mencionadas.
3. Planificación de los coordinadores de cátedra: para cada semestre la planificación dada por los coordinadores es distinta, esto se debe a que todos los semestres la cantidad de alumno por materia es diferente, lo que hace que sea necesario la apertura o cierre de secciones.

4. Disponibilidad horaria del personal docente: En las Instituciones de Educación Superior se encuentran profesores contratados a tiempo completo, los cuales pasan la mayor parte del tiempo en las mismas, y profesores contratados a tiempo convencional, esto es, quienes generalmente asisten a la Universidad para cubrir las horas asignadas. Este tipo de profesores abarca la mayoría del total de docentes de las instituciones. Para la planificación de horarios, los profesores a tiempo completo no generan mayores inconvenientes, ya que el período de disponibilidad es bastante amplio, en tanto que para los profesores contratados, este período depende en gran parte del resto de sus ocupaciones personales o profesionales, por lo tanto el tiempo que les es posible dedicar a impartir clases en las instituciones es menor, generalmente en horas extremas - principios de la mañana o finales de la tarde – lo cual es un factor determinante en la cantidad de horas libres en los horarios y la cantidad de profesores requeridos por materia.

5. Artículos importantes: La escuela de Ingeniería Informática cuenta con artículos que deben ser tomados en cuenta para la generación horaria de los profesores, estos artículos ofrecen una condición diferente en ciertas materias y son capaces de alterar algunos de los factores antes mencionados. Estos artículos son:

a. En cumplimiento con el artículo 7 del "Reglamento sobre Régimen de Estudios de la Facultad de Ingeniería" los alumnos que tuviesen que cursar su último semestre adicional con una asignatura, la cual tuviere una prelación perteneciente a cualquier de los últimos tres periodos del plan de estudios de la carrera, dicha prelación será exonerada.

6. Lista de electivas a abrir con su disponibilidad: Esta restricción se relaciona directamente con la cantidad de electivas a abrir para un periodo académico, es importante tener que cuenta que para las electivas también existen prelaciones y es necesario tomar en cuenta los factores antes mencionados para su asignación dentro de un bloque de horario.

5.2 Fase #2: Diseño Conceptual

En esta etapa se evaluaron los resultados obtenidos en la fase anterior ya que se debe modelar el proyecto desde el principio pero teniendo en

cuenta las restricciones obtenidas, la información que se va a almacenar y de qué forma se hará, distribución de la data, roles en el sistema e interfaces a realizar, siempre y cuando todo los resultados se encuentren orientados al desarrollo de un software necesario para la Escuela de Ingeniería Informática.

Como se describe en la metodología los resultados en esta fase son los siguientes: Ver Apéndice A.

- Diagrama de Entidad Relación.
- Diccionario de Datos.
- Diagrama de casos de uso.
- Interfaces de usuario.

Lo documentos anteriormente nombrados permitieron mantener la trazabilidad del proyecto y determinar pasos o tareas a seguir para la implementación correcta del mismo.

5.3 Fase # 3: Desarrollo del Prototipo

Para esta fase como se explicó en el capítulo de desarrollo, se dividió el prototipo en dos fases con la finalidad de obtener un sistema administrativo para la gestión de entradas al sistema y luego implementar el algoritmo genético para obtener una solución al problema.

En el desarrollo de la gestión de entradas fue necesario describir los componentes que se requieren para mantener la trazabilidad del desarrollo, es por ello que se realizó la descripción de los casos de uso, diseño físico de la base de datos y para finalizar la implementación de los módulos que permitan la gestión de entradas al sistema. Ver Apéndice B.

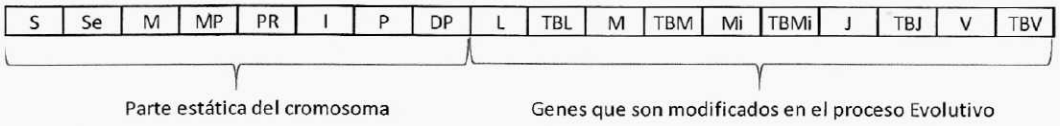
Es importante destacar que tras cada desarrollo de cada módulo se realizaron las pruebas unitarias de cada componente validando que el funcionamiento sea consistente.

Tras la implementación del sistema para la gestión de entradas, se procedió a la definición y la implementación del algoritmo genético.

RepresentaciónCromosómica

Para la representación del cromosoma fue considerada toda aquella información que debería ser mostrada en el horario final y además aquellas restricciones cuyo cálculo afectan el desempeño del algoritmo genético para la obtención de una solución.

A continuación se muestra la representación cromosómica que fue utilizada en el presente trabajo:



S = Sección
Se = Semestre
M = Materia
MP = Materias que prela
PR = Practica o Teórica
I = Cromosoma tocable o intocable
P = Profesor
DP = Disponibilidad del profesor
L = Lunes Bloque Horario
TBL = Tamaño del bloque lunes
M = Martes Bloque Horario
TBM = Tamaño del bloque Martes
Mi = Miércoles Bloque Horario
TBMi = Tamaño del bloque Miércoles
J = Jueves Bloque Horario
TBJ = Tamaño del bloque Jueves
V = Viernes Bloque Horario
TBV = Tamaño del bloque Viernes

Figura # 12 RepresentaciónCromosómica. Fuente:Elaboración Propia

Función Aptitud (Cromosoma)

Para la definición de la función aptitud, se tomó en cuenta aquellas restricciones que afectan directamente a un cromosoma o individuo de la población y para el cálculo de esta se evalúa cada cromosoma individualmente utilizando la función aptitud que es mostrada a continuación:

$$\sum_{1}^{N}(Incumplimiento\ de\ disponibilidad\ profesor)$$

$$N = Cantidad\ de\ individuos.$$

Figura # 13 Función Aptitud (Cromosoma). Fuente: Elaboración Propia.

Para la función aptitud se toma como premisa la disponibilidad del profesor y los cumplimientos a esta. Si un cromosoma C tiene un bloque de horario H que cumple con la disponibilidad del profesor en su prioridad más alta (1) se le asignan dos puntos, Si un cromosoma C tiene un bloque de horario H que cumple con la disponibilidad del profesor en su prioridad intermedia (2) se le asigna un punto, a continuación se muestra un ejemplo del cálculo de la función aptitud para un cromosoma.

Nota : 4 Nota : 2 Nota : 0

Horas / Días	Lunes		Martes		Miércoles		Jueves	Viernes
7 - 8	1	1	0	0	0	0	0	0
8 - 9	1	1	0	0	0	0	0	0
9 - 10	0	0	2	1	0	1	0	0
10 - 11	0	0	2	1	0	1	0	0
11 - 12	0	0	0	0	0	0	0	0
12 - 13	0	0	0	0	0	0	0	0
13 - 14	0	0	0	0	0	0	0	0
14 - 15	0	0	0	0	0	0	0	0
15 - 16	0	0	0	0	0	0	0	0
16 - 17	0	0	0	0	0	0	0	0
17 - 18	0	0	0	0	0	0	0	0
18 - 19	0	0	0	0	0	0	0	0
19 - 20	0	0	0	0	0	0	0	0

Disponibilidad del Profesor
Bloque Horario del cromosoma
Nota Calificación del cromosoma para el Bloque

Calificación para la función Aptitud del Cromosoma
6

Figura # 14 Calculo de la Función Aptitud. Fuente: Elaboración Propia.

Función Aptitud (Población).

Para el cálculo de la Función Fitness se recorre secuencialmente toda la población y por cada individuo se calcula el total de incumplimientos de las restricciones fuertes del problema.

Dado a que las restricciones fuertes del problema deben ser cumplidas a cabalidad para obtener una solución, la Función Fitness es de los elementos más importantes en el Algoritmo Genético y es por esto que el estudio de esta puede considerarse como punto clave.

A continuación se muestra la Función Fitness utilizada en el presente trabajo:

$$\sum_{1}^{N}\left(\sum_{1}^{M} I\right)$$

$N = Cantidad\ de\ individuos.$

$M = Cantidad\ de\ restricciones.$

Figura # 15. Función Aptitud (Población). Fuente: Elaboración Propia

Algoritmo Genético

Tras la definición de todos los componentes del algoritmo genético y como fue expresado en el capítulo de desarrollo, se realizaron distintas pruebas de implementación del Algoritmo Genético, sin embargo,el modelo con el cual se obtuvo una solución fue con la Función Comprobadora, dicha función simula el proceso actual llevado por la escuela para la generación del horario.

El algoritmo genético en su ejecución tiene una duración promedio de aproximadamente 12 horas, haciendo una cantidad de 20000 iteraciones aproximadamente, obteniendo una solución aceptable y con un total de 0 incumplimientos en las restricciones del problema.

A continuación se muestran los resultados obtenidos aplicando Algoritmo Genético.

Electivas

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8					Programacion PLC
8 - 9	Desarrollo IOS	Administracion Linux			
9 - 10					
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3					
3 - 4			SAP BI	Presentaciones de negocio	
4 - 5					
5 - 6					
6 - 7					
7 - 8					

10mo Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Eval de sist. Informaticos Seccion 1 - Mijares H Marize D			
8 - 9					
9 - 10				Pract. Eval de sist. Informaticos Seccion 1	
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3					Eval de sist. Informaticos Seccion 1
3 - 4					
4 - 5		Gestion de proyectos de software Seccion 1 - Honack Alberto		Gestion de proyectos de software Seccion 1 - Honack Alberto	
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Eval de sist. Informaticos Seccion 2 - Alfredo Luis García Rodriguez			
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3				Gestion de proyectos de software Seccion 2 - Nelson Antonio Rodriguez Lopez	Pract. Eval de sist. Informaticos Seccion 2 - Alfredo Luis García Rodriguez
3 - 4					
4 - 5	Gestion de proyectos de software Seccion 2 - Nelson Antonio Rodriguez Lopez				
5 - 6					
6 - 7					
7 - 8					

9no Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Etica y ejercicio Prof - Olga Bravo Seccion 1				Pract. Redes III - Seccion 1 Luis Molner
8 - 9					
9 - 10		Multimedia y aplicaciones web - Rafael Lara Campos	Redes III - Seccion 1 Luz Karime Murillo Gonzalez		
10 - 11					Multimedia y aplicaciones web - Rafael Antonio Lara Campos - Seccion:1
11 - 12					
12 - 1					
1 - 2					
2 - 3					
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8				Multimedia y aplicaciones web - Viviana Trujillo	
8 - 9					
9 - 10		Multimedia y aplicaciones web - Viviana Trujillo		Pract. Redes III - Seccion 2 Luis Molner	
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3					
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

8vo Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8				Seguridad Computacional - Seccion 1 - Profesor:Wilmer Pereira	
8 - 9					
9 - 10					
10 - 11					
11 - 12	Analisis de Inversiones - Profesor:Pedro Jose Mijares Savino				
12 - 1					Investigación de Operaciones - Seccion 1 - Evelenir Barreto
1 - 2				Pract . Seguridad Computacional - Seccion 1 - Profesor:Wilmer Pereira	
2 - 3	Investigación de Operaciones - Seccion 1 - Evelenir Barreto		Desarrollo del Software - Carlos Magurnos		Desarrollo del Software - Carlos Magurnos
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8			Sistemas Distribuidos - Seccion 2 - Valeria Leon		Sistemas Distribuidos - Seccion 2 - Valeria Leon
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 1	Sistemas Distribuidos - Profesor:Wilmer Pereira				
1 - 2					
2 - 3					Pract . Seguridad Computacional - Seccion 2 - Profesor:Wilmer Pereira
3 - 4					
4 - 5		Seminario de TG - Profesor:Lúcia Isabel Cardoso Militao			Sistemas Distribuidos - Profesor:Wilmer Pereira
5 - 6					
6 - 7	Desarrollo del Software - Carlos Barroeta		Desarrollo del Software - Carlos Barroeta		
7 - 8					

7mo Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Contabilidad General - Seccion 1 - Profesor:Juan de la Cruz Guaita Giron	Redes de comp II - Seccion 1 - Profesor:Luz Karime Murillo Gonzalez			Redes de comp II - Seccion 1 - Profesor:Luz Karime Murillo Gonzalez
8 - 9					
9 - 10					
10 - 11				Computación Grafica - Wilmer Pereira - Seccion 1	
11 - 12			Contabilidad General - Seccion 1 - Profesor:Juan de la Cruz Guaita Giron		
12 - 1					
1 - 2		Pract. Redes de comp II - Profesor:Rafael E Andara G - Seccion:1			
2 - 3					
3 - 4				Sistemas de B. de Datos II - Seccion 1- Profesor:Lúcia Isabel Cardoso Militao	
4 - 5	Metodologia de Software - Seccion 1 - Profesor:Lúcia Isabel Cardoso Militao				Sistemas de B. de Datos II - Seccion 1- Profesor:Lúcia Isabel Cardoso Militao
5 - 6				Metodologia de Software - Seccion 1 - Profesor:Lúcia Isabel Cardoso Militao	
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Computación Grafica - Esmitt Ramirez - Seccion 2	Redes de comp II - Seccion 2 - Profesor:Biagio Domenico Cante Mata			
8 - 9					
9 - 10		Pract. Redes de comp II - Jorge García - Seccion:2		Redes de comp II - Seccion 2 - Profesor:Biagio Domenico Cante Mata	
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3					
3 - 4					
4 - 5	Metodologia de Software - Seccion 2 - Profesor:Quinorbis Milagros Peña Figueroa			Metodologia de Software - Seccion 2 - Profesor:Quinorbis Milagros Peña Figueroa	
5 - 6					
6 - 7		Sistemas de B. de Datos II - Seccion 2 - Profesor:Marcos Salazar		Sistemas de B. de Datos II - Seccion 2 - Profesor:Marcos Salazar	
7 - 8					

6to Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8			Probabilidad y Estadistica - Profesor:Rosaura Paladino Alvarado - Seccion:1	Probabilidad y Estadistica - Profesor:Rosaura Paladino Alvarado - Seccion:1	
8 - 9					
9 - 10	Pract. Redes de Comp. I - Profesor:Igor Alberto Chavez Porra - Seccion:1	Sistema de B. de Datos I - Profesor:Francisco Javier Moreno Fernández - Seccion:1	Redes de Comp. I - Profesor:Luz Karime Murillo Gonzalez - Seccion:1		Redes de Comp. I - Profesor:Luz Karime Murillo Gonzalez - Seccion:1
10 - 11					
11 - 12					
12 - 1	Pract. Arquitectura del computador - Profesor:Biagio Domenico Cante Mata - Seccion:1		Métodos Numericos - Profesor:Carlos Guerreiro - Seccion:1		
1 - 2					
2 - 3	Arquitectura del computador - Profesor:Ramón Porras - Seccion:1				
3 - 4		Arquitectura del computador - Profesor:Ramón Porras - Seccion:1	Sistema de B. de Datos I - Profesor:Francisco Javier Moreno Fernández - Seccion:1		
4 - 5					
5 - 6		Métodos Numericos - Profesor:Carlos Guerreiro - Seccion:1			
6 - 7				Sistema de B. de Datos I - Profesor:Ana Karina Fernandes Agrela - Seccion:3	Sistema de B. de Datos I - Profesor:Ana Karina Fernandes Agrela - Seccion:3
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8					Arquitectura del computador - Profesor:Biagio Domenico Cante Mata - Seccion:2
8 - 9					
9 - 10					
10 - 11			Redes de Comp. I - Profesor:Igor Alberto Chavez Porra - Seccion:2		Redes de Comp. I - Profesor:Igor Alberto Chavez Porra - Seccion:2
11 - 12	Sistema de B. de Datos I - Profesor:Jesus Mendoza - Seccion:4			Sistema de B. de Datos I - Profesor:Jesus Mendoza - Seccion:4	
12 - 1					
1 - 2	Arquitectura del computador - Profesor:Biagio Domenico Cante Mata - Seccion:2	Sistema de B. de Datos I - Profesor:Quinorbis Milagros Peña Figueroa - Seccion:2		Sistema de B. de Datos I - Profesor:Quinorbis Milagros Peña Figueroa - Seccion:2	
2 - 3			Pract. Redes de Comp. I - Profesor:Henry Córdova - Seccion:2		
3 - 4					
4 - 5		Pract. Arquitectura del computador - Profesor:Ramón Porras - Seccion:2	Probabilidad y Estadistica - Profesor:Jose Miguel Campos - Seccion:2		Probabilidad y Estadistica - Profesor:Jose Miguel Campos - Seccion:2
5 - 6					
6 - 7					
7 - 8					

5to Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Circuitos Electronicos - Profesor:Ramón Porras - Seccion:1				
8 - 9					
9 - 10					
10 - 11				Economia General - Profesor:Delfin Delfin - Seccion:1	
11 - 12	Sistemas de operacion - Profesor:Jose Ricardo Ric - Seccion:1	Circuitos Electronicos - Profesor:Ramón Porras - Seccion:1			
12 - 1					
1 - 2				Pract. Circuitos Electronicos - Profesor:Biagio Domenico Cante Mata - Seccion:1	
2 - 3			Sistemas de operacion - Profesor:Jose Ricardo Ric - Seccion:1		Cálculo IV - Profesor:Luis Estrada - Seccion:1
3 - 4					
4 - 5	Cálculo IV - Profesor:Luis Estrada - Seccion:1				
5 - 6					
6 - 7	Ingenieria del software - Profesor:Manuel Rivas García - Seccion:1		Ingenieria del software - Profesor:Manuel Rivas García - Seccion:1		
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Circuitos Electronicos - Profesor:Biagio Domenico Cante Mata - Seccion:2				
8 - 9			Sistemas de operacion - Profesor:Viviana Eugenia Trujillo Osuna - Seccion:2	Sistemas de operacion - Profesor:Viviana Eugenia Trujillo Osuna - Seccion:2	
9 - 10	Cálculo IV - Profesor:Jorge Luis Omar - Seccion:2				
10 - 11					
11 - 12		Pract. Circuitos Electronicos - Profesor:Ramón Porras - Seccion:2			
12 - 1					
1 - 2					
2 - 3					Circuitos Electronicos - Profesor:Biagio Domenico Cante Mata - Seccion:2
3 - 4				Cálculo IV - Profesor:Jorge Luis Omar - Seccion:2	
4 - 5					
5 - 6					
6 - 7	Ingenieria del software - Profesor:Jacinto Fung Leon - Seccion:2	Ingenieria del software - Profesor:Jacinto Fung Leon - Seccion:2			
7 - 8					

4to Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8				Fisica General II - Profesor:Juan Carlos Lavado - Seccion:1	
8 - 9					
9 - 10			Humanidades III - Profesor:Marielena Nestas - Seccion:1		Algoritmos y programacion III - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:1
10 - 11					
11 - 12					
12 - 1					
1 - 2		Calculo III - Profesor:Miguel Angel Rodriguez - Seccion:1	Calculo III - Profesor:Miguel Angel Rodriguez - Secc:1		
2 - 3	Pract. Algoritmos y programacion III - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:1				
3 - 4					
4 - 5	Calculo III - Profesor:Miguel Angel Rodriguez - Seccion:1	Estructura del computador - Profesor:Marcos Salazar - Seccion:1	Algoritmos y programacion III - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:1		Fisica General II - Profesor:Juan Carlos Lavado - Seccion:1
5 - 6				Estructura del computador - Profesor:Marcos Salazar - Seccion:1	
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Estructura del computador - Profesor:Carlos Gabriel González Pinto - Seccion:2			
8 - 9					
9 - 10	Calculo III - Profesor:Dayana Villegas - Seccion:2	Algoritmos y programacion III - Profesor:Lia Roxana Mendoza - Seccion:2			
10 - 11					
11 - 12				Fisica General II - Profesor:Santiago Suarez - Seccion:2	
12 - 1	Fisica General II - Profesor:Santiago Suarez - Seccion:2				
1 - 2		Calculo III - Profesor:Dayana Villegas - Seccion:2			
2 - 3	Estructura del computador - Profesor:Carlos Gabriel González Pinto - Seccion:2				
3 - 4			Pract. Algoritmos y programacion III - Profesor:Lia Roxana Mendoza - Seccion:2		
4 - 5	Algoritmos y programacion III - Profesor:Lia Roxana Mendoza - Seccion:2				
5 - 6					
6 - 7					
7 - 8					

3er Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Cálculo II - Profesor:Miguel Angel Rodriguez - Seccion:1	Cálculo II - Profesor:Miguel Angel Rodriguez - Seccion:1	Cálculo II - Profesor:Miguel Angel Rodriguez - Seccion:1		
8 - 9					Algoritmos y programación II - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:1
9 - 10	Humanidades II - Profesor:P Franklin - Seccion:1				
10 - 11		Algoritmos y programación II - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:1			
11 - 12	Fisica General I - Profesor:Jorge Benedicto - Seccion:1			Fisica General I - Profesor:Jorge Benedicto - Seccion:1	
12 - 1		Humanidades II - Profesor:P Franklin - Seccion:1			
1 - 2		Fisica General I - Profesor:Jorge Benedicto - Seccion:1			
2 - 3			Pract. Algoritmos y programación II - Profesor:Omar Jesus Hernandez Barrera - Seccion:1	Matematicas Discretas - Profesor:Silvina Fatima Maria de Jesus Da Mata - Seccion:1	
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Cálculo II - Profesor:Kahela Bastidas - Seccion:2	Matematicas Discretas - Profesor:Silvina Fatima Maria de Jesus Da Mata - Seccion:2	Matematicas Discretas - Profesor:Silvina Fatima Maria de Jesus Da Mata - Seccion:2	
8 - 9					
9 - 10			Cálculo II - Profesor:Kahela Bastidas - Seccion:2	Cálculo II - Profesor:Kahela Bastidas - Seccion:2	
10 - 11					
11 - 12		Algoritmos y programación II - Profesor:Omar Jesus Hernandez Barrera - Seccion:2			Fisica General I - Profesor:Jose Manuel Marino - Seccion:2
12 - 1			Fisica General I - Profesor:Jose Manuel Marino - Seccion:2		
1 - 2					
2 - 3					
3 - 4		Pract. Algoritmos y programación II - Profesor:Ricardo Casanova Nébola-Kopak - Seccion:2			
4 - 5					
5 - 6					
6 - 7					
7 - 8					

2do Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8				Humanidades I - Profesor:Arapé Elizabeth - Sección:1	
8 - 9					
9 - 10			Algoritmos y Programación I - Profesor:Lia Roxana Mendoza - Seccion:3		
10 - 11					
11 - 12			Cálculo I - Profesor:Augusto Cardenas - Seccion:1		
12 - 1					
1 - 2					Algoritmos y Programación I - Profesor:Evelenir Barreto González - Seccion:1
2 - 3	Algoritmos y Programación I - Profesor:Evelenir Barreto González - Seccion:1	Cálculo I - Profesor:Augusto Cardenas - Seccion:1	Pract. Algoritmos y Programación I - Profesor:Roquefelix Alexander Benal Vera - Seccion:3		
3 - 4					Cálculo I - Profesor:Augusto Cardenas - Seccion:1
4 - 5		Algoritmos y Programación I - Profesor:Lia Roxana Mendoza - Seccion:3	Cálculo I - Profesor:Lisset De Gouccia - Seccion:2	Cálculo I - Profesor:Augusto Cardenas - Seccion:1	
5 - 6	Pract. Algoritmos y Programación I - Profesor:Ana Maria Bossio Barcelo - Seccion:1				
6 - 7		Lógica Computacional - Profesor:Oswel Jesus Sanchez Salazar - Seccion:1		Lógica Computacional - Profesor:Oswel Jesus Sanchez Salazar - Seccion:1	
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Algoritmos y Programación I - Profesor:Ana Maria Bossio Barcelo - Seccion:2				
8 - 9					
9 - 10		Cálculo I - Profesor:Lisset De Gouccia - Seccion:2	Cálculo I - Profesor:Lisset De Gouccia - Seccion:2	Cálculo I - Profesor:Lisset De Gouccia - Seccion:2	
10 - 11	Cálculo I - Profesor:Lisset De Gouccia - Seccion:2				
11 - 12	Lógica Computacional - Profesor:Gloria Tarrío Villaverde - Seccion:2				
12 - 1				Lógica Computacional - Profesor:Gloria Tarrío Villaverde - Seccion:2	
1 - 2					
2 - 3	Pract. Algoritmos y Programación I - Profesor:Lia Roxana Mendoza - Seccion:4	Algoritmos y Programación I - Profesor:Ana Maria Bossio Barcelo - Seccion:2			
3 - 4					
4 - 5	Algoritmos y Programación I - Profesor:Roquefelix Alexander Benal Vera - Seccion:4	Pract. Algoritmos y Programación I - Profesor:Omar Jesus Hernandez Barrera - Seccion:2	Humanidades I - Profesor:Arapé Elizabeth - Seccion:2	Algoritmos y Programación I - Profesor:Roquefelix Alexander Benal Vera - Seccion:4	
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8				Algoritmos y Programación I - Profesor:Omar Jesus Hernandez Barrera - Seccion:5	
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 1					
1 - 2					
2 - 3					
3 - 4					Pract. Algoritmos y Programación I - Profesor:Evelenir Barreto González - Seccion:5
4 - 5			Algoritmos y Programación I - Profesor:Omar Jesus Hernandez Barrera - Seccion:5		
5 - 6					
6 - 7					
7 - 8					

1er Semestre

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Introduccion a la informatica - Profesor:Rosaura Paladino Alvarado - Seccion:1	Introduccion a la informatica - Profesor:Rosaura Paladino Alvarado - Seccion:1		
8 - 9					
9 - 10					
10 - 11				Lenguaje y Comunicacion - Profesor:Profesor Generado - Seccion:1	
11 - 12					
12 - 1			Lenguaje y Comunicacion - Profesor:Profesor Generado - Seccion:1		
1 - 2	Matematica Basica - Profesor:Norkys Sanchez - Seccion:1				
2 - 3		Matematica Basica - Profesor:Norkys Sanchez - Seccion:1	Matematica Basica - Profesor:Norkys Sanchez - Seccion:1		
3 - 4					
4 - 5	Trigonometria - Profesor:Milagros Vilacha - Seccion:1				
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8	Matematica Basica - Profesor:Mary Braña - Seccion:2	Introduccion a la informatica - Profesor:Lia Roxana Mendoza - Seccion:2		Introduccion a la informatica - Profesor:Lia Roxana Mendoza - Seccion:2	
8 - 9					
9 - 10		Matematica Basica - Profesor:Mary Braña - Seccion:2			
10 - 11					
11 - 12					
12 - 1					
1 - 2			Matematica Basica - Profesor:Mary Braña - Seccion:2		
2 - 3					
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

	Lunes	Martes	Miercoles	Jueves	Viernes
7 - 8		Introduccion a la informatica - Profesor:Ana Maria Bossio Barcelo - Seccion:3			
8 - 9			Introduccion a la informatica - Profesor:Katiuska Del Carmen Zambrano Camero - Seccion:4		
9 - 10					Introduccion a la informatica - Profesor:Ana Maria Bossio Barcelo - Seccion:3
10 - 11					
11 - 12					
12 - 1	Introduccion a la informatica - Profesor:Katiuska Del Carmen Zambrano Camero - Seccion:4				
1 - 2					
2 - 3					
3 - 4					
4 - 5					
5 - 6					
6 - 7					
7 - 8					

Figura # 16 Resultados de Ejecución del algoritmo genético con la función comprobadora

5.4 Tabla de Resultados Vs Objetivos

Objetivos	Estudiar y elegir los módulos para la administración del sistema.	Estudiar y seleccionar la tecnología a utilizar basada en los estándares del DTI.	Evaluar los parámetros de entrada y sus posibles variables y determinar cuáles deben ser considerados como factores determinantes en la generación de horarios.	Implementar los módulos para el manejo de entradas y administración del sistema que satisfagan las necesidades y peticiones.	Diseñar un modelo de Base de Datos que se ajuste a los requerimiento s del sistema.	Modelar y construir el Algoritmo Genético a implementar .	Realizar las pruebas del modelo necesarias para monitorear desde la factibilidad de los parámetros hasta la obtención de una solución aceptable siendo esta aquella que ofrece una o varias respuestas que cumpla con todas las restricciones impuestas para la generación de un horario.
Producto							
Levantamiento de Información	X		X				
Limitaciones	X						

Alcance	X						
Entradas al Sistema			X				
Restricciones	X		X				
Herramientas Utilizadas		X					
Diagrama Entidad Relación					X		
Diccionario de datos					X		
Interfaces del Sistema				X			
Representación Cromosómica						X	
Función Aptitud						X	
Función Fitness						X	
Algoritmo Genético						X	X

CAPITULO VI. CONCLUSIONES

El presente trabajo demuestra de manera directa que el uso de la informática y la tecnología en la actualidad proporciona soluciones satisfactorias a problemas grandes de optimización que requieren de tiempo y personas dedicadas, en este caso para la elaboración de los horarios de los profesores de la Escuela de IngenieríaInformática de la Universidad CatólicaAndrés Bello, por ello dicho proceso se evaluó en busca de una solución factible que cumpla con las restricciones de la Escuela, así como también el reducir el tiempo y el personal necesarios para el proceso.

Para la obtención de un producto final satisfactorio se llevaron a cabo ciertos estudios para problemas de optimización, en este caso problemas NP – Completos y poder así plantear una representación del problema donde se puedan cumplir todas las restricciones presentes y las entradas al sistema.

Después de los estudios realizados y que se llevaron a cabo desde la selección adecuada del lenguaje de programación hasta el modelado correcto del Algoritmo Genético, se pudo apreciar que el algoritmo genético permite conseguir mejores soluciones en un menor tiempo de ejecución, así como también una buena selección del lenguaje de programación en este caso C# que fue el idóneo para implantar el algoritmo ya que sus características ayudaron a que el sistema tenga un alto rendimiento en todos los aspectos relevantes del proyecto para luego ejecutar pruebas durante y después del desarrollo del sistema comprobando de manera satisfactoria que el algoritmo permite obtener una solución para la generación de horarios,mejorando tiempo en la obtención de una solución en comparación con el proceso que se maneja actualmente.

Se puede concluir que la solución propuesta basada en un sistema computacional que permite la automatización del proceso actual para la generación de horarios fue finalizado con éxito cumpliendo con las características y restricciones obtenidas en el levantamiento de información, dando así una solución factible para la generación de horarios de los

profesores de la Escuela de IngenieríaInformática de la Universidad CatólicaAndrés Bello y superando resultados obtenidos de manera manual.

RECOMENDACIONES

- Una Integración con el Departamento de Tecnología para la integración del sistema de generación horaria con Zona Sistemas y aplicaciones existentes, con la finalidad de permitir la obtención de las entradas del sistema de fuentes externas.
- Una adaptación en el sistema para que pueda ser útil en otras Escuelas de la Universidad CatólicaAndrés Bello siguiendo las restricciones exigidas en cada escuela.
- La integración con Zona sistemas para profesores y poder ingresar la disponibilidad horaria en línea.
- La consulta de los horarios aprobados por profesor a través de Zona Sistemas.
- Solicitar a los profesores para que proporcionen el doble de la disponibilidad requerida por las materias que desea dictar ya que esto mejora el desempeño del algoritmo.

GLOSARIO DE TERMINOS.

2.9 Horario

Es una definición fácil y rápida de la distribución de clases, aulas y horas semanales dictadas por un profesor.

2.10 Pensum

Corresponde al detalle del total de materias a dictarse por carrera.

2.11 Prelaciones

Se define como la conexión que tiene una materia con otra y a su vez como el requisito que debe cumplirse para cursar una materia del semestre siguiente, es decir, un estudiante puede cursar una materia siempre y cuando haya aprobado la materia que antecede a esta, siempre y cuando tenga una prelación o conexión con otra de semestres inferiores.

2.12 Profesor Tiempo Completo

Cumplen una jornada laboral en la universidad, de 40 horas semanales y se deben dictar 12 horas de clases.

2.13 Profesor Tiempo Parcial

Son aquellos que solo asisten para cubrir las horas asignadas.

BIBLIOGRAFIA

Alfaro, E. (s.f.). http://eddyalfaro.galeon.com. Recuperado el Junio de 2013, de http://eddyalfaro.galeon.com/geneticos.html

Bertona, L. F. (2005). *algoritmos evolutivos.* http://dharma.frm.utn.edu.ar/cursos/ia/2012/material/APUNTES-FILMINAS/U4/algoritmo_genetico.htm.

Casanovas, J. (31 de julio de 2004). Usabilidad y arquitectura del software. Obtenido de http://www.alzado.org/articulo.php?id_art=355

Cortéz, A. (2004). *Teoria de la Complejidad Computacional y Teoria de la Computabilidad.* Lima, Peru: Universidad Nacional Mayor de San Marcos.

Frittelli, V. (7 de Noviembre de 2012). Complejidad Computacional.

Garcia, G. (2010). *Introduccion a la Resolucion de problemas con Algoritmos Geneticos.*Caracas: Universidad Catolica Andres Bello.

Intelligent Systems Group. (s.f.). *Intelligent Systems Group.*Recuperado el Junio de 2013, de http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf

Parra Truyol, A. (1997). *Algoritmos Geneticos.* http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf.

Lara Rafael, A. P. (2004). *Desarrollo de un Modelo para la asignación de horarios de clase a través de Algoritmo Genético.* Caracas.

Sequeira, T. (16 de Noviembre de 2010). *http://www.slideshare.net/.* Recuperado el 25 de Septiembre de 2013, de http://www.slideshare.net/NoeGonzalezMendoza/arquitectura-cliente-servidor

Will, A. (27 de Septiembre de 2002). *http://www.herrera.unt.edu.ar/.* Recuperado el 25 de Septiembre de 2013, de http://www.herrera.unt.edu.ar/gapia/Curso_AG/Curso_AG_08_Clase_3.pdf

Sanchez, Jiménez (29 de agosto de 2000). *http://www.sidar.org/. Recuperado el 25 de Septiembre de 2013, de* http://www.sidar.org/recur/desdi/traduc/es/visitable/quick-prototype.htm

Scaridad. (s.f). Recuperado el 12 de enero del 2011, de http://www.scaridad.com/files/MPR.pdf.

APENDICES

APENDICE A. DEFINICION DE ESPECIFICACIONES.

Apéndice A.1. Diccionario de datos

Tabla o entidad	Atributo o campo	Tipo de dato	Long	Llave primaria	Descripción	Tabla o Entidad Foránea
Pensum	idPensum	INT		PK	Identificador de Pensum	
	pe_fechaInicio	DATE			Fecha Inicio	
	pe_fechaFin	DATE			Fecha Fin	
Pensum_materia	idPensumMateria	INT		PK	Identificador de Pensum Materia	
	pm_numero	INT			Numero	
	pm_hora_teorica	INT			Cantidad de horas teóricas	
	pm_hora_practica	INT			Cantidad de horas de practica	
	pm_hora_laboratorio	INT			Cantidad de horas de laboratorio	
	pm_unidadesCredito	INT			Unidades de crédito	
	pm_numero_semestre	INT			Numero de semestre	
	fk_pensum	INT		FK	Foránea de Pensum	Pensum (idPensum)
	fk_materia	INT		FK	Foránea de Materia	Materia (codigoMateria)

Requisito	idRequisito	INT		PK	Identificador de Requisito	
	fk_materia	INT		FK	Foránea de Pensum Materia (representa la materia)	Pensum_materia (idPensumMateria)
	fk_prelacion	INT		FK	Foránea de Pensum Materia (representa la prelación de la materia)	Pensum_materia (idPensumMateria)
Materia	codigoMateria	INT		PK	Identificador de Materia	
	ma_nombre	VARCHAR	100		Nombre de la Materia	
	ma_status	INT			Status de la Materia	
	fk_coordinacion	INT		FK	Foránea de coordinación	Coordinacion (idCoordinacion)
Coordinacion	idCoordinacion	INT		PK	Identificador de Coordinación	
	co_nombre	VARCHAR	100		Nombre Coordinación	
	co_status	INT			Status Coordinación	
	co_coordinador	VARCHAR	100		Nombre Coordinador	

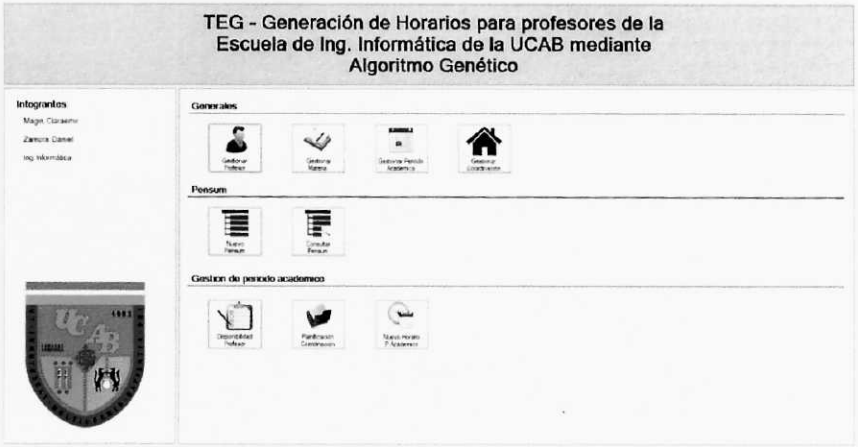
Disponibilidad_materia	DPM_ID	INT		PK	Identificador de disponibilidad materia	
	fk_disponibilidad	INT		FK	Foránea de Disponibilidad	Disponibilidad (idDisponibilidad Profesor)
	fk_pensumMateria	INT		FK	Foránea de Pensum Materia	Pensum_materia (idPensumMateria)
Disponibilidad	idDisponibilidad_Profesor	INT		PK	Identificado de Disponibilidad de profesor	
	dp_otrasEscuelas	VARCHAR	100		Nombre otras escuelas	
	dp_telefono	VARCHAR	45		teléfono	
	fk_periodo	INT		FK	Foránea de Periodo	Periodo (idPeriodo)
	fk_profesor	INT		FK	Foránea de Profesor	Profesor (idProfesor)
	fk_tipoDisponibilidad	INT		FK	Foránea de Tipo Disponibilidad	Tipo_Disponibilidad (idTipo_Disponibilidad)
Tipo_disponibilidad	idTipo_Disponibilidad	INT		PK	Identificador de tipo de disponibilidad	
	td_nombre	VARCHAR	100		Nombre	
	td_horaInicio	TIME			hora inicio	
	td_HoraFi	TIME			Hora Fin	

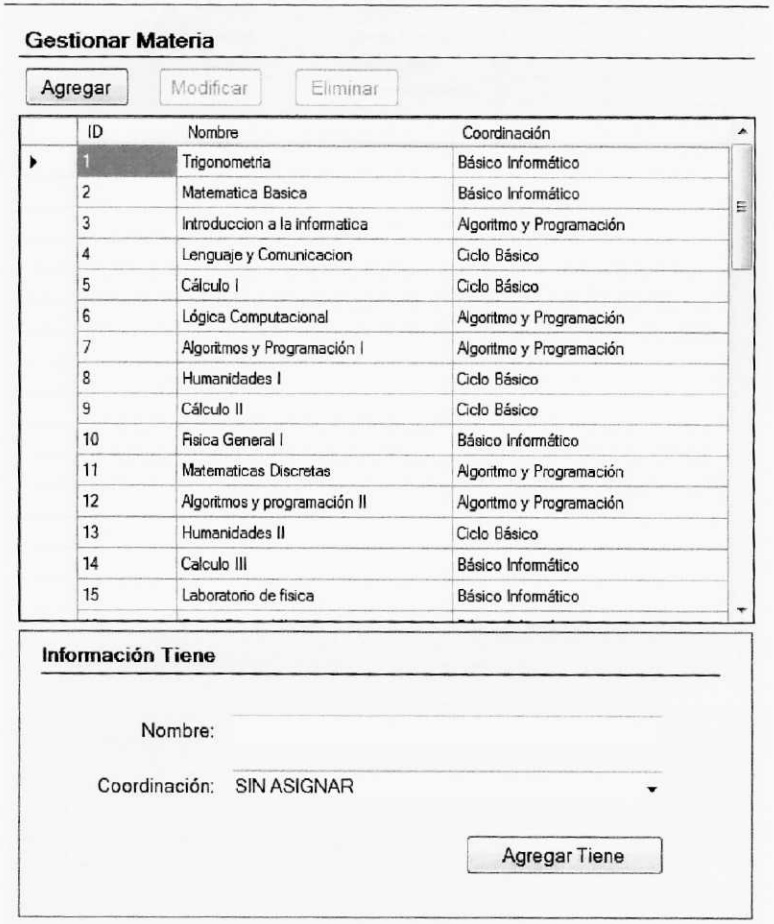
	n					
	td_Tamano_bloque	INT			Tamaño del Bloque	
bloque_disponible	idHorario	INT		PK	Identificador del bloque disponible	
	BDP_HoraInicio	INT			hora inicio	
	bdp_HoraFin	INT			hora fin	
	bdp_prioridad	INT			prioridad de bloque	
	bdp_dia	INT			día	
	fk_disponibilidad	INT		FK	Foránea de Disponibilidad	Disponibilidad (idDisponibilidad Profesor)
Profesor	idProfesor			PK	Identificador del Profesor	
	po_nombre	VARCHAR	200		Nombre del Profesor	
	po_apellido	VARCHAR	100		Apellido del Profesor	
	po_telefono	VARCHAR	100		Teléfono del Profesor	
Periodo	idPeriodo	INT			Identificador de Periodo	
	pe_nombre	VARCHAR	100		Nombre del Periodo	
	pe_fechalnicio	DATE			Fecha inicio	
	pe_FechaFin	DATE			Fecha Fin	

Planificacion_Coordinacion	idPlanificacion	INT		PK	Identificador de planificación coordinación	
	pl_term	INT			Termino	
	pl_Fecha Emision	DATE			fecha de emisión	
	pl_Comentario	VARCHAR	500		Comentarios	
	fk_Coordinacion	INT		FK	Foránea de Coordinación	Coordinacion (idCoordinacion)
	fk_periodo	INT		FK	Foránea de Periodo	Periodo (idPeriodo)
Planificacion_materia_profesor	IdPlanificacionMateriaProfesor	INT		PK	Identificador de Planificación Materia Profesor	
	pmp_num_seccionesTeoricas	INT			Numero de secciones teóricas	
	pmp_num_alumnos Teoricas	INT			Número de alumnos teoría	
	pmp_horasTeoricas	INT			Número de horas teóricas	
	pmp_num_seccionesPracticas	INT			Numero de secciones practicas	
	pmp_num_Alumnos Practicas	INT			Número de Alumnos Practicas	
	pmp_horasPracticas	INT			Número de Horas	

					practicas	
	fk_planificacion	INT		FK	Foránea de planificación	Planificacion_Coordinación(idPlanificacion)
	fk_profesor	INT		FK	Foránea de Profesor	Profesor (idProfesor)
	fk_materia	INT		FK	Foránea de Materia	Materia (codigoMateria)

Apéndice A.2. Interfaces de Usuario





Gestionar Coordinación

Agregar | Modificar | Eliminar | Agregar Materias

ID	Nombre	Nombre Coordinador
1	Ciencias de la computación	Roseura Paladino
2	Algoritmo y Programación	Gloria Tarrio
3	Básico Informático	Ricardo Casanova
4	Ingeniería del Software	Lúcia Cardoso
5	Telemática	Wilmer Pereira
6	Matemáticas Aplicadas	Evelenir Barreto
7	Gerencia	Susana García Martínez
8	Electivas	Simy Blomer
9	Ciclo Básico	Milagros Boschetti

Información Coordinación

Nombre:

Coordinador nombre:

Agregar Coordinación

Gestionar Profesor

Agregar | Modificar

ID	Nombre	Apellido	Cedula	Telefono
1	Evelenir	Barreto González	0	
2	Lúcia Isabel	Cardoso Militao	0	
3	Ricardo	Casanova Nébola-Ko...	0	
4	Susana Coromoto	García Martínez	0	
5	Rosaura	Paladino Alvarado	0	
6	Quinorbis Milagros	Peña Figueroa	0	
7	Wilmer	Pereira	0	
8	Gloria	Tarrío Villaverde	0	
9	Roquefelix Alexander	Bernal Vera	0	
10	Ana Maria	Bossio Barcelo	0	
11	Jose Gregorio	Fajardo Ramos	0	

Información Profesor

*Nombre:

*Apellido:

Cedula:

Telefono:

Agregar Profesor

Scheduale TIG - Generar Horario

Generar Nuevo Horario

Periodo Academico

Periodo Academico

-- Seleccione periodo academico ---

Generar Horario

Verificación de entradas

Pensum -

Planificación Coordinación

	Nombre Coordinación	Estado

Disponibilidad Profesores -

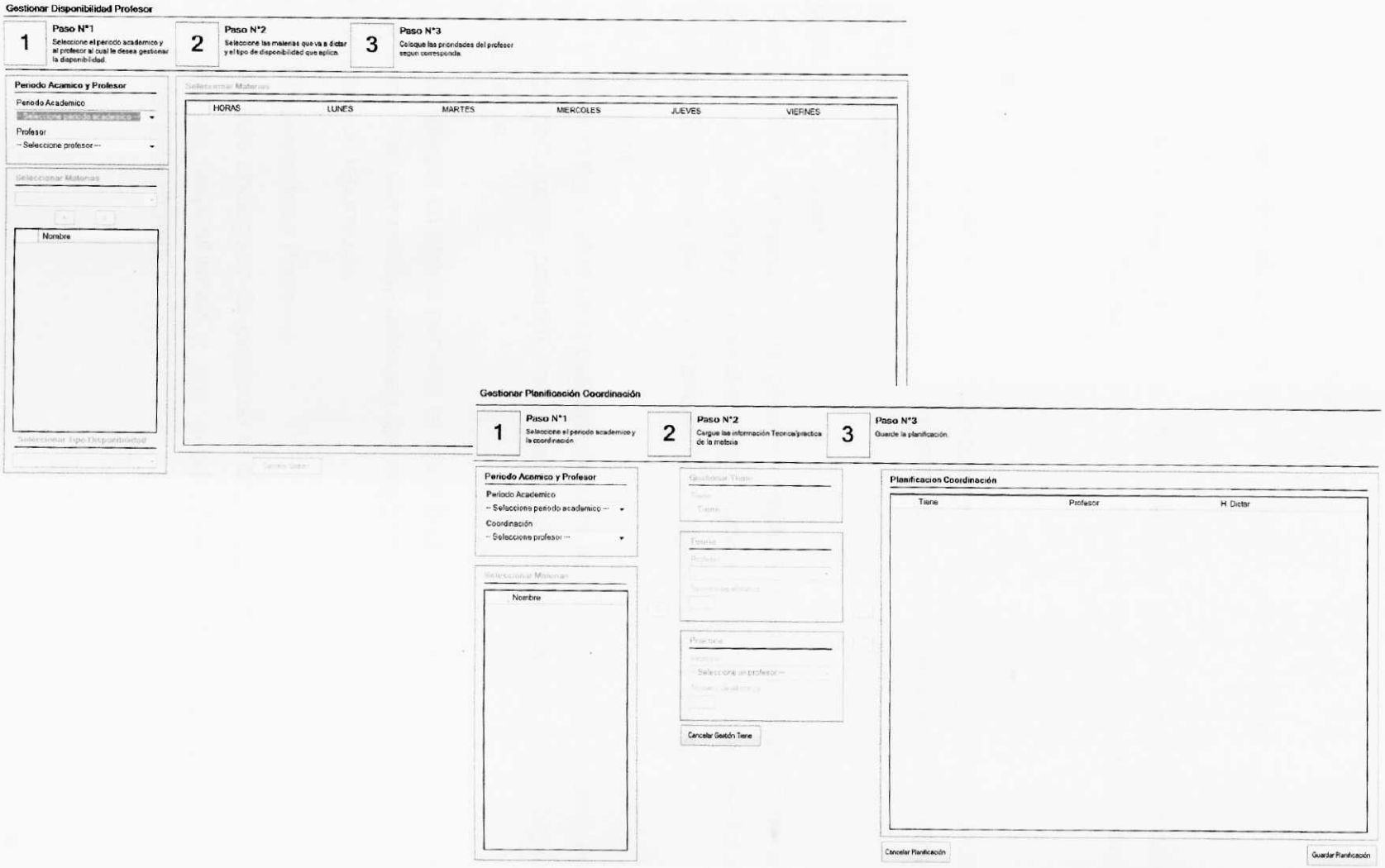
Consultar Pensum

1° Semestre 1 2 3 4 5 6 7 8 9 10

Codigo	Tiene	T	P	L	U
401	Trigonometria	3	0	0	3
402	Matematica Basica	6	0	0	6
407	Introduccion a la informatica	4	0	0	4
403	Lenguaje y Comunicacion	4	0	0	4

Prelaciones

Codigo	Tiene	T	P	L	U


Gestionar Disponibilidad Profesor
1
Paso N°1
Seleccione el periodo academico y al profesor al cual le desea gestionar la disponibilidad.
2
Paso N°2
Seleccione las materias que va a dictar y el tipo de disponibilidad que aplica.
3
Paso N°3
Coloque las prioridades del profesor segun corresponda.
Periodo Acamico y Profesor
Periodo Academico
Profesor
-- Seleccione profesor --
Seleccionar Materias
Nombre
HORAS
LUNES
MARTES
MIERCOLES
JUEVES
VIERNES
Gestionar Planificación Coordinación
1
Paso N°1
Seleccione el periodo academico y la coordinación
2
Paso N°2
Cargue la información Teorica/practica de la materia
3
Paso N°3
Guarde la planificación.
Periodo Acamico y Profesor
Periodo Academico
-- Seleccione periodo academico --
Coordinación
-- Seleccione profesor --
Seleccionar Materias
Nombre
Cancelar Gestión Tiene
Planificacion Coordinación
Tiene
Profesor
H. Dictar
Cancelar Planificación
Guardar Planificación

APENDICE B.

Apéndice B.1. Descripción de los Módulos del Sistema

B.2.1 Profesor

Modulo encargado de la gestión completa del profesor, en el se puede agregar un profesor y modificar su información, a través de una sola ventana de administración.

B.2.2 Materia

Modulo encargado de la gestión completa de la materia, en el se puede agregar una materia y modificar su información a través de una sola ventana de administración.

B.2.3. Coordinación

Modulo encargado de la gestión completa de la coordinación de la escuela, en el se agregan coordinaciones nuevas y las materias que dicha coordinación maneja, así como también la modificación de la misma.

B.2.4 Pensum

Modulo encargado de la gestión completa del pensum de la escuela, en él se agregan nuevos pensum y se consulta la información del mismo según su semestre.

Al agregar un nuevo pensum se maneja la información de las materias que tiene por semestres, unidades de créditos, códigos y horas teóricas, prácticas y de laboratorio.

B.2.5 Disponibilidad Profesor

Modulo encargado de gestionar la disponibilidad completa del profesor y el tipo de disponibilidad y sus prioridades basándose en el periodo académico seleccionado.

B.2.6 Planificación Coordinación

Modulo encargado de la gestión de las planificaciones de cada coordinación, en él se maneja el periodo académico a seleccionar, las materias que se dictaran, cantidad de secciones a abrirse por materia y la asignación de profesores.

B.2.7 Horario

Modulo encargado de generar el horario de los profesores de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello.