



Universidad del Valle

Propuesta de trabajo de grado:

**Sistema de asignación de horarios utilizando
herramientas de computación evolutiva para la
distribución de las aulas de clases en la
Universidad del Valle sede Zarzal**

PRESENTADO ANTE:

Comité de Pregrado

Lino Alexander Sinisterra Asprilla

Estudiante de Ingeniería Industrial.

**Universidad del Valle
Regionalización sede Zarzal
Zarzal, Valle del Cauca**

2020



Universidad del Valle

**Sistema de asignación de horarios utilizando
herramientas de computación evolutiva para la
distribución de las aulas de clases en la
Universidad del valle sede Zarzal**

Lino Alexander Sinisterra Asprilla

Director

MSc. Juan Carlos Galeano Garibello.

Codirector

Ing. Milton Alexander Londoño

Universidad del Valle

Regionalización sede Zarzal

Zarzal, Valle del Cauca

2020

Involucrados en el proyecto:

ELABORADO POR:	
Lino Alexander Sinisterra A. <i>Estudiante de Ingeniería Industrial</i> Cédula: 1.116.434.287, Zarzal, Valle del Cauca Celular: 312 873 1138 e-mail: lino.sinisterra@correounivalle.edu.co Facultad: Ingeniería Programa: Ingeniería Industrial Universidad del Valle	
DIRIGIDO POR:	
MSc. Juan Carlos Galeano Garibello <i>Director del Trabajo de Grado</i>	Ing. Milton Alexander Londoño <i>Codirector del Trabajo de Grado</i>
REVISADO POR:	
<i>Jurado del Trabajo de Grado.</i>	
OBSERVACIONES	
<i>Jurado del Trabajo de Grado.</i>	
REVISADO POR:	
<i>Jurado del Trabajo de Grado.</i>	
OBSERVACIONES	

AGRADECIMIENTOS

Expreso mis agradecimientos a:

A Dios por brindarme todos lo necesario para obtener este logro y guiarme en todo momento, a mí padre Marcelino Sinisterra, Madre María Luz Dary Asprilla y a todos los integrantes de mi familia por estar a mi lado en este camino y por su apoyo.

A los y las directivas de la Universidad del Valle, a la directora de la Universidad del Valle sede Zarzal Dr. Cecilia Madriñan Polo.

Quiero extender mis agradecimientos a mi director de tesis el MSc. Juan Carlos Galeano y al ingeniero y candidato a grado industrial Milton Alexander Londoño por su guía, dedicación y compromiso en este proceso, al coordinador de programa o profesor de apoyo del programa de ingeniería industrial Julián González, a la profesora nombrada del programa de ingeniería industrial Vivian Lorena Chud, Al MSc. Mag. Andrés Garzón Valencia, al ingeniero Jairo Echeverry, al técnico Oscar Ossa y a toda la comunidad universitaria de la sede Zarzal por sus aportes en esta investigación.

TABLA DE CONTENIDO

RESUMEN	10
1. INTRODUCCIÓN	11
2. CONTEXTO	12
2.1 PLANTEAMIENTO DEL PROBLEMA	12
2.2 FORMULACIÓN DEL PROBLEMA	13
2.3 PREGUNTA DE INVESTIGACIÓN	17
2.4 HIPÓTESIS	17
3. MARCO TEÓRICO Y ESTADO DEL ARTE	17
3.1 MARCO DE ANTECEDENTES	17
3.2 MARCO CONCEPTUAL	19
3.3 MARCO HISTÓRICO	24
4. MOTIVACIÓN Y JUSTIFICACIÓN	25
5. OBJETIVOS	27
5.1 OBJETIVOS GENERALES	27
5.2 OBJETIVOS ESPECÍFICOS	27
5.3 RESULTADOS ESPERADOS	27
5.4 METODOLOGÍA	28
5.5 ALCANCES Y LIMITACIONES	28
5.6 IMPACTOS ESPERADOS	29
5.7 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN	29
5.8 POBLACIÓN DE MUESTRA	29
6. SISTEMA PARA LA ASIGNACIÓN DE ESPACIOS Y HORARIOS DE CLASE	30
6.1 REQUISITOS DEL SISTEMA	30
6.1 FORMULACIÓN DEL MODELO MATEMÁTICA	33
6.2 DISEÑO ARQUITECTÓNICO	35
6.3 DISEÑO DETALLADO	37
6.4 IMPLEMENTACIÓN.	48
6.4.1. CONFIGURACIÓN DE LOS PARÁMETROS DE EJECUCIÓN.	49
6.4.2. FUNCIÓN DE CARGAR PROGRAMACIÓN ACADÉMICA.	49
6.4.3. FUNCIÓN DE CRUCE.	50
6.4.4. FUNCIÓN DE SELECCIÓN PARA OPERACIÓN DE MEJOR TORNEO.	50
6.4.5. REQUERIMIENTOS HARDWARE Y SOFTWARE.	51

7. PRUEBA Y RESULTADOS	51
8. CONCLUSIONES Y TRABAJOS FUTUROS	63
8.1. CONCLUSIONES.	63
8.2. TRABAJOS FUTUROS.	64
9. REFERENCIAS	65
10. ANEXOS	67

TABLA DE IMÁGENES

Imagen 1 - Población Estudiantil sede Zarzal	12
Imagen 2 - Árbol De Problema Del Proceso	14
Imagen 3 - Pág. 1 Proceso Actual De La Asignación De Aulas Y Horarios	15
Imagen 4 - Pág. 2 Proceso Actual De La Asignación De Aulas Y Horarios	16
Imagen 5 - Terminología algoritmos genéticos.....	23
Imagen 6 - Árbol De Objetivos Del Proceso.....	27
Imagen 7 - Descripción lineal del sistema funcional.....	30
Imagen 8 - Proceso Propuesto De La Asignación De Aulas Y Horarios	31
Imagen 9 - Proceso Propuesto De La Asignación De Aulas Y Horarios	32
Imagen 10 - Diseño Arquitectónico	36
Imagen 11 - Representación del cromosoma para diseño timetabling	37
Imagen 12 - Modelo aplicado Software AMPL IDE Demo	52
Imagen 13 - Matriz de programación de asignaturas	54
Imagen 14 - Tiempo promedio de ejecución	60
Imagen 15 - Promedio de soluciones validas según las población y generaciones	60
Imagen 16 - Total número de asignaturas por aula y día de la semana.....	61
Imagen 17 - Total de asignaturas por aula programadas en la semana	62
Imagen 18 - uso de aulas en según resultado de AMPL.....	62
Imagen 19 - Campus de la Universidad sede Zarzal.....	67

TABLA DE TABLAS

Tabla 1 – pseudocódigo Algoritmo evolutivo	36
Tabla 2 - Representación del gen	37
Tabla 3 - Representación de la programación del gen.....	38
Tabla 4 - Resultados de la programación del gen	39
Tabla 5 - Programación académica.....	42
Tabla 6 - Integración de los genotipos	48
Tabla 7 - Parámetros de configuración de los programas.....	49
Tabla 8 - Cargar programación académica	49
Tabla 9 - Función de cruce.....	50
Tabla 10 - Función de selección y mejor torneo.....	50
Tabla 11 - Requisitos tecnológicos.....	51
Tabla 12 - Resultados de la implementación del modelo, Elaboración propia	57
Tabla 13 - Resultados de la implementación del modelo, Elaboración propia	58
Tabla 14 - Resultados de la implementación del modelo, Elaboración propia	59
Tabla 15 - Descripción de los campus de la Universidad del Valle Sede Zarzal ...	67
Tabla 16 - Franjas de horarias y jornadas.....	68
Tabla 17 - Pág. 1 Aulas de Clase.....	69
Tabla 18 - Pág. 2 Aulas de Clase.....	70

TABLA DE ANEXO

Anexo 1 - Campus de la Universidad del Valle Sede Zarzal.	67
Anexo 2 - Descripción Detallada De Los Campus.	67
Anexo 3 - Calendario Académico.	68
Anexo 4 - Aulas De Clases Sede Bolivar.	69
Anexo 5 - Aulas de Clases Sede Balsas.	70

RESUMEN

El presente trabajo de grado, integra un modelo lógico matemático para la optimización de la asignación de aulas y horarios de clases para la Universidad del Valle sede Zarzal, por lo tanto, se incorporan técnicas de programación evolutiva (**PE**) que ayudan para cumplir con las especificaciones funcionales que buscan el mejor aprovechamiento de los recursos para garantizar su mayor aprovechamiento.

Se utilizaron técnicas de análisis, para poder identificar la forma como se realizaba el proceso de asignación de horarios y aulas de clase por parte del área encargada en la Universidad del Valle sede Zarzal, de esta forma se construyó un algoritmo único que incorpora todas actividades para minimizar el proceso por medio de un algoritmo evolutivo (**AG**) y aprovechar un modelo matemático apropiado para cumplir con las restricciones y exigencias del problema de asignación.

El software del prototipo se desarrolló en el lenguaje RUBY y AMPL, que es de libre licenciamiento, el cual ayudó a incorporar secuencias de código de forma más sencillas y aprovechar el uso del modelo matemático.

Los resultados generados por el prototipo y las referencias bibliográficas con las que se realizó este trabajo se encuentran referencias en las secciones 6.3 de pruebas y resultados.

Palabras clave:

Asignación de aulas de clases, programación evolutiva (**PE**), método de programación simplex (**MPS**), algoritmos evolutivos (**AG**).

1. INTRODUCCIÓN

La asignación de los horarios y aulas de clases de la Universidad del Valle sede Zarzal, entra en los problemas de complejidad NP-HARD, se busca el planteamiento de una solución para la asignación de horarios y aulas de clases para los programas académicos y los cursos que se ofrecen desde la oficina de proyección social y extensión; este tipo de problemas es reconocido en el mundo organizacional dentro del área “University Course Timetabling”. (Programación de Cursos Universitarios), como “timetabling” (Programación de horarios; dentro de un ambiente universitario usando modelos de computación evolutiva, este problema contempla maximizar la optimización de la distribución de aulas y horarios.

La solución se fundamenta en enfrentar el problema del proceso de asignaciones por medio de la optimización de las jornadas de clases, considerando que el proceso manual es difícil por el cruce de información que es distribuida en los horarios y aulas de clases.

Este tipo de problemas ha sido materia de innumerables investigaciones, que han generado soluciones considerando el contexto donde lo proponen. Siguiendo esta línea, este trabajo presenta los antecedentes de la forma como se maneja el proceso relevantes para esta investigación enmarcados en el entorno de la Universidad del Valle Sede Zarzal, luego se elabora la descripción detallada del problema, presentando las principales características y requerimientos funcionales, continuando con el análisis de la situación actual y formulación del problema, se presentaran los resultados, conclusiones y trabajos futuros, finalizando con las referencias bibliográficas en relación con el problema de programación de la asignación de horarios, presentando los principales métodos de la solución.

2. CONTEXTO

2.1 PLANTEAMIENTO DEL PROBLEMA

La asignación de los horarios y espacios de las clases de los programas académicos, especializaciones y otros cursos que se ofrecen a la comunidad universitaria y en general, por parte de la Universidad del Valle sede Zarzal, se ha convertido en un problema, debido a la creciente demanda de los estudiantes entre el primer semestre del 2015 al primer semestre del 2019 (imagen 1), lo que ha generado una dificultad en la asignación de los espacios y las jornadas de clases.

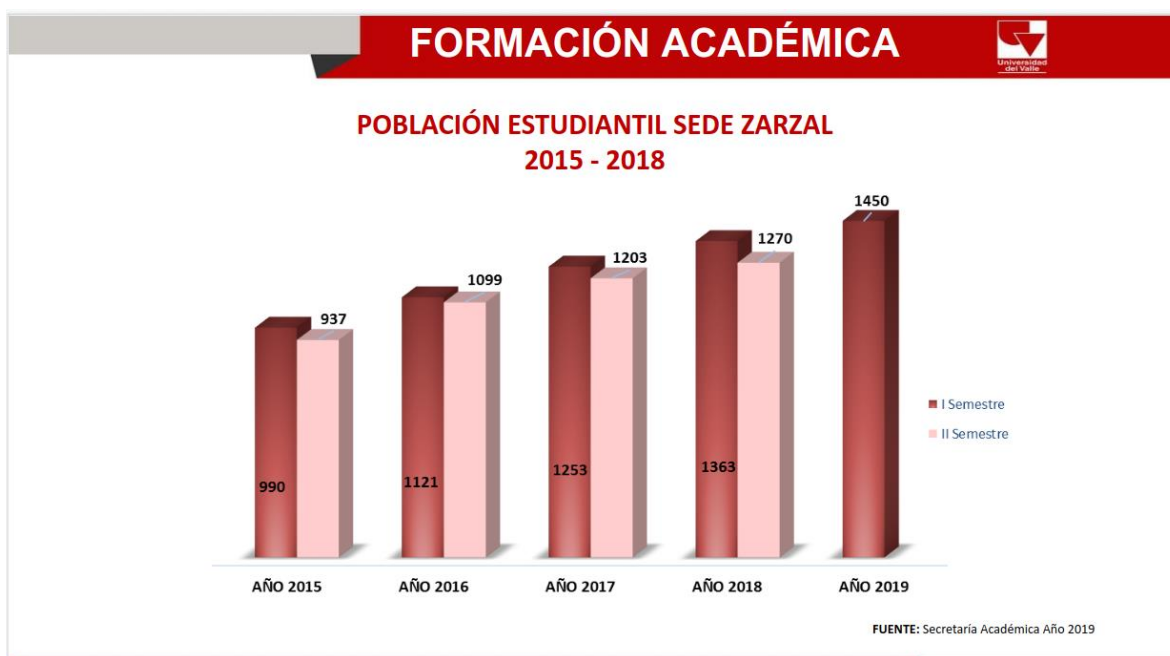


Imagen 1 - Población Estudiantil sede Zarzal
Universidad del Valle, Secretaría Académica 2015 - 2019

La distribución de horarios de las asignaturas y aulas de clases de la Universidad del Valle sede Zarzal, se realiza de forma manual, a través del conocimiento adquirido tras años de manejar procesos académicos por parte del secretario académico de la sede y del grupo de trabajo acompañados de los coordinadores de programas académicos, el área de proyección social y extensión quienes son los encargados de los procesos de asignación. Existen inconvenientes en su proceso de asignación de horarios y aulas, que generan cruces entre horarios de asignaturas en las aulas, evidenciando extensos intervalos entre clases donde no son aprovechados para optimización de espacios. Hay restricciones que deben tomarse en cuenta para la implementación de la propuesta del prototipo funcional que impone la universidad introducen complicadas y dificultosas limitaciones, a la hora de implementar una solución para optimizar el proceso de asignación de horarios y aulas de clases.

El problema de programación óptima de horarios que se propone está conformado por la asignación de los horarios de las aulas de clases donde actualmente se cuenta con 2 sedes Campus Bolívar y Campus Las Balsas (Anexo 4 y 5), campus Bolívar cuenta con 12 aulas de clase, 5 aulas especializadas (laboratorios y salas de cómputo) y 1 auditorio; con una jornada de 6 días (lunes a sábado) y un horario de 7 am. – 9 pm. (Los días festivos y domingos con permiso de las directivas) y Campus las Balsas cuenta con 13 aulas de clase, 4 aulas especializadas (salas de cómputo); con una jornada de 6 días (lunes a sábado) y un horario de 7 am. – 5 pm. (Los días festivos y domingos con permiso de las directivas); El objetivo es elegir un aula y un bloque de tiempo para cada clase de forma que se maximicen las preferencias de los estudiantes sin crear conflictos en la programación de estudiantes o aulas. La técnica empleada para resolver este problema se basa en la modificación y adecuación del algoritmo de computación evolutiva (CE) que ayuden a buscar una solución óptima al problema de asignación horarios y aulas de clase.

En la dificultad de la programación de los horarios y aulas de clases, la motivación de trabajar en desarrollar un modelo para la universidad del Valle sede Zarzal mediante computación evolutiva (CE); este modelo tendrá en cuenta no solo la programación de asignaturas de los programas académicos sino también los diversos cursos, diplomados, especializaciones, entre otros programados por la oficina de proyección social y extensión, sin olvidar los convenios que la Universidad del Valle realiza con diversas instituciones.

Elaborar una asignación de horarios óptimos, es un problema por el que se atraviesa constantemente en cualquier institución educativa, sobre todo si su población de estudiantes crece desproporcionadamente con relación a su infraestructura. En el presente trabajo se describe cómo este problema puede ser resuelto utilizando un Algoritmo Genético (AG). Esta técnica ha sido muy utilizada en problemas similares. En este caso se decidió utilizarla ya que combina rapidez y eficiencia, produciendo así buenos resultados. Actualmente, esta técnica de búsqueda ha alcanzado un impresionante éxito debido a que su aplicación en problemas de optimización no requiere de conocimiento auxiliar sobre el problema.

2.2 FORMULACIÓN DEL PROBLEMA

El problema se analizó, partiendo de unas preguntas que se les realizaron a los Coordinadores de programas y secretaria académica.

La asignación de aulas y horarios de clases de la Universidad del Valle sede Zarzal, para los programas académicos, desde los inicios de la sede hasta la actualidad este proceso se viene realizando de manera manual teniendo en cuenta cada una de las restricciones y la numerosa cantidad de cambios que se realizan por diferentes razones:

- La asignación de los horarios de clases empieza programando las asignaturas partiendo de la malla o pensum programático de cada uno de los programas académicos, esta labor se realiza una vez por semestre.
- Cada coordinador de programa se encarga de seleccionar y distribuir a los profesores de acuerdo con las asignaturas y su experiencia en ellas, y distribuye los horarios de acuerdo con la disponibilidad de estos.
- Los horarios provisionales convergen en la oficina de Secretaria Académica, hay el secretario académico procede a asignar las aulas, a los programas académicos; empieza en una hoja a distribuir los horarios en la jornada académica de 6 días (lunes a sábado) y un horario de 7 am. – 9 pm. (los días festivos y domingos con permiso de las directivas).

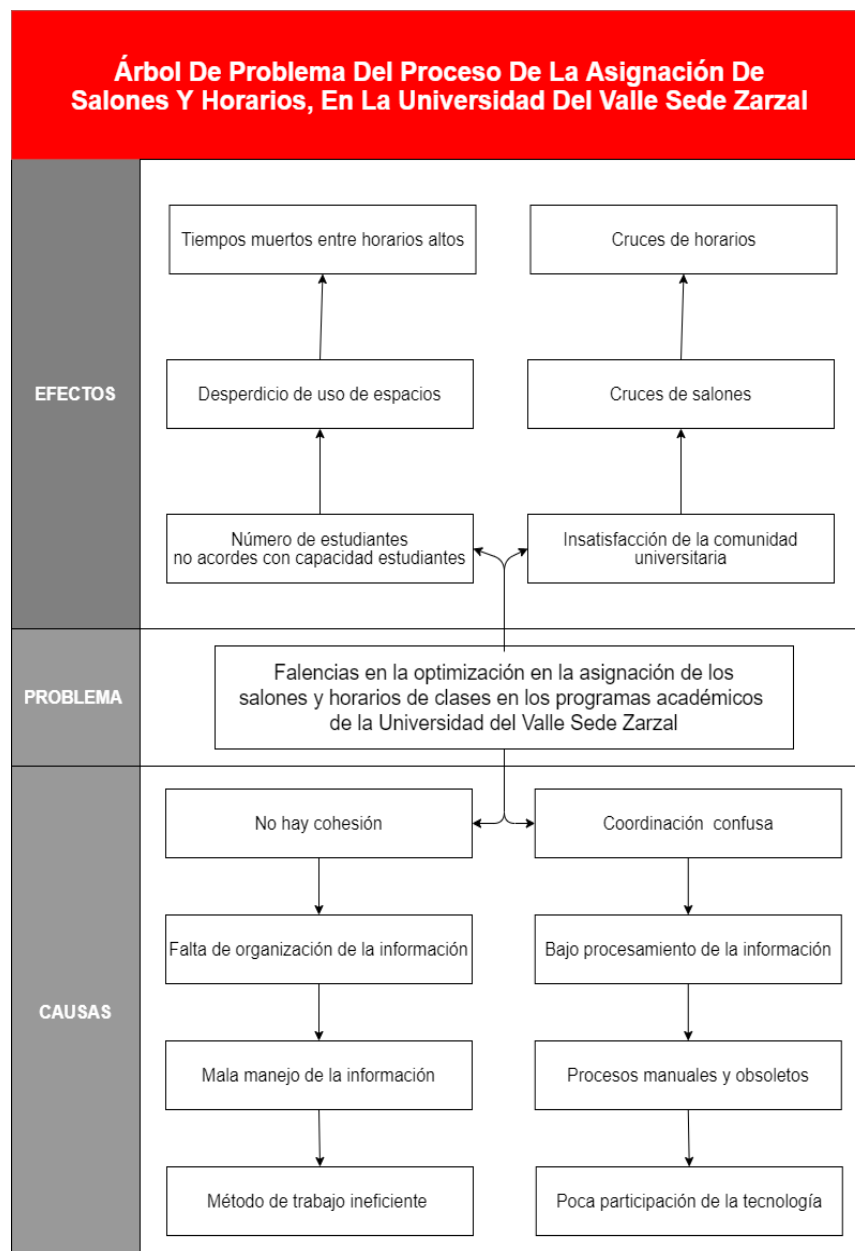


Imagen 2 - Árbol De Problema Del Proceso De La Asignación De Aulas Y Horarios, En La Universidad Del Valle Sede Zarzal

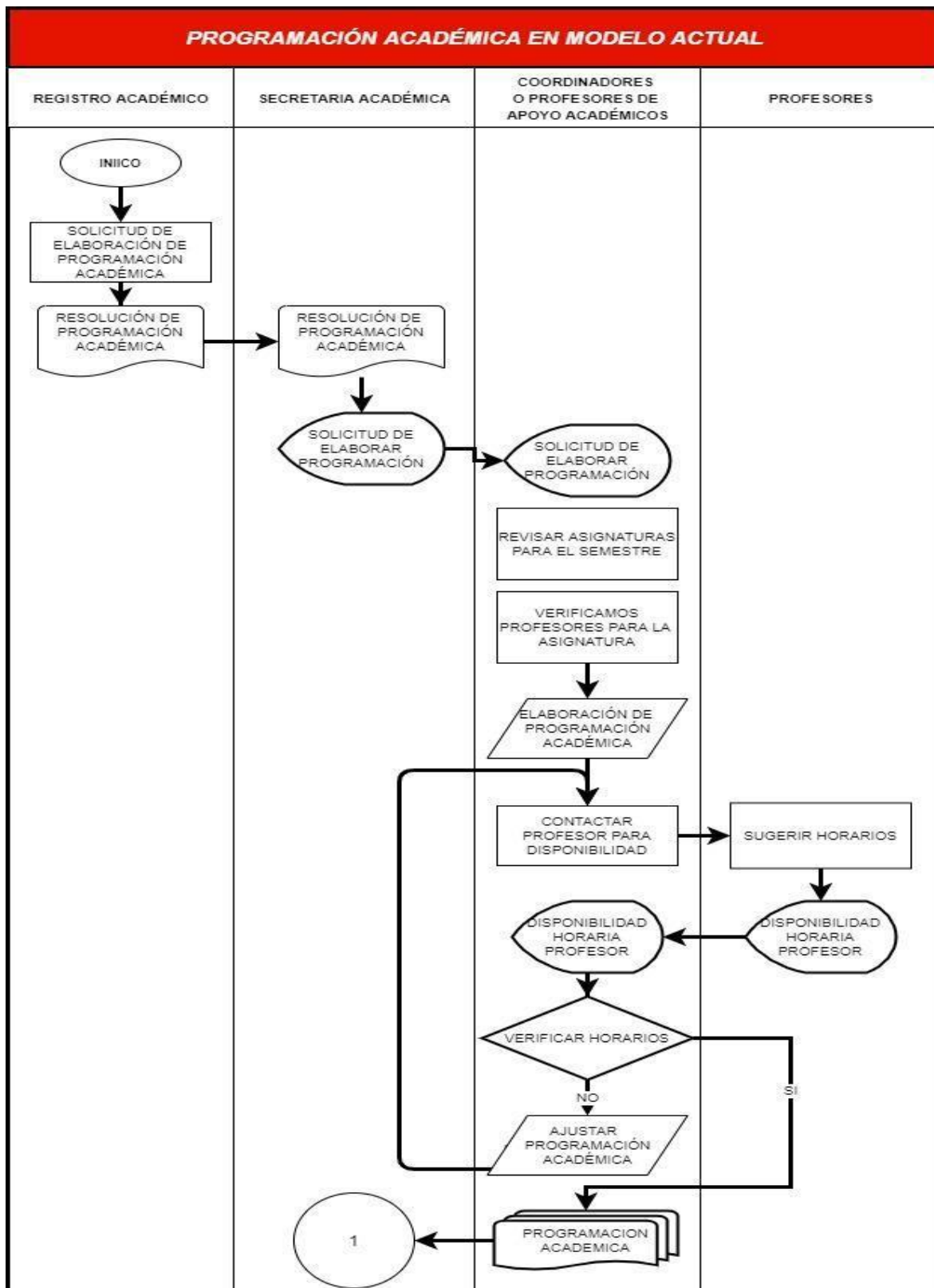


Imagen 3 - Pág. 1 Proceso Actual De La Asignación De Aulas Y Horarios En La Universidad Del Valle Sede Zarzal, Elaboración propia.

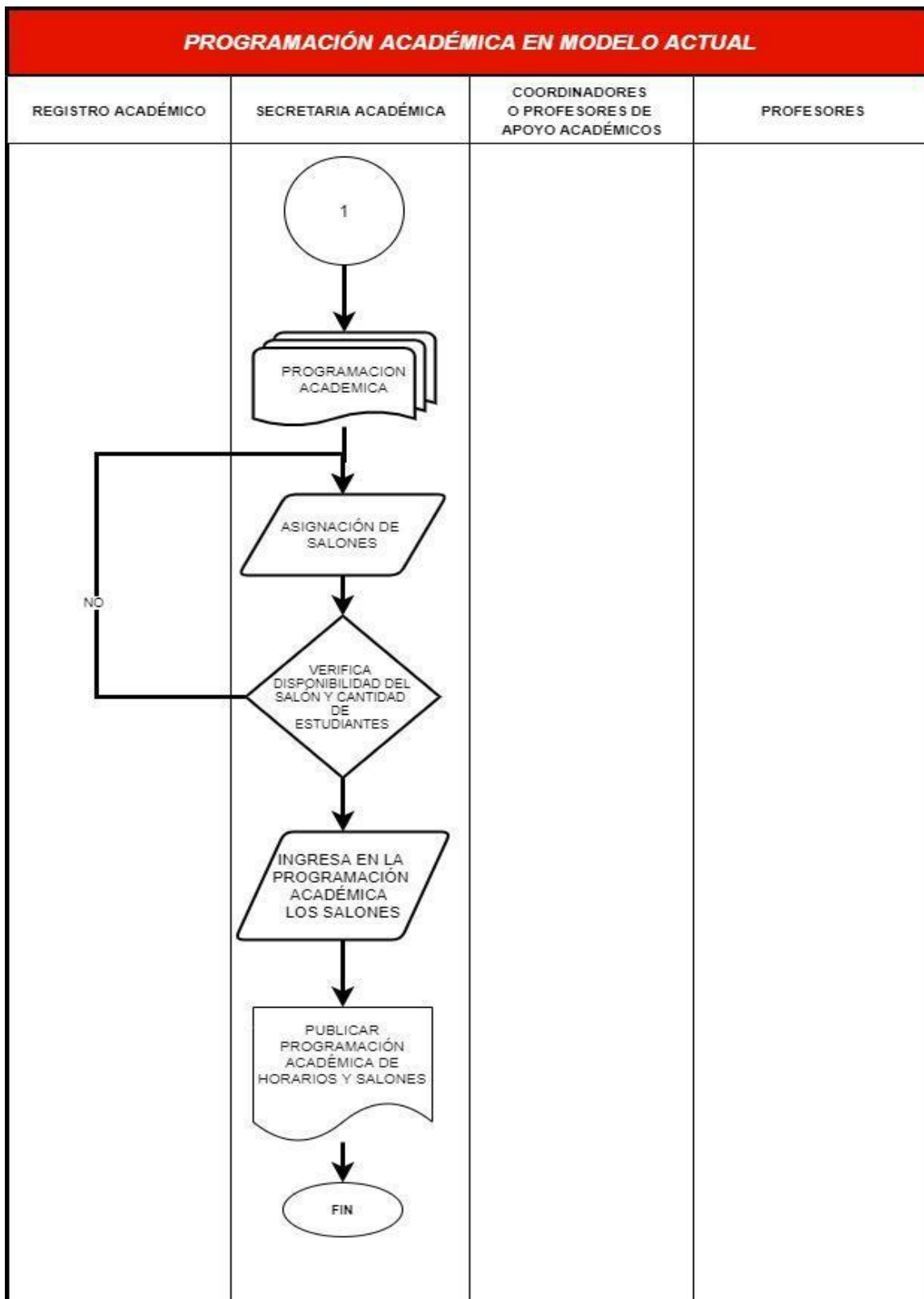


Imagen 4 - Pág. 2 Proceso Actual De La Asignación De Aulas Y Horarios En La Universidad Del Valle Sede Zarzal, Elaboración propia

2.3 PREGUNTA DE INVESTIGACIÓN

¿Existe un algoritmo que maximice la optimización de la distribución de horarios y aulas de clases?

2.4 HIPÓTESIS

Se pueden identificar un modelo matemático para buscar la alternativa más viable para dar solución al problema de asignación de aulas y horarios de clases de la Universidad del Valle sede Zarzal, que integre técnicas en programación evolutiva que ayuden a optimizar los procesos académicos – administrativos considerando mejor rendimiento que los modelos y algoritmos propuestos o disponibles en la literatura.

3. MARCO TEÓRICO Y ESTADO DEL ARTE

3.1 MARCO DE ANTECEDENTES

Actualmente existen numerosos estudios e investigaciones, sobre “Timetabling” (Programación de horarios), atendiendo todas las problemáticas y restricciones, encontrando una solución óptima, para la asignación de aulas y horarios de clases, citaremos un par de investigaciones:

- “Sistema de apoyo a la generación de horarios basado en algoritmos genéticos”; El artículo presenta un estudio para solucionar el problema de elaboración de horarios o timetabling en una facultad. Propone un modelo matemático en el cual se definen las restricciones del problema, y luego se establece el diseño de la solución y la adaptación del algoritmo a esta. [1]
- “Modelo de asignación de salones y horarios mediante algoritmos evolutivos La asignación de horarios de clases y salones en el programa de Ingeniería Industrial de la Facultad de Ingeniería de la Universidad de La Guajira. Ingeniero José María Mejía Caballero y el ingeniero Carlos Paternina Arboleda Universidad del Norte. Aplicación de algoritmos evolutivos para la asignación de horarios de clases y salones en el programa de Ingeniería Industrial de la Facultad de Ingeniería de la Universidad de La Guajira.[2]
- “Modelo matemático para la programación de un horario escolar con multi-localización de docentes, para el Ministerio Nacional de Educación. Ingeniera Industrial Linda Lucía Esquivel T. un modelo matemático para la construcción de un horario escolar, cuyo entorno es una Institución Educativa del sector público, conformada por varias sedes. El horario tendrá en cuenta el Proyecto Educativo Institucional (PEI) y algunas necesidades y requerimientos de los docentes de secundaria, a través del desarrollo de un modelo metodológico utilizando una herramienta basada en la programación. [3]

- “Programación óptima de horarios de clase usando un algoritmo memético”: El objetivo es elegir un salón y un bloque de tiempo para cada clase de forma que se maximicen las preferencias de los estudiantes sin crear conflictos en la programación de estudiantes o salones. La técnica empleada para resolver este problema se basa en la modificación y adecuación del algoritmo genético propuesto por Chu-Beasley. [4]
- “Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación entera. Un aspecto importante en la gestión académica de las universidades es la generación de horarios y la asignación de salas de clase para los distintos cursos que realizan. [5]
- “Problema de asignación óptima de salones resuelto con Búsqueda Tabú”; En este artículo se presenta un modelo de programación entera el cual decide simultáneamente los horarios de los cursos y la asignación de salas.[6]
- “Aplicación De Programación Lineal Para La Asignación De Horarios En Una Institución Educativa Mexicana”; En este trabajo se pretende optimizar la asignación de cursos a grupos en un conjunto de periodos de tiempo con determinados requerimientos. [7]
- “Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos”; Elaborar una asignación de horarios óptimos, es un problema por el que se atraviesa constantemente en cualquier institución educativa, sobre todo si su población de estudiantes crece desproporcionadamente con relación a su infraestructura. En el presente trabajo se describe cómo este problema puede ser resuelto utilizando un Algoritmo Genético. [8]
- “Implantación Del Módulo De Asignación De Horarios Para Los Docentes En La Facultad De Ingeniería De La Universidad De San Buenaventura Sede Bogotá”; el cual tiene como objeto académico mejorar y fortalecer los procesos que se ejecutan en la Facultad de Ingeniería entre ellos está la asignación y consulta de horarios Docentes la inscripción de materias, consulta de notas, entre otros. [9]
- “Diseño de una herramienta de software que asigne eficientemente las aulas para todos los cursos de pregrado de la Universidad Tecnológica de Bolívar usando un algoritmo genético”; La asignación de aulas es uno de los subproblemas a solucionar en la programación de horarios universitarios (University Course Timetable Programming, UCTP), en conjunto con la asignación de docentes y la programación de los cursos (Chaudhuri & Kajal, 2010). [10]
- “Diseño de un modelo de programación lineal para la asignación de aulas en la Institución Universitaria Esumer, Colombia; este trabajo se centra en la aplicación de un modelo de programación lineal para la asignación de aulas que minimice los costos ociosos de la Institución Universitaria Esumer y optimice los recursos académicos a utilizar”. [11]

3.2 MARCO CONCEPTUAL

Tratándose de modelos de programación con un alto grado de complejidad, se requiere una estrategia que recurrirá a diversas técnicas de encontrar la solución. Los modelos de referencia se encaminan a la heurística y metaheurística, que dan un gran abanico de variables adecuadas, efectivas y satisfactorias en relación con el tiempo que demanda llegar a una solución a través de un método óptimo:

Computación Evolutiva (CE)

Bajo el término de Computación Evolutiva se engloba a un amplio conjunto de técnicas de resolución de problemas complejos basados en la emulación de los procesos naturales de evolución. [12]

El principal aporte de la Computación Evolutiva a la metodología de resolución de problemas consiste en el uso de mecanismos de selección de soluciones potenciales y de construcción de nuevos candidatos por recombinación de características de otros ya presentes, de modo parecido a como ocurre en la evolución de los organismos naturales.

Las implementaciones concretas de Computación Evolutiva se conocen como Algoritmos Evolutivos. El propósito de los Algoritmos Evolutivos es guiar una búsqueda estocástica haciendo evolucionar un conjunto de estructuras y seleccionando de modo iterativo a las más aptas. Un Algoritmo Evolutivo se ejecuta sobre una población de individuos, que representan un conjunto de candidatos a soluciones de un problema. Dichos individuos son sometidos a una serie de transformaciones, con las que se actualiza la búsqueda, y después a un proceso de selección que favorece a los mejores individuos. Cada ciclo de transformación-selección constituye una generación. Se espera del Algoritmo Evolutivo que tras cierto número de generaciones (iteraciones), el mejor individuo esté razonablemente próximo a la solución buscada.

Los Operadores Evolutivos son los que realizan los cambios de la población durante la ejecución de un Algoritmo Evolutivo. Clásicamente, existen dos operadores genéticos: mutación (Es un operador unario que simula el proceso evolutivo que ocurre en los individuos cuando cambian su estructura genética.) y cruce (Es un operador normalmente binario, que permite representar los procesos de apareamiento natural usando operaciones sencillas. toman diversos componentes de distintos individuos para generar con ellos nuevos individuos, de tal manera que hereden características de sus padres). Sin embargo, han sido propuestos otros operadores que se acoplan a problemas particulares, por ejemplo: Dominación, Segregación, Translocación, Inversión, entre otros.

Los pasos de un Algoritmo Evolutivo son los siguientes:

1. Generación de una población inicial, generalmente en forma aleatoria.
2. Evaluación de los individuos

3. Selección de algunos individuos de la población, a través de algún mecanismo.
4. Modificación de los genes de los progenitores seleccionados usando los operadores genéticos.
5. Generación de una nueva población mediante algún mecanismo de reemplazo.
6. Verificación del criterio de parada del Algoritmo, y se regresa al paso 3, si es el caso.

Las principales técnicas de la Computación Evolutiva son: los Algoritmos Genéticos propuestos por Holland, las Estrategias Evolutivas propuestas por Rochenberg y Schwefel, la Programación Genética propuesta por Koza, y la Programación Evolutiva propuesta por Fogel. [13]

Programación Evolutiva (PE)

La Programación Evolutiva fue originalmente concebida por L. Fogel en 1960. Es un mecanismo que hace evolucionar un conjunto de máquinas de estado finito. Esta técnica desarrolla un conjunto de soluciones, las cuales muestran un comportamiento óptimo con respecto a un ambiente y a una función deseada. A continuación, se hace una explicación exhaustiva de como la Programación Evolutiva hace evolucionar las máquinas de estado finito. [19]

- InitPopulation $P(t)$: Comienza con una población de MEF que representan soluciones al problema en cuestión.
- evaluate $P'(t)$: Cada máquina es evaluada por medio de una función específica, que calcula la capacidad del individuo para resolver el problema.
- Mecanismo de Selección: todos los individuos son seleccionados, ya que cada máquina genera un descendiente a través de un proceso de mutación que se aplica sobre ella.
- $P'(t) = \text{mutate } P(t)$: cada miembro de la población es alterado a través de un proceso de mutación, el cruce no es utilizado. Clásicamente existen cinco tipos de mutaciones aleatorias:
 - Cambiar un símbolo de salida.
 - Cambiar la transición de un estado a otro.
 - Agregar un estado.
 - Eliminar un estado.
 - Cambiar el estado inicial.
 - $P(t+1) = \text{survive } P(t), P'(t)$: se escogen los individuos que van a sobrevivir para la siguiente generación. Normalmente se toman las k mejores máquinas de la población total. Generalmente el tamaño de la población permanece constante, pero no hay restricción en este caso.
- Stopping Criterion: El proceso termina cuando la solución alcanza una calidad predeterminada, cuando ha sido obtenido un número específico de iteraciones (generaciones), o algún otro criterio de parada.

El primero en discutir problemas evolutivos con un espíritu científico (100 años antes que Darwin), p.ej. consideró las similitudes entre el hombre y los simios y especuló sobre la posible existencia de un ancestro común (aunque él mismo negó esa posibilidad).

La computación evolutiva tiene una historia larga y fascinante. En sus orígenes, varios investigadores propusieron de manera totalmente independiente diversos modelos computacionales con objetivos específicos, los cuales tenían en común su inspiración en la evolución natural. Con el tiempo, las evidentes similitudes entre estas técnicas llevaron a acuñar el término genérico “Algoritmo Evolutivo” en la década de los noventa del siglo pasado. Fogel, D. B. *Evolutionary Computation: The Fossil Record*, The Institute of Electrical and Electronic Engineers, New York, USA, 1998. [13], [14], [18]

Espero haber presentado una visión aceptable del mecanismo de la evolución. Es el momento de pasar estas ideas a nuestros programas. ¿Por dónde empezar? ¿Qué tipos de Computación Evolutiva existen? Las nuevas ciencias siempre pasan por una fase inicial de desconcierto e inconsistencia hasta llegar a unas convenciones aceptadas por todos. Más o menos la clasificación aceptada en nuestros días es la del esquema siguiente.

- Inteligencia Artificial.
 - Enfoque simbólico o ‘top-down’.
 - Enfoque sub-simbólico o ‘bottom-up’.
 - Redes neuronales artificiales.
 - Computación evolutiva o algoritmos evolutivos.
 - Solidificación o recocido simulado (simulated annealing).
 - Algoritmos evolutivos.
 - Estrategias evolutivas.
 - Clasificadores genéticos.

Programación Genética (PG).

El enfoque sub-simbólico de la IA se caracteriza por crear sistemas con capacidad de aprendizaje. Éste se puede obtener a nivel de individuo imitando el cerebro (Redes Neuronales); o a nivel de especie imitando la evolución, lo que se ha denominado Computación Evolutiva (CE); término relativamente nuevo que intenta agrupar un batiburrillo de paradigmas muy relacionados cuyas competencias no están aún muy definidas. Hasta hace poco era común hablar de Algoritmos Genéticos (AG) en general, en vez de identificar diferentes tipos de CE, ya que el resto de los algoritmos se pueden interpretar como variaciones o mejoras de los AG, más conocidos. En un AG los elementos de las cadenas (genes) son típicamente bits, y en ciertos casos, valores numéricos o strings. En la Programación Genética, los genes son instrucciones en un lenguaje de programación, y en las Estrategias Evolutivas, valores reales. Las Estrategias Evolutivas surgieron inicialmente para resolver problemas de optimización paramétrica; con el paso del tiempo fueron incorporando procedimientos propios de la computación evolutiva, con lo que han llegado a convertirse en una disciplina más. La agregación simulada (Simulated

Annealing) se puede considerar una simplificación de los AG cuyo origen está en los procedimientos físicos de solidificación controlada. Los Clasificadores Genéticos solucionan problemas de reconocimiento de patrones mediante un entrenamiento basado en ejemplos, **almacenando** en las cadenas información que relacione los datos de entrada y la salida deseada. Finalmente, a mí me gusta considerar la Vida Artificial como un súper conjunto de todas estas técnicas [13] [17].

Los algoritmos evolutivos pueden trabajar hasta con 20 soluciones, por lo tanto, no son reducibles a un modelo general explicativo de todas las realidades, aunque sí se pueden adaptar algunas particularidades que, guardando las proporciones y ajustándose a cambios, pueden intervenir en situaciones concretas. De todas maneras, los algoritmos evolutivos, al permitir series de combinaciones (hibridación) entre sus diversas expresiones, generan resultados óptimos, por ejemplo, en las pruebas porcentuales cuando se le asigna el número máximo de 20 en mutación y de 80 en recombinación, propiciando con esto que en la tercera instancia se vaya expresando la optimización. De esta manera, los horarios quedan correctamente asignados y se evidencia un menor impacto en los cruces de esos horarios. [2]

Terminología

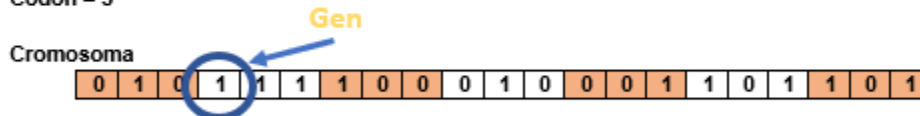
En este punto es necesario definir una serie de conceptos para poder entender los términos utilizados en los algoritmos genéticos.

- **Cromosoma.** Vector de números enteros o bits que codifica un resultado del problema.
- **Gen.** Cada una de las posiciones del vector que representa el cromosoma, es decir, la mínima unidad de información genética, representada por un número entero o bit.
- **Población.** Consiste en una serie de cromosomas. La población inicial es el conjunto de cromosomas habitualmente generados de forma aleatoria y las sucesivas serán el resultado de la evolución. En nuestra implementación se representa mediante un vector que contiene cromosoma.
- **Generación.** Fracción de la población originada tras el mismo número de iteraciones en el algoritmo genético, por lo que están en el mismo nivel del árbol de soluciones.
- **Fitness.** La función de fitness o función de idoneidad indica la calidad de cada individuo. En nuestro caso el mejor fitness será máximo valor de los resultados de la búsqueda, lo que significa que el individuo poseedor del mismo proporciona exactamente la salida esperada. Sin embargo, no siempre es conocida la solución óptima al problema que se pretende resolver.

Población: 8 individuos

0	1	0	1	1	1	1	0	0	0	1	0	0	0	1	1	0	1	1	0	1
0	1	0	1	0	1	1	0	0	0	1	0	0	0	1	1	0	1	1	0	1
0	1	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	0	0	0	1	1
0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1
0	1	0	1	1	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1
1	1	0	1	1	1	1	0	1	0	1	0	0	1	1	0	0	1	1	0	1

Codón = 3



Esto permite generar el cromosoma extendido

Grupo de genes

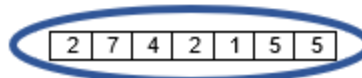


Imagen 5 - Terminología algoritmos genéticos
Elaborado por el directo

Algoritmo de Método Simplex

Algoritmo Simplex es un método que permite encontrar la mejor solución de la función objetivo. Este tipo de metodología su resultado final se da cuando no es posible continuar mejorando su resultado, es decir, cuando se llega a la solución óptima (el mayor o menor valor posible, según el caso, para el que se satisfacen todas las restricciones). La búsqueda se realiza mediante desplazamientos por las aristas del polígono, desde el vértice actual hasta uno adyacente que mejore el valor de la función objetivo. Siempre que exista región factible, como su número de vértices y de aristas es finito, será posible encontrar la solución. El método Simplex se basa en la siguiente propiedad: si la función objetivo Z no toma su valor máximo en el vértice A , entonces existe una arista que parte de A y a lo largo de la cual el valor de Z aumenta.¹

El método simplex empieza con una función a minimizar o maximizar y un conjunto de restricciones que normalmente no verifican estas condiciones. En la etapa reguladora se transforma el conjunto de restricciones en otro con términos independientes no negativos, y en la conocida como de interacciones estándar se persigue conseguir coeficientes no negativos en la función transformada que debe optimizarse, mientras que al mismo tiempo se busca conservar los términos independientes no negativos. Si esto es posible, se obtiene la solución óptima. Si no, el problema es no acotado o no factible. [15]

¹ "PHPSimplex Optimizando recursos con Programación Lineal, 15 de agosto de 2020, disponible en internet: http://www.phpsimplex.com/teoria_metodo_simplex.htm".

Bases de Datos

“Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron”. [16]

Una base de datos la podemos definir como un conjunto de datos informativos organizados y estructurados en un mismo argumento para su administración, consulta e integración. En la actualidad es fundamental para la industria 4.0, encontramos inmersos en un ambiente donde las actividades que requieren algún tipo de interacción con la información, esta se administra a través del uso de bases de datos, en una universidad para el manejo de la información de estudiantes, docentes, bibliotecas físicas y digitales, en un banco, reserva de una entrada para el cine, historias clínicas de pacientes, compra y venta de productos y muchas actividades del día a día de la humanidad.

3.3 MARCO HISTÓRICO

Universidad del Valle

Durante los años cuarenta del siglo XX, el Valle del Cauca se afianzaba como una región con un cierto nivel de desarrollo, poseía una relativa y eficiente infraestructura de transporte, un incipiente ensamblaje agroindustrial y manufacturero y una gran vocación agropecuaria.

En esa época la mayoría de los habitantes del Valle del Cauca tenían una baja preparación académica y profesional, factor que influía negativamente en el crecimiento del sector industrial. El alto volumen poblacional y lo restringido del mercado laboral por su baja preparación técnica y académica, deprimieron los salarios y aceleraron la recuperación de los capitales, permitiendo adelantar nuevas experiencias industriales en asocio con capitales internacionales.²

Sistema de Regionalización de la Universidad del Valle

Desde su fundación en 1945, la Universidad del Valle se concibió como una universidad territorial del Departamento del Valle del Cauca y no sólo de Cali, con el propósito de impulsar el desarrollo integral de toda la región. En 1959 se plantea la necesidad de conformar un sistema universitario regional desconcentrado. En aquel entonces, cuando se empezaba a consolidar la agroindustria regional, existía

² “Historia de la Universidad del Valle, Reseña Histórica 60 años, 15 de agosto de 2020, disponible en internet: <http://aniversario60.univalle.edu.co/historia/resenha/index.html>”.

todavía un cierto equilibrio entre la población urbana y la rural; el metropolitana de Cali frente al resto del Valle no era tan marcada y la Universidad contaba apenas con 566 estudiantes.³

Universidad del Valle Sede Zarzal

Mediante Resolución No. 105 de septiembre 05 de 1986 emanada del Consejo Superior de la Universidad del Valle, se crean los Centros Universitarios Regionales, entre ellos la SEDE ZARZAL. Inicio labores en 1986 con la conformación de la Fundación de Apoyo por los sectores cívico y gubernamental del municipio. La zona de influencia se destaca por su gran actividad agrícola, por ello la Sede Zarzal ofrece programas Académicos para fortalecer el desarrollo de esta región.⁴

La sede Universidad del Valle en Zarzal, por su localización esta llamada a prestarle los servicios de educación superior de excelencia a un territorio habitado por cerca de 400.000 personas provenientes principalmente de los municipios de Andalucía, Bolívar, Buga la Grande, El Dovio, La unión, Obando, Roldanillo, Versalles, Toro y Zarzal.⁵

4. MOTIVACIÓN Y JUSTIFICACIÓN

La asignación de aulas y horarios de clases ha sido objeto de intensa investigación durante varias décadas, estos pueden ser capaz de encontrar la asignación óptima de los horarios y aulas de clases a través de la computación evolutiva y algoritmos genitivos, dando solución a determinados tipos de problemas, como por ejemplo la programación de actividades, asignación de personal, en transporte, programación de citas de una clínica y administración de inventarios, hacen parte de los desafíos de Distribución y logística aplicando los modelos incluidos en la industria 4.0.

La programación atreves de computación evolutiva y algoritmos evolutivos para el proceso de asignación de aulas y horarios de clases, reduce las inconsistencias, ayudando a una mayor satisfacción a todos los actores involucrados en este proceso, que el manejo de programación académica de acuerdo a restricciones, ya que la sede actual cuenta con una cantidad limitada de aulas de clases, un aumento en la oferta de programas académicos por parte de la sede y el aumento de la demanda de los estudiantes, por lo tanto realizar la asignación de horario y aulas a mano, significa analizar un conjunto de probabilidades de asignación, se hace muy

³ “Universidad del Valle, Dirección de Regionalización, Reseña Histórica, Antecedentes, Página 1 de 5, texto completo, 15 de julio de 2020, disponible en internet: <http://regionalizacion.univalle.edu.co/resena-historica>”.

⁴ “Universidad del Valle, Secretaría Académica, Consejo Académico, Resoluciones, Página 1 de 5, texto completo, 15 de agosto de 2020, disponible en internet: <http://proxse16.univalle.edu.co/~secretariageneral/consejo-academico/resoluciones/1986/RCA-134.pdf>”.

⁵ “Universidad del Valle, Dirección de Regionalización, Sede Zarzal, Reseña Histórica, Antecedentes, Página 1 de 5, texto completo, 15 de agosto de 2020, disponible en internet: <http://zarzal.univalle.edu.co/la-sede/informacion/historia-sede>”

complejo, desgastante y de mucha pérdida de tiempo, permanece limitado a ser un ejercicio de prueba y error, para encontrar una solución que se adapte las necesidades de la comunidad universitaria; además hay que considerar que exploran diferentes investigaciones que plantean soluciones a la asignación de aulas y horarios de clases,

Es materia de investigación la asignación de aulas mediante un procedimiento de optimización metaheurística para el problema de timetabling, mediante los algoritmos evolutivos, son técnicas de optimización basadas en los principios de evolución natural. Aunque estos algoritmos han sido utilizados de manera exitosa en la solución de varios problemas de optimización, el beneficio de tales algoritmos, se han venido utilizando para acercarse a soluciones a problemas NP-hard.

Se plantear una solución donde se reduce el tiempo y la presencia de errores en la asignación manual de los horarios de clase por parte de los coordinadores de programas o profesores de apoyo y secretaria académica, y satisfacer al máximo las necesidades de los usuarios, además optimizando tanto la calidad de sus soluciones como los recursos requeridos por el algoritmo. Esto conlleva a tener un avance tecnológico y organizativo cuyo objetivo es el de minimizar los cruces entre asignaturas, aulas y horas.

La investigación busca un progreso administrativo y tecnológico que provoque directamente un beneficio para toda la Universidad del Valle sede Zarzal, la apropiación e innovación abierta, además de una estrategia digital, administrativas y científicas, hacen parte de la busca diaria por la excelencia académica.

Beneficios

- Incrementar el campo y manejo del tema.
- Preparar las bases para futuros trabajos de grado sobre el tema.
- Fortalecer la Línea de investigación de la asignación aulas de clases.

Beneficiarios

- Estudiantes de la Universidad del Valle sede Zarzal
- Universidad del Valle sede Zarzal
- Campo del desarrollo de aplicaciones en la asignación aulas de clases basados en computación evolutiva y algoritmos evolutivos

5. OBJETIVOS

Árbol De Objetivos Del Proceso De La Asignación De Aulas Y Horarios, En La Universidad del Valle Sede Zarzal				
OBJETIVOS GENERALES	Proponer un modelo de asignación de horarios y aulas de clases en la Universidad del Valle Sede Zarzal, con los objetivos que contribuya al proceso de toma de decisiones en la institución.			
OBJETIVOS ESPECIFICOS	Analizar el proceso de asignación aulas y horarios de clases de la Universidad del Valle Sede Zarzal	Formular un modelo matemático para el proceso de asignación de aulas y horarios de clases de la institución	Implementar un prototipo que incorpore los modelos, diseños y técnicas desarrolladas en el proyecto	Validar el modelo mediante casos de pruebas para verificar el funcionamiento
ACTIVIDADES	• Análisis • Manejo de la información • Análisis del proceso	• Restricciones • Modelo Matemático • Variables • Método de programa	• Población • Reproducción, cruce y mutación • Evaluación	• Resultados • Evaluar • fortalezas y debilidades • mejoras
RESULTADOS	La capacidad salón estudiantes adecuada	Maximizar el uso de espacios	Optimización de horarios	Sin cruces de horarios ni de aluas

Imagen 6 - Árbol De Objetivos Del Proceso De La Asignación De Aulas Y Horarios, En La Universidad del Valle Sede Zarzal, Elaboración propia

5.1 OBJETIVO GENERAL

Proponer un modelo de asignación de horarios y aulas de clases en la Universidad del Valle Sede Zarzal, con los objetivos que contribuya al proceso de toma de decisiones en la institución.

5.2 OBJETIVOS ESPECÍFICOS

- Analizar el proceso de asignación aulas y horarios de clases de la Universidad del Valle Sede Zarzal.
- Formular un modelo matemático para el proceso de asignación de aulas y horarios de clases de la institución.
- Implementar un prototipo que incorpore los modelos, diseños y técnicas desarrolladas en el proyecto.
- Validar el modelo mediante casos de pruebas para verificar el funcionamiento.

5.3 RESULTADOS ESPERADOS

Este proyecto pretende integrar un sistema para la optimización del proceso de asignación de aulas y horarios de clases de la Universidad del Valle sede Zarzal

basándose en programación evolutiva para la búsqueda de coincidencias en las secuencias de horarios para generar un mejor aprovechamiento de los espacios asignados a las aulas de clases, que soporten los requerimientos funcionales de las condiciones de mejoramiento continuo.

En la entrega final y en una sección del documento se especificará las pruebas que se realizarán sobre el prototipo, con sus respectivos resultados y comentarios.

El proponente aquí nombrado es consciente del riesgo que se asume con la realización de este proyecto y asume toda responsabilidad en las consecuencias de este.

5.4 METODOLOGÍA

El problema de asignación de aulas y horarios de clases presenta una alta complejidad lógico-matemática, lo que estimula al desarrollo de diversas metodologías que permitan encontrar soluciones óptimas a sus múltiples aplicaciones. En esta investigación se desarrolla una metodología de administración de información, que da forma al modelo lógico matemático propuesto, los resueltos implementados, gestionar los datos e interacción utilizando el software. Se presenta la metodología del proyecto y de trabajo propuesto para resolver el problema de asignación de aulas y horarios de clases optimizando los espacios en la Universidad del Valle sede Zarzal.

Se ha acogió para la investigación, la aplicación de un modelo lógico matemático en un sistema de información, que permite integrar las metodologías de los algoritmos evolutivos, métodos de programación simplex, el diseño de modelo lógico matemático, la construcción y evaluación de un modelo de ingeniería con el fundamento investigativo científico.

5.5 ALCANCES Y LIMITACIONES

Como el alcance de este proyecto de investigación se obtiene, un modelo matemático de asignación de aulas y horarios de clases en la Universidad del Valle Sede Zarzal.

Permite la administración de la información relacionada con aulas, docentes, estudiantes asignaturas y requisitos (recursos tecnológicos) por ser datos fundamentales para el modelo, se presenta una solución por el modelo, la cual puede no ser la mejor debido a las características del problema y del algoritmo el cual no asegura obtener la mejor solución aun si esta existe.

No se realiza la ejecución ni pruebas de calidad del software ya que por ser un modelo funcional podría no cumplir con todas las características necesarias para asegurar calidad y éxito.

5.6 IMPACTOS ESPERADOS

Impactos científicos y tecnológicos del proyecto

Este proyecto contribuirá a generar nuevas herramientas e investigaciones en el campo de las expresiones regulares y programación evolutiva permitiendo aumentar nuevas proyecciones para la universidad en cuanto:

- Incrementar el campo y manejo del tema.
- Preparar las bases para futuros trabajos de grado sobre el tema.
- Fortalecer la Línea de investigación de Computación Evolutiva.
- Reducir el esfuerzo y los costos de la implementación de aplicaciones en Expresiones Regulares.

Impactos sobre el medio ambiente y la sociedad

- Este proyecto contribuirá a la conservación del medio ambiente reduciendo el impacto de a través de formas como: la documentación que se genera se realizara de forma digital y la herramienta a desarrollar es de tipo software la cual ayuda a la reducción de papel.
- Los documentos que sean necesarios para las revisiones se entregaran de forma digital o si es necesario se imprimirán en papel reciclado o biodegradable.
- Adicionalmente se buscarán estrategias para la reducción del consumo de energía, para cuando no se necesite el uso de dispositivos de cómputo.

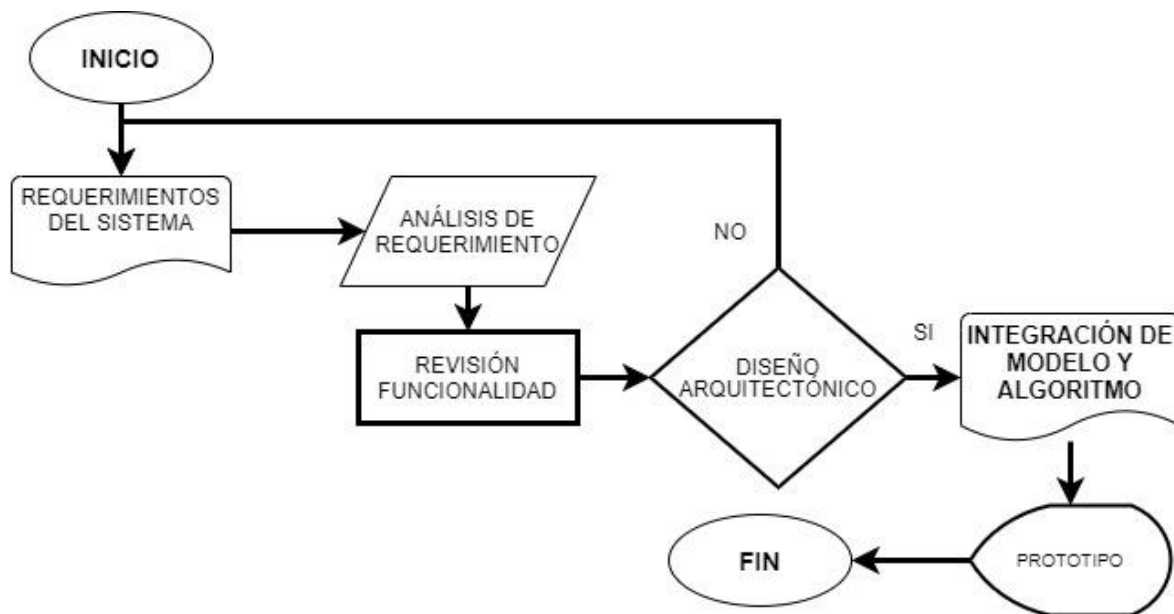
5.7 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Se realizaron entrevistas en las personas de las áreas involucradas en el proceso de asignación de horarios y aulas de clases, como lo son secretarias académicas, docentes de apoyo o coordinadores de programa y directivas de la universidad de Valle sede Zarzal, como también se manifestó la importancia de sistematizar el proceso.

5.8 POBLACIÓN DE MUESTRA

El personal de las áreas involucradas en el proceso de asignación de horarios y aulas de clases, como lo son secretarias académicas, docentes de apoyo o coordinadores de programa y directivas de la universidad de Valle sede Zarzal.

6. SISTEMA PARA LA ASIGNACIÓN DE ESPACIOS Y HORARIOS DE CLASE



*Imagen 7 - Descripción lineal del sistema funcional
Elaboración propia*

6.1 REQUISITOS DEL SISTEMA

En esta parte del documento, se identificaron las funcionalidades utilizadas del sistema actual, los requerimientos planteados que no son cubiertos por el sistema actual y las necesidades expresadas, teniendo en cuenta la estructura y las reglas asociadas a la ejecución de este prototipo, podemos plantear una serie de requerimientos básicos para la generación automática por medio de algoritmos evolutivos; este análisis independiente los componentes del sistema y los elementos que interactúan en las pruebas del sistema, para que sean aceptables a los modelos implementados en esta parte de la investigación, prevalece lo fundamental y facilita la evaluación de productos candidatos.

El objetivo en esta fase de la investigación es analizar el estado inicial en que se encuentra el sistema, donde buscará realizar la asignación de un horario y aulas. El ingreso de los datos necesarios, identificar los factores involucrados y su interrelación generan un grado de complejidad para el sistema. Se debe considerar el número de estudiantes por asignatura que se pueden matricular y teniendo en cuenta la capacidad de cada una de las aulas. Adicionalmente se debe tener claridad sobre el conjunto de restricciones que impone restricciones que corresponde a los requerimientos de la universidad, optimizar los recursos de la organización, distribución de horarios de clases y asignación de las aulas.

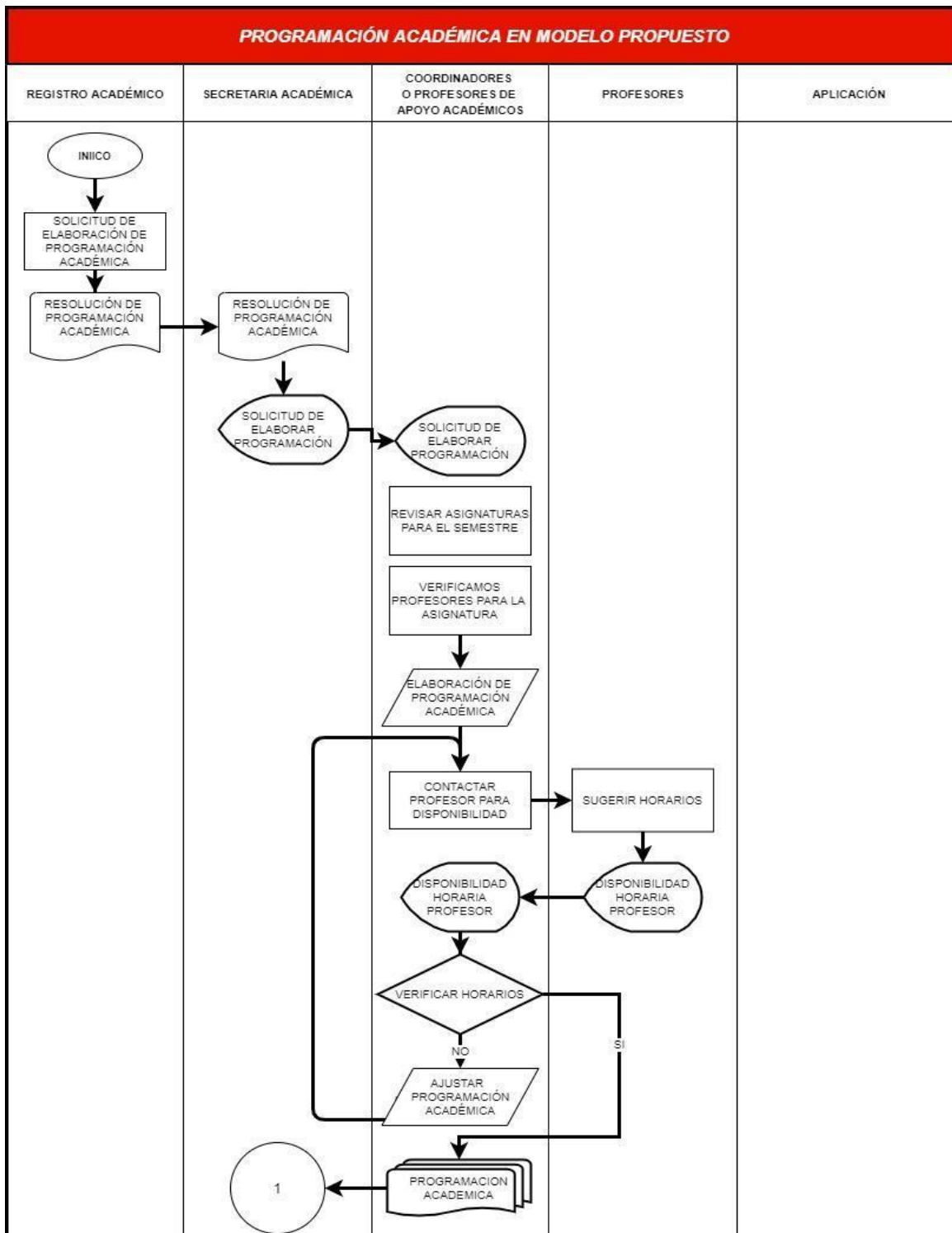


Imagen 8 - Proceso Propuesto De La Asignación De Aulas Y Horarios En La Universidad Del Valle Sede Zarzal, Elaboración propia.

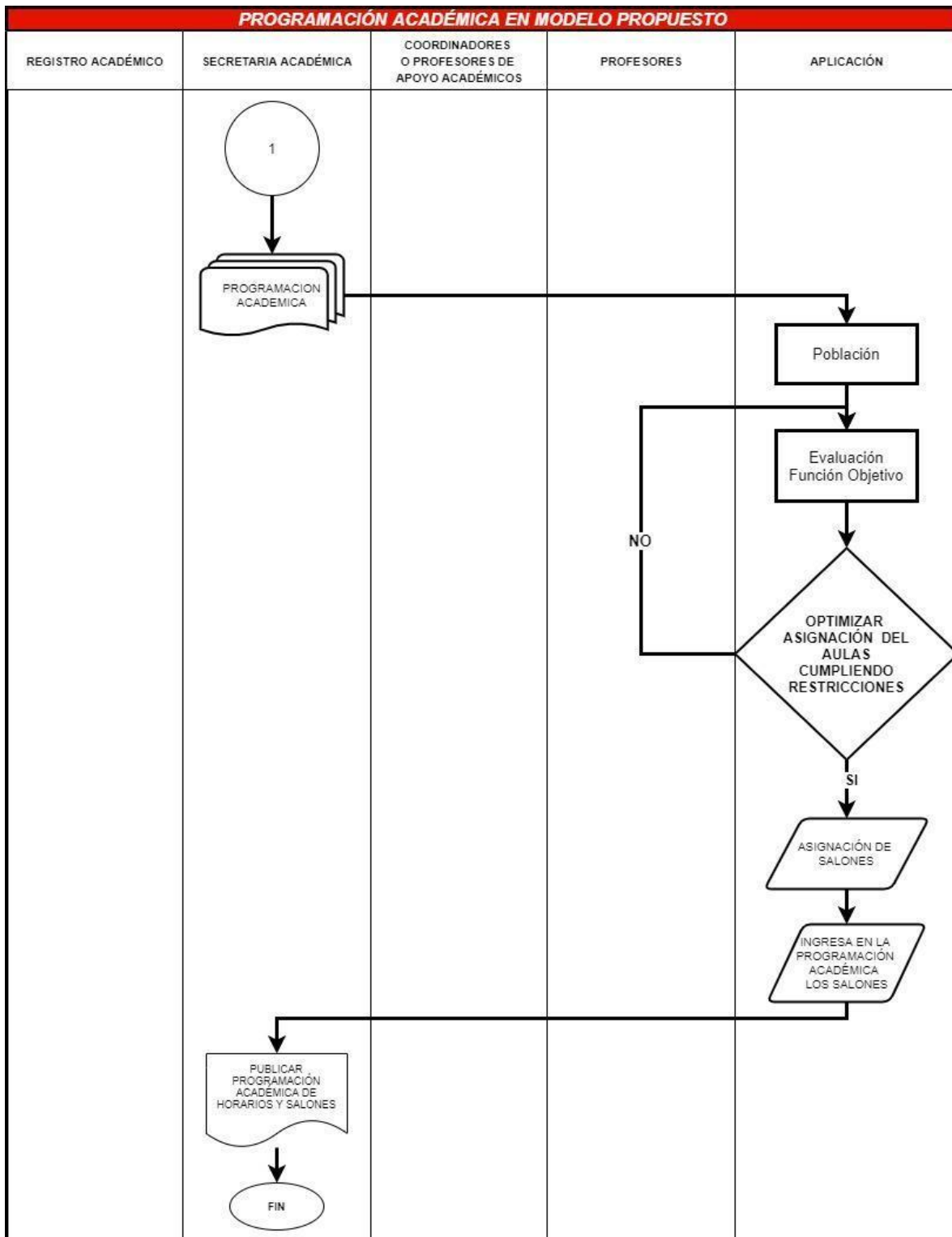


Imagen 9 - Proceso Propuesto De La Asignación De Aulas Y Horarios En La Universidad Del Valle Sede Zarzal, Elaboración propia.

6.1 FORMULACIÓN DEL MODELO MATEMÁTICA

Un modelo de programación utilizando algoritmos evolutivos para la asignación de aulas y horarios de las asignaturas de la Universidad del Valle sede Zarzal, cuyo problema de optimización metaheurística, tomando como documentos guía “*Asignación de horarios de clases Universitarias mediante algoritmos evolutivos*” [2] realizado por el Ing. José Mejía Caballero y “*Diseño de un modelo de programación lineal para la asignación de aulas en la Institución Universitaria ESUMER, Colombia*” [11] realizado por magister en ingeniería administrativa Eduardo Alexander DUQUE Grisales, magíster en gestión financiera Juan David González Ruíz e ingeniero industrial y magister en ingeniería John Dairo RAMÍREZ Aristizábal .

Debemos que tener en cuenta que los profesores de apoyo o coordinadores de programas realizan la programación de las asignaturas se establece el día, la hora, el profesor y la cantidad de estudiantes; quedando pendiente la asignación del aula de las asignaturas, el modelo matemático será el siguiente:

Conjuntos e índices:

$M, i \in \{1.. M \}$	Conjunto de aulas , indexado por el subíndice i .
$N, j \in \{1.. N \}$	Conjunto de asignaturas a programar, indexado por el subíndice j .
$F, f \in \{1.. F \}$	Conjunto de franjas horarias, indexado por el subíndice f .
$D, d \in \{1.. D \}$	Conjunto de franjas horarias, indexado por el subíndice d .
$Dia_Asigna_Franja_{\{d,j,f\}}$	Conjunto inducido por el día d , asignatura j y la franja f .

Parámetros:

k_i Capacidad en número de sillas del aula i

c_j Cantidad de estudiantes matriculados en la asignatura j

$$\sigma_{i,j} \begin{cases} k_i - c_j, \text{ Si la capacidad del aula } i \text{ es menor a la cantidad de alumnos de la asignatura } j. \\ 50, \text{ de lo contrario.} \end{cases}$$

$$\forall_i \in \{1, \dots, M\}, \forall_j \in \{1, \dots, N\}$$

$$x_{idjf} = \begin{cases} 1, & \text{Si el aula } i \text{ es asignada al día } d \text{ a la asignatura } j \text{ en la franja } f. \\ 0, & \text{de lo contrario} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{Si el aula } i \text{ es asignada a la asignatura } j. \\ 0, & \text{de lo contrario.} \end{cases}$$

La capacidad de estudiantes que dispone las aula, se puede obtener por medio de la dirección de infraestructura universitaria (DUI) de la universidad del valle y el área de mantenimiento e infraestructura de la universidad del Valle sede Zarzal, y la cantidad de estudiantes de cada asignatura obtiene mediante el análisis de la información el proceso de matrícula académica que realiza coordinador de programa o docente de apoyo del programa y secretaria académica de la Universidad del Valle sede Zarzal, con esto establecemos una función que representa por capacidad ociosas que quedan en una determinada aula cuando programa una asignatura.

Función objetivo:

$$\text{Minimizar } z : \sum_{i \in M} \sum_{j \in M} y_{ij} * \sigma_{i,j}$$

La función objetivo hace referencia al aprendizaje y evolución del algoritmo evolutivo que determina la calidad de la solución, la función objetivo consiste en minimizar el número de asientos universitarios disponibles en la asignación, así como los conflictos entre asignaturas, horarios y aulas; partiendo de estas premisas exponemos la siguiente expresión:

Restricciones:

- Toda asignatura j debe ser asignada.

$$\sum_{i \in M} \sum_{d1 \in D / d1=d} \sum_{f1 \in F / f1=f} x_{idjf} = 1 \quad \forall (d,j,f) \in \text{Dia_Asigna_Franja}$$

- Toda asignatura j debe ser asignada a lo máximo una vez.

$$\sum_{i \in M} y_{ij} \leq 1 \quad \forall j \in N$$

- En un día, en una franja y en un aula, no deben de existir más de una asignatura

$$\sum_{(d,j,f) \in \text{Dia_Asigna_Franja} / d1=d, f1=f} x_{idjf} \leq 1 \quad \forall d1 \in D, f1 \in F, i \in M$$

- Conexión de variables x e y

$$\sum_{d1 \in D / d1=d} \sum_{f1 \in F / f1=f} x_{idjf} = y_{ij} \quad \forall i \in M, (d,j,f) \in \text{Dia_Asigna_Franja}$$

Finalmente se procede a utilizar el software AMPL demo en la programación del modelo, genera un número de soluciones de prueba y este software utiliza métodos basados en programación simplex, mejorar constantemente los resultados de cada prueba. Para cada solución de prueba, evaluando las funciones de distribución de probabilidad y generando un nuevo valor para la casilla objetivo, una y otra vez. El resultado de cada solución de prueba se almacena en una base de datos.

6.2 DISEÑO ARQUITECTÓNICO

De manera detallada se presenta un diseño arquitectónico, se analizan los datos que provienen de secretaria académica y los profesores de apoyo o coordinadores de programa, la importancia de la selección de los criterios, la identificación de los patrones y la implementación de un modelo acorde a los criterios y patrones es fundamental para llegar a una solución de calidad, se incorpora un nuevo nivel de abstracción.

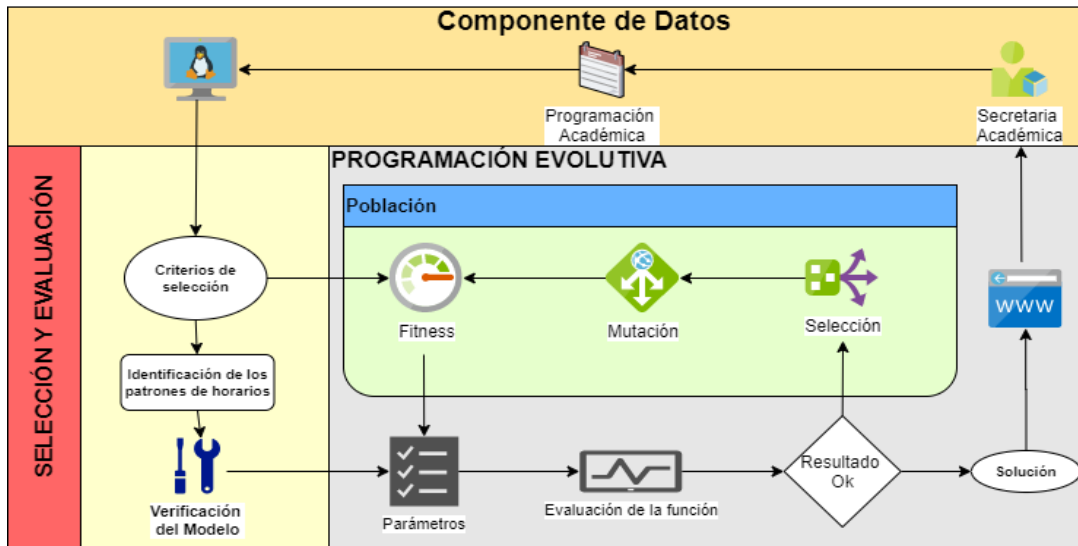


Imagen 10 - Diseño Arquitectónico
Elaboración Propia

Los pasos básicos para seguir para implementar un algoritmo evolutivo (AE) son:

- **Inicializar:** se inicializa el genotipo y los parámetros de la estrategia de cada individuo y se calcula el fitness.
- **Cruzar:** se genera una nueva generación de hijos a partir de la generación de padres haciendo uso de un operador de cruza.
- **Evaluar:** se calcula el fitness de cada individuo de la nueva generación.
- **Seleccionar:** los individuos con mejor fitness son seleccionados como la nueva generación de padres por medio de la técnica de torneo.

Una (AE) $(\mu + \lambda)$ genera λ hijos a partir de μ padres, donde $1 \leq \mu \leq \lambda < \infty$. La siguiente generación está conformada por los mejores individuos de μ y λ . El (AE) $(\mu + \lambda)$ implementa elitismo para conservar a los mejores padres.

Algoritmo evolutivo

```

1:   Inicializar población de padres  $\mu$ 
2:   while Criterio de paro no alcanzado do
3:     while hijos  $< \lambda$  do
4:       Cruzar individuos de población  $\mu$  para obtener un hijo  $h$ 
5:       Mutar  $h$ 
6:       Evaluar  $h$ 
7:     end while
8:     Seleccionar mejores individuos de  $\mu$  y  $\lambda$  para nueva población  $\mu$ 
9:   end while

```

Tabla 1 – pseudocódigo Algoritmo evolutivo
Elaboración propia

6.3 DISEÑO DETALLADO

Los algoritmos evolutivos, debemos tener en cuenta varios ítems, primero es necesario definir una población inicial y ésta será nuestro conjunto de cromosomas, la relación de los cromosomas será representados en una matriz tridimensional, sus ejes coordenados están representados en:

- El eje x, son los días de las semanas.
- El eje y, son las franjas horarias de cada día
- El eje z, las aulas de clases

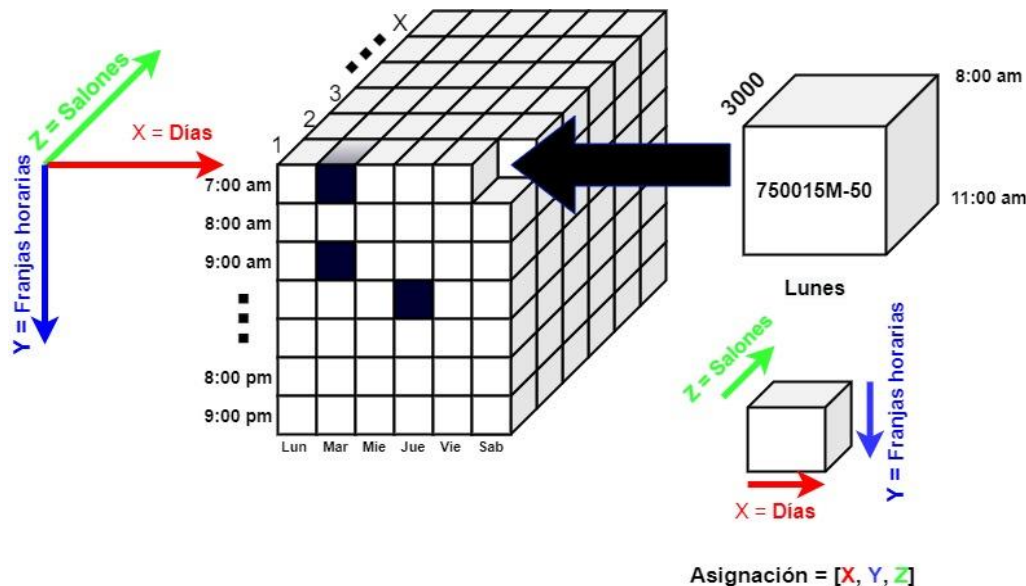


Imagen 11 - Representación del cromosoma para diseño timetabling
Universidad del Valle sede Zarzal, Elaboración Propia

Las variables horario, aula y asignatura tomarán sus valores propios, es decir, la variable aula tendrá el número del aula que le corresponde a la asignatura a el aula en un horario, la variable asignatura tendrá el código la asignatura que será impartida en el aula en un horario, la variable horaria tendrá la hora de la asignatura en el aula. El gen, ésta representada como el valor que contiene cada elemento de la matriz, como se exponen en la siguiente tabla:

Aula	Asignatura	Día	Franja	Variable binaria

Tabla 2 - Representación del gen
Elaboración propia

El cromosoma estará formado por la información correspondiente de la asignatura que estructuran el genotipo del modelo propuesto para cada día de la semana, considerando los parámetros que lo conforman, es decir se debe crear una matriz

que corresponda a la información del cromosoma con un tamaño equivalente al número de franjas horarias y su distribución.

Lunes

07:00	710205M		730069M				
08:00	710205M	710203M	730069M				
09:00	710205M	710203M	730069M				
10:00		710203M	730069M	710202M	710198M		
11:00		714202M		710202M	710198M		
12:00		714202M		710202M	710198M		

Tabla 3 - Representación de la programación del gen
Elaboración propia

Con esta información ya se puede crear el primer modelo del comportamiento del fenotipo y definir los parámetros del sistema a desarrollar considerando las franjas horarias y la ocupación por salón.

Fenotipo

Hora: 7: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	0	0

Hora: 8: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	710203M	0

Hora: 9: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	710203M	0

Hora: 10: 00 am

1062	1064	2000	2002	2004	3000
0	710202M	710198M	730069M	710203M	0

Hora: 11: 00 am

1062	1064	2000	2002	2004	3000
0	710202M	710198M	0	714202M	0

Hora: 12: 00 am

1062	1064	2000	2002	2004	3000
0	710202M	710198M	0	714202M	0

Estos valores representan la ocupación de las asignaturas según los horarios el cual permiten generar la matriz para evaluación de los Aulas.

Horarios	Aulas					
	1062	1064	2000	2002	2004	3000
07:00	710205M			730069M		
08:00	710205M			730069M	710203M	
09:00	710205M			730069M	710203M	
10:00		710202M		730069M	710203M	710198M
11:00		710202M		714202M		710198M
12:00		710202M		714202M		710198M

Tabla 4 - Resultados de la programación del gen
Elaboración propia

Una vez generados los resultados, una función de fitness evalúa su adecuación a la solución del problema considerando el método de selección que es realmente importante en los algoritmos evolutivos [20], pues determina en gran medida qué individuos sobreviven y cuáles se extinguen. Para ello es clave la función de fitness, que señalará los cromosomas más adecuados para su supervivencia. En una primera aproximación general se suelen utilizar dos métodos para el cruce: la ruleta y el torneo.

La selección por ruleta consiste en dar una probabilidad de selección a cada individuo de la forma:

$$p_i = \frac{f(i)}{\sum_{j=1}^n f(i)}$$

donde f(i) es el valor fitness del individuo i. Para cada individuo i se define una puntuación acumulada de la forma:

$$q_i = \sum_{j=1}^i p_j$$

A continuación, se genera un número aleatorio $\alpha \in [0; 1]$ y se selecciona el individuo i que cumpla $q_i - 1 < \alpha < q_i$, este proceso se repite hasta seleccionar el porcentaje de individuos deseados.

La selección por torneo consiste en elegir un conjunto de individuos al azar de la población, normalmente 2 o 3 individuos, y seleccionar el individuo con mejor valor fitness, este proceso se repite hasta seleccionar todos los individuos deseados.

En el presente trabajo se ha utilizado el método de selección por torneo ya que es el más apropiado para una implementación eficiente de (AE).

La independencia entre el algoritmo evolutivo y los horarios generados proporcionada por el proceso de transformación de genotipo a fenotipo permite que el arreglo se beneficie de todos los avances en el área de los algoritmos evolutivos.

Descripción del funcionamiento del timetabling con el algoritmo evolutivo (AE)

La separación entre el espacio de búsqueda (genotipos) y el espacio de soluciones (fenotipos o programas) permite que no exista ninguna restricción sobre los genotipos y por lo tanto tampoco sobre la búsqueda en su correspondiente espacio, manteniéndose en paralelo la validez de las soluciones [21]. Se puede encontrar una descripción de distintos métodos de generación de fenotipos válidos a partir de genotipos.

El sistema para asignación de horarios y espacios de la Universidad del Valle sede Zarzal utilizan como método de búsqueda de la solución un algoritmo evolutivo (AE), pero será válido cualquier método de búsqueda capaz de trabajar con cadenas de textos que permiten estructurar el cromosoma extendido [22] Según la propuesta original, utilizando un algoritmo evolutivo como método de búsqueda de las mejores soluciones como se muestra a continuación:

- Se inicializa la población, los genotipos de sus individuos, de manera secuencial; cada genotipo es una cadena de textos que representa el código de las asignaturas de una longitud inicialmente limitada pero que variará en el proceso evolutivo.
- A partir de cada genotipo se genera un fenotipo con un algoritmo de transformación.
- Los fenotipos de toda la población se evalúan con una función de fitness que cuantifica el acercamiento a la solución del problema que se esté tratando.
- Una vez que está inicializada la población, comienza el proceso de evolución en el que, en cada paso evolutivo, se seleccionan los mejores individuos (aquellos con mejor función de fitness) para que actúen como padres y generen una progenie que sustituiría a los peores individuos de la población (aquellos con peor fitness). La creación de los hijos a partir de los padres se

realiza con los genotipos, aplicando los operadores de cruce de un punto y selección propios de los algoritmos genéticos.

- El proceso termina cuando se genera un cromosoma con los horarios con mejor fitness el cual será la mejor opción para establecer los horarios por día, o se alcanza algún límite preestablecido como por ejemplo un número de generación, un número de ejecuciones de la función de fitness, un tiempo de proceso, etc.

La técnica (AE), según la plantean sus creadores, tiene una característica muy importante para la asignación de aulas y horarios porque este modelo permite generar un componente del sistema, y por lo tanto se puede sustituir de una manera muy sencilla. Esta característica de la técnica permite generar soluciones al problema que se esté tratando, en distintos lenguajes sin ningún coste/esfuerzo añadido.

Conversión de genotipo a fenotipo

Uno de los componentes de la técnica del (AE) es la utilización del modelo simplex que permite generar los resultados de las evaluaciones de las soluciones posibles del problema que se pretende resolver.

El algoritmo de transformación de genotipo a fenotipo consiste básicamente en construir, a partir de la distribución de la programación académica, una derivación y evaluación del fenotipo. La manera de seleccionar la regla que se va a utilizar se explica con detalle más adelante. La transformación, por lo tanto, comienza con la estructura de la asignación de los horarios por día.

Los genes se utilizan en orden secuencial, comenzando por el primero, es decir, en general, el primer gen que representa la información de una asignatura es el que se utiliza para decidir cuál de las posiciones del cromosoma va a evaluarse en la primera derivación del algoritmo y así sucesivamente.

Durante el proceso de traducción de genotipo a fenotipo, puede ocurrir que se agoten los genes del genotipo antes de que el fenotipo esté completamente terminado, es decir, esté formado únicamente por valores vacíos por tal razón se otorga el valor de 0 que quiere decir que los horarios para ese día están disponibles. La reutilización se implementa volviendo a utilizar el genotipo desde su primer gen una vez que se haya consumido el último. Este operador se inspira en el mecanismo de solapamiento de genes que ocurre en algunos organismos como por ejemplo las bacterias. Si ocurriese que el fenotipo no se termina de construir correctamente incluso después de aplicar una o más veces el nuevo operador, se le asignaría al individuo el peor valor de fitness y los mecanismos de selección y reemplazamiento se encargarían de hacer desaparecer ese individuo de la población.

El proceso de transformación de genotipo a fenotipo termina cuando en uno de los pasos del algoritmo, el fenotipo en construcción tiene un valor el cual es determinado por el fitness, es decir, está formado únicamente por asignaciones de horarios y aulas válidos.

El proceso se repite hasta que ocurre una de las dos siguientes situaciones:

- Se obtiene una programación de aulas y horarios con valores válidos y, por tanto, un fenotipo correcto cuando los valores de fitness son iguales o aproximados a cero.
- Se consume los genes y no se genera ninguna asignación de aulas correcta y, por tanto, sin obtener un fenotipo correcto.

Ejemplo

Programación académica por día de todas las asignaturas

N	Cod. Asig	Asignatura	Grupo	Horas	Día	Horario	No. Est Mat
1	207009M	DESARROLLO DE LA PERSONALIDAD I	50	3	LUN	07:00:00	5
2	801007M	TÉCNICAS DE COMUNICACIÓN	50	3	MAR	09:00:00	40
3	802082M	FUNDAMENTOS DE CONTABILIDAD FINANCIERA	50	4	LUN	07:00:00	43
4	802002M	INTRODUCCIÓN A LA CONTADURÍA PÚBLICA	50	2	JUE	13:00:00	40
5	760175M	INTRODUCCIÓN A LOS SISTEMAS DE PRODUCCIÓN LEAN	50	3	VIE	18:00:00	26
6	801157M	MATEMÁTICA BÁSICA-CONTADURÍA	50	4	SAB	14:00:00	41
7	204101M	LECTURA DE TEXTOS ACADÉMICOS EN INGLÉS I	51	3	LUN	10:00:00	11
N	XXXXXXX	XXXXXX	XXX	X	XXX	XXXXX	X

Tabla 5 - Programación académica
Secretaría académica

Se organiza los cromosomas considerando el número de aulas por día y franjas horarias para su evaluación.

Lunes

Hora: 7: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	0	0

Hora: 8: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	710203M	0

Hora: 9: 00 am

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	710203M	0

Se continúa para todas las franjas horarias en el día. Se cambian los valores correspondientes considerando las diferencias en capacidad y número de estudiantes.

La asignatura 730069M tiene 20 estudiantes y el salón 2002 es para 30 el valor resultado es de 10, en el caso de la asignatura 710205M tiene 30 estudiantes y el salón 1062 es de 15 el resultado es -20 por tal motivo se le otorga el máximo valor por ser negativo de 100, en el caso contrario que no hay asignaturas en el salón se toma como 0 porque hay disponibilidad.

Configuración del Algoritmo:

- Numero de Generaciones: 2
- Tamaño de la población 3
- % de Cruce 50
- Número de Genes: 6 (Tamaño del número de aulas disponibles)

Generación 1:

Lunes - Hora: 7: 00 am

Primera Generación

Genotipo:

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	0	0

Individuo 1:

1062	1064	2000	2002	2004	3000
710205M	0	760155M	730069M	0	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
100	0	5	10	0	0

Total, evaluación: $100+0+5+10+0+0=115$

Individuo 2:

1062	1064	2000	2002	2004	3000
0	760155M	0	730069M	710205M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	5	0	10	0	0

Total, evaluación: $0+5+0+10+0+0=15$

Individuo 3:

1062	1064	2000	2002	2004	3000
0	760155M	0	0	710205M	730069M

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	5	0	0	0	20

Total, evaluación: $0+5+0+0+0+20=25$

La mejor opción de distribución de aulas considerando el torneo es el individuo 2 con un resultado de 15.

Se genera el arreglo ordenado de los valores de los individuos con los mejores fitness

Individuo 2	Individuo 3	Individuo 1
15	25	115

El mejor individuo pasa hacer el genotipo del algoritmo por medio de la operación de selección

Genotipo:

1062	1064	2000	2002	2004	3000
0	760155M	0	730069M	710205M	0

Se realiza la operación de cruce para los demás individuos

Individuo 2:

1062	1064	2000	2002	2004	3000
730069M	710205M	0	0	760155M	0

Individuo 3:

1062	1064	2000	2002	2004	3000
0	0	760155M	730069M	710205M	0

Se genera la población para el fitness de la generación 2

Generación 2:**Individuo 1:**

1062	1064	2000	2002	2004	3000
0	760155M	0	730069M	710205M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	5	0	10	0	0

Total, evaluación: $0+5+0+10+0+0=15$

Individuo 2:

1062	1064	2000	2002	2004	3000
730069M	710205M	0	0	760155M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
5	0	0	0	15	0

Total, evaluación: $5+0+0+0+15+0=20$

Individuo 3:

1062	1064	2000	2002	2004	3000
0	0	760155M	730069M	710205M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	0	0	0	5	0

Total, evaluación: $0+0+0+0+5+0=5$

La mejor opción de distribución de aulas considerando el torneo es el individuo 2 con un resultado de 5.

Se genera el arreglo ordenado de los valores de los individuos con los mejores fitness

Individuo 3	Individuo 1	Individuo 2
5	15	20

El mejor individuo pasa hacer el genotipo del algoritmo por medio de la operación de selección

Genotipo:

1062	1064	2000	2002	2004	3000
0	0	760155M	730069M	710205M	0

Luego se hace la evaluación para las otras franjas horarias y se genera el arreglo con la distribución con las mejores horarios y asignaciones de aulas.

También se debe considerar que para la asignación de la siguiente franja horaria si la asignatura que se encuentra en la anterior franja debe estar incluida en el genotipo en ese mismo salón

Lunes - Hora: 8: 00 am

Primera Generación

Genotipo:

1062	1064	2000	2002	2004	3000
0	710202M	760155M	730069M	710205M	0

En este caso la única asignatura que se le aplicaran las operaciones del algoritmo sea la del código 710202M porque las otras ya se encuentran optimizadas por el algoritmo.

Nuevamente se crean los individuos con las restricciones.

Individuo 1:

1062	1064	2000	2002	2004	3000
0	710202M	760155M	730069M	710205M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	0	0	0	5	0

Total, evaluación: $0+0+0+0+5+0=5$

Individuo 2:

1062	1064	2000	2002	2004	3000
710202M	0	760155M	730069M	710205M	0

Valores de evaluación:

1062	1064	2000	2002	2004	3000
100	0	0	0	5	0

Total, evaluación: $100+0+0+0+5+0=105$

Individuo 3:

1062	1064	2000	2002	2004	3000
0	0	760155M	730069M	710205M	710202M

Valores de evaluación:

1062	1064	2000	2002	2004	3000
0	0	0	0	5	25

Total, evaluación: $100+0+0+0+5+25=30$

Se genera el arreglo ordenado de los valores de los individuos con los mejores fitness

Individuo 1	Individuo 3	Individuo 2
5	30	105

El mejor individuo pasa hacer el genotipo del algoritmo por medio de la operación de selección

Genotipo:

1062	1064	2000	2002	2004	3000
0	710202M	760155M	730069M	710205M	0

Y por último se integran los genotipos en la distribución y en consecuencia queda organizado el fenotipo.

Horarios	Aulas					
	1062	1064	2000	2002	2004	3000
07:00			760155M	730069M	710203M	
08:00		710202M	760155M	730069M	710203M	
09:00		710202M	760155M	730069M	710203M	
10:00		710202M		730069M		710198M
11:00				714202M		710198M
12:00				714202M		710198M

Tabla 6 - Integración de los genotipos
Elaboración propia

6.4 IMPLEMENTACIÓN.

En esta sección de la investigación se describe el bosquejo del modelo propuesto de la asignación de aulas y horarios de las asignaturas de la Universidad del Valle sede Zarzal, encontramos los requerimientos funcionales para implementar el modelo, el lenguaje de programación en RUBY es el software seleccionado para realizar la compilación y ejecución del algoritmo evolutivo, este software cumple con los requisitos organizativos, compatibles, computacionales y funcionales para la implementación del modelo. Los casos de uso que serán implementados son:

- Gestionar el algoritmo evolutivo.
- Iniciar algoritmo evolutivo.
- Componente de Datos.
- Selección y evaluación.
- Parámetros de algoritmo evolutivo.
- Mostrar Asignación.
- Exportar Asignación.
- Mostrar resultados

En esta parte mostraremos las funciones claves para la codificación del algoritmo evolutivo.

6.4.1. CONFIGURACIÓN DE LOS PARÁMETROS DE EJECUCIÓN.

Al empezar la ejecución del prototipo, se pueden cambiar los parámetros de inicio del programa considerando los resultados que se deseen para el rendimiento de las pruebas y la complejidad del sistema de gestión de aulas.

<code>max_generaciones = 50</code>	<code>#Máximas Generaciones</code>
<code>poblacion = 100</code>	<code>#Tamaño de la población</code>
<code>num_gen = 30</code>	<code>#Número total de aulas</code>
<code>pcruce = 0.30</code>	<code>#Porcentaje de Cruce</code>

*Tabla 7 - Parámetros de configuración de los programas.
Elaboración del director*

6.4.2. FUNCIÓN DE CARGAR PROGRAMACIÓN ACADÉMICA.

La función de cargar archivo permite ingresar la información de la programación académica en el sistema para poder empezar la ejecución del algoritmo.

```
#Metodo que permite cargar el archivo de textos de la programación académica
#En este caso se puede cargar cualquier tipo de archivo (txt, html, json, php, etc)

def cargar_archivo_p()
  # archivo = File.new("programacionAcademica.txt","r")
  lineas=File.readlines("programacionAcademica.txt")
  archivo=""
  lineas.each do |id|
    archivo+= id.to_s
  end
  return archivo
end
```

*Tabla 8 - Cargar programación académica
Elaboración propia.*

6.4.3. FUNCIÓN DE CRUCE.

La función de cruce se comporta a nivel de cadenas de textos que corresponden a los códigos de las asignaturas, donde se establece un valor de cruce para la probabilidad de que se realice el intercambio de un gen entre dos padres, se seleccionan aleatoriamente dos puntos de cruce, uno en cada genotipo y luego se procede a intercambiar los segmentos de los valores que no se encuentren en la excepción horaria para generar un nuevo genotipo o individuo de la población.

```
def cruce(padre1, padre2, pcruce=0.30)
  return ""+padre1[:cadenabits] if rand()>=pcruce
  cut = rand([padre1[:cadenabits].size, padre2[:cadenabits].size].min)
  p2size = padre2[:cadenabits].size
  val=padre1[:cadenabits][0...cut]+padre2[:cadenabits][cut...p2size]
  return val
end
```

Tabla 9 - Función de cruce.
Elaboración propia.

6.4.4. FUNCIÓN DE SELECCIÓN PARA OPERACIÓN DE MEJOR TORNEO.

La función de mejor torneo evalúa los resultados generados por el fitness y determina las posiciones de los mismos, generando un orden de mayor a menor, para cada uno de los valores obtenidos para el arreglo de selección.

```
#Llamado a la función de torneo para seleccionar los mejores resultados de la
evolución de #funcionamiento

seleccion = Array.new(poblacion){|i| torneo_mejores(pop)}

-----
#Este método maneja la selección de los mejores valores evaluados y escoge los
mejores cromosomas

def torneo_mejores(pop)
  i, j = rand(pop.size), rand(pop.size)
  while j==i
    return (pop[i][:fitness] < pop[j][:fitness]) ? pop[i] : pop[j]
  end
end
```

Tabla 10 - Función de selección y mejor torneo.
Elaboración propia.

6.4.5. REQUERIMIENTOS HARDWARE Y SOFTWARE.

Modelo	Lenovo G40-80
Procesador	Intel ® Core TM i5-5200U CPU @ 2.20Hz
Memoria RAM	DDR3 8 GB
Sistema operativo	Windows 10 Profesional 64 bits
Softwares empleados	Microsoft Office Professional 2013
	AMPL Demo
	RUBY

Tabla 11 - Requisitos tecnológicos,
Elaboración propia

7. PRUEBA Y RESULTADOS

7.1. IMPLEMENTACIÓN DE MODELO MATEMÁTICO EN EL SOFTWARE AMPL.

Como se ha planteado en capítulos anteriores, se utilizara la herramienta de programación para plantear, aplicar y resolver de modelos de optimización, fundamentalmente modelos de Programación Lineal, Programación Entera y programación no lineal, el software de SOLVER AMPL demo IDE Versión: 3.6.1.201912161614, para encontrar un valor óptimo (mínimo o máximo), donde la función objetivo está sujeta a restricciones o limitaciones, basado en un modelo simplex, este está conformado por los siguientes ficheros:

- RUM: donde se programa los elementos de configuración y funcionamiento del prototipo.

Código

```
reset;
model salones.mod;      # carga él .mod
data salones.dat;       # carga los datos del .dat
option solver gurobi;    # llama la función solver gurobi
option show_stats 1;
option solution_precision 0; # precisión en el valor de la solución en 0 decimales
option omit_zero_cols 1;
option display_width 50;
option omit_zero_rows 1;
option gurobi_options $gurobi_options 'outlev 1'; #Informe del modelo

param cal ;
for {i in M}           # ciclo
{
    for {j in N}       # ciclo
    {
        let cal := k[i]-c[j];
        if cal >= 0 then # condicional
```

```

    {
        let sigma[i,j]:=cal;
    }
else {                                     # condicional
    let sigma[i,j]:=50;
}
}

solve;
display x;      # muestra los valores de la variable x
display y;      # muestra los valores de la variable y

printf "\n" > RESULTS.txt;               # imprime los resultados en el directorio raíz

for {i in M, (d,j,f) in Dia_Asigna_Franja}
{
    option omit_zero_cols 1;
    option omit_zero_rows 1;
    printf "%9d", i >> RESULTS.txt;
    printf " %s", j >> RESULTS.txt;
    printf " %s", d >> RESULTS.txt;
    printf " %9d", f >> RESULTS.txt;
    printf "%9d", x[i,d,j,f] >> RESULTS.txt;
    printf "\n" > RESULTS.txt;
}

```

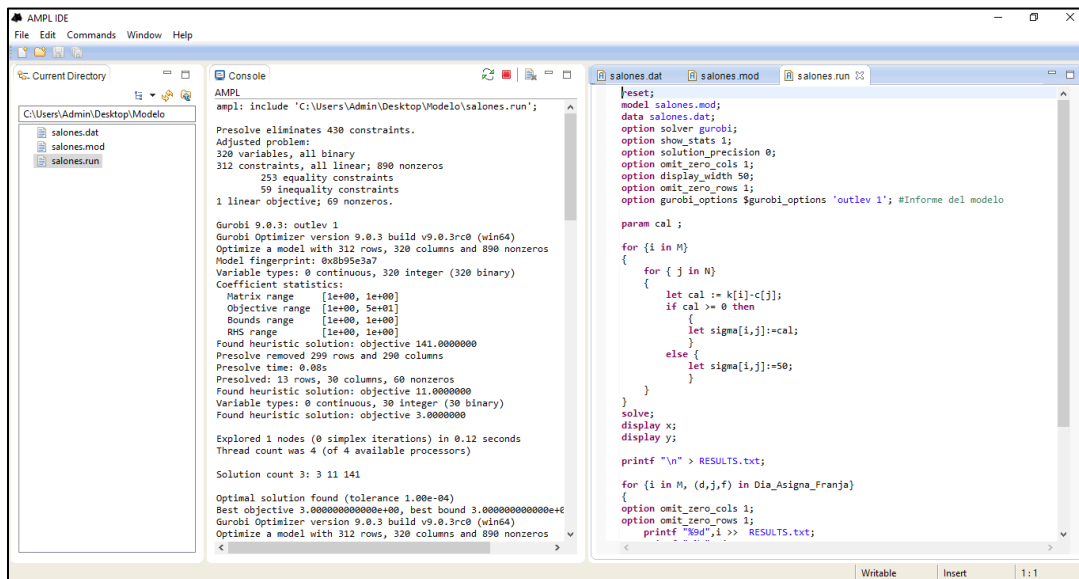


Imagen 12 - Modelo aplicado Software AMPL IDE Demo
Elaboración propia

- MOD: en esta parte se encuentran los conjuntos, parámetros, variables, función objetivo y restricciones.

Código

```

set M;      # Conjunto de aulas, indexado por el subíndice i
set N;      # Conjunto de asignaturas a programar, indexado por el subíndice j.
set F;      # Conjunto de franjas horarias, indexado por el subíndice f.

```

```

set D; # Conjunto de franjas horarias, indexado por el subíndice d.
set Dia_Asigna_Franja within {D,N,F}; # Conjunto inducido por el día d ,asignatura j y la franja f.

param k{i in M};
param c{j in N};
param sigma{i in M, j in N};

var x{i in M, (d,j,f) in Dia_Asigna_Franja} binary;
var y{i in M, j in N} binary;

minimize z: sum{i in M, j in N}(y[i,j]*sigma[i,j]);

subject to r1_1 {(d,j,f) in Dia_Asigna_Franja}: sum{i in M, d1 in D, f1 in F: d1=d and
f1=f}(x[i,d1,j,f1])=1; #Toda asignatura debe ser asignada solo una vez
subject to r1_2 {j in N}:sum{i in M} y[i,j]<=1; #Toda asignatura debe ser asignada solo una vez
subject to r2 {d1 in D, f1 in F, i in M}: sum{(d,j,f) in Dia_Asigna_Franja: d=d1 and
f=f1}(x[i,d,j,f])<=1; # En un día, en una franja y en un aula, no deben de existir más de una
asignatura
subject to r3 {i in M,(d,j,f) in Dia_Asigna_Franja}: sum{d1 in D, f1 in F:f=f1 and
d=d1}x[i,d,j,f]=y[i,j]; # Conexión de variables x - y

```

- DAT: donde encontramos la base de datos, donde esta consignada la información del MOD.

Código

```

param k:= # Capacidad del aula m
1062 55
1064 60
2000 55
2002 20
2004 25
3000 55
3006 30
3008 30
3010 55
3014 12;

param c:= # Número de alumnos de la asignatura n
745037M 25 # Elementos de proc. Industria de Alimentos
111050M 55 # Cálculo I
111051M 55 # Cálculo II
111048M 55 # Álgebra Lineal
720009M 55 # Mecánica y resistencia
760010M 30 # Gerencia de proyectos
760056M 10 # Trabajo de Grado I
760049M 30 # Salud Ocupacional
760037M 19; # Logística Industrial

# Programación de las aulas y sus franjas horarias

```

```

set Dia_Asigna_Franja:= (Lun,*,*): 07 08 09 10 11 12 13 14:=
    745037M + + + - - - - -
    111051M - - - + + - - -
    760037M - - - - - - + +
(Mar,*,*): 07 08 09 10 11 12 13 14:=
    111050M - + + + - - - -
    111048M - + + - - - - -
    720009M - - - - + + - -
(Mie,*,*): 07 08 09 10 11 12 13 14:=
    760010M - + + + - - - -
    760056M - - - + + + - -
    760049M - - + + + - - - ;

```

Imagen 13 - Matriz de programación de asignaturas
, Elaboración propia

7.2. OPERACIÓN DEL SISTEMA.

ampl: include 'C:\Users\Admin\Desktop\Modelo\salones.run'; # directorio de ejecución

Presolve eliminates 430 constraints. # Presolve elimina restricciones

Adjusted problem:

320 variables, all binary # cantidades de variables binarias procesadas

312 constraints, all linear; 890 nonzeros # restricciones, todas las líneas y no ceros

253 equality constraints # restricciones de igualdad

59 inequality constraints # restricciones de desigualdad

1 linear objective; 69 nonzeros. # Objetivo

Gurobi 9.0.3: outlev 1 # características del modelo y software

Gurobi Optimizer version 9.0.3 (win64)

Optimize a model with 312 rows, 320 columns and 890 nonzeros

Model fingerprint: 0x8b95e3a7

Variable types: 0 continuous, 320 integer (320 binary)

Coefficient statistics: # estadísticas de coeficientes

Matrix range [1e+00, 1e+00]

Objective range [1e+00, 5e+01]

Bounds range [1e+00, 1e+00]

RHS range [1e+00, 1e+00]

Found heuristic solution: objective 141.0000000 # soluciones heurísticas encontradas

Presolve removed 299 rows and 290 columns

Presolve time: 0.08s

Presolved: 13 rows, 30 columns, 60 nonzeros

Found heuristic solution: objective 11.0000000

Variable types: 0 continuous, 30 integer (30 binary)

Found heuristic solution: objective 3.0000000

Explored 1 nodes (0 simplex iterations) in 0.12 seconds

Thread count was 4 (of 4 available processors)

Solution count 3: 3 11 141

Optimal solution found (tolerance 1.00e-04) # **solución óptima encontrada**
 Best objective 3.000000000000e+00, best bound 3.000000000000e+00, gap 0.0000%
 Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (win64)
 Optimize a model with 312 rows, 320 columns and 890 nonzeros
 Model fingerprint: 0x62d3df57
 Coefficient statistics: # **estadísticas de coeficientes**
 Matrix range [1e+00, 1e+00]
 Objective range [1e+00, 5e+01]
 Bounds range [1e+00, 1e+00]
 RHS range [1e+00, 1e+00]
 Iteration Objective Primal Inf. Dual Inf. Time
 0 3.0000000e+00 0.000000e+00 0.000000e+00 0s
 Solved in 0 iterations and 0.02 seconds
 Optimal objective 3.000000000e+00
 Times (seconds): # **tiempo en segundos**
 Input = 0.015625
 Solve = 0.03125 (summed over threads)
 Output = 0.03125
 Elapsed < 1
 Gurobi 9.0.3: optimal solution; objective 3
 1 branch-and-cut nodes

7.3. RESULTADO.

Nota: 50 representa la penalización al no cumplir con las restricciones

:	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9:=
1062	0	0	0	0	30	25	36	25	45
1064	5	5	5	5	35	30	41	30	50
2000	0	0	0	0	30	25	36	25	45
2002	50	50	50	50	50	50	1	50	10
2004	50	50	50	50	0	50	6	50	15
3000	0	0	0	0	30	25	36	25	45
3006	50	50	50	50	5	0	11	0	20
3008	50	50	50	50	5	0	11	0	20
3010	0	0	0	0	30	25	36	25	45
3014	50	50	50	50	50	50	50	50	2

x	[:,Lun,*,7]								
:	745037M	:=							
2004	1								

	[:,Lun,*,8]								
:	745037M	:=							
2004	1								

	[:,Lun,*,9]								
:	745037M	:=							
2004	1								

	[:,Lun,*,10]								
:	111051M	:=							
2000	1								

	[:,Lun,*,11]								
:	111051M	:=							
2000	1								

	[:,Lun,*,13]								
:	760037M	:=							
2002	1								

```

    [* ,Lun ,*,14]
:      760037M      :=
2002      1

```

```

    [* ,Mar ,*,8]
:      111048M 111050M      :=
1062      1      0
2000      0      1

```

```

    [* ,Mar ,*,9]
:      111048M 111050M      :=
1062      1      0
2000      0      1

```

```

    [* ,Mar ,*,10]
:      111050M      :=
2000      1

```

```

    [* ,Mar ,*,11]
:      720009M      :=
2000      1

```

```

    [* ,Mar ,*,12]
:      720009M      :=
2000      1

```

```

    [* ,Mie ,*,8]
:      760010M      :=
3008      1

```

```

    [* ,Mie ,*,9]
:      760010M 760049M      :=
3006      0      1
3008      1      0

```

```

    [* ,Mie ,*,10]
:      760010M 760049M 760056M      :=
3006      0      1      0
3008      1      0      0
3014      0      0      1

```

```

    [* ,Mie ,*,11]
:      760049M 760056M      :=
3006      1      0
3014      0      1

```

```

    [* ,Mie ,*,12]
:      760056M      :=
3014      1
;

```

```

y :=
1062 111048M 1
2000 111050M 1
2000 111051M 1
2000 720009M 1
2002 760037M 1
2004 745037M 1
3006 760049M 1
3008 760010M 1
3014 760056M 1
;

```

Aula	Asignatura	Día	Franja	Variable binaria	Aula	Asignatura	Día	Franja	Variable binaria
1062	745037M	Lun	7	0	1064	760056M	Mie	10	0
1062	745037M	Lun	8	0	1064	760056M	Mie	11	0
1062	745037M	Lun	9	0	1064	760056M	Mie	12	0
1062	111051M	Lun	10	0	1064	760049M	Mie	9	0
1062	111051M	Lun	11	0	1064	760049M	Mie	10	0
1062	760037M	Lun	13	0	1064	760049M	Mie	11	0
1062	760037M	Lun	14	0	2000	745037M	Lun	7	0
1062	111050M	Mar	8	0	2000	745037M	Lun	8	0
1062	111050M	Mar	9	0	2000	745037M	Lun	9	0
1062	111050M	Mar	10	0	2000	111051M	Lun	10	1
1062	111048M	Mar	8	1	2000	111051M	Lun	11	1
1062	111048M	Mar	9	1	2000	760037M	Lun	13	0
1062	720009M	Mar	11	0	2000	760037M	Lun	14	0
1062	720009M	Mar	12	0	2000	111050M	Mar	8	1
1062	760010M	Mie	8	0	2000	111050M	Mar	9	1
1062	760010M	Mie	9	0	2000	111050M	Mar	10	1
1062	760010M	Mie	10	0	2000	111048M	Mar	8	0
1062	760056M	Mie	10	0	2000	111048M	Mar	9	0
1062	760056M	Mie	11	0	2000	720009M	Mar	11	1
1062	760056M	Mie	12	0	2000	720009M	Mar	12	1
1062	760049M	Mie	9	0	2000	760010M	Mie	8	0
1062	760049M	Mie	10	0	2000	760010M	Mie	9	0
1062	760049M	Mie	11	0	2000	760010M	Mie	10	0
1064	745037M	Lun	7	0	2000	760056M	Mie	10	0
1064	745037M	Lun	8	0	2000	760056M	Mie	11	0
1064	745037M	Lun	9	0	2000	760056M	Mie	12	0
1064	111051M	Lun	10	0	2000	760049M	Mie	9	0
1064	111051M	Lun	11	0	2000	760049M	Mie	10	0
1064	760037M	Lun	13	0	2000	760049M	Mie	11	0
1064	760037M	Lun	14	0	2002	745037M	Lun	7	0
1064	111050M	Mar	8	0	2002	745037M	Lun	8	0
1064	111050M	Mar	9	0	2002	745037M	Lun	9	0
1064	111050M	Mar	10	0	2002	111051M	Lun	10	0
1064	111048M	Mar	8	0	2002	111051M	Lun	11	0
1064	111048M	Mar	9	0	2002	760037M	Lun	13	1
1064	720009M	Mar	11	0	2002	760037M	Lun	14	1
1064	720009M	Mar	12	0	2002	111050M	Mar	8	0
1064	760010M	Mie	8	0	2002	111050M	Mar	9	0
1064	760010M	Mie	9	0	2002	111050M	Mar	10	0
1064	760010M	Mie	10	0	2002	111048M	Mar	8	0

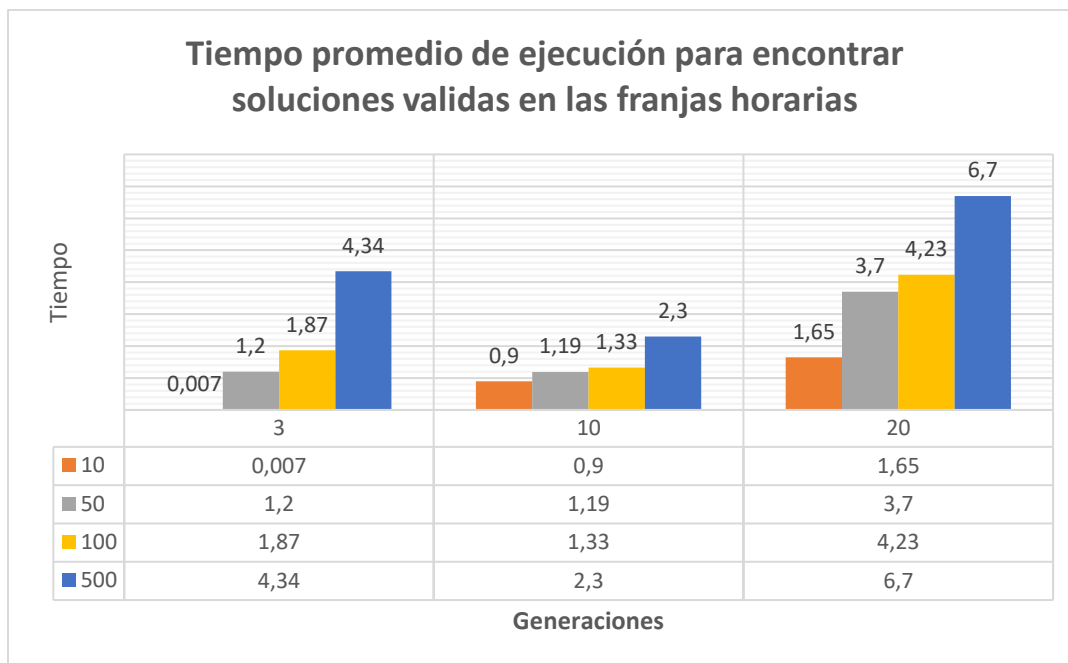
Tabla 12 - Resultados de la implementación del modelo, Elaboración propia

Aula	Asignatura	Día	Franja	Variable binaria	Aula	Asignatura	Día	Franja	Variable binaria
2002	111048M	Mar	9	0	3000	760037M	Lun	13	0
2002	720009M	Mar	11	0	3000	760037M	Lun	14	0
2002	720009M	Mar	12	0	3000	111050M	Mar	8	0
2002	760010M	Mie	8	0	3000	111050M	Mar	9	0
2002	760010M	Mie	9	0	3000	111050M	Mar	10	0
2002	760010M	Mie	10	0	3000	111048M	Mar	8	0
2002	760056M	Mie	10	0	3000	111048M	Mar	9	0
2002	760056M	Mie	11	0	3000	720009M	Mar	11	0
2002	760056M	Mie	12	0	3000	720009M	Mar	12	0
2002	760049M	Mie	9	0	3000	760010M	Mie	8	0
2002	760049M	Mie	10	0	3000	760010M	Mie	9	0
2002	760049M	Mie	11	0	3000	760010M	Mie	10	0
2004	745037M	Lun	7	1	3000	760056M	Mie	10	0
2004	745037M	Lun	8	1	3000	760056M	Mie	11	0
2004	745037M	Lun	9	1	3000	760056M	Mie	12	0
2004	111051M	Lun	10	0	3000	760049M	Mie	9	0
2004	111051M	Lun	11	0	3000	760049M	Mie	10	0
2004	760037M	Lun	13	0	3000	760049M	Mie	11	0
2004	760037M	Lun	14	0	3006	745037M	Lun	7	0
2004	111050M	Mar	8	0	3006	745037M	Lun	8	0
2004	111050M	Mar	9	0	3006	745037M	Lun	9	0
2004	111050M	Mar	10	0	3006	111051M	Lun	10	0
2004	111048M	Mar	8	0	3006	111051M	Lun	11	0
2004	111048M	Mar	9	0	3006	760037M	Lun	13	0
2004	720009M	Mar	11	0	3006	760037M	Lun	14	0
2004	720009M	Mar	12	0	3006	111050M	Mar	8	0
2004	760010M	Mie	8	0	3006	111050M	Mar	9	0
2004	760010M	Mie	9	0	3006	111050M	Mar	10	0
2004	760010M	Mie	10	0	3006	111048M	Mar	8	0
2004	760056M	Mie	10	0	3006	111048M	Mar	9	0
2004	760056M	Mie	11	0	3006	720009M	Mar	11	0
2004	760056M	Mie	12	0	3006	720009M	Mar	12	0
2004	760049M	Mie	9	0	3006	760010M	Mie	8	0
2004	760049M	Mie	10	0	3006	760010M	Mie	9	0
2004	760049M	Mie	11	0	3006	760010M	Mie	10	0
3000	745037M	Lun	7	0	3006	760056M	Mie	10	0
3000	745037M	Lun	8	0	3006	760056M	Mie	11	0
3000	745037M	Lun	9	0	3006	760056M	Mie	12	0
3000	111051M	Lun	10	0	3006	760049M	Mie	9	1
3000	111051M	Lun	11	0	3006	760049M	Mie	10	1

Tabla 13 - Resultados de la implementación del modelo, Elaboración propia

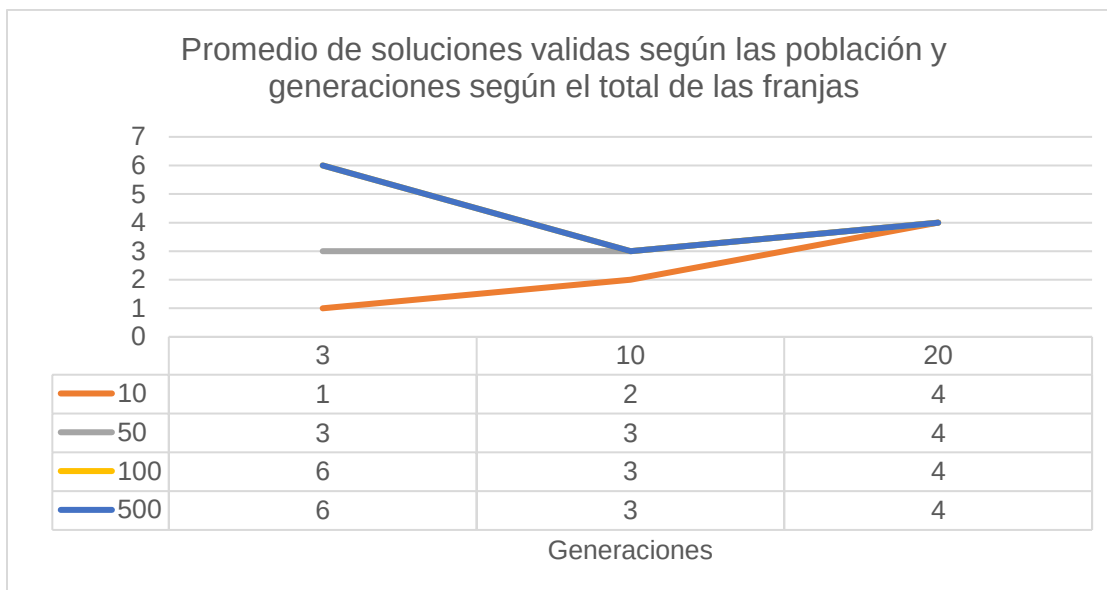
Aula	Asignatura	Día	Franja	Variable binaria	Aula	Asignatura	Día	Franja	Variable binaria
3006	760049M	Mie	11	1	3010	760010M	Mie	10	0
3008	745037M	Lun	7	0	3010	760056M	Mie	10	0
3008	745037M	Lun	8	0	3010	760056M	Mie	11	0
3008	745037M	Lun	9	0	3010	760056M	Mie	12	0
3008	111051M	Lun	10	0	3010	760049M	Mie	9	0
3008	111051M	Lun	11	0	3010	760049M	Mie	10	0
3008	760037M	Lun	13	0	3010	760049M	Mie	11	0
3008	760037M	Lun	14	0	3014	745037M	Lun	7	0
3008	111050M	Mar	8	0	3014	745037M	Lun	8	0
3008	111050M	Mar	9	0	3014	745037M	Lun	9	0
3008	111050M	Mar	10	0	3014	111051M	Lun	10	0
3008	111048M	Mar	8	0	3014	111051M	Lun	11	0
3008	111048M	Mar	9	0	3014	760037M	Lun	13	0
3008	720009M	Mar	11	0	3014	760037M	Lun	14	0
3008	720009M	Mar	12	0	3014	111050M	Mar	8	0
3008	760010M	Mie	8	1	3014	111050M	Mar	9	0
3008	760010M	Mie	9	1	3014	111050M	Mar	10	0
3008	760010M	Mie	10	1	3014	111048M	Mar	8	0
3008	760056M	Mie	10	0	3014	111048M	Mar	9	0
3008	760056M	Mie	11	0	3014	720009M	Mar	11	0
3008	760056M	Mie	12	0	3014	720009M	Mar	12	0
3008	760049M	Mie	9	0	3014	760010M	Mie	8	0
3008	760049M	Mie	10	0	3014	760010M	Mie	9	0
3008	760049M	Mie	11	0	3014	760010M	Mie	10	0
3010	745037M	Lun	7	0	3014	760056M	Mie	10	1
3010	745037M	Lun	8	0	3014	760056M	Mie	11	1
3010	745037M	Lun	9	0	3014	760056M	Mie	12	1
3010	111051M	Lun	10	0	3014	760049M	Mie	9	0
3010	111051M	Lun	11	0	3014	760049M	Mie	10	0
3010	760037M	Lun	13	0	3014	760049M	Mie	11	0
3010	760037M	Lun	14	0					
3010	111050M	Mar	8	0					
3010	111050M	Mar	9	0					
3010	111050M	Mar	10	0					
3010	111048M	Mar	8	0					
3010	111048M	Mar	9	0					
3010	720009M	Mar	11	0					
3010	720009M	Mar	12	0					
3010	760010M	Mie	8	0					
3010	760010M	Mie	9	0					

Tabla 14 - Resultados de la implementación del modelo, Elaboración propia



*Imagen 14 - Tiempo promedio de ejecución
Elaboración propia*

Esto es para un cromosoma con tamaño de 9 que es igual al número de aulas y para un total de 9 asignaturas, en la **imagen 14** se muestra el promedio total de ejecución para encontrar soluciones validas en las franjas horarias para construir las expresiones candidatas, la efectividad del número de generaciones según la efectividad como se muestra en esta imagen, para cada una de las generaciones se realizaron las pruebas dentro de las cuales dio como promedio resultados.

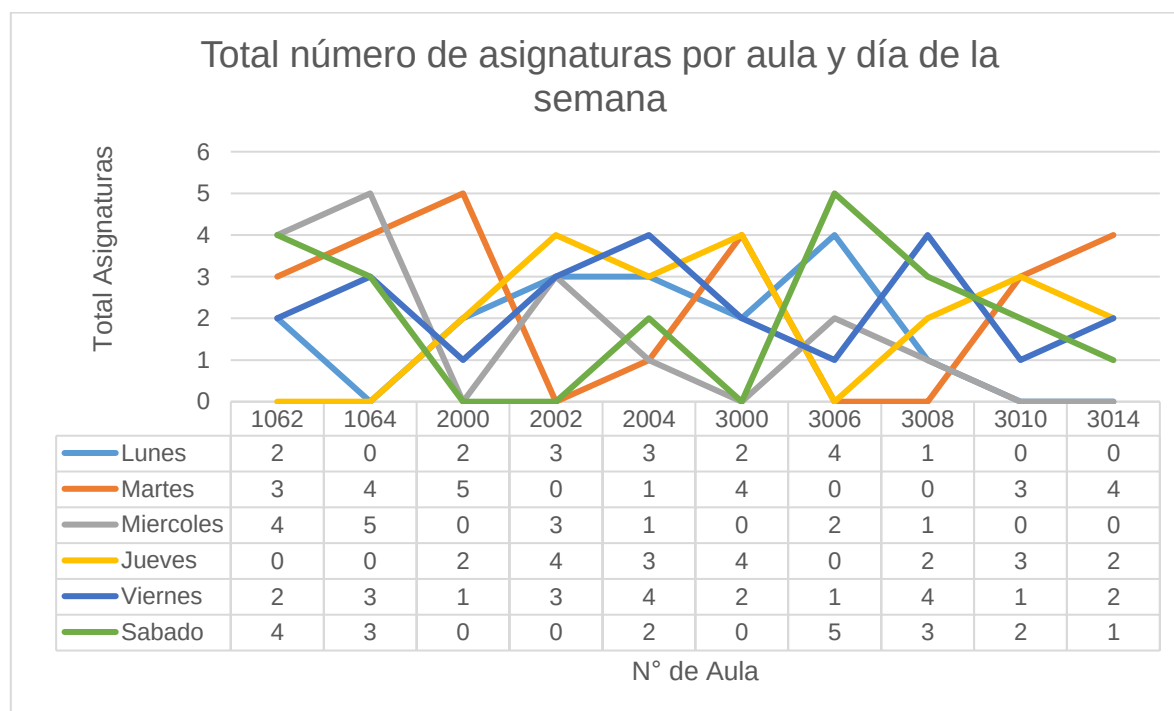


*Imagen 15 - Promedio de soluciones validas según las población y generaciones
Elaboración propia*

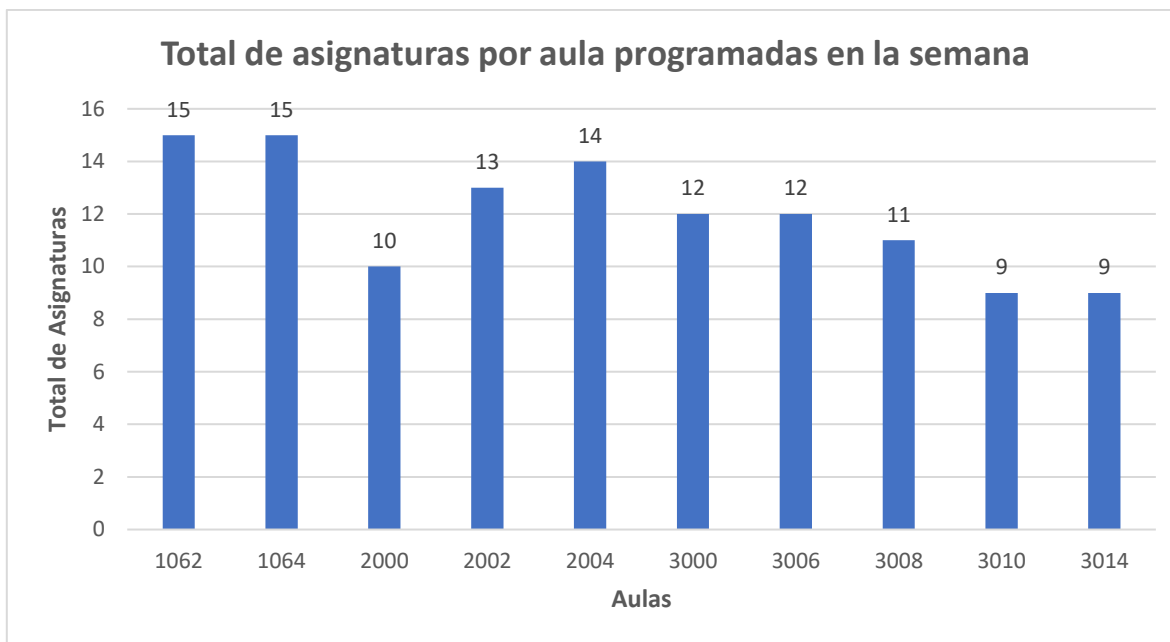
En este caso se tomó en cuenta el promedio de las soluciones validas según población y las generaciones según la cantidad total de las franjas la expresión regular, el análisis no ubica en una convergencia final de las pruebas realizadas como se muestra en la imagen 15, donde a pesar de que se aumenten tanto la población como el número de generaciones el prototipo mantiene su eficiencia y un tiempo corto de operación para encontrar la solución óptima.

En este prototipo el rendimiento en tiempo de ejecución en cada una de las soluciones validas es un elemento fundamental, para este tipo de modelos timetabling.

Para evaluar el rendimiento del prototipo se realizó una prueba con un total de 85 asignaturas en 10 aulas, en la **imagen 16** podemos ver el comportamiento del prototipo en el número de asignaturas programadas por aula y día de la semana.

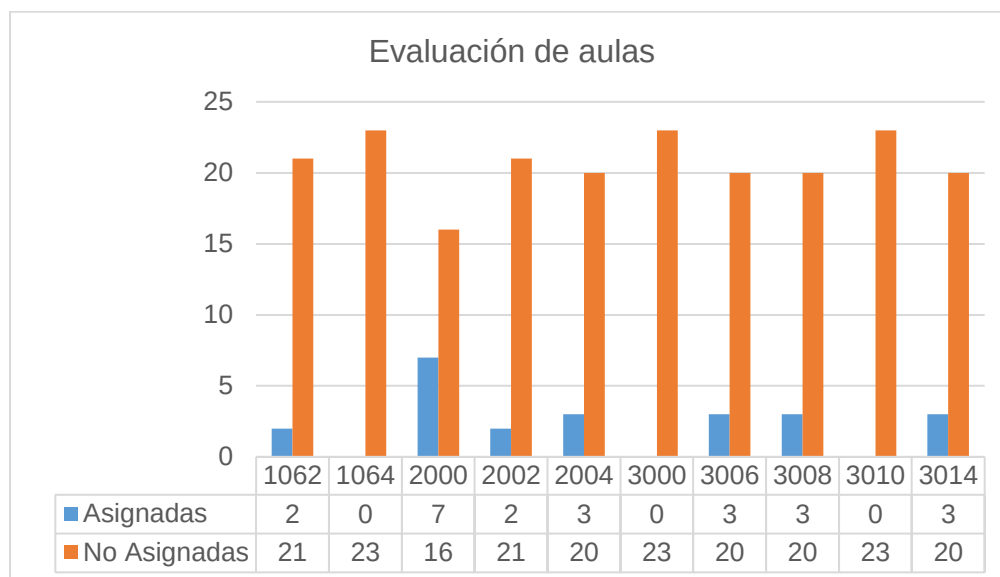


*Imagen 16 - Total número de asignaturas por aula y día de la semana
Elaboración propia*



*Imagen 17 - Total de asignaturas por aula programadas en la semana
Elaboración propia*

Al evaluar el prototipo en AMPL las asignaturas fueron evaluadas en las aulas respetando las restricciones, cada aula fue evaluada 23 veces, este valor se obtiene de análisis de los resultados, podemos observar en la **imagen 18** la cantidad de franjas asignas a cada aula en el modelo.



*Imagen 18 - uso de aulas en según resultado de AMPL
Elaboración propia*

8. CONCLUSIONES Y TRABAJOS FUTUROS

8.1. CONCLUSIONES.

En esta investigación va dirigida a Proponer un modelo de asignación de horarios y aulas de clases en la Universidad del Valle Sede Zarzal, con los objetivos que contribuya al proceso de toma de decisiones en la institución, resolver un problema utilizando modelos lógico matemáticos y metaheurísticos, que permite resolver problemas complejos.

Analizar el proceso, Formular un modelo matemático, Implementar un prototipo que incorpore los modelos, diseños y técnicas de asignación aulas y horarios de clases de la Universidad del Valle Sede Zarzal y evaluar el modelo, es necesario para implementar el algoritmo evolutivo, donde se expresan las restricciones para el modelo, así como las variables de decisión, criterio de optimización, conjuntos y subconjuntos, donde empleo el software AMPL. En el método metaheurístico se plasmó la representación del problema, así también se describe la función objetivo y de ahí la puesta en marcha del prototipo que entrega la solución, en este punto donde se decidió utilizar el software RUBY con base de datos, la primera herramienta permite evaluar el modelo lógico matemático y el segundo permite gestionar el manejo de la información.

La solución encontrada, se realizó partiendo del proceso de programación académica organizada por los coordinadores de programa o profesores de apoyo donde se encuentra las asignaturas que se van a programar como la hora de inicio, la duración, el docente, el grupo y la cantidad de estudiantes, que ingresan al prototipo, obteniendo resultados coherentes con las expectativas generada por la investigación, la hipótesis fue evaluada encontrando una solución óptima para el programa. Si se analizan los resultados obtenidos, estos acertados, podemos exponer la viabilidad de uso de los algoritmos evolutivos para solucionar un problema de tipo timetabling, las buenas soluciones óptimas dependen en gran medida del algoritmo evolutivo y las herramientas implementadas.

La problemática que presenta la asignación de horarios y aulas, también denominados problema de tipo timetabling, aplicado en variados modelos de asignación en institución de educación, es complejo de resolver, debido a la cantidad de variables y restricciones presentes que varían dependiendo las características que acompañan a la institución, si como el proceso y las actividades inherentes, no obstante, esto no es impedimento para ser motivo de su investigación, al contrario aumenta el por afrontar el reto de entrar en materia sobre este tema y encontrar soluciones al problema. El objetivo de la investigación es llegar a una solución de optimización metaheurística al problema de asignación de horarios y aulas de clases para de la Universidad del Valle sede Zarzal, a través de los algoritmos evolutivos.

Los elementos para la asignación de aulas aplicado en este trabajo de grado, el modelo lógico matemático se basa en la programación lineal en primera parte de la investigación, que da partida al algoritmo evolutivo para encontrar la solución en un problema más complejo de asignación, el prototipo se ha implementado en un ambiente de pruebas y utilizando datos reales suministrados por el área de secretaria académicos de la universitaria del Valle sede Zarzal. Este prototipo ha exhibido un mejor desempeño confrontado con el método manual que es implementado actualmente en la universidad, la optimización de los recursos académicos, la disminución de la presencia de los cruces en la programación de las asignaturas y las aulas, garantizando que el número de estudiantes que tiene cada grupo se acorde con la capacidad en sillas para los estudiantes

De la solución encontrada, el modelo y el algoritmo aplicados obtuvimos resultados coherentes. No obstante se programaron pruebas del prototipo y resolvió el problema específicamente para el programa de las asignaturas de la Universidad del Valle sede Zarzal, tanto este tipo de modelos lógico matemáticos, como los algoritmos evolutivos, permiten adaptarse a este tipo de problemas de asignación de horarios y aulas de clases en instituciones de educación, facilitando una flexibilidad en los resolución, La importancia de la administración de la información adecuada del proceso y cada una de sus actividades, permite realizar análisis de exhaustivo con respecto a los actores principales (asignaturas, aulas y franjas horarias).

8.2. TRABAJOS FUTUROS.

Se plantea como plan de mejora la de implementar un aplicativo Web con este algoritmo evolutivo, que les permita a los actores de este proceso interactuar con la solución y beneficiarse de sus resultados, que se pueda tener acceso a él, de esta manera que sea usado por otras sedes de la universidad del Valle. Otra ventaja de implementar el aplicativo sería que se estaría destinando la carga del procesamiento a un servidor para liberar al usuario de realizar este proceso de forma manual.

Al implementar procesamiento del producto final de la investigación en un servidor, podrá ser establecido sobre una arquitectura de procesamiento distribuida. En este punto se presentan varias opciones, ya que proporciona una cantidad de recursos según se necesite, claro que este modelo sería independiente al algoritmo y tendría que realizarse un trabajo previo para implementar esta arquitectura. Otra opción unida a la implementación de la solución sería elaborar un modelo de procesamiento servidor cliente, donde un servidor procese el algoritmo evolutivo, concentre la información ingresada y resultante. Dicha carga de trabajo podría ser la evaluación por parte del modelo aplicado sobre la población, dado que esto es lo que más tiempo toma en procesar.

9. REFERENCIAS

- [1] A. Cortez Vásquez, G. Rosales Gerónimo, R. Naupari Quiroz, y H. Vega Huerta, «Sistema de apoyo a la generación de horarios basado en algoritmos genéticos», *Rev. Investig. Sist. e Informática*, vol. 7, n.º 1, pp. 37-55, 2010, [En línea]. Disponible en: <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3264/2729>.
- [2] J. Mejía y C. Paternina, «Asignación de horarios de clases universitarias mediante algoritmos evolutivos», *Rev. Educ. en Ing.*, vol. 9, pp. 140-149, 2010.
- [3] L. L. Esquivel T, «Modelo Matemático Para La Programación De Un Horario Escolar Con Multi-Localización De Docentes», *Univ. del Val.*, p. 95, 2014, doi: 10.1017/CBO9781107415324.004.
- [4] M. Granada, E. M. Toro, y J. F. Franco, «Programación óptima de horarios de clase usando un algoritmo memético», *Sci. Tech.*, vol. XII, n.º 30, pp. 255-260, 2006, [En línea]. Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/6531>.
- [5] R. Hernández, J. Miranda P, y P. A. Rey, «Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera», *Control*, pp. 121-141, 2008.
- [6] J. F. Franco y B. Tabú, «Problema de asignación óptima de salones resuelto con Búsqueda Tabú», *Ing. y Desarrollo*, n.º 24, pp. 149-175, 2008.
- [7] J. C. Marín Ángel y P. A. Maya Duque, «Modelo lineal para la programación de clases en una institución educativa», *Ing. y Cienc.*, vol. 12, n.º 23, pp. 47-71, 2016, doi: 10.17230/ingciencia.12.23.3.
- [8] C. L. Pacheco Agüero, «Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos», p. 89, 2000.
- [9] dan A. F. M. Dhamar Yudho Aji onesia, «IMPLANTACIÓN DEL MÓDULO DE ASIGNACIÓN DE HORARIOS PARA LOS DOCENTES EN LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN BUENAVENTURA SEDE BOGOTÁ.», *Cell*, vol. 151, n.º 4, pp. 1-46, 2008, doi: 10.1016/j.cell.2009.01.043.
- [10] D. P. C. C. L. E. Y. N. to K. in 20 Weeks, «Diseño de una herramienta de software que asigne eficientemente las aulas para todos los cursos de pregrado de la Universidad Tecnológica de Bolívar usando un algoritmo genético.», *Dk*, vol. 53, n.º 9, pp. 1689-1699, 2015, doi: 10.1017/CBO9781107415324.004.
- [11] E. A. Duque Grisales, J. D. Gonzalez Ruíz, y J. D. Ramírez Aristizabal,

«Diseño de un modelo de programación lineal para la asignación de aulas en la Institución Universitaria Esumer, Colombia», *Espacios*, vol. 38, n.º 10, p. 1, 2017.

- [12] P. L. Pettis, «Belgrano», 2003.
- [13] A. María, I. Solano, A. Bezaleel, I. Carbonell, U. Simón, y B. Barranquilla - Atlántico, «La Computación Evolutiva y sus Paradigmas Paradigms of Evolutionary Computing», *Investig. y Desarro. en TIC*, vol. 2, pp. 29-38, 2011.
- [14] L. V. Santana Quintero y C. A. Coello, «Una introducción a la computación evolutiva y algunas de sus aplicaciones en economía y finanzas», *Rev. Metod. Cuantitativos para la Econ. y la Empres.*, vol. 2, n.º 1, pp. 3-26, 2006.
- [15] E. Castillo, A. J. Conejo, P. Pedregal, R. García, y N. Alguacil, «Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia», *España Univ. Castilla-La Mancha*, 2002.
- [16] B. Maegraith, *Tropical diseases. 2. Malaria. (II). Management and treatment.*, vol. 64, n.º 12. 1968.
- [17] La Computación Evolutiva [Heitkötter, Beasley, January 2000].
- [18] David B. Fogel, *Computación evolutiva. Toward a New Philosophy of Machine Intelligence*, The Institute of Electrical and Electronic Engineers, Nueva York, 1995.
- [19] Fogel, DB (1995). *Computación evolutiva: hacia una nueva filosofía de la inteligencia artificial* (3. ed.).
- [20] Noraini, M. R., & Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving TSP.
- [21] Keller, R.E. y Banzhaf, W. (1996). ((Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes.)) En: Koza.J.R.; D.E. Goldberg; D.B. Fogel y Riolo R.L. (Eds.), *Genetic Programming 1996: Proceedings Of The First Annual Conference*, pp. 116{122. MIT Press, Stanford University, CA, USA.
- [22] O'Sullivan, J. y Ryan, C. (2002). ((An Investigation into the Use of Different Search Strategies with Grammatical Evolution)). En: *Genetic Programming*, volumen 2278 de *Lecture Notes in Computer Science*, pp. 268{277. Springer, Berlin / Heidelberg.

10. ANEXOS

Anexo 1 - Campus de la Universidad del Valle Sede Zarzal.

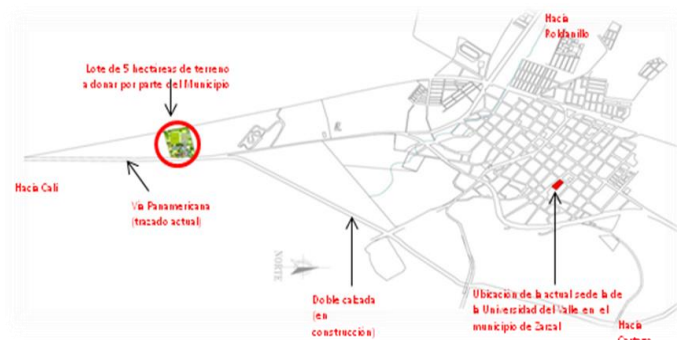


Imagen 19 - Campus de la Universidad sede Zarzal
Campus Bolívar y Campus Las Balsas; Fuente: Universidad del Valle – Sede Zarzal, 2015.

Anexo 2 - Descripción Detallada De Los Campus.

Campus	Bolívar		Las Balsas	
Imagen				
Fuente	https://www.facebook.com/ZarzalUnivalle/photos/a.1807951292773852/2748400812062224/ Universidad del Valle (Sede Bolívar) - Zarzal Valle del Cauca.		https://www.facebook.com/RumbosUnivalle Vista aérea del campus Las Balsas Universidad del Valle sede Zarzal. Fotografía: Servidrone.	
Área	2.283 M ² aproximados		5 hectáreas o 50.000 M ² aproximados	
Edificios	1 académico Administrativo de 1.916 M ² 1 biblioteca de 367 M ²		1 académico administrativo de 2.204,32 M ² 1 cafetería de 410,39 M ² 1 área deportiva de 2.682,09 M ² 1 parqueaderos, andenes y portería de 5.589,54M ²	
Dirección	Cl. 14 #7-134, Zarzal, Valle del Cauca		La vía Zarzal - La Paila, Zarzal, Valle del Cauca.	
Espacios	12 aulas de clases 1 auditorio 4 salas de computo 2 laboratorios 1 sala de juntas	16 oficinas 1 sala de docentes 1 cafetería 1 cocineta 9 baterías sanitarias	13 aulas de clases 4 salas de computo 1 sala de Juntas 8 oficinas 1 sala de docentes	1 piscina 1 cafetería 2 cocineta 2 canchas múltiples 14 baterías sanitarias

Tabla 15 - Descripción de los campus de la Universidad del Valle Sede Zarzal
Fuente DIU-División de Infraestructura Universitaria, Técnico de Infraestructura de la Sede Zarzal

Anexo 3 - Calendario Académico.

El periodo académico es comprendido por 2 semestres en el año, los cuales cada uno es constituido por 18 semanas académicas, con una franja horaria de 7 am a 10 pm de lunes a viernes y los sábados de 7 am a 8 pm.

Franja Horaria		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Inicio	Fin						
7:00 a. m.	8:00 a. m.	1	16	31	46	61	76
8:00 a. m.	9:00 a. m.	2	17	32	47	62	77
9:00 a. m.	10:00 a. m.	3	18	33	48	63	78
10:00 a. m.	11:00 a. m.	4	19	34	49	64	79
11:00 a. m.	12:00 m.	5	20	35	50	65	80
12:00 m.	1:00 p. m.	6	21	36	51	66	81
1:00 p. m.	2:00 p. m.	7	22	37	52	67	82
2:00 p. m.	3:00 p. m.	8	23	38	53	68	83
3:00 p. m.	4:00 p. m.	9	24	39	54	69	84
4:00 p. m.	5:00 p. m.	10	25	40	55	70	85
5:00 p. m.	6:00 p. m.	11	26	41	56	71	86
6:00 p. m.	7:00 p. m.	12	27	42	57	72	87
7:00 p. m.	8:00 p. m.	13	28	43	58	73	88
8:00 p. m.	9:00 p. m.	14	29	44	59	74	
9:00 p. m.	10:00 p. m.	15	30	45	60	75	

	Jornada Diurna
	Jornada Nocturna

Tabla 16 - Franjas de horarias y jornadas
Elaboración Propia

Anexo 4 - Aulas De Clases Sede Bolivar.

No.	Sede	Edificio	Espacio	Tipo de ambiente de aprendizaje	Objetivo del ambiente de aprendizaje	Área en Mts2	Dotado	Sillas	Movilidad reducida
1	Bolívar	510	1062	SALON 140	Salón de Clase	58,7	Si	55	Si
2	Bolívar	510	1064	SALON 145	Salón de Clase	59,1	Si	60	Si
3	Bolívar	510	2000	SALON 200	Salón de Clase	58,7	Si	50	No
4	Bolívar	510	2002	SALON 205	Salón de Clase	35,7	Si	30	No
5	Bolívar	510	2004	SALON 210	Salón de Clase	44	Si	35	No
6	Bolívar	510	2006	SALA DE SISTEMAS 1 (215)	Sala de Computó	70,4	Si	46	No
7	Bolívar	510	2008	SALA DE SISTEMAS 2 (220)	Sala de Computó	41,2	Si	25	No
8	Bolívar	510	2010	SALA DE SISTEMAS 3 - IDIOMAS (225)	Sala de Computó	59,9	Si	31	No
9	Bolívar	510	3000	SALON 300	Salón de Clase	58,7	Si	55	No
10	Bolívar	510	3002	LABORATORIO DE ALIMENTOS – 305	Laboratorio	59,1	Si	15	No
11	Bolívar	510	3004	LABORATORIO DE ELECTRÓNICA - 310	Laboratorio	49,8	Si	15	No
12	Bolívar	510	3006	SALON 315	Salón de Clase	44,5	Si	37	No
13	Bolívar	510	3008	SALON 320	Salón de Clase	36,9	Si	40	No
14	Bolívar	510	3010	SALON 325	Salón de Clase	59,6	Si	57	No
15	Bolívar	510	3014	SALON 335	Salón de Clase	16,4	Si	12	No
16	Bolívar	510		SALON 1 (SÓTANO BIBLIOTECA)	Salón de Clase	56,4	Si	41	Si
17	Bolívar	510		SALON 2 (SÓTANO BIBLIOTECA)	Salón de Clase	80,1	Si	60	Si
18	Bolívar	510		SALA DE SISTEMAS 4 – BIBLIOTECA	Sala de Computó	32,6	Si	10	No
19	Bolívar	510	1054	AUDITORIO	Salones de Práctica	243	Si	280	Parcial

Tabla 17 - Pág. 1 Aulas de Clase
Elaboración Propia

Anexo 5 - Aulas de Clases Sede Balsas.

No.	Sede	Edificio	Espacio	Tipo de ambiente de aprendizaje	Objetivo del ambiente de aprendizaje	Área en Mts2	Dotado	Sillas	Movilidad reducida
20	Balsas	510	102	SALÓN AUDITORIO 101	Salón de Clase	64,8	Si	36	Si
21	Balsas	510	103	SALÓN AUDITORIO 102	Salón de Clase	64,8	Si	36	Si
22	Balsas	510	104	SALÓN AUDITORIO 103	Salón de Clase	64,8	Si	36	Si
23	Balsas	510	105	SALÓN AUDITORIO 104	Salón de Clase	64,8	Si	36	Si
24	Balsas	510	106	SALÓN AUDITORIO 105	Salón de Clase	64,8	Si	36	Si
25	Balsas	510	202	SALON 201	Salón de Clase	64,8	Si	36	Si
26	Balsas	510	203	SALON 202	Salón de Clase	64,8	Si	36	Si
27	Balsas	510	204	SALON 203	Salón de Clase	64,8	Si	36	Si
28	Balsas	510	205	SALON 204	Salón de Clase	64,8	Si	36	Si
29	Balsas	510	206	SALON 205	Salón de Clase	64,8	Si	36	Si
30	Balsas	510	213	SALA DE SISTEMAS 1	Sala de Computó	90,72	Si	40	Si
31	Balsas	510	214	SALA DE SISTEMAS 2	Sala de Computó	90,72	Si	40	Si
32	Balsas	510	302	SALON 301 – 302	Salón de Clase	129,6	Si	72	Si
33	Balsas	510	304	SALON 303	Salón de Clase	64,8	Si	36	Si
34	Balsas	510	306	SALON 304 – 305	Salón de Clase	129,6	Si	54	Si
35	Balsas	510	313	SALA DE SISTEMAS 3	Sala de Computó	90,72	Si	40	Si
36	Balsas	510	314	SALA DE SISTEMAS 4	Sala de Computó	90,72	Si	40	Si

Tabla 18 - Pág. 2 Aulas de Clase
Elaboración Propia