	<b>UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA</b>			
	Documento	Código	Fecha	Revisión
FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	10-04-2012	A	
Dependencia	Aprobado	Pág.		
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO	i(116)		

## RESUMEN – TRABAJO DE GRADO

AUTORES	<b>ANDRES MAURICIO VERGEL MIRANDA</b>		
FACULTAD	<b>INGENIERÍAS</b>		
PLAN DE ESTUDIOS	<b>INGENIERÍA DE SISTEMAS</b>		
DIRECTOR	<b>ANDRES MAURICIO PUENTES VELASQUEZ</b>		
TÍTULO DE LA TESIS	<b>ALGORITMO PARA LA ASIGNACIÓN DE HORARIOS ACADÉMICOS EN LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL</b>		
<b>RESUMEN</b> (70 palabras aproximadamente)			
<p style="text-align: center;"><b>LA INVESTIGACIÓN DE ALGORITMOS PARA LA ASIGNACIÓN DE HORARIOS ACADÉMICOS EN LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL, SURGE COMO UNA PROPUESTA TECNOLOGIA QUE AYUDE A LOS DIRECTORES DE PLAN DE ESTUDIO DE LA FACULTAD DE INGENIERIAS EN LA OPTIMIZACIÓN DEL PROCESO DE ASIGNACIÓN DE HORARIOS.</b></p>			
<b>CARACTERÍSTICAS</b>			
PÁGINAS: 88	PLANOS:	ILUSTRACIONES:	CD-ROM: 1



VÍA ACOLSURE, SEDE EL ALGODONAL. OCAÑA N. DE S.  
Línea Gratuita Nacional 018000 121022 / PBX: 097-5690088  
[www.ufpso.edu.co](http://www.ufpso.edu.co)



ALGORITMO PARA LA ASIGNACIÓN DE HORARIOS ACADÉMICOS EN LA  
UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA UTILIZANDO  
TÉCNICAS DE INTELIGENCIA ARTIFICIAL

AUTOR

ANDRÉS MAURICIO VERGEL MIRANDA

DIRECTOR

ANDRES MAURICIO PUENTES VELASQUEZ

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

Ocaña, Colombia

Febrero, 2017

## Índice

<b>Resumen.....</b>	<b>1</b>
<b>Introducción .....</b>	<b>3</b>
Capítulo 1. Algoritmo para la asignación de horarios académicos en la Universidad Francisco de Paula Santander Ocaña utilizando técnicas de inteligencia artificial.....	<b>5</b>
1.1 Planteamiento del Problema .....	5
1.2 Formulación del Problema.....	7
1.3 Objetivos de la Investigación.....	7
1.3.1 Objetivo General.....	7
1.3.2 Objetivos Específicos.....	7
1.4 Justificación .....	8
1.5 Delimitaciones .....	9
1.5.1 Delimitación Operativa.....	9
1.5.2 Delimitación Conceptual. ....	10
1.5.3 Delimitación Geográfica.....	10
1.5.4 Delimitación Temporal. ....	10
<b>Capítulo 2. Marco Referencial.....</b>	<b>11</b>
2.1 Marco Histórico .....	11
2.1.1 Antecedentes Nacionales .....	14
2.1.2 Antecedentes Internacionales.....	17
2.2 Marco Teórico.....	20
2.2.1 Alternativas de Solución.....	25
2.2.2 Selección de la Alternativa de Solución .....	27
2.3 Marco Conceptual.....	29
2.3.1 Inteligencia Artificial.....	29
2.3.2 Algoritmos .....	31
2.3.3 Diagrama de flujo .....	31
2.3.4 Algoritmos genéticos .....	31
2.3.5 Clasificación de problema P y NP .....	43
2.3.6 Asignación de Horario .....	44
2.3.7 Software .....	45
2.4 Marco Legal.....	46
Ley n.1341 2009 .....	46
Artículo 71 de la constitución política de Colombia .....	50
<b>Capítulo 3. Diseño Metodológico.....</b>	<b>51</b>
3.1 Metodología de Desarrollo de Software .....	51
3.2 Tipo de Investigación.....	51
<b>Capítulo 4. Resultados y Discusión.....</b>	<b>54</b>

4.1 Descripción Del Problema .....	54
4.2 Organización y Representación de los Datos de Entrada .....	56
4.3 Organización Y Representación De Estructuras De Validación.....	63
4.4 Estructura de Solución o Cromosoma.....	70
4.5 Definición de la Función De Evaluación .....	71
4.6 Preparación de Datos de Entrada .....	73
4.7 Algoritmo.....	74
4.7.1 Generación de la población inicial.....	75
4.7.2 Operadores genéticos .....	76
4.8 Parámetros del Algoritmo Genético.....	79
4.9 Descripción del Sistema.....	80
<b>Capítulo 5. Recolección de Información .....</b>	<b>95</b>
<b>Conclusiones .....</b>	<b>96</b>
<b>Recomendaciones .....</b>	<b>99</b>
<b>Referencias.....</b>	<b>100</b>
<b>Apéndice.....</b>	<b>104</b>

**Lista de tablas**

Tabla 1 .....	28
Tabla 2 .....	30
Tabla 3 .....	54
Tabla 4 .....	55
Tabla 5 .....	80
Tabla 6 .....	81

## Lista de figuras

Figura 1. Funcionamiento de un algoritmo.....	34
Figura 2. Cruce de un punto.....	38
Figura 3. Cruce de dos puntos.....	38
Figura 4. Estructura de los datos manejados por un algoritmo genético. ....	43
Figura 5. Problemas de complejidad.....	44
Figura 6. Representación de la tabla grupos en la base de datos. ....	56
Figura 7. Representación de la tabla sede en la base de datos. ....	57
Figura 8. Representación de la tabla de salón en la base de datos.....	58
Figura 9. Representación de la tabla docente en la base de datos.....	59
Figura 10. Representación de la tabla horarios en la base de datos. ....	61
Figura 11. Representación de la tabla carreras en la base de datos. ....	61
Figura 12. Representación de la tabla materias en la base de datos. ....	62
Figura 13. Representación de la tabla horario docentes en la base de datos. ....	64
Figura 14. Horario disponible del docente utilizado por el prototipo CES. ....	65
Figura 15. Representación de grupos por materia en la base de datos. ....	66
Figura 16. Vista de asignación de grupos del prototipo CES. ....	67
Figura 17. Representación de la tabla docente materia en la base de datos.....	68
Figura 18. Vista de materias asignadas a un docente en el prototipo CES Educativo.....	69
Figura 19. Representación no binaria de un gen.....	70
Figura 20. Representación no binaria de un cromosoma.....	71
Figura 21. Datos de entrada. ....	73
Figura 22. Diagrama de flujo algoritmo genético. ....	74
Figura 22. Ejemplo cruce uniforme. ....	77
Figura 23. Ejemplo de mutación.....	78
Figura 24. Pantalla de Bienvenida. ....	82
Figura 25. Interfaz principal del prototipo CES.....	83
Figura 26. Interfaz docente del prototipo CES. ....	85
Figura 27. Interfaz materia docente del prototipo CES. ....	86
Figura 28. Interfaz Horario del prototipo CES. ....	87
Figura 29. Interfaz carreras del prototipo CES. ....	88
Figura 30. Interfaz materia del prototipo CES.....	89
Figura 31. Interfaz salones del prototipo ces. ....	90
Figura 32. Generando horarios en el prototipo CES.....	91
Figura 33. Ejecución del prototipo CES. ....	92
Figura 34. Visualización de horarios en el prototipo CES.....	92
Figura 35. Generando reportes en Excel.....	93
Figura 36. Reporte en Excel de horarios.....	94
Figura 37. Diagrama de flujo población inicial. ....	105
Figura 38. Diagrama de flujo Operador selección. ....	106
Figura 39. Diagrama de flujo operador cruce uniforme. ....	107
Figura 40. Diagrama de flujo operador mutación.....	108

## Resumen

La programación de horarios académicos es un problema particular que se encuentra dentro de la clasificación de la teoría de la complejidad como NP-completo. Este problema de horarios se conoce en la comunidad científica como *timetabling* o en español como problema de asignación de horarios universitario. Debido a su importancia y complejidad ha sido estudiado desde los años 70 hasta la actualidad. Hoy en día, la mayoría de las instituciones educativas que realizan el trabajo de la organización de horarios de clases de manera manual, ya sea por cuestiones de desconocimiento o simplemente de no querer dar un salto tecnológico en ese sentido, en cualquiera de los casos el trabajo manual se convierte en un problema de tiempo, personal y costos.

Como institución de educación superior la Universidad Francisco de Paula Santander Ocaña presenta problemas en la asignación de sus horarios de clase, generar estos horarios consumen tiempo y recursos humanos y el resultado obtenido no es el más óptimo, teniendo que dedicar días o semanas en el peor de los casos para satisfacer los requerimientos de los docentes, salones y estudiantes. Esto genera inconformidad y situaciones no deseadas a las personas involucradas en el mismo.

Es por ello que se hace necesario proponer una solución que permita mitigar las molestas situaciones que genera este problema. En este trabajo se presenta un prototipo diseñado para aplicar técnicas de inteligencia artificial que permitan optimizar el tiempo empleado en generar horarios para la facultad de ingenierías, mejorar la organización y sincronización de los

encargados de esta tarea como son los directores de programa, y darle un buen uso a los salones de la Universidad.

Para ello se realizó un estudio de las técnicas utilizadas en los trabajos de otras universidades lo que permite dar a conocer como dan solución a su problema particular. En nuestro trabajo se propuso una solución con la técnica de algoritmos genéticos, considerando que esta entrega mucha flexibilidad al momento de adaptarse a la generación de horarios del problema en particular. Además, tiene la ventaja de entregar varias soluciones y al ser una técnica estándar se puede poner en práctica rápidamente.

Una vez seleccionada la técnica a utilizar se realizó la etapa de representación de la información de nuestro problema en particular, como relacionar los datos involucrados de la Universidad para definir la representación de cromosomas utilizados por el algoritmo genético.

Este documento muestra el diseño y la implementación de un algoritmo genético que logra resolver el problema en tiempos razonables y con una calidad aceptable de soluciones. Al final se construyó un prototipo en JAVA llamado CES Educativo que permite generar horarios de clases por semestre para la facultad de ingenierías de la Universidad, teniendo en cuenta restricciones generales presentadas en problemas de *timetabling* y la disponibilidad de salones y docentes utilizando datos de prueba.



## Introducción

Los algoritmos genéticos son métodos de búsqueda pertenecientes a una metodología de meta heurísticas desarrollada en la rama de inteligencia artificial y es utilizada para dar solución a problemas de búsqueda y optimización, se basan en los mecanismos de selección que utiliza la naturaleza para permitir que los individuos más fuertes y mejor adaptados de una población sobrevivan y generen nuevos individuos. Los algoritmos genéticos han demostrado ser muy rápidos y confiables en las soluciones generadas para este tipo de problemas de optimización.

El problema de asignación de horarios de la Universidad Francisco de Paula Santander Ocaña es uno de esos problemas que necesitan una solución de optimización. Para encontrar dicha solución se aplicaron algoritmos genéticos con el fin de cumplir la mayoría de restricciones que la Universidad propone en cuanto a la generación de horarios se refiere. El término de restricciones hace referencia a las condiciones que debe satisfacer el horario de clases.

Adicionalmente se realizó una clasificación de restricciones las cuales se aplican a la mayoría de estos problemas de horarios y las restricciones específicas que son propias de la Universidad.

Esta investigación espera dar solución al problema de asignación de horarios en la facultad de ingenierías de la Universidad Francisco de Paula Santander Ocaña mediante la incorporación de tecnología de inteligencia artificial que permita generar de manera automática los horarios de

clases disminuyendo el tiempo que actualmente los directores de programa invierten en este proceso.

El principal objetivo de esta investigación es la construcción de un prototipo que realice esta tarea de forma automática mediante la técnica de algoritmos genéticos y que genere soluciones aceptables. También se busca tener en cuenta el horario disponible de los docentes encargados de dictar clases en la facultad y que los salones sean asignados a materias que de verdad necesiten los recursos de clase que estos ofrecen, como lo son los laboratorios, salas de cómputo y salas de dibujo.

# Capítulo 1. Algoritmo para la asignación de horarios académicos en la Universidad Francisco de Paula Santander Ocaña utilizando técnicas de inteligencia artificial.

## 1.1 Planteamiento del Problema

El proceso de asignación de horarios que tiene la Universidad Francisco de Paula Santander Ocaña implica la gestión de diferentes variables de manera simultánea como son: la distribución de los salones, las materias, los horarios de los profesores, los grupos de estudiantes y los periodos de tiempo. Este trabajo es realizado por los directores de programa el cual toma como base las asignaciones de semestres anteriores y se le realizan los cambios necesarios que permitan satisfacer las necesidades del semestre en curso. Estas necesidades hacen referencia al cumplimiento de una gran variedad de restricciones que se clasifican en dos grupos: reglas de obligatorio cumplimiento y restricciones de optimización, algunas de estas restricciones tienen que ver con situaciones como la disponibilidad de los profesores, la asignación correcta de profesores a sus respectivas materias, la disponibilidad de salones, las características de las aulas en cuanto a tamaño, disponibilidad de equipos y la cercanía entre ellas, entre otras.

Todas las situaciones y restricciones anteriores se presentan al mismo tiempo, generando un problema muy complejo, que se ha abordado de manera parcial, generando soluciones de asignación manual de horarios. Actualmente Díaz, García, y Hernández (2015), “la asignación manual toma bastante tiempo y es poco eficiente” (p.10) pues los problemas que se presentan, por ejemplo el sobrecupo, la cantidad de grupos a generar para cada materia y encontrar los

docentes para esos nuevos grupos, son complicados de resolver. Teniendo en cuenta solo un subconjunto de las condiciones previamente mencionadas; se generan asignaciones de horarios que en algunos casos han derivado en estados no deseados para estudiantes y docentes, y para los recursos asignados: tiempo, salones, laboratorios.

En entrevistas realizadas a directores de programa y al jefe de planeación académica de la Universidad se han podido detectar una serie de situaciones específicas que requieren una solución de optimización:

- El ritmo de crecimiento actual de la Universidad requiere que se asignen los salones de clase de manera eficiente (tamaño del salón asignado de acuerdo a la cantidad de estudiantes, características de seguridad y confort de las aulas de acuerdo al tipo de materia y al horario de clase).
- Dificultad en la organización y sincronización entre las personas encargadas de asignar salones y horarios de clase (hay descentralización de la asignación, pero no hay una administración centralizada eficiente).
- Evitar bloques de clase con asignaciones en diferentes sedes de la Universidad.
- Asignación de recursos de clases a materias que no las necesitan.

El problema de *Timetabling* o problema de programación horaria se define como la asignación de un número de eventos, cada uno con ciertas características a un número limitado de recursos sujetos a restricciones. En este caso específico, se abordará el problema de

*educational timetabling* que hace referencia a los problemas de programación de horarios en instituciones educativas.

Actualmente los directores de programa son los encargados de lidiar con este problema todos los semestres, ellos son los responsables de realizar este proceso de asignación de forma manual. Se ha determinado que los resultados de los horarios generados utilizando esta técnica dependen de forma muy subjetiva a la experiencia de quien valida y asigna. (Torres, 2013).

## **1.2 Formulación del Problema**

¿Realizar el proceso de asignación de salones y horarios de clases apoyándose en un algoritmo de inteligencia artificial optimizará la forma en que se asignan los recursos académicos?

## **1.3 Objetivos de la Investigación**

### **1.3.1 Objetivo General**

Desarrollar un algoritmo de optimización para la asignación de salones y horarios de clases en la Universidad Francisco de Paula Santander Ocaña utilizando técnicas de inteligencia artificial

### **1.3.2 Objetivos Específicos**

- Conceptualizar el conocimiento del dominio partiendo de diversas técnicas de recolección y representación de la información

- Formalizar la solución algorítmica con las reglas y restricciones particulares del dominio
- Construir un prototipo funcional de software a partir del algoritmo previamente definido, teniendo en cuenta las pruebas necesarias con datos reales de la facultad de ingeniería.

#### **1.4 Justificación**

El problema de asignación de horarios académicos presente en todas las universidades radica en la complejidad que presenta la programación de eventos a un número limitado de recursos sujetos a una serie de restricciones. La facultad de ingenierías de la Universidad Francisco de Paula Santander Ocaña padece este problema todos los semestres y es por esto que es pertinente desarrollar una herramienta de software que ayude a suplir estas necesidades, buscando la optimización del proceso de asignación de horarios con el objetivo de facilitar la toma de decisiones que permita gestionar de forma eficiente los recursos académicos disponibles. También evitar las molestias que puede generar para estudiantes y docentes la asignación de horarios que no tengan en cuenta la posición geográfica de los salones y la asignación de asignaturas en tiempos donde el docente no está disponible.

Los beneficios que implicaría el desarrollo de una solución algorítmica que asigne de manera automática los horarios académicos de la Universidad serían: la optimización del tiempo de las personas (estudiantes y profesores) al establecer reglas que ubiquen las materias en horarios contiguos (disminuyendo el número de desplazamientos); se optimizará el uso de los recursos (salones, recursos de apoyo) al asignar a cada materia los salones de acuerdo a la necesidad; se disminuirá el tiempo empleado al comienzo de cada semestre para el proceso de

asignación de horarios; la Universidad contará con una herramienta avanzada de software que le permita estar preparada para el crecimiento, ya que será más fácil gestionar el tiempo y los recursos académicos, sin importar la cantidad de estudiantes.

En el ámbito nacional, la Universidad Distrital Francisco José de Caldas (Díaz, García, & Hernández, 2015) y la Universidad de los Andes (Castro & Medaglia, 2004), han desarrollado múltiples soluciones al problema de asignación de horarios, en las cuales los resultados de la implementación de técnicas de inteligencia artificial han demostrado tener un gran impacto en la optimización de los horarios. A nivel internacional trabajos realizados en instituciones de educación superior de Chile, México, España y Uruguay consiguieron implementar estas técnicas, logrando resultados satisfactorios en cuanto a la reducción del tiempo destinado en la realización de los horarios y la optimización de los recursos académicos y la disponibilidad de los docentes.

## **1.5 Delimitaciones**

### **1.5.1 Delimitación Operativa**

El algoritmo a desarrollar tendrá en cuenta un conjunto de restricciones aplicables a toda la Universidad, sin embargo, el alcance del presente proyecto contempla el diseño de un algoritmo de optimización y el desarrollo de una prueba piloto funcional con los programas de la facultad de ingeniería, debido a la cantidad de información que se requiere capturar, procesar y almacenar. Se recomienda a la universidad dedicar personas y recursos para la implantación de la solución para el resto de programas de la Universidad.

### **1.5.2 Delimitación Conceptual.**

En este proyecto se realizará un estudio de las tecnologías de inteligencia artificial utilizadas a nivel nacional e internacional para el tratamiento de problemas de asignación de horarios en instituciones educativas para luego seleccionar la más adecuada y poder dar solución a los problemas de la facultad de ingenierías de la Universidad Francisco de Paula Santander Ocaña.

La base conceptual que trabaja este proyecto comenzará de lo general a lo específico. En lo general se trabajara problemas de *timetabling* o asignación de horarios, su clasificación, su complejidad computacional, las técnicas de inteligencia artificial desarrolladas para darle solución, hasta llegar a especificar la técnica de algoritmos utilizada para desarrollar el aplicativo. Las técnicas que hasta el momento se han contemplado son Algoritmos genéticos, Redes neuronales, Meta-heurísticas.

### **1.5.3 Delimitación Geográfica.**

Este proyecto se desarrollará en las instalaciones del Tecnoparque nodo Ocaña y en la oficina del grupo de investigación en tecnología y desarrollo en ingeniería GITYD de la facultad de ingenierías de la Universidad Francisco de Paula Santander Ocaña.

### **1.5.4 Delimitación Temporal.**

Este trabajo tendrá una duración de dieciséis (16) semanas.



## Capítulo 2. Marco Referencial

### 2.1 Marco Histórico

Los orígenes de la historia de la computadora se remontan a épocas donde el ser humano sintió la necesidad de realizar procedimientos de cálculo de una manera más rápida.

A partir del siglo XVII los principales matemáticos y hombres de ciencia crearon máquinas que permitían realizar cálculos matemáticos como sumas y restas. Johnston (2000) “Desde el origen de las computadoras, constantemente se han hecho investigaciones científicas y tecnológicas con la finalidad de facilitar algunas actividades propias de los seres humanos.”(p.24). Algunas de estas actividades eran realizar el procesamiento de cálculos matemáticos a una gran velocidad y complicados procesos que requerían una toma rápida de decisión.

El término inteligencia artificial fue primeramente utilizado por John McCarthy en 1956 cuando se llevó a cabo la primera conferencia académica sobre el tema, en la cual se reunieron los personajes más prominentes en el área de teoría de autómatas, redes neuronales y el estudio de la inteligencia, cuyos aportes realmente notables fue proponer el concepto de inteligencia artificial para este campo de investigación el cual entre otras cosas se encargaba de entender el tema sobre si las computadoras podían realmente pensar.

Se abrió el camino a una nueva etapa, donde la computadora hacia parte de la ciencia y la ingeniería, permitió el estudio de modelos de computo capaces de realizar actividades de los seres humanos en base a sus características de razonamiento y conducta, por eso se dice que Smith (2006) "Ninguno puede refutar la habilidad de las computadoras para procesar lógicamente pero no mucho se sabe sobre si una maquina puede pensar" (p.4).

Los autores reconocidos del primer trabajo de inteligencia artificial fueron Warren McCulloch y Walter Pitts, quienes partieron de tres fuentes: conocimientos sobre la fisiología básica y funcionamiento de las neuronas del cerebro, el análisis formal de la lógica proposicional de Russell y Whitehead y la teoría de la computación de Turing. Propusieron un modelo constituido por neuronas artificiales, en el que cada una de ellas se caracterizaba por estar <<activada>> o <<desactivada>>. Mostraron, por ejemplo, que cualquier función de computo podría calcularse mediante una red de neuronas interconectadas y que todos los conectores lógicos (and, or, not, etc.) se podrían implementar utilizando estructuras de red sencillas. (Russell & Norvig, 1996, p. 19).

Los primeros años de la inteligencia artificial (desde ahora IA) estuvieron llenos de éxitos (aunque con ciertas limitaciones), teniendo en cuenta lo primitivo de los computadoras y las herramientas de programación de aquella época. Los más importantes avances realizados los últimos 60 años han sido sobre avanzados algoritmos de búsqueda, algoritmos de aprendizaje de máquina y la integración del análisis estadístico para comprender el mundo en general.

La primera década de investigación de la IA se enfocó en el desarrollo de mecanismos de búsqueda de propósito general, en busca de aplicar la IA a problemas más amplios o más complejos aparece una nueva alternativa, sistemas basados en conocimiento, que aplicaban conocimiento específico del dominio que facilitarían la resolución de problemas. Esto permitía afirmar que “el conocimiento específico del dominio del problema garantizaba en la práctica saber de antemano la correspondiente respuesta.” (Russell & Norvig, 1996, p. 26).

Los primeros sistemas basados en conocimiento o expertos eran programas de computación con el objetivo de modelar la experiencia humana en una o más áreas específicas de conocimiento, que por lo general consisten de tres componentes básicos; base de datos de conocimiento de hechos y reglas que representan el conocimiento y la experiencia humana; un motor de inferencia para procesar y determinar cómo se están haciendo las inferencias; y una interfaz de entrada / salida para la interacción con el usuario.

Los avanzados algoritmos de búsqueda generados como resultado de investigaciones realizadas a la resolución de problemas de optimización o de gran complejidad, problemas como la programación horaria que consisten en la asignación de ciertos eventos a distintos bloques horarios respetando una serie de requerimientos y condiciones. Hernandez, Miranda, & Rey (2008) “Esta área es conocida como *Timetabling* y dentro de ella existe una rama específica llamada *Class Scheduling*, que estudia problemas relacionados con la programación horaria para entidades educativas.” (p.122).

Dentro de este contexto existen tres tipos de problemas:

- Programación de horarios de evaluaciones y exámenes (*Examination Timetabling*).
- Programación de horarios de clases para colegios (*School Course Timetabling*).
- Programación de horarios de clases para instituciones de educación superior o universidades (*University Course Timetabling*). (Schaerf, *Artificial Intelligence Review*, 1999, p. 127).

Posteriormente se comenzó el desarrollo de nuevas ideas que faciliten la optimización de estos problemas, la teoría de algoritmos evolutivos, el desarrollo de algoritmos genéticos por Holland y las estrategias evolutivas fueron algunas de ellas.

Descubiertos por John H. Holland en la década de los 60 y definidos como algoritmos de búsqueda basados en los mecanismos de selección natural y genética natural. Combinan la supervivencia de los más compatibles entre las estructuras de cadenas, con una estructura de información ya aleatorizada, intercambiada para construir un algoritmo de búsqueda con algunas de las capacidades de innovación de la búsqueda humana (Goldberg, 1989).

### **2.1.1 Antecedentes Nacionales**

En la actualidad es posible encontrar la aplicación de estas técnicas de inteligencia artificial en la mayoría de problemas complejos que afrontamos (transito, transporte y envío de carga y

asignación de recursos de clases), en Colombia varias universidades han dedicado estudios a técnicas de inteligencia artificial para optimizar los resultados en la asignación de horarios de clases, los cuales se describen a continuación:

En Colombia existen desarrollos relacionados con este proyecto, se encontraron trabajos realizados que destacaron y fueron tenidos en cuenta para el desarrollo del nuestro, por ejemplo, el trabajo de asignación de horarios de clases universitarias mediante algoritmos evolutivos, trabajo que realizo mediante procedimientos de meta heurística, con la técnica no tradicional llamada algoritmos evolutivos en la Universidad del Norte Barranquilla, Colombia (Mejia, 2008), propuso tratar la problemática de organizar y distribuir los horarios de clases utilizando para ello la disponibilidad de los docentes como uno de los pilares fundamentales. También establece un modelo matemático donde se describen los recursos necesarios a utilizar, esto nos permitió identificar los parámetros necesarios en la resolución de nuestro problema y permite que nuestro modelo tenga en cuenta variables como salones especiales para cada asignatura, su capacidad y ubicación geográfica.

En la Universidad de los Andes se formuló un modelo de heurística basada en programación entera binaria para el problema de asignación de salones, que tiene en cuenta los objetivos administrativos de la oficina de admisiones y registro como las preferencias de los profesores por un horario y salones específicos. (Castro & Medaglia, 2004). En el se mencionan variables o restricciones administrativas que deben ser tenidas en cuenta, también definen las combinaciones profesor grupo y el espacio que este debe tener en los días de la semana, estas definiciones ayudaron a complementar nuestro modelo ya que la realización de un buen horario

de clases no debe solo considerar las preferencias de horario o salón específico de los profesores o estudiantes, también es necesario satisfacer algunos procesos administrativos de la universidad. Estas preferencias de salones especiales y disponibilidad de horarios de profesores son tenidas en cuenta en nuestro trabajo, y han demostrado ser un ítem importante que se tiene en cuenta al momento de evaluar la aceptación de los horarios de clases.

En 2010, la Universidad Tecnológica de Pereira (Gonzalez & Quintero, 2010), presento un análisis, diseño e implementación de un sistema informático que apoyaba el proceso de asignación de carga académica usando programación con restricciones, las cuales eran mayormente enfocadas en los profesores, teniendo en cuenta su intensidad horaria, sobrecarga y jornada lo cual permitió delimitar la asignación de horario con respecto a los toques de cada profesor, es decir, su clasificación (catedráticos o de contrato).

Finalmente, en el año 2011 en la Universidad ICESI (Gonzales & Trujillo, 2011), se formuló un modelo de gestión del conocimiento para la programación de horarios que se compone de ocho etapas con el fin de convertir el conocimiento tácito asociado a la programación de horarios en conocimiento explícito. También incluye la identificación de los procesos asociados a la generación de un plan de horarios y especifica las entradas, variables y salidas de estos. La validación del modelo se realizó mediante la aplicación de este en la Universidad ICESI, logrando obtener beneficios importantes para la institución en Cali, Colombia.

Los trabajos anteriores presentaron una idea de la importancia de la disponibilidad horaria de los profesores, resaltan que gran parte de la complejidad que presenta este problema de asignación de horarios es por los diferentes valores que tiene esta variable para cada profesor.

### **2.1.2 Antecedentes Internacionales**

En el ámbito internacional también se han desarrollado proyectos que involucran problemas de asignación de horarios en universidades. Algunos de los trabajos presentados a continuación permitieron realizar la representación de las variables que intervienen en nuestro problema.

En los años 2005 se aborda el problema de programación de horarios utilizando algoritmos genéticos, la Universidad del País Vasco (Gomez, Artaza, Irigoyen, & Iriando, 2005), realizó un análisis del problema y estableció un modelo general del mismo pero ajustándolo a su problema particular, esto con el fin de implementar un algoritmo que genere soluciones cuasi óptimas en tiempos razonables. Ese mismo año la Universidad de la República en Uruguay (Costabel, 2005), realizó la investigación de la aplicación de las meta heurísticas en el desarrollo de problemas de asignación de horarios de clases con el fin de construir un motor capaz de apoyar la herramienta de software que utiliza la facultad de ingenierías de la universidad, este motor construido incluye la combinación de las meta heurísticas más importantes como los algoritmos genéticos, búsqueda tabú y dentro de optimización por colonia de hormigas. Al final se busca realizar una comparación en cuanto al desempeño de cada una de las meta heurísticas utilizadas en la solución del mismo problema.

A pesar de que los algoritmos genéticos utilizados en la solución de los problemas de la investigación anterior obtuvieron las peores soluciones, estudios más recientes como el realizado en la Universidad Nacional Mayor de San Marcos en Lima, Perú (Cortez, Rosales, Naupuri, & Vega, 2010), propone que la efectividad en las soluciones generadas por un algoritmo depende en buena medida a una excelente definición de su función de evaluación y la codificación que se realiza del problema. Definieron un modelo matemático que sirviera como un sistema de apoyo a la generación de horarios en una facultad de la universidad, dicho modelo definía las restricciones del problema y luego establecía un diseño de la solución permitiendo la adaptación de los algoritmos genéticos.

A nivel internacional se estableció un modelo de asignación de carga académica usando algoritmos genéticos capaces de encontrar la asignación óptima de clases, profesores y horarios en el Instituto Tecnológico de Nuevo Laredo, México. (López & Johnston, 2007). Dicho modelo fue simplificado y fue adaptado para mostrar la relación grupo-materia, profesor y hora que hace referencia al gen del cromosoma.

La Universidad del Caribe (Castilla, Jacinto, & Joel, 2008), presentó una nueva forma de realizar la representación de los individuos (cromosoma) en la aplicación de algoritmos genéticos en el problema de calendarización de horarios y fue desarrollado por el programa de estudios de ingeniería telemática. Esta representación simplificada del cromosoma nos permitió empezar con la codificación de nuestro problema, realizar la representación no binaria de un gen de nuestro trabajo.



En el año 2013, el Politécnico di Milano (Colorni, Dorigo, & Maniezo, 2013), presento un artículo con los resultados de una investigación realizada que buscaba medir las posibilidades que ofrecían los algoritmos genéticos en la solución de problemas de asignación de horarios y la comparación con otras meta heurísticas. Los resultados presentados fueron comparados con los resultados de otros algoritmos, la conclusión a la que llegaron fue que los algoritmos genéticos producían mejores soluciones que el algoritmo de enfriamiento lento simulado, pero no eran tan buenos como los resultados del algoritmo búsqueda tabú, sin embargo los algoritmos genéticos mostraron tener una ventaja sobre los demás algoritmos pues permitían flexibilidad para adaptarlos y resolver otros problemas de asignación de horarios. El problema utilizado para esta investigación fue un problema real de una institución de educación superior de Italia.

Es preciso resaltar dos trabajos diferentes realizados en Chile y México: El primero en la institución de educación superior INACAP donde se desarrolló el proyecto de generación de horarios académicos utilizando algoritmos genéticos (Ahumada, 2014), donde se propuso la organización y representación de los datos de entrada mediante tablas en una base de datos relacional que luego permitía realizar la combinación de los datos para así generar los cromosomas del individuo, en este trabajo se especifican las tablas utilizadas y el modelo relacional entre ellas, este modelo ha sido tomado como referencia para nuestro trabajo así como también la base teórica. El segundo tuvo lugar en 2000, en Nuevo León (Johnston, 2000), donde también se aplicaron algoritmos genéticos para la asignación de carga académica en instituciones de educación superior pero este determinaba la carga académica más apropiada para cada profesor de la institución, considerando los diversos factores involucrados. Desarrollaron un prototipo funcional con un número muy limitado de restricciones que permitía darle valores de

penalización, utilizando la minimización como función de evaluación. Este modelo de evaluación es implementado en nuestro trabajo.

Estas investigaciones realizadas en el área de la asignación de horarios para universidades, permitieron definir nuestro modelo, los parámetros que debemos tener en cuenta para nuestro problema, si bien la mayoría de los trabajos tienen algunas características en común, los ítems a tener en cuenta al momento de evaluar el resultado de nuestras soluciones deben ser más particulares. También permitieron dar una idea de las interfaces del nuestro prototipo y de la interacción que debe tener el usuario con el software, el poder gestionar los recursos disponibles en la universidad.

Generar investigación que permita seleccionar la mejor alternativa para resolver el problema de asignación de horarios de la Universidad Francisco de Paula Santander Ocaña en su Facultad de Ingenierías. Se hace mención a los diferentes métodos que han sido utilizados en Colombia y en el mundo con mayor o menor éxito para resolver el problema siendo el uso de algoritmos genéticos los más recurrentes y los cuales se eligió en esta investigación.

## **2.2 Marco Teórico**

El problema de asignación de horarios, es un problema de gran complejidad que se presenta en todas las instituciones de educación superior, la creación de horarios de clases es un procedimiento administrativo que consiste en asignar para cada curso un docente, una sala y un hora para ser dictado en la semana, este proceso debe hacerse para cada semestre pues las condiciones varían, aunque actualmente la facultad utiliza la información de semestres anteriores

para asignar nuevos horarios. Este no es un proceso simple, en este proceso existen muchos factores que se convierten en restricciones que deben ser consideradas al momento de generar un horario.

Cuando hablamos de las restricciones a tener en cuenta existen dos tipos:

- **Restricciones Obligatorias:** Son aquellas condiciones que deben cumplirse, la violación de alguna de estas restricciones generan un horario no válido.

- **Restricciones deseables:** Son aquellas condiciones que definen preferencias del usuario, el no cumplimiento de estas no genera horarios no válidos, pero entre más condiciones se cumplan más óptimo será el resultado del mismo.

Para el caso de la facultad de ingenierías de la Universidad se consideran las siguientes restricciones:

### **Restricciones Obligatorias Generales**

- **ROG1.** Un curso debe tener asignado a lo más un docente en un aula en un periodo específico

- **ROG2.** Un docente debe tener asignado a lo más un curso en un aula en un periodo específico

- **ROG3.** Un aula debe tener asignado a lo más un curso en un aula en un periodo específico

- ROG4. No es posible asignar a un docente a un curso por el cual no posea preferencia.
- ROG5. No se puede asignar un curso a un docente fuera de su disponibilidad de tiempo.
- RGO6. La cantidad de estudiantes de un curso no puede sobrepasar la capacidad del aula
- RGO7. Un curso debe cumplir con una cantidad de horas semanales requeridas. Según sea el caso, si el número de horas académicas es par se asignará dos horas diarias hasta cubrir la totalidad, sino se asignará un día de tres horas y el resto de dos horas.
- ROG8. Las horas de un bloque para un determinado curso en un mismo día deben ser consecutivas además de ser asignados a un docente y aula.
- ROG9. Cada bloque de un determinado curso solo puede asignarse a lo más una sola vez cada día.
- ROG10. Los cursos no deben solaparse entre ellos, ni en duración ni ubicación física, una vez seleccionada la hora de inicio de estos y el aula correspondiente.

### **Restricciones Obligatorias Específicas**

- ROE1. Un docente debe impartir una cantidad determinada de horas de dictados de clases como máximo y como mínimo según su categoría.
- ROE2. Los cursos de teoría deben asignarse a aulas de tipo no laboratorio.
- ROE3. Los cursos de laboratorio deben asignarse a aulas tipo laboratorio.

- ROE4. No deben coincidir los horarios de cursos que correspondan a un mismo semestre y grupo.
- ROE5. Se debe minimizar el número de horas libres entre el fin de una clase y el inicio de otra.
- ROG6. No se deben asignar cursos en horario de almuerzo (12:00 – 14:00).
- ROG7. No se deben asignar más de cuatro cursos por día.

Actualmente la mayoría de estas reglas no se cumplen en la asignación de horarios, las características de este proceso lo transforman en un problema de optimización, ya que consiste en asignar recursos limitados y que deben organizarse de la mejor forma posible. Así, el objetivo en este problema es minimizar el incumplimiento de las restricciones.

El problema de asignación de horarios que se presenta en todas las instituciones de educación llega a ser diferentes por sus restricciones individuales pero mantienen también una serie de restricciones generales lo que permite clasificar el problema para facilitar su comprensión. La siguiente clasificación realizada por (Schaerf A. , 1999).

- Programación de clases y docentes: Considera la programación semanal para todas las clases de una escuela evitando que los docentes se asignen a dos clases en el mismo tiempo y viceversa. Este tipo de problema se considera *scheduling*.
- Programación de clases y salones: Considera la programación semanal de todas las clases de un conjunto de cursos universitarios, minimizando los cruces de clases de cursos con alumnos comunes. Este tipo de problema se considera *timetabling*.

- **Asignación de horario de exámenes:** Consiste en la calendarización de exámenes de los estudiantes, evitando los cruces y permitiendo una buena separación entre los mismos asignando docentes y salones.

El problema que abordamos en la Universidad corresponden a una suma de los dos primeros tipos: programación de clases y docentes y programación de clases y salones. Por esto mismo, muchas veces el problema se aborda en dos etapas, una asignando los docentes a los cursos en un horario y la otra la combinación docente- curso- a el salón. (Granada, Toro, & Franco, 2006).

Estos son algunos de los enfoques que se tienen en cuenta para dar solución a este problema:

- **Aproximación:** Un algoritmo que rápidamente encuentra una solución no necesariamente óptima, pero dentro de un cierto rango de error. En algunos casos encontrar una buena aproximación es suficiente para resolver el problema.
- **Probabilidad:** Un algoritmo probabilístico obtiene en promedio una buena solución al problema planteado, para una distribución de los datos de entrada dada.
- **Heurísticas:** Son algoritmos que no entregan soluciones exactas y que se basan en una estrategia de búsqueda para llegar más rápido a una solución. (Cormen, Leiserson, Rivest, & Stein, 2001).

### 2.2.1 Alternativas de Solución

La complejidad de este problema de asignación de horarios que se aplica de manera general a todas las Universidades, lo hacen un problema muy estudiado y para el cual existe mucha investigación. A continuación algunas de las metodologías que se han utilizado para buscar su solución.

#### **Meta-heurísticas**

Las metodologías de meta-heurísticas son métodos probabilísticos que a partir de una solución inicial se van generando nuevas soluciones con la característica de que las nuevas soluciones serán mejor que la anterior, esto mediante la ejecución de un algoritmo. Este tipo de soluciones están basadas en un principio de aumento sucesivo e intentan evitar la generación de soluciones de pobre calidad.

Dentro de la meta-heurística se encuentran los siguientes métodos:

- **Búsqueda Tabú (Tabu Search).** Es un método heurístico de búsqueda local, comienzan con una solución inicial que puede ser aleatoria y a través de ciclos exploran el espacio de búsqueda, realizando una modificación a la solución inicial pasando iterativamente desde una solución a otra. Fue diseñada por Fred Glover. (Glover, 1989).
- **Algoritmos Genéticos (Genetic Algorithm).** Los algoritmos genéticos son métodos de búsqueda basado en el mecanismo de selección natural, comienzan con una

solución o población inicial que puede ser generada aleatoriamente también y luego se le aplica dos procesos básicos de la evolución, la selección natural y la reproducción. La selección se encarga de tomar los mejores individuos existentes en la población y prepararlos para su reproducción o mutación, lo cual hace que las soluciones o poblaciones vayan mejorando a medida que el algoritmo avanza. (Holland, 1992). Este método permite que se explore un espacio de búsqueda mayor al espacio de búsqueda inicial lo que da una gran ventaja sobre los métodos tradicionales.

- Algoritmos de enfriamiento lento simulado (Simulated Annealing). La familia de enfriamiento lento simulado tratan sobre un modelo de resolución para la optimización de problemas de tipo combinatorio con mínimos locales y su aproximación consiste en generar de forma aleatoria una solución cercana a la solución actual y aceptarla como buena si consigue reducir la función de costo. Este algoritmo se inicia con una solución que se va modificando a medida que el algoritmo avanza o itera. (Ahumada, 2014).

- Colonia de hormigas (ACO Ant Colony Optimization). El principio de este método se basa en el estudio del comportamiento que realizan las hormigas para marcar el camino más corto entre su colonia y el alimento. En la implementación de un algoritmo de colonia de hormigas, las posibles soluciones son denominadas hormigas artificiales, que puede ser una solución factible o no. Al comienzo existe una colonia o conjunto de soluciones posibles y en cada iteración las hormigas crean alternativas en base a la información recopilada por las soluciones antecesoras. La cantidad de hormigas es un parámetro importante a definir en el diseño del algoritmo. (Ahumada, 2014).



### **Programación lineal entera**

Utiliza métodos matemáticos que permitan formular los problemas mediante un sistema de inecuaciones lineales para luego definir una función objetivo, que se debe maximizar o minimizar según corresponda. Las variables de la función objetivo se encuentran sujetas a una serie de restricciones que tienen distinto peso según su importancia en el problema. Las restricciones son expresadas mediante el sistema de inecuaciones. (Ahumada, 2014).

### **Sistemas expertos**

Los sistemas expertos son sistemas basados en la simulación de los procedimientos que realizan los humanos para resolver un problema, buscan modelar el conocimiento y los pasos que un experto humano realiza para obtener conclusiones y resolver el problema determinado. Los sistemas expertos pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Estos sistemas razonan pero basándose en un conocimiento adquirido y no tienen sitio para la subjetividad. (Schaerf, Artificial Intelligence Review, 1999).

#### **2.2.2 Selección de la Alternativa de Solución**

Los algoritmos genéticos a pesar de que no son la técnica que genere mejores soluciones en comparación con otros algoritmos de búsqueda y optimización, ellos han demostrado poseer ventajas en la aplicación de soluciones en la asignación de horarios por sobre otros algoritmos. Teniendo en cuenta algunas de las siguientes características tomadas del informe de tesis (Ahumada, 2014):

- Son algoritmos estocásticos. Cada ejecución del algoritmo puede entregar a su vez soluciones distintas.
- Son algoritmos de búsqueda múltiple, por lo cual entregan varias soluciones.
- Son algoritmos que exploran un amplio espacio de soluciones.

### **Ventajas y desventajas de los algoritmos genéticos.**

- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se unan para problemas de optimización de una función de evaluación, resultan menos afectados por los óptimos locales (falsas soluciones).
- Usan operadores probabilísticos en vez de los típicos operadores determinísticos de las otras técnicas.
- La mayoría de las otras técnicas de optimización son también estocásticas.

(Ahumada, 2014).

La siguiente tabla muestra una comparación sobre las características que poseen los algoritmos genéticos con respecto a otros métodos de optimización basadas en el artículo de Goldberg. (Goldberg, 1989).

#### **Tabla 1.**

*Comparación de algoritmos genéticos con técnicas tradicionales.*

<b>Algoritmos genéticos</b>	<b>Métodos de optimización tradicionales</b>
-----------------------------	--

<ul style="list-style-type: none"> <li>• Trabajan con parámetros codificados, es decir que deben codificarse como cadenas de longitud finita sobre algún alfabeto finito.</li> </ul>	<ul style="list-style-type: none"> <li>• Trabajan directamente con los parámetros.</li> </ul>
<ul style="list-style-type: none"> <li>• Utilizan poblaciones de puntos, es decir, se usa una base de datos de puntos simultáneamente, de tal forma que la probabilidad de quedar atrapados en óptimos locales se reduce (falsas soluciones).</li> </ul>	<ul style="list-style-type: none"> <li>• Operan sobre puntos individuales, ya que sus movimientos en el espacio de búsqueda se hacen de un punto a otro usando reglas de transición determinística. Esto puede ocasionar que se encuentren óptimos locales en lugar de óptimos globales.</li> </ul>
<ul style="list-style-type: none"> <li>• No necesitan conocimientos auxiliares sobre el problema, ya que usan información de la función de evaluación con respecto a los cromosomas.</li> </ul>	<ul style="list-style-type: none"> <li>• Requieren de mucha información auxiliar para trabajar adecuadamente.</li> </ul>
<ul style="list-style-type: none"> <li>• Utilizan reglas de transición probabilísticas.</li> </ul>	<ul style="list-style-type: none"> <li>• Usan reglas determinísticas.</li> </ul>

Nota Fuente: (Ahumada, 2014).

## 2.3 Marco Conceptual

Para el desarrollo de las actividades planteadas para el proyecto, es esencial tener conocimiento sobre el área de estudio inteligencia artificial, algoritmos, programación y aplicación de algoritmos genéticos que permitan la construcción de un prototipo que genere horarios de clases para la facultad de ingenierías de la Universidad, usamos un conjunto de conceptos básicos que revisaremos a continuación.

### 2.3.1 Inteligencia Artificial

La inteligencia artificial es una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos en base a dos de sus características primordiales: el razonamiento y la conducta. (Johnston, 2000). Existen distintas definiciones de inteligencia artificial de acuerdo a distintos enfoques, a continuación alguna de ellas sacadas del libro de inteligencia artificial un enfoque moderno (Russell & Norvig, 1996) y representadas en la siguiente tabla tomada de un trabajo de tesis.

**Tabla 2**

*Definiciones de IA (Russell & Norvig, 1996)*

“La interesante tarea de lograr que la computadoras piensen... maquina con mente, en su amplio sentido literal.”(Haugeland, 1985).	“El estudio de las facultades mentales mediante el uso de modelos computacionales.”(Charniak y McDermott, 1985).
“La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como la toma de decisiones, resolución de problemas, aprendizaje...” (Bellman, 1978).	“El estudio de los cálculos que permiten razonar y actuar.”(Winston, 1992).
Sistemas que piensan como humanos	Sistemas que piensan racionalmente
Sistemas que actúan como humanos	Sistemas que actúan racionalmente
“El arte de crear maquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia.”(Kurzweil, 1990).	“Un campo de estudio que se enfoca a la explicación y emulación de la conducta inteligente en función de procesos computacionales.”(Schalkoff, 1990).
“El estudio de cómo lograr que las computadoras realicen tareas que por el momento los humanos hacen mejor.”(Rich y Knight, 1991).	“La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente.”(Luger y Stubblefield, 1993).

**Nota Fuente:** (Johnston, 2000).

De la tabla se tiene las definiciones que el autor asocio a las cuatro características fundamentales las cuales son: las de la parte superior se refieren a los procesos de la mente y el razonamiento, mientras que los de la parte inferior hacen alusión a la conducta. Por otro lado, las

definiciones de la izquierda evalúan la condición deseable en función de la eficiencia humana, mientras que las de la derecha lo hacen en base al concepto de inteligencia ideal denominado racionalidad. (Johnston, 2000, p. 26).

### 2.3.2 Algoritmos

Un algoritmo puede ser definido como la secuencia ordenada de pasos, sin ambigüedades, que conducen a la solución de un problema dado y expresado en lenguaje natural. Todo algoritmo debe ser:

- Preciso. Indicando el orden de realización de cada uno de los pasos.
- Definido. Si se sigue el algoritmo varias veces proporcionándole los mismos datos, se deben obtener siempre los mismos resultados.
- Finito. Al seguir el algoritmo, este debe terminar en algún momento, es decir tener un número finito de pasos. (Aguilar, Baena, & Azuela, 1996)

### 2.3.3 Diagrama de flujo

Los diagramas de flujo se utilizan tanto para la representación gráfica de las operaciones ejecutadas sobre los datos a través de todas las partes de un sistema de procesamiento de información como para la representación de la secuencia de pasos necesarios para describir un procedimiento particular. (Aguilar, Baena, & Azuela, 1996, p. 24).

### 2.3.4 Algoritmos genéticos

Los algoritmos genéticos son una clase de algoritmos evolutivos que se usan para resolver problemas de búsqueda y optimización. Estos algoritmos son capaces de hacer evolucionar los

individuos de una población hacia soluciones del mundo real, siempre que se encuentre en una codificación adecuada del problema, siendo esta una de las claves principales de éxito cuando se aplica este tipo de técnicas. (Gamo, 2006)

Los algoritmos genéticos (en adelante AG) simulan mediante poblaciones de individuos la evolución sufrida a través de diferentes operadores. En la naturaleza, los individuos de una población compiten entre ellos por diferentes recursos necesarios para la supervivencia como el agua y la comida, y los miembros de una misma especie compiten por aparearse. De todos ellos, los que tienen más éxito en sobrevivir y aparearse conseguirán un mayor número de descendientes. Los individuos con peores aptitudes producirán muy poca o ninguna descendencia. (Gamo, 2006).

Esto significa que los genes de los individuos de mayor adaptación al medio serán heredados por un número creciente de individuos en cada generación. La combinación de buenas características de diferentes antepasados puede generar una descendencia altamente adaptada al medio. De la misma forma, es posible también que la combinación de genes de individuos bien adaptados genere otros con peores características, con lo que estos individuos tenderán a desaparecer. (Gamo, 2006).

Los AG presentan una analogía directa a este comportamiento natural. A cada individuo se le asigna un valor dependiendo de lo buena que sea la solución que aporta al problema. La potencia de los AG viene de que la técnica que usan es robusta y puede tratar con éxito un gran número de tipos de problemas, incluyendo y destacando aquellos que son difícilmente

solucionables utilizando otro tipo de métodos clásicos. Gamo (2006)“Los AG no garantizan la localización exacta de la solución óptima del problema, pero proporcionan una solución alternativa aceptablemente buena en un tiempo corto.”(p.12).

Los AG trabajan sobre una población de individuos, cada uno de ellos representan una posible solución al problema que se desea resolver. Una generación se obtiene a partir de la anterior por medio de los operadores de reproducción. Existen dos tipos:

**Cruce.** Se trata de una reproducción de tipo sexual. Se genera una descendencia a partir del mismo número de individuos (generalmente dos) de la generación anterior.

**Copia.** Se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010, p. 16).

```

Inicializar población actual aleatoriamente
MIENTRAS no se cumpla el criterio de terminación
    crear población temporal vacía
    SI elitismo: copiar en población temporal mejores individuos
    MIENTRAS población temporal no llena
    seleccionar padres
    cruzar padres con probabilidad  $P_c$ 
    SI se ha producido el cruce
        mutar uno de los descendientes (prob.  $P_m$ )
        evaluar descendientes
        añadir descendientes a la población temporal
    SINO
        añadir padres a la población temporal
    FIN SI
    FIN MIENTRAS
    aumentar contador generaciones
    establecer como nueva población actual la población temporal
FIN MIENTRAS

```

**Figura 1. Funcionamiento de un algoritmo.**

**Nota Fuente:** (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

El funcionamiento genérico de un algoritmo genético puede apreciarse en el pseudocódigo, reflejado en la Figura 1.

A continuación comienza a generarse la nueva población en base a la aplicación de los operadores genéticos de cruce o copia. Una vez generados los nuevos individuos se realiza la mutación con una probabilidad definida que suele ser muy baja, generalmente entre el 0.5% y 2%.

Es recomendable utilizar AG en la solución de problemas con espacios de búsquedas discretos, aunque sean muy grandes. La función de aptitud es aquella que permite evaluar las diferentes respuestas y determina el objetivo del problema de optimización. Aunque los Ag únicamente maximizan, la minimización puede implementarse aplicando el recíproco de la



maximización. La particularidad que presenta la función aptitud es que debe castigar a las malas soluciones y debe premiar a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez. (Johnston, 2000, p. 31).

La codificación más común es por medio de cadenas binaria, aunque se han utilizado también números reales y letras. Esta forma de codificación originalmente fue propuesta por John Holland y actualmente es muy popular debido a que resulta muy sencilla de implementar.

#### *2.3.4.1 Operadores genéticos*

La mecánica de los AG es muy simple, para el paso de una generación a la siguiente se aplican algunos operadores. Los operadores básicos utilizados en un AG son los siguientes:

##### **Selección**

Es un proceso donde se van a escoger que individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto, la selección de un individuo estará relacionada con su valor de ajuste. No se debe, sin embargo, eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea. (Gestal, Rivero, Rabuñal, Dorado, y Pazos, 2010, p. 18).

- Selección por ruleta. Propuesto por Dejong, es posiblemente el método más utilizado. A cada uno de los individuos de la población se le asigna una parte

proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción la ruleta mayor que la recibida por los peores. Generalmente, la población esta ordenada en base al ajuste, por lo que las porciones más grandes se encuentran al inicio de la ruleta.

Es un método muy sencillo pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es  $O(n^2)$ ). Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

- Selección por torneo. La idea principal de este método de selección consiste en escoger a los individuos genéticos en base a comparaciones directas entre sus genotipos. Existen dos versiones de selección mediante torneo, el torneo determinístico y el torneo probabilístico. En la versión determinística se selecciona al azar un número  $p$  de individuos (generalmente se escoge  $p=2$ ). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

La versión probabilística únicamente se diferencia en el paso de selección al ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo  $[0..1]$ , si es mayor que un parámetro  $p$  (fijado para todo el proceso evolutivo) se escoge el individuo más alto y en caso contrario el menos apto. Generalmente  $p$  toma valores en el rango  $0.5 < p \leq 1$  variando el número de individuos que participan en cada

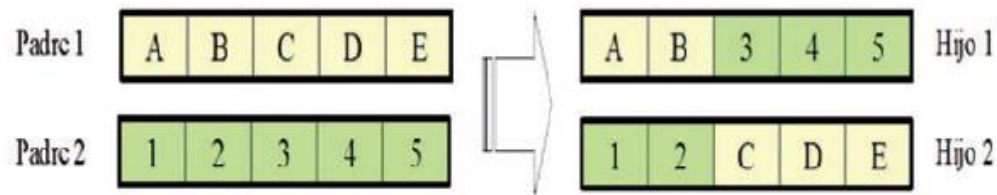
torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

## **Cruce**

Una vez realizada la selección de individuos, estos son recombinados para producir los nuevos individuos que van a pertenecer a la nueva generación. Lo que busca el cruce es garantizar que los nuevos individuos generados sean más aptos que sus antecesores o padres, esto se debe a que comparten las características buenas de dos individuos. La tasa de cruce con la que se trabaja es de 90%. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

Los algoritmos de cruce más empleados son los que aparecen a continuación:

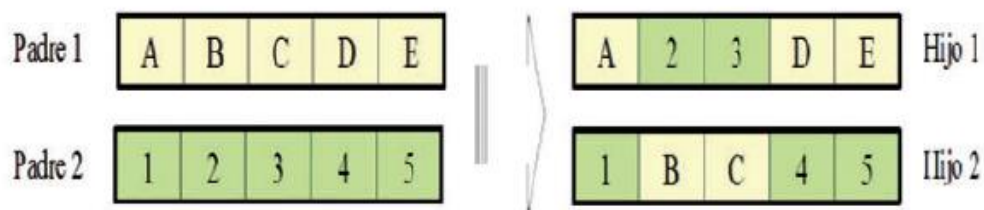
- **Cruce de 1 punto.** Es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres.



**Figura 2.** Cruce de un punto

**Nota Fuente:** (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

- Cruce de 2 puntos. Se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres, se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre.



**Figura 3.** Cruce de dos puntos.

**Nota Fuente:** (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

**Copia**

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación. El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo. Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce y si este finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

### **Mutación**

La mutación de un individuo provoca que alguno de sus genes, generalmente uno solo, varíe su valor de forma aleatoria. Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se suele utilizar de manera conjunta con el operador de cruce. Primeramente se seleccionan dos individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad  $p_m$ . Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

La probabilidad de mutación es muy baja, generalmente menor al 1%. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser exterminado.

“Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Este consiste en variar aleatoriamente un gen de un cromosoma. Si se trabaja con codificaciones binarias, consistirá simplemente en negar un bit.” (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010, p. 26).

#### *2.3.4.2 Evaluación*

Para el correcto funcionamiento de un AG se debe de poseer un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto, para cada tipo de problema que se desee resolver deberá derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos.

De esto se encarga la función de evaluación, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste. En la naturaleza el ajuste de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad deberá estar ponderada con el número de individuos de la población genética. (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

La función de evaluación debe diseñarse de tal modo que encapsule todas las restricciones, fuertes y suaves, del problema. Estas restricciones son representadas con un costo dentro de la función, asignando un costo más alto a las restricciones fuertes y uno más bajo a las restricciones

suaves. La aptitud de un cromosoma se obtiene sumando los costos de las restricciones no cumplidas, siendo los cromosomas o soluciones más aptos aquellos que incumplan el menor número de restricciones. Al igual que en el diseño de la representación del individuo para cada problema particular se debe diseñar una función objetivo. Una función objetivo bien diseñada conduce al algoritmo a seleccionar siempre los mejores individuos para la reproducción.

En el mundo de los AG se empleará esta medición para controlar la aplicación de los operadores genéticos. Es decir, permitirá controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo. La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se le asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

#### *2.3.4.3 Población inicial*

Como se dijo antes, este algoritmo se basa en la selección natural por lo cual debe comenzar con una población inicial que corresponde a posibles soluciones del problema y a partir de ellos construye nuevas soluciones. La población inicial (como también las “futuras” posibles soluciones) es sometida a la evaluación y se seleccionan probabilísticamente aquellos individuos más aptos según la función objetivo. Los cromosomas más aptos son seleccionados como candidatos a reproducirse, es decir dar origen a nuevos cromosomas, o como una solución al problema.

La población inicial puede ser generada de forma aleatoria, mediante una heurística particular, o tomada de una base de conocimiento de alguna solución anterior de otra instancia del problema. En la literatura se observa que al iniciar la población mediante heurísticas, se tiende a tener problemas de convergencia prematura, es decir, el algoritmo converge hacia soluciones óptimas locales. Es importante que población inicial tenga un tamaño lo suficientemente grande para garantizar la diversidad de soluciones. (Ahumada, 2014).

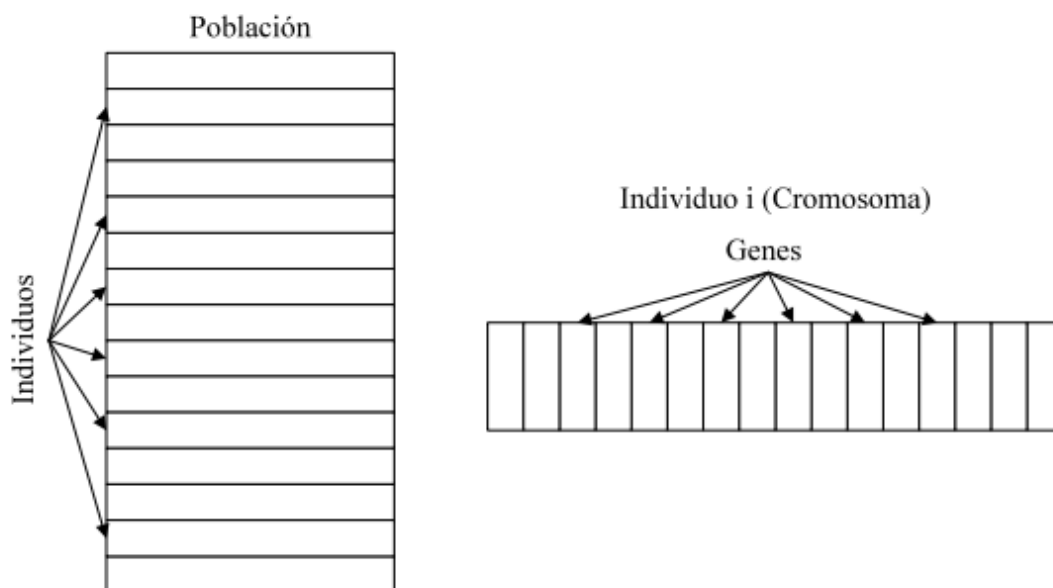
#### *2.3.4.4 Cromosoma*

El cromosoma es la representación de una solución al problema, la cual puede estar compuesta de uno o más genes. El cromosoma se conforma de una o más estructuras de datos que deben contener toda la información relevante del problema. La forma que se defina para el cromosoma es de extrema relevancia para asegurar la rapidez de la ejecución del algoritmo genético y es factor preponderante la velocidad con la cual la función objetivo puede evaluar si el cromosoma es apto o no. Los cromosomas deben ocupar el menor espacio posible y deben ser fáciles de preservar durante la ejecución del algoritmo genético. (Ahumada, 2014).

#### **Gen**

Se denomina gene o gen a cualquier posición a lo largo de una cadena que represente a un individuo.





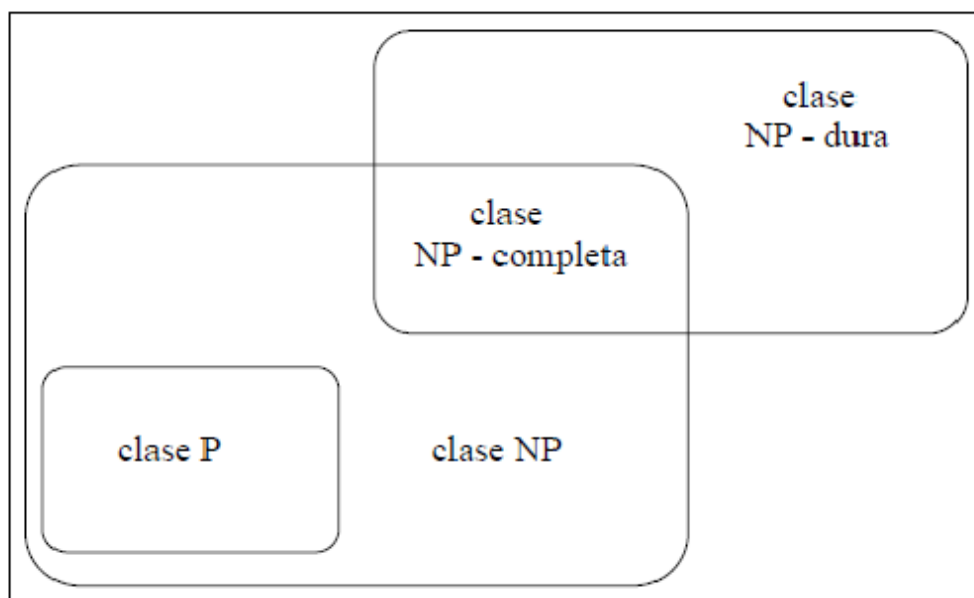
**Figura 4.** Estructura de los datos manejados por un algoritmo genético.

Nota Fuente: (Gamo, 2006).

### 2.3.5 Clasificación de problema P y NP

Es posible encontrar una clasificación para la complejidad, así se tiene que P es el conjunto de los problemas de decisión que pueden ser resueltos en una máquina determinista en tiempo polinómico, lo que corresponde intuitivamente a problemas que pueden ser resueltos aun en el peor de sus casos, por otra parte existen la clase de complejidad NP, que es el conjunto de los problemas de decisión que pueden ser resueltos por una maquina no determinística en tiempo polinómico. (Mejia, 2008)

Un conjunto X de problemas es hard con respecto a un conjunto de problemas Y si todo problema en Y puede ser transformado sencillamente en algún problema de X con la misma respuesta, el termino sencillo se define precisamente en cada caso, el conjunto hard más importante es NP-hard. El conjunto X es completo para Y si es hard para Y y es también un subconjunto de Y. el conjunto completo más importante es NP completo.



**Figura 5. Problemas de complejidad.**

Fuente: (Weber, 2000, p. 20).

### 2.3.6 Asignación de Horario

Mejía (2008) “Una definición general para un problema de *timetabling* es aquella donde se asignan un conjunto de eventos( encuentros, asignaturas, salones, profesores, entre otros) dentro de un número limitado de periodos de tiempo sujeto a un conjunto de restricciones.”(p.33). Las restricciones duras bajo ninguna circunstancia deben ser infringidas y son utilizadas para encontrar soluciones optimas, y las restricciones blandas son deseables de cumplir, pero dicha exigencia puede ser relajada si es necesario y generalmente son utilizadas para entregar un costo a la función objetivo la que será optimizada. (Burke & Erben, 2000).

Timetabling pertenece a la categoría de problemas NP-hard, en los cuales existe una proporción directa entre el tiempo y la calidad de la solución, lo cual impulsa a utilizar métodos de búsqueda que entreguen una solución óptima en un tiempo adecuado. En particular en los

problemas de timetabling o scheduling, los algoritmos genéticos han sido usados con mucho éxito encontrando soluciones muy cercanas a las óptimas y adaptándose muy bien a las condiciones particulares de cada problema.

### 2.3.7 Software

El software corresponde a la inmaterial o lógica de un sistema informático, son los datos y programas necesarios para que la parte física de un ordenador funcione y produzca resultados. Para que un ordenador pueda realizar una tarea es necesario que se le indique qué y cómo debe hacerlas, estas órdenes se denomina programa. El software es el conjunto de programas diseñados para controlar las actividades del ordenador que transformen datos de entrada en resultados. (Heredero, 2004).

Para que un programa pueda ser creado, es necesario que se concluya en que lenguaje de programación ofrecerá mayor rendimiento y será más rentable para el cliente, el lenguaje de programación corresponde al idioma de comunicación entre el hombre y el ordenador, mediante el cual se transmiten a este las instrucciones e información en un formato que sea comprensible para la máquina. (Heredero, 2004).

Para el desarrollo de este proyecto se utilizara el lenguaje de programación JAVA, se utilizara una base de datos creada en mysql utilizando el lenguaje de consulta '*Structured Query Language*, SQL', este lenguaje es de alto nivel para la recuperación de datos almacenados en bases de datos.

## 2.4 Marco Legal

### **Ley n.1341 2009**

“por la cual se definen principios y conceptos sobre la Sociedad de la información y la organización de las Tecnologías de la información y las comunicaciones - tic-, se Crea la agencia nacional de espectro y se dictan otras disposiciones”

El Congreso de Colombia

Decreta:

Título I Disposiciones Generales

Capítulo I - Principios Generales

**ARTICULO 1.- OBJETO.** La presente Ley determina el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, su ordenamiento general, el régimen de competencia, la protección al usuario, así como lo concerniente a la cobertura, la calidad del servicio, la promoción de la inversión en el sector y el desarrollo de estas tecnologías, el uso eficiente de las redes y del espectro radioeléctrico, así como las potestades del Estado en relación con la planeación, la gestión, la administración adecuada y eficiente de los recursos, regulación, control y vigilancia del mismo y facilitando el libre acceso y sin discriminación de los habitantes del territorio nacional a la Sociedad de la Información. Parágrafo. El servicio de televisión y el servicio postal continuarán rigiéndose por las normas especiales pertinentes, con las excepciones específicas que contenga la presente ley. Sin perjuicio de la aplicación de los principios generales del derecho.

**ARTÍCULO 2.- PRINCIPIOS ORIENTADORES.**

La investigación, el fomento, la promoción y el desarrollo de las Tecnologías de la Información y las Comunicaciones son una política de Estado que involucra a todos los sectores y niveles de la administración pública y de la sociedad, para contribuir al desarrollo educativo, cultural, económico, social y político e incrementar la productividad, la competitividad, el respeto a los derechos humanos inherentes y la inclusión social. Las Tecnologías de la Información y las Comunicaciones deben servir al interés general y es deber del Estado promover su acceso eficiente y en igualdad de oportunidades, a todos los habitantes del territorio nacional.

Son principios orientadores de la presente Ley:

Prioridad al acceso y uso de las Tecnologías de la Información y las Comunicaciones. El Estado y en general todos los agentes del sector de las Tecnologías de la Información y las Comunicaciones deberán colaborar, dentro del marco de sus obligaciones, para priorizar el acceso y uso a las Tecnologías de la Información y las Comunicaciones en la producción de bienes y servicios, en condiciones no discriminatorias en la conectividad, la educación los contenidos y la competitividad.

Libre competencia. El Estado propiciará escenarios de libre y leal competencia que incentiven la inversión actual y futura en el sector de las TIC y que permitan la concurrencia al mercado, con observancia del régimen de competencia, bajo precios de mercado y en condiciones de igualdad. Sin perjuicio de lo anterior, el Estado no podrá fijar condiciones

distintas ni privilegios a favor de unos competidores en situaciones similares a las de otros y propiciará la sana competencia.

Uso eficiente de la infraestructura y de los recursos escasos. El Estado fomentará el despliegue y uso eficiente de la infraestructura para la provisión de redes de telecomunicaciones y los servicios que sobre ellas se puedan prestar, y promoverá el óptimo aprovechamiento de los recursos escasos con el ánimo de generar competencia, calidad y eficiencia, en beneficio de los usuarios, siempre y cuando se remunere dicha infraestructura a costos de oportunidad, sea técnicamente factible, no degrade la calidad de servicio que el propietario de la red viene prestando a sus usuarios y a los terceros, no afecte la prestación de sus propios servicios y se cuente con suficiente infraestructura, teniendo en cuenta la factibilidad técnica y la remuneración a costos eficientes del acceso a dicha infraestructura. Para tal efecto, dentro del ámbito de sus competencias, las entidades de orden nacional y territorial están obligadas a adoptar todas las medidas que sean necesarias para facilitar y garantizar el desarrollo de la infraestructura requerida, estableciendo las garantías y medidas necesarias que contribuyan en la prevención, cuidado y conservación para que no se deteriore el patrimonio público y el interés general.

Protección de los derechos de los usuarios. El Estado velará por la adecuada protección de los derechos de los usuarios de las Tecnologías de la Información y de las Comunicaciones, así como por el cumplimiento de los derechos y deberes derivados del Habeas Data, asociados a la prestación del servicio. Para tal efecto, los proveedores y/u operadores directos deberán prestar sus servicios a precios de mercado y utilidad razonable, en los niveles de calidad establecidos en los títulos habilitantes o, en su defecto, dentro de los rangos que certifiquen las entidades

competentes e idóneas en la materia y con información clara, transparente, necesaria, veraz y anterior, simultánea y de todas maneras oportuna para que los usuarios tomen sus decisiones.

Promoción de la Inversión. Todos los proveedores de redes y servicios de telecomunicaciones tendrán igualdad de oportunidades para acceder al uso del espectro y contribuirán al Fondo de Tecnologías de la Información y las Comunicaciones.

Neutralidad Tecnológica. El Estado garantizará la libre adopción de tecnologías, teniendo en cuenta recomendaciones, conceptos y normativas de los organismos internacionales competentes e idóneos en la materia, que permitan fomentar la eficiente prestación de servicios, contenidos y aplicaciones que usen Tecnologías de la Información y las Comunicaciones y garantizar la libre y leal competencia, y que su adopción sea armónica con el desarrollo ambiental sostenible.

El Derecho a la comunicación, la información y la educación y los servicios básicos de las TIC: En desarrollo de los artículos 20 y 67 de la Constitución Nacional el Estado propiciará a todo colombiano el derecho al acceso a las tecnologías de la información y las comunicaciones básicas, que permitan el ejercicio pleno de los siguientes derechos: La libertad de expresión y de difundir su pensamiento y opiniones, la de informar y recibir información veraz e imparcial, la educación y el acceso al conocimiento, a la ciencia, a la técnica, y a los demás bienes y valores de la cultura. Adicionalmente el Estado establecerá programas para que la población de los estratos desarrollará programas para que la población de los estratos menos

favorecidos y la población rural tengan acceso y uso a las plataformas de comunicación, en especial de Internet y contenidos informáticos y de educación integral.

Masificación del gobierno en línea. Con el fin de lograr la prestación de servicios eficientes a los ciudadanos, las entidades públicas deberán adoptar todas las medidas necesarias para garantizar el máximo aprovechamiento de las Tecnologías de la Información y las Comunicaciones en el desarrollo de sus funciones. El Gobierno Nacional fijará los mecanismos y condiciones para garantizar el desarrollo de este principio. Y en la reglamentación correspondiente establecerá los plazos, términos y prescripciones, no solamente para la instalación de las infraestructuras indicadas y necesarias, sino también para mantener actualizadas y con la información completa los medios y los instrumentos tecnológicos.

#### **Artículo 71 de la constitución política de Colombia**

La búsqueda del conocimiento y la expresión artística son libres. Los planes de desarrollo económico y social incluirán el fomento a las ciencias y, en general, a la cultura. El Estado creará incentivos para personas e instituciones que desarrollen y fomenten la ciencia y la tecnología y las demás manifestaciones culturales y ofrecerá estímulos especiales a personas e instituciones que ejerzan estas actividades. (Colombia, 1991).



## **Capítulo 3. Diseño Metodológico**

### **3.1 Metodología de Desarrollo de Software**

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir de sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo”.

Para el desarrollo de este proyecto se tuvo en cuenta la metodología de desarrollo ágil SCRUM, la cual comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo en bloques temporales o cortos, cada bloque debe proporcionar un resultado completo.

### **3.2 Tipo de Investigación**

El tipo de investigación que se utilizará en el desarrollo del Algoritmo para la asignación de horarios académicos, corresponde a una investigación aplicada, de tipo descriptivo, ya que explora las técnicas de optimización requeridas para un problema real y describe la mejor solución aplicable a un entorno y unas condiciones particulares. El proceso investigativo abordará un enfoque cuantitativo.

El presente estudio aborda una solución a un problema que afecta a la totalidad de las personas de la facultad de Ingenierías de la Universidad Francisco de Paula Santander Ocaña (directores de plan de estudios, docentes), debido a que ellos son partícipes en la asignación de

los horarios académicos; teniendo en cuenta que esta población es muy grande se seleccionará una muestra de materias y docentes de diferentes semestres de la facultad que permita entender el problema y poder formular una solución algorítmica.

Para el presente estudio se seleccionó como muestra a: las personas relacionadas con el proceso de asignación de horarios (directores de programa) para algunas de las materias del nuevo pensum de los tres programas de pregrado de la facultad de ingeniería (Sistemas, Civil y Mecánica); un subgrupo representativo de profesores de estos programas.

En el caso concreto de este trabajo, el proceso se dividió en 4 actividades importantes. Las actividades son las siguientes:

**Identificación y planteamiento del problema:** se requiere identificar el problema para identificar su alcance y establecer los objetivos que darán solución al problema. En este trabajo, la descripción del problema comprende los problemas que se generan en la asignación de horarios de clases y como utilizar técnicas de inteligencia artificial como los algoritmos genéticos para construir un prototipo que genere horarios de manera automática utilizando estos algoritmos.

**Creación del estado del arte:** Dentro de esta actividad se realiza una revisión de las propuestas existentes, para analizar el problema, conocer lo que está hecho y como lo desarrollaron y proponer nuestra solución. La revisión del estado del arte realizada en este

trabajo comprende las técnicas de inteligencia artificial aplicadas para la solución de problemas de asignación de horarios, teoría de algoritmos genéticos, complejidad del problema *timetabling*.

**Definición de la propuesta:** En esta actividad se hace la formulación de las reglas del algoritmo genético que son las restricciones generales que presentan todos los problemas de asignación horaria.

**Construcción y pruebas del prototipo:** Dentro de esta actividad se realiza lo que son las interfaces del prototipo, construcción de la base de datos, integración del algoritmo genético, ingresar la información a la base de datos y las pruebas al prototipo.

**Publicación de los resultados:** con los resultados obtenidos y su evaluación, se inicia el proceso de divulgación.

## Capítulo 4. Resultados y Discusión.

### 4.1 Descripción Del Problema

La elaboración de un horario de clases consiste en la asignación de un docente, un lugar físico en el cual realizar las clases y el periodo de tiempo. Para entender el proceso de asignación de horarios de la facultad de ingenierías de la Universidad Francisco de Paula Santander seccional Ocaña (en adelante UFPSO) es necesario mencionar algunos términos y conceptos.

La UFPSO cuenta con dos sedes, algodonal y primavera, en las cuales se dictan las clases de la facultad de ingenierías.

#### **Sede Algodonal.**

**Tabla 3**

*Recursos físicos sede algodonal UFPSO*

Cantidad de salones	Capacidad Estudiantes	Tipo de salón
21	40	Teoría
19	40	Teoría
6	30	Sala de computación
5	30	Laboratorio
2	30	Dibujo

**Nota Fuente.** Autor del proyecto.

#### **Sede primavera**

**Tabla 4**

*Recursos físicos sede primavera UFPSO.*

Cantidad de salones	Capacidad Estudiantes	Tipo de salón
3	40	Teoría
8	30	Laboratorio

Nota Fuente. Autor del proyecto.

En estas dos sedes la facultad asigna las clases de los tres programas de ingeniería.

Cada carrera cuenta con un pensum que define las asignaturas que los alumnos de la carrera deben cursar. Según esto, cada materia está asociada a un semestre en el cual cada alumno le corresponde cursar.

La UFPSO divide el año académico en dos semestres y los horarios de las clases son de lunes a viernes y se dividen en 16 bloques de horarios de 60 minutos cada uno. Los sábados se dictan clases en 6 bloques de 60 minutos también.

Cada semestre se planifica en base a la proyección de matrícula de alumnos, esto permite saber la cantidad de cursos necesarios para cada materia de cada carrera. También se define los requerimientos que las materias solicitan, por ejemplo, el espacio físico donde esta materia se va a dictar debe ser un salón teórico, una sala de computación, un laboratorio o una sala de dibujo.

La infraestructura se encuentra clasificada de tal forma que se sabe el número de salones, salas de cómputo, laboratorios y salas de dibujo que existen en cada sede de la Universidad. En

cuanto a los docentes estos pueden indicar cuál es su horario de preferencia para dictar sus clases.

#### 4.2 Organización y Representación de los Datos de Entrada

Ahumada (2014)“Para el diseño de un AG es muy importante la representación de los datos, no solo de la solución o de los cromosomas, sino también de los datos de entrada.”(p.43).

Los datos se encuentra en una base de datos creada por el autor del proyecto llamada ces.

#### **Grupos**

Se define un grupo como la cantidad necesaria de cursos de una materia a crear para satisfacer la cantidad de estudiantes que van a cursar dicha materia.



**Figura 6. Representación de la tabla grupos en la base de datos.**

**Nota Fuente.** Autor del proyecto.

Donde:

idGrupo: identificador único en la tabla. Sirve para identificar el grupo dentro del AG.

Nombre: nombre que reciben los grupos (A, B, C, D...) de las materias.

Las restricciones asociadas a los grupos son las siguientes:

- A cada grupo se le debe asignar un docente que pueda impartir la materia.
- A cada grupo se le debe asignar un salón que cumpla los requerimientos de la materia.

El algoritmo del prototipo CES Educativo define la cantidad de grupos a crear para cada materia dependiendo del número de estudiantes matriculados en esa materia y la capacidad del salón o espacio físico donde esa materia se va a dictar, por ejemplo, la materia cálculo diferencial asignada a un salón teórico con capacidad de 40 estudiantes, va a ser cursada por 120 estudiantes, como no se puede tener un solo grupo con esa cantidad de estudiantes, se crean tres grupos de cálculo diferencial, cálculo diferencial A, cálculo diferencial B y cálculo diferencial C.

### **Sede**

Se define la sede como el espacio físico al cual pertenecen los salones que posee la UFPSO para asignar clases.



**Figura 7.** Representación de la tabla sede en la base de datos.

**Nota Fuente:** Autor del proyecto.

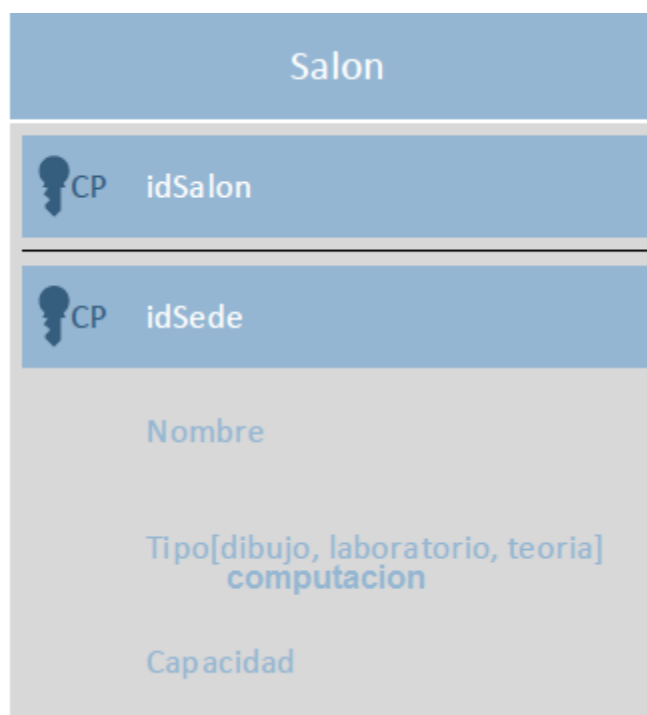
Donde:

idSede: identificador único de la tabla. Sirve para identificar la sede dentro del AG.

Nombre: nombre que recibe la sede por parte de la UFPSO.

### Salones

Los Salones se definen como el espacio físico existente en una sede perteneciente a la UFPSO para dictar una asignatura.



**Figura 8.** Representación de la tabla de salón en la base de datos.

**Nota Fuente.** Autor del proyecto.

Donde:

idSalon: identificador del salón. Sirve para identificar el salón dentro del AG.



idSede: identificador único de pertenencia de sede del salón.

Nombre: Nombre del salón de clases.

Tipo: selecciona el tipo de salón (dibujo, laboratorio, teoría o computación).

Capacidad: cantidad de estudiantes que pueden asignarse al salón.

Las restricciones asociadas a los salones son las siguientes:

- No se puede asignar más de un grupo de clase a un mismo salón en el mismo bloque horario.

## Docentes

Se define a los docentes como el personal de la UFPSO encargado de dictar las materias del pensum de las carreras de la facultad de ingenierías.



**Figura 9.** Representación de la tabla docente en la base de datos.

**Nota Fuente.** Autor del proyecto.

Donde:

idDocente: identificador único en la tabla. Sirve para identificar un docente dentro del AG.

idCategoriaDocente: identificador de la categoría a la que pertenece el docente (Tiempo completo o contratado).

Nombres: nombre del docente.

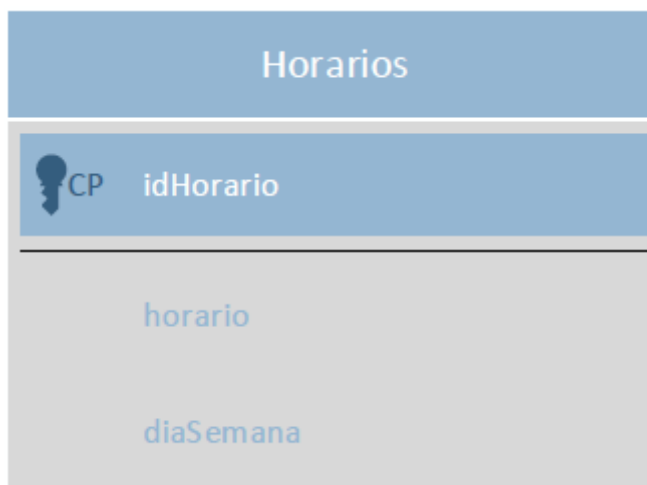
Apellidos: apellidos del docente.

Las restricciones asociadas a los docentes son las siguientes:

- No se puede asignar un docente a más de un grupo en el mismo horario.
- No se puede asignar un docente a un horario que no tenga disponible.
- No se puede asignar docente a materias que no tenga asignadas.

## Horarios

Se define los horarios como la cantidad de bloques en los cuales se puede dictar clases repartidos en seis días de la semana.



**Figura 10. Representación de la tabla horarios en la base de datos.**

**Nota Fuente.** Autor del proyecto.

Donde:

idHorario: identificador único de la tabla. Sirve para identificar un horario dentro del AG.

Horario: bloques del día que se pueden dictar clases. Sirve para almacenar los números de bloques que se pueden asignar ese día, siendo 16 si es de lunes a viernes y de 6 los sábados.

diaSemana: valor de la semana. Sirve para almacenar los días de la semana.

### **Carreras**

Se definen carreras como el conjunto de materias que conforman un pensum académico de la UFPSO.



**Figura 11. Representación de la tabla carreras en la base de datos.**

**Nota Fuente.** Autor del proyecto.

Donde:

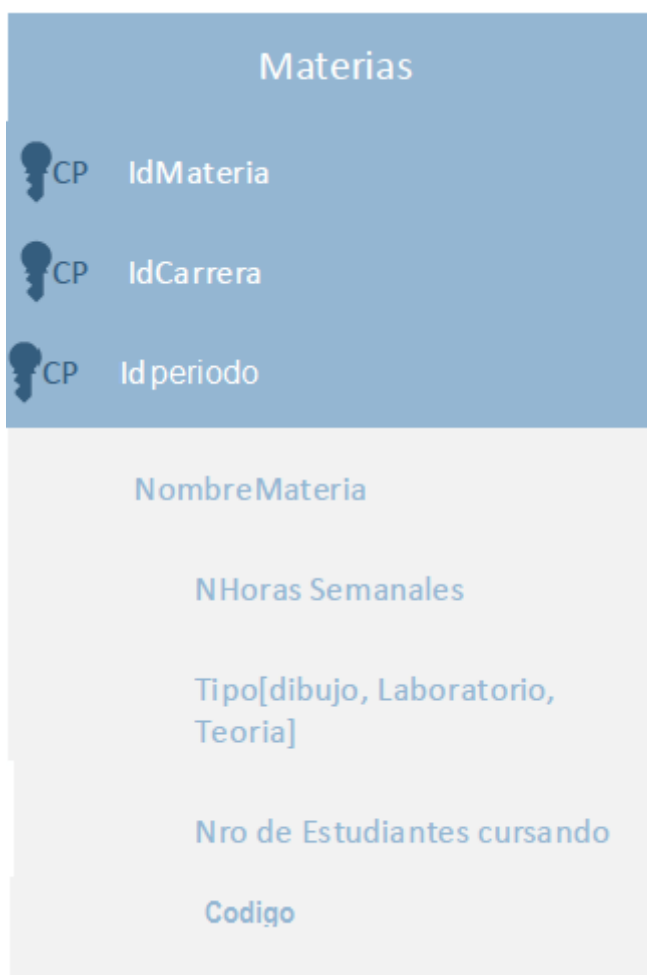
idCarrera: identificador único en la tabla. Sirve para identificar una carrera dentro del AG.

idSede: identificador único en la tabla. Sirve para identificar la sede en la que se dicta la carrera dentro del AG.

Nombre: Nombre que recibe la carrera dentro de la facultad de ingenierías de la UFPSO.

### **Materias**

Se definen las materias como las asignaturas pertenecientes al pensum académico de una carrera universitaria de la UFPSO.



**Figura 12. Representación de la tabla materias en la base de datos.**

**Nota Fuente.** Autor del proyecto.

Donde:

idMateria: identificador único en la tabla. Sirve para identificar una materia dentro del AG.

idCarrera: identificador único en la tabla. Sirve para identificar a que carrera pertenece la materia dentro del AG.

idPeriodo: identificador único en la tabla. Sirve para identificar a que semestre pertenece la materia dentro del AG.

NombreMateria: Nombre que recibe la materia en el pensum de la carrera.

NHorasSemanales: valor que recibe del número de horas que se dicta esa materia semanalmente.

Tipo: valor que recibe sobre la preferencia de la materia en cuanto a salón (laboratorio, teoría, computo o dibujo).

NEstudiantesCursando: valor que recibe el número de estudiantes cursando la materia. Sirve para definir el número de grupos a crear de la materia dentro del AG.

Codigo: Código asignado a la materia en el pensum de la carrera de la UFPSO.

#### **4.3 Organización Y Representación De Estructuras De Validación**

Además de las tablas que almacenan los datos de entrada, se definen una serie de estructuras para ser consultadas por la función de evaluación. Estas estructuras ayudan al cumplimiento de algunas restricciones.

A continuación se describen las relaciones entre la información de la base de datos.

### Combinación docentes-horario

Se definen las combinaciones validas entre docentes y horario, esta estructura se basa en la disponibilidad horaria que se le asignó a cada docente.



**Figura 13. Representación de la tabla horario docentes en la base de datos.**

**Nota Fuente.** Autor del proyecto.

Donde:

idHorario: identificador único de la tabla. Sirve para identificar el horario dentro del AG.

idDocente: identificador único de la tabla. Sirve para identificar el docente dentro del AG.

Horarioinicial: Valor de la hora exacta donde empieza la disponibilidad del docente.

Horariofinal: Valor de la hora donde termina la disponibilidad del docente.

Diasemana: Valor del día de la semana que se le asignan las horas disponibles.

La siguiente figura muestra un ejemplo del horario que tiene disponible un docente de la UFPSO. Estos datos fueron creados para realizar las pruebas en el prototipo CES Educativo.

id	Día	H. Inicial	H. Final
4	Lunes	07:00	12:00
5	Miercoles	09:00	11:00
6	Jueves	15:00	22:00
16	Viernes	07:00	22:00
17	Sabado	07:00	13:00

**Figura 14.** Horario disponible del docente utilizado por el prototipo CES.

**Nota Fuente.** Autor del proyecto.

La información de esta tabla permite al AG jugar con las horas disponibles y asignar un curso a dictar dentro de las mismas. La asignación de buenas horas disponibles para docentes ayuda a generar buenas soluciones finales de horarios de clases.

### **Combinación Grupos-Materia**

Se define el conjunto de combinaciones validas entre grupos y materias. Esta estructura permite almacenar la cantidad de grupos que se asignan a una materia y la cantidad de estudiantes por grupo.



**Figura 15. Representación de grupos por materia en la base de datos.**

**Nota Fuente:** Autor del proyecto.

Donde:

idGrupoMateria: Identificador único de la tabla. Sirve para identificar el grupo de la materia dentro del AG.

idMateria: Identificador único de la tabla. Sirve para identificar la materia a la cual se le creo un grupo dentro del AG.

idGrupo: identificador único de la tabla. Sirve para identificar el grupo seleccionado de la tabla grupos dentro del AG.



La siguiente figura 16 muestra un ejemplo de asignación de grupos para una materia de ingeniería de sistemas en su primer semestre.

The screenshot shows the 'CES EDUCATIVO V1.0' interface. At the top, there are filters for 'Sede' (Algodonal) and 'Semestre' (I). Below these are buttons for 'Docente', 'Carreras', 'Salones', and 'Reporte'. The main area is a table with columns for 'Lista de Carreras', 'Grupo', 'Materia', 'Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado', 'Profesor', 'Id', and 'Lista de Horarios'. The 'Lista de Carreras' column has a dropdown menu with 'Ing. SISTEMAS', 'Ing. MECANICA', and 'Ing. CIVIL' selected. The table shows three groups (A, B, C) for the 'Ing. SISTEMAS' program, each with a specific schedule and professor.

Lista de Carreras	Grupo	Materia	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Profesor	Id	Lista de Horarios
Ing. SISTEMAS	A	FUNDAMENTOS DE PR...	09:00 09:00 COMPUTO ...				08:00 11:00 COMPUTO ...		BADWIN	101	Horario 101
Ing. MECANICA	B	FUNDAMENTOS DE PR...			14:00 16:00 COMPUTO ...	07:00 09:00 COMPUTO ...			SIR ALEXICI	99	Horario 99
Ing. CIVIL	C	FUNDAMENTOS DE PR...		13:00 15:00 COMPUTO S...			14:00 17:00 COMPUTO ...		SIR ALEXICI	97	Horario 97

**Figura 16.** Vista de asignación de grupos del prototipo CES.

**Nota Fuente:** Autor del proyecto.

El AG del prototipo generó tres grupos de la materia fundamentos de programación de la carrera de sistemas teniendo en cuenta:

- Cantidad de estudiantes: El número de estudiantes que se proyectó van a cursar esa materia.
- Capacidad del tipo del salón: El salón elegido para dictar la materia fundamentos de programación es un salón de sala de cómputo el cual tiene una capacidad de 30 estudiantes.

Al generarse tres grupos de esta materia podemos afirmar que la cantidad de estudiantes proyectados a cursar esta materia es mayor a 60 pero menor a 90, generando grupos menores o iguales a 30 estudiantes.

## Combinación Docentes-Materia

Se define el conjunto de combinaciones válidas entre docentes y materias. Esta estructura permite asignar un docente a una materia. El AG asigna entonces la materia en las horas disponibles del docente.



**Figura 17. Representación de la tabla docente materia en la base de datos.**

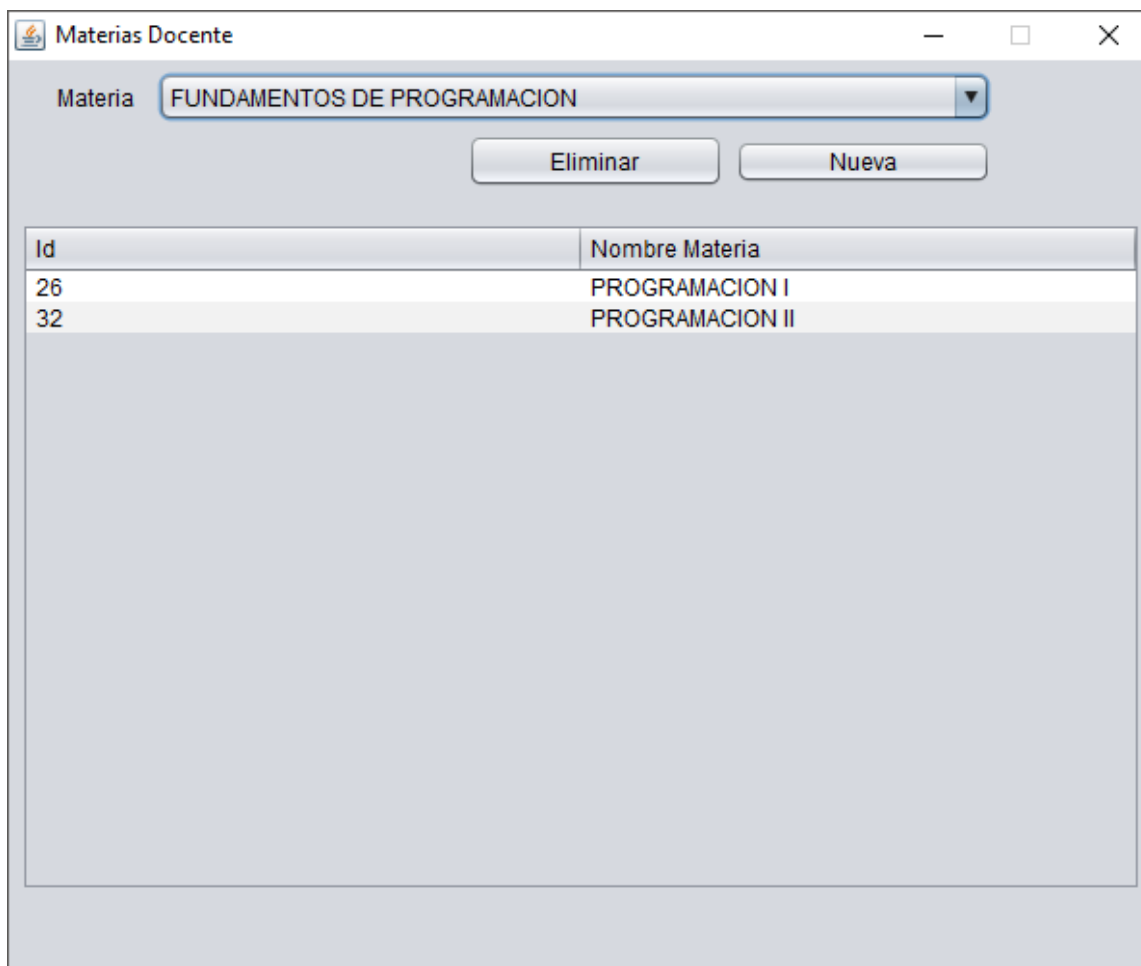
Nota Fuente: Autor del proyecto.

Donde:

idMateria: Identificador único de la tabla. Sirve como identificador de la materia dentro del AG.

idDocente: Identificador único de la tabla. Sirve como identificador del docente dentro del AG.

La representación de las combinaciones validas de docentes y materia, se muestra en la siguiente figura 18.



**Figura 18.** Vista de materias asignadas a un docente en el prototipo CES Educativo

**Nota Fuente:** Autor del proyecto.

La vista de la figura nos muestra que el docente tiene asignadas dos materias (programación I y programación II). El AG debe generar un horario de clases de estas dos materias dentro de las horas disponibles por dicho docente.

### **Disponibilidad horaria de salones**

Los salones existentes en las sedes que tiene la universidad para dictar clases, se les asigno un horario de disponibilidad que corresponde a la tabla horarios. La disponibilidad de los salones

es de lunes a viernes de 6:00 – 22:00 y los sábados de 6:00 – 12:00, cumpliendo con los 16 bloques de 60 minutos que se dictan por día y los 6 bloques que se dictan los sábados en la UFPSO.

#### 4.4 Estructura de Solución o Cromosoma

Ahumada (2014)“La codificación del cromosoma es uno de los factores más influyentes en el rendimiento de un algoritmo genético.”(p.56).

En el diseño del algoritmo genético se optó por un cromosoma no binario, facilitando la comprensión del problema, debido a que los datos involucrados se representan mediante variables no homogéneas (de diferente tipo y longitud). La relación [grupo-materia, docente, horarios (hora)] que indica la materia a impartir, el profesor titular y la hora respectivamente, conforman así mismo un gen del cromosoma a formar. Por ejemplo, la combinación [A1, 10, 1] indica que la materia 1 (fundamentos de programación) del grupo A, será impartida por el docente 10(Badwin Arévalo Vera) en la hora 1(6:00 – 7:00) de un día determinado.



**Figura 19. Representación no binaria de un gen.**

**Nota Fuente:** Autor del proyecto.

Sin embargo, para indicar la distribución completa de todos los grupos-materias de un semestre, es necesario concatenar un gen por cada grupo-materia requerido en ese semestre, formando un cromosoma como el que se muestra a continuación. En él se aprecia un cromosoma

de un semestre (semestre 1 al 10), la lista de genes que lo conforman, número de puntos asignados por horas libres, número de puntos asignados por la unión de todas las clases de un mismo grupo, el número de puntos asignados por la unión de todas las clases de un docente, el número de puntos asignados por la unión de todos los salones de un grupo. Esta fue la información que se utilizó para generar el horario de un semestre. Al final de cada cromosoma se incluye un campo denominado aptitud, que indica el valor numérico entero de la aptitud total del cromosoma. (Johnston, 2000).



**Figura 20. Representación no binaria de un cromosoma.**

**Nota Fuente.** Autor del proyecto.

#### 4.5 Definición de la Función De Evaluación

La función de evaluación es el parámetro que evalúa una solución y permite que el AG eleve su rendimiento al mejorar la aptitud de los cromosomas conforme avanza el proceso.

En el problema de la facultad de ingenierías de la UFPSO, los cromosomas serán evaluados con un conjunto de penalizaciones o castigos, teniendo en cuenta los siguientes puntos:

- Los empalmes o uniones de salones de un grupo. No deben asignarse varios grupos de clases al mismo salón en la misma hora.

- Los empalmes o uniones de clases de un grupo. No deben asignarse varias clases al mismo grupo en la misma hora.
- Los empalmes o uniones de clases de un profesor. No deben asignarse varias clases al mismo profesor en la misma hora.
- Las horas libres entre clases de un grupo. No es deseable periodos inactivos en la secuencia de las clases del grupo.

Estos puntos sirven para determinar la aptitud de cada cromosoma dentro de un conjunto de posibles soluciones y son los parámetros tenidos en cuenta al momento de evaluar las soluciones. (Johnston, 2000).

El AG del prototipo funciona atacando un problema de minimización, en donde el objetivo es encontrar una solución o cromosoma cuya aptitud óptima sea cero, es decir, hallar el cromosoma que no presente penalizaciones (o el menor número posible). Se evalúan los cromosomas tomando en cuenta las siguientes penalizaciones:

- 4 puntos por cada empalme o unión de salones de un grupo.
- 3 puntos por cada empalme o unión de clases de un mismo grupo.
- 2 puntos por cada empalme o unión de clases de un profesor.
- 1 punto por cada hora libre de un grupo.

Johnston (2000) La aptitud de un cromosoma es inversamente proporcional a la suma de las penalizaciones mencionadas, es decir, los cromosomas con el menor número de penalizaciones son los más aptos, además, estos valores reflejan directamente el orden de importancia en la aplicación de tales penalizaciones a los cromosomas, lo que significa que es más importante

tratar de evitar las uniones de salones de un grupo que las uniones de clases de un mismo grupo y que las horas libres que puedan presentar una distribución de [grupo-materia, docente, horario].

#### 4.6 Preparación de Datos de Entrada

Corresponde a la extracción de datos desde la base de datos. En esta etapa se realizan las estructuras definidas en la sección anterior, organización y representación de estructuras de validación. Esta información es la planificación de los grupos a crear por cada materia, la disponibilidad en los horarios de los docentes, la disponibilidad de los salones de la UFPSO y los pensum académicos.

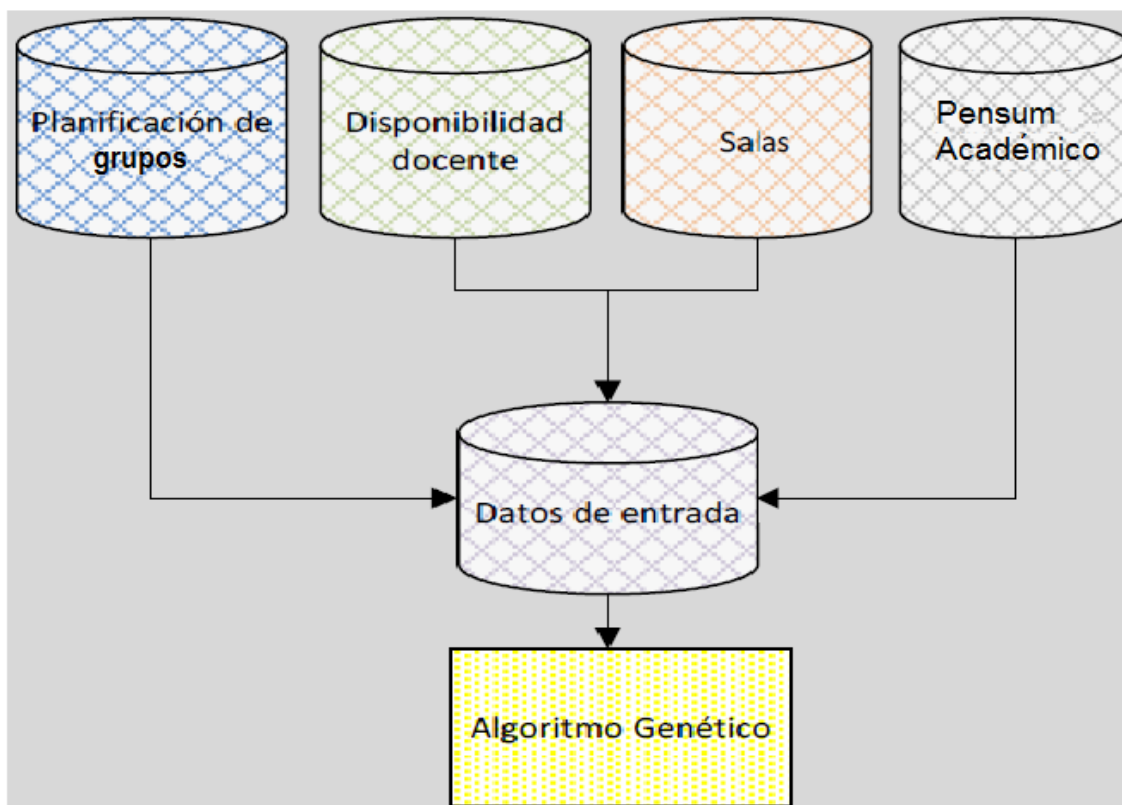
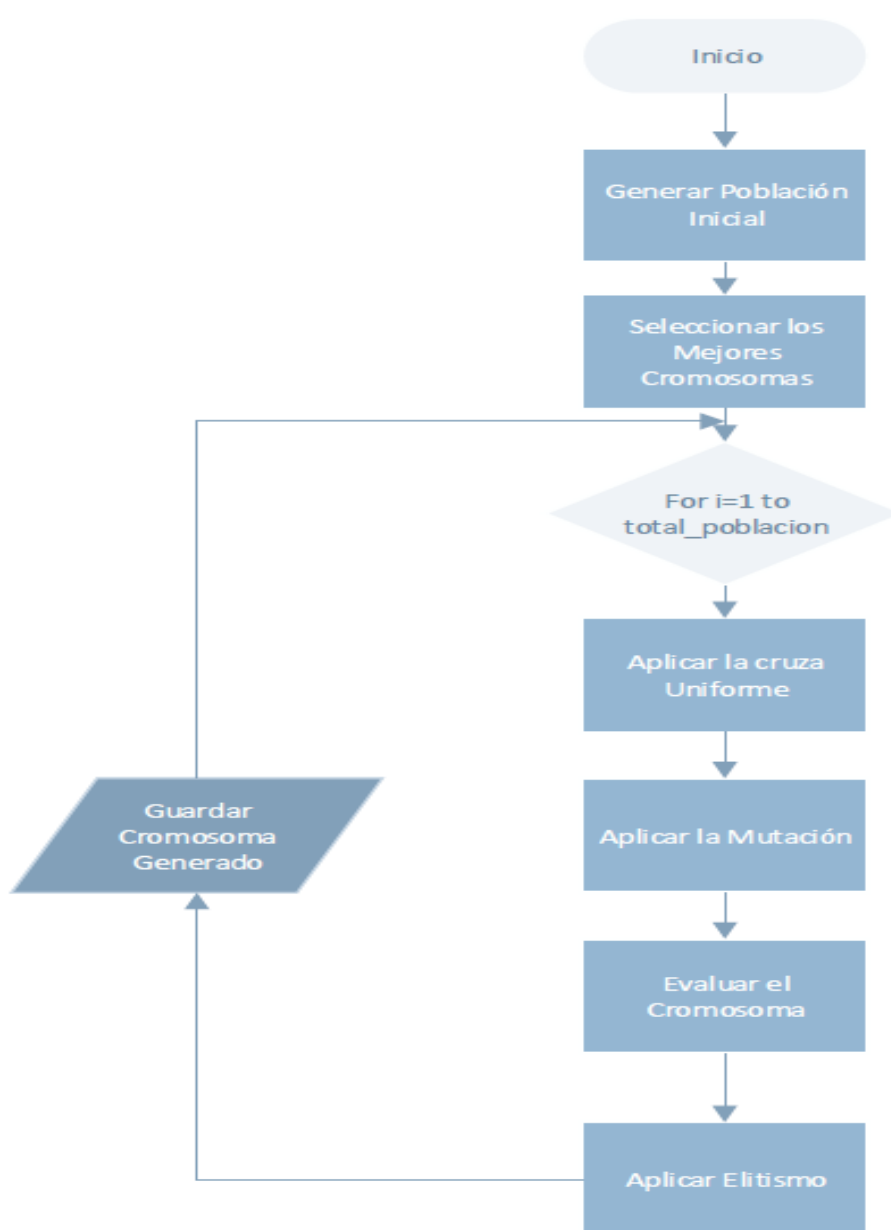


Figura 21. Datos de entrada.

Nota Fuente: (Ahumada, 2014).

## 4.7 Algoritmo

A continuación la figura 22 muestra el diagrama de flujo de la estructura principal del algoritmo genético desarrollado.



**Figura 22. Diagrama de flujo algoritmo genético.**

**Nota Fuente:** Autor del proyecto.



#### 4.7.1 Generación de la población inicial

La primera etapa de la ejecución de un AG es generar la población inicial, en la cual se van a generar  $n$  cromosomas de forma aleatoria, en algunos casos se aplican otras meta heurísticas para generar la población inicial, mejorando considerablemente la calidad de la población. Para nuestro caso se ha seleccionado una forma aleatoria.

Por razones de espacio de almacenamiento de los cromosomas generados estos cromosomas de la población inicial se almacenan en archivos temporales. Los cromosomas generados en la población inicial se almacenan secuencialmente y se van formando con la selección aleatoria de un profesor disponible por grupo-materia. Ver figura 37 del diagrama de flujo de generación de la población inicial en la sección de apéndice.

Con la generación de la población inicial con este método aleatorio se cumple lo siguiente:

- Los grupos-materias tengan asignado un maestro a una hora de clase.
- Las materias sean impartidas por docentes cuyo perfil académico sea el apropiado, es decir, los docentes tengan asignada esa materia para dictar.
- Las horas de clase de los maestros se encuentren dentro de su disponibilidad de horario.
- Las materias tengan asignado un salón, sea de teoría, dibujo, laboratorio o sala de computo.

### 4.7.2 Operadores genéticos

Una vez se genera la población inicial, se procede a aplicar los operadores genéticos de selección, cruce, mutación y elitismo para generar nuevas poblaciones, tal como se indica en la figura diagrama de flujo del AG.

#### *4.7.2.1 Selección de los mejores cromosomas*

El tipo de selección escogido corresponde al método de la ruleta. Nuestro AG está diseñado para problemas de minimización y aplica el método de selección creando una ruleta que asigna partes proporcionales de acuerdo a la aptitud de cada cromosoma, en la cual el cromosoma con el valor de aptitud más bajo tendrá una mayor área de la ruleta.

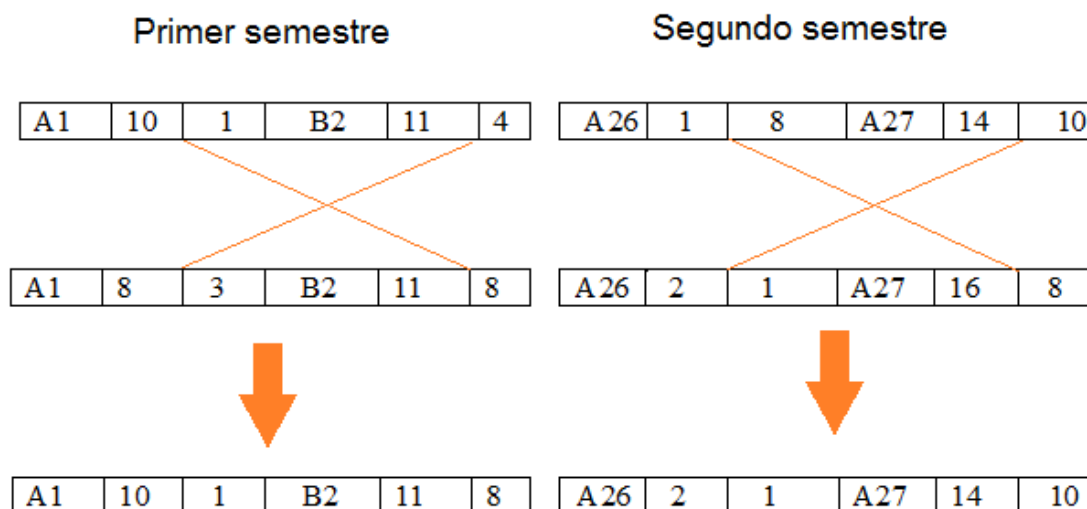
La ruleta se gira  $n$  veces para generar la siguiente población, cuyos cromosomas seleccionados se almacenan temporalmente. Ver figura 38 del diagrama de flujo de operador selección en la sección apéndice.

#### *4.7.2.2 Cruce*

El tipo de cruce escogido corresponde a una variante del operador cruce llamado cruce uniforme, el cual consiste en aplicar con cierta probabilidad el operador a bloques específicos del cromosoma, es decir, se aplica el cruce uniforme solo a semestres correspondientes de dos cromosomas seleccionados. Aleatoriamente se seleccionan dos cromosomas de la población actual, y a cada semestre del cromosoma se le aplica la probabilidad de cruce para determinar si

se efectúa la operación, en caso de ser así, se cruzan las horas donde se asignaron a dictar correspondientes, después se evalúan y finalmente se almacenan. (Johnston, 2000).

La figura muestra el funcionamiento de este operador. En ella se muestran dos cromosomas seleccionados aleatoriamente para aplicarles el operador, en los cuales se aprecian dos semestres que a su vez tienen dos genes cada uno. Con este operador, se cruza únicamente los genes que pertenezcan a los mismos semestres, es decir, solo se cruzan los genes del semestre 1 del primer cromosoma con los genes del mismo semestre del segundo cromosoma seleccionado. Esto mismo se aplica con los genes de los semestres siguientes.



**Figura 22. Ejemplo cruce uniforme.**

**Nota Fuente:** Autor del proyecto.

De este proceso se obtienen dos descendientes que se van a incorporar en la población a pesar de que su aptitud no sea necesariamente mejor que la de sus padres. Ver figura 39 del diagrama de flujo del operador cruce en la sección apéndice.

#### 4.7.2.3 Mutación

La operación de mutación consiste en modificar aleatoriamente algunos genes de acuerdo a un valor probabilístico, sin embargo, para el cromosoma representado en este problema particular, se cambia solamente una sección de los cromosomas, es decir, se sustituye aleatoriamente un docente por otro disponible en algunos genes. Al hacerlo, es necesario modificar también el horario de clases de esa materia, esto significa que la materia ahora se va a asignar en el horario disponible del nuevo docente. Del total de genes de la población actual, se escoge aleatoriamente un porcentaje de ellos (determinado por la probabilidad de mutación) y solamente a ellos se les cambiará el docente y el horario de clase dentro del rango que tiene disponible. (Ahumada, 2014).

En la figura 23 se muestra un ejemplo de mutación de un gen seleccionado de forma aleatoria.

Docentes disponibles para Fundamentos de Programación (A 1) y sus horarios			
Id	Nombre	Horario disponible	
8	Sir Alexci	6:00 – 10:00	1-2-3-4
9	Edinho	14:00 – 18:00	8-9-10-11-12
10	Badwin	14:00 – 18:00	8-9-10-11-12

A1	8	1
----	---	---



A1	9	8
----	---	---

**Figura 23. Ejemplo de mutación.**

**Nota Fuente:** Autor del proyecto.

La figura 40 del apéndice muestra el diagrama de flujo del operador mutación.

#### *4.7.2.4 Elitismo*

Un AG aplica una técnica de búsqueda basada en la teoría de la supervivencia de los individuos más capaces de adaptarse al medio ambiente y mejorar su rendimiento utilizando medidas heurísticas. Esto significa que conforme avanza el proceso de reproducción de los individuos, se debe mejorar constantemente la aptitud de los mismos, es decir, a medida que el AG genera nuevas poblaciones de individuos, estos deben mostrar características superiores a las de sus antecesores o en el peor de los casos, mantener el nivel de aptitud de la población anterior. (Mejia, 2008).

El elitismo es un criterio que se aplica en AG con la finalidad de mantener el mejor cromosoma de cada población insertándolo directamente en la siguiente población asegurando el nivel de aptitud del mejor cromosoma de cada generación conforme se avanza en la generación de poblaciones. (Johnston, 2000).

### **4.8 Parámetros del Algoritmo Genético**

Existen algunos parámetros importantes para el AG que deben tenerse en cuenta para mejorar los resultados obtenidos en la ejecución. Es muy importante seleccionar los parámetros adecuados, sin embargo, no existe una metodología que indique los valores exactos que deben asignarse ya que varían de acuerdo a la naturaleza del problema.

De Jong (1975) sugiere un valor alto para la probabilidad de cruza, bajo para la probabilidad de mutación y moderado para el tamaño de la población.

**Tabla 5**

*Parámetros*

<b>Parámetro</b>	<b>Mínimo</b>	<b>Máximo</b>
Tamaño de población	50	300
Probabilidad de cruce	70%	90%
Puntos de cruce	1	2
Probabilidad de mutación	0.001%	0.2%

**Nota Fuente:** (De Jong, 1975).

El criterio de terminación está dado por el número de poblaciones que se desean generar, este parámetro lo asignamos en un valor cercano a 100 debido a factores de tiempo en la ejecución del AG.

#### 4.9 Descripción del Sistema

El proyecto “ALGORITMO PARA LA ASIGNACIÓN DE HORARIOS ACADÉMICOS EN LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL” desarrollado dentro de las instalaciones de la Universidad Francisco de Paula Santander Seccional Ocaña fue diseñado de forma sencilla.

Se creó un prototipo llamado CES Educativo encargado de la asignación de horarios de clases para la facultad de ingenierías de la UFPSO, es decir, sus tres programas académicos (Ingeniería civil, Ingeniería de Sistemas e Ingeniería mecánica) utilizando AG. Desarrollado en

lenguaje JAVA, permite el ingreso de los datos descritos en este documento para ser almacenados en la base de datos Mysql mediante el lenguaje SQL.

**Tabla 6**

*Arquitectura prototipo CES Educativo*

<b>Software</b>	<b>Versión</b>
<b>Java SE Development Kit 8</b>	1.8.0_101-b13
<b>Myql server 5</b>	5.6.12 MySQL Community Server (GPL)
<b>Eclipse Java EE IDE</b>	Luna Service Release 2 (4.4.2)
<b>Sistema Operativo</b>	Linux o Windows

**Nota Fuente:** Autor del proyecto.

Los programas locales:

- Java jre 1.8.
- Mysql server 5.6 o superior.
- Jar ejecutable CES Educativo.

Entradas y salidas

- Ingreso de datos de docentes, materias, salones y carreras a través de formularios.
- Datos almacenados en base de datos MySql.
- Reportes en Excel.

## Diseño de interfaz

A continuación se muestra el diseño de interfaces del prototipo.

### Pantalla de Bienvenida:

Esta pantalla indica que el software CES está cargando todas las librerías y componentes necesarios para iniciar.

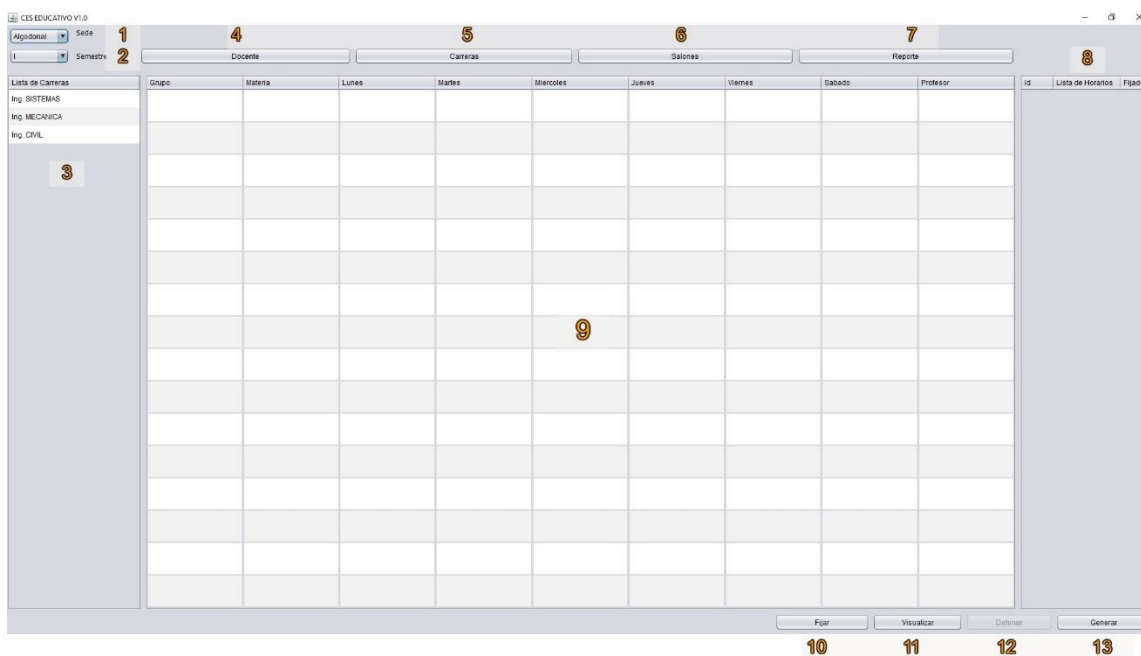


Figura 24. Pantalla de Bienvenida.

**Nota Fuente:** Autor del Proyecto.



**Pantalla primera interfaz:** Es la pantalla a la cual se accede una vez cargados los componentes necesarios para la ejecución del prototipo, en esta primera interfaz encontramos el menú a través del cual se pueden acceder a diferentes gestiones que puede hacer el prototipo.



**Figura 25. Interfaz principal del prototipo CES.**

**Nota Fuente:** Autor del proyecto

En esta interfaz encontramos los siguientes campos:

1. Sedes: Muestra la sede donde se generan los horarios.
2. Selector de Semestres: Permite seleccionar entre el primer y el último semestre.
3. Tabla lista de Carreras: Muestra la lista de las carreras cargadas en el sistema y permite seleccionar una.
4. Botón Docente: Abre una ventana que gestiona los datos del Docente.
5. Carreras: Gestiona las carreras de la facultad de Ingenierías.

6. Salones: Gestiona los salones a utilizar.
7. Reportes: Permite exportar los horarios a Excel
8. Lista de horarios: Vista de los horarios generados.
9. Vista horario: Muestra la asignación de los horarios.
10. Fijar: Fija el horario seleccionado en la lista de horarios.
11. Visualizar: Carga el horario seleccionado a la vista de horario.
12. Detener: Cancela la ejecución del AG.
13. Generar: Ejecuta el algoritmo para generar horarios.

**Pantalla Interfaz Docente:** En esta interfaz docente encontramos la totalidad de los docentes almacenados en la base de datos, también podemos gestionar (Agregar, buscar, modificar y eliminar), agregar las materias que va a dictar dicho docente, agregar la disponibilidad horaria y finalmente podemos agregar una nueva categoría docente (categorías existentes: tiempo completo y contratado).

**1. Campos de la Tabla Docente**

Nombre  Telefono

Apellidos

e-mail  Categoría Crontratado **2**

**3** Materias Docente **4** Disponibilidad Horario **5** Insertar nueva categoria

**6** Buscar **7** Eliminar **8** Modificar **9** Nuevo

Id	Nombres y Apellidos	Categoría	Estudios
1	ALVEIRO ALONSO ROSADO GOM...	Crontratado	
2	MAGRETH ROSSIO SANGUINO R...	Crontratado	
3	LAURA PATRICIA SANTIAGO ARE...	Crontratado	<b>Docentes almacenados</b>
4	HARVEY ENRIQUE CRIADO ANG...	Tiempo Completo	
5	LUIS ADOLFO OROZCO CARVAJA...	Crontratado	
6	LUIS EDUARDO PALACIO RUEDA	Tiempo Completo	
7	EDWIN BARRIENTOS AVENDANO	Crontratado	
8	SIR ALEXCI SUAREZ CASTRILLON	Tiempo Completo	
9	EDINHO REINALDO DIAZ BUSTOS	Tiempo Completo	
10	BADWIN AREVALO VERA	Crontratado	
11	OLGA LUCIA REYES PENARANDA	Tiempo Completo	
12	MONICA LILIANA VERGEL MOGOL...	Crontratado	
13	EDUAR BAYONA IBANEZ	Crontratado	
14	YEGNY KARINA AMAYA TORRADO	Crontratado	
15	JORGE ERNESTO CHAVES PENA	Crontratado	
16	ORLANDO AREVALO ASCANIO	Crontratado	
17	LUCENITH LEHERICITH GALAN	Crontratado	
18	MARIA DEL PILAR PEREZ CASADI...	Crontratado	
19	MIGUEL ANTONIO MUNOZ NORIE...	Crontratado	
20	DIANA MARCELA RODRIGUEZ AR...	Crontratado	
21	YEIGNE PEREZ CASADIEGO	Crontratado	
22	CARLOS DANIEL ECHAVEZ VERG...	Crontratado	
23	LINA FERNANDA PEREZ CASADI...	Crontratado	
24	ALBERTO EMIRO MONTANO GOM...	Crontratado	
25	RODRIGO PENARANDA JACOME	Crontratado	
26	JORGE HUMBERTO OVALLOS MA...	Crontratado	
27	SAID HUMBERTO JAIME BARBOSA	Crontratado	

**Figura 26. Interfaz docente del prototipo CES.**

**Nota Fuente:** Autor del proyecto

4.1. Campos de la tabla docente: Los campos de la tabla docente de la base de datos

4.2. Categoría: Permite seleccionar la categoría de los docentes que se van a almacenar (tiempo completo, contratado).

4.3. Materias Docente: Permite asignar o eliminar materias.

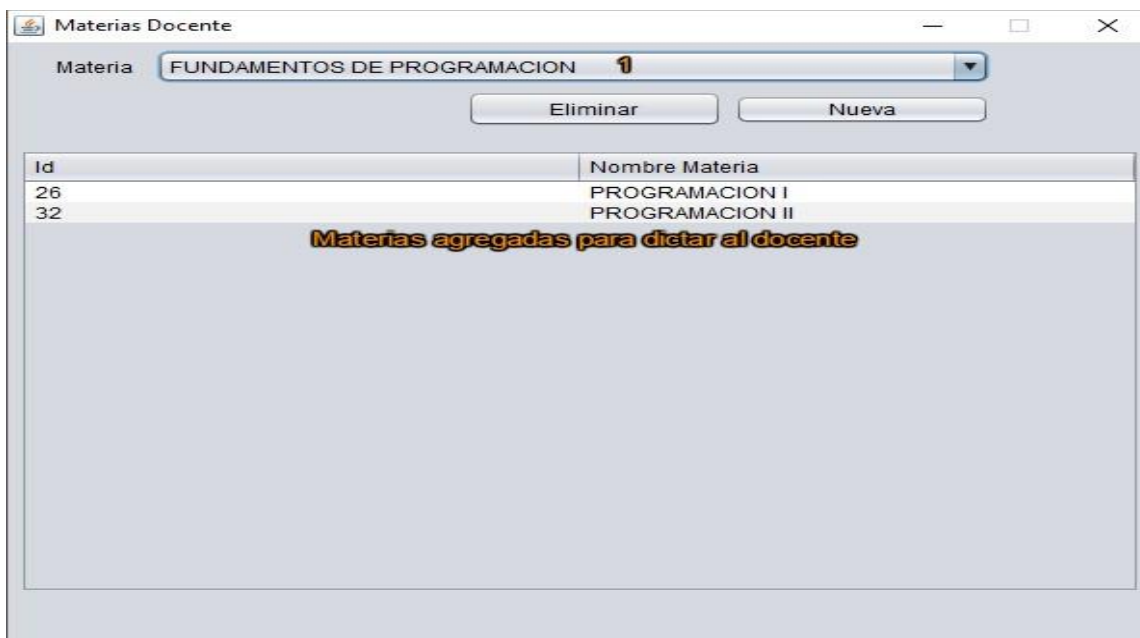
4.4. Disponibilidad Horario: Permite gestionar el horario disponible de los docentes en la semana.

4.5. Insertar nueva categoría: Inserta una nueva categoría para los docentes.

4.6. Buscar: Permite realizar la búsqueda de docentes registrados en la base de datos, digitando su nombre en los campos y dando clic en buscar.

4.7, 4.8 y 4.9. Gestionar Docentes: Permite agregar, modificar y eliminar los docentes de la base de datos.

**Pantalla Interfaz Materia Docente:** En esta interfaz se pueden agregar o eliminar las materias a dictar a un docente



**Figura 27.** Interfaz materia docente del prototipo CES.

**Nota Fuente:** Autor del proyecto.

**Pantalla Interfaz Horarios:** En esta interfaz se puede gestionar la disponibilidad horaria de los docentes seleccionando el día de la semana y a continuación la hora de inicio y la hora fin disponible presionando el botón nuevo. Para eliminar y modificar selecciona el horario de la tabla de abajo.



The screenshot shows a window titled "Horarios" with a standard Windows-style title bar. Below the title bar, there are three dropdown menus: "Día de la Sema..." (set to "Lunes"), "Hora de Inicio" (set to "06:00"), and "Hora de Fin" (set to "07:00"). Below these are three buttons: "Eliminar", "Modificar", and "Nuevo". Underneath the buttons is a table with the following data:

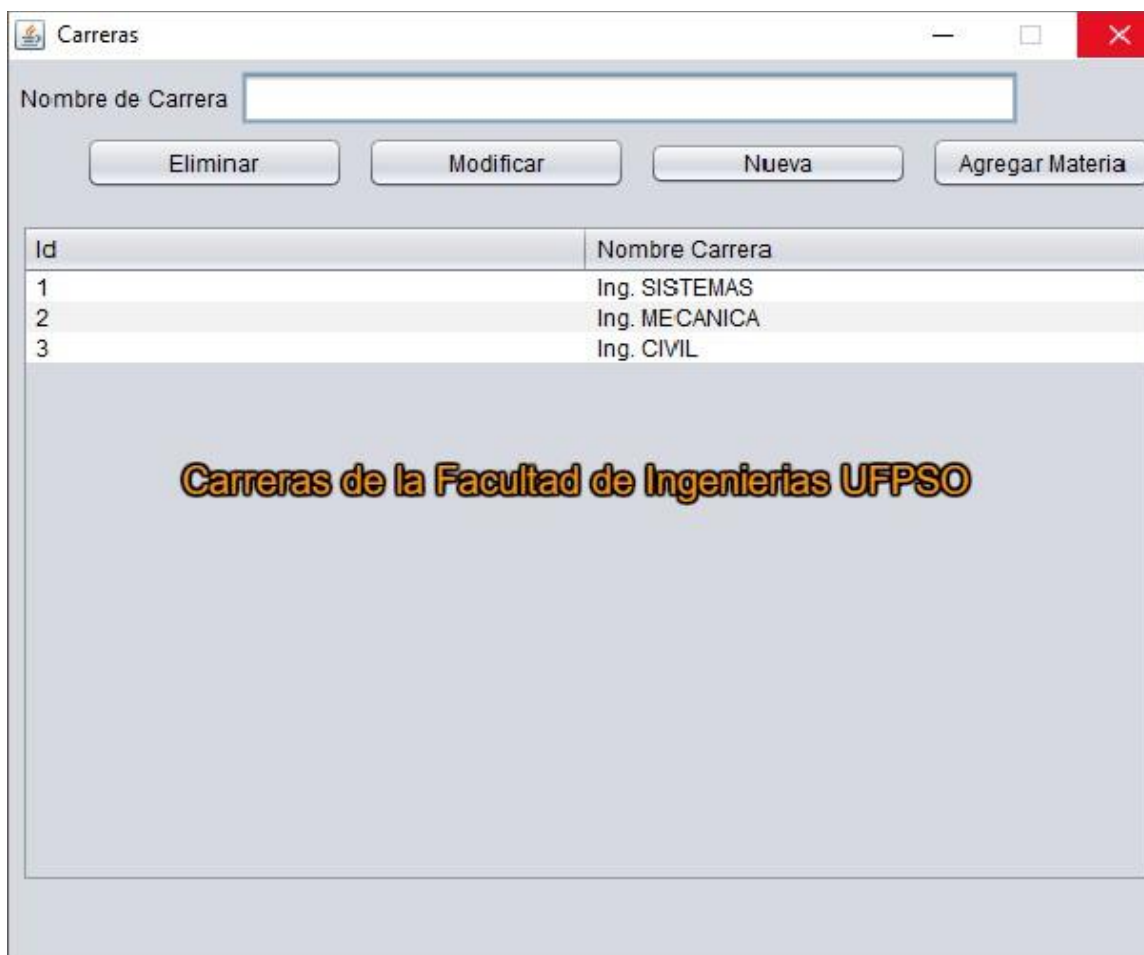
id	Día	H. Inicial	H. Final
4	Lunes	07:00	12:00
5	Miercoles	09:00	11:00
6	Jueves	15:00	22:00
16	Viernes	07:00	22:00
17	Sabado	07:00	13:00

Below the table, there is a large text area with the text "Disponibilidad en la semana del docente seleccionado" in a bold, orange, outlined font.

**Figura 28. Interfaz Horario del prototipo CES.**

**Nota Fuente:** Autor del proyecto

**Interfaz Carreras:** En esta pantalla se puede hacer la gestión de carreras: agregar, modificar y eliminar. También permite agregar las materias que conforman las carreras.



**Figura 29.** Interfaz carreras del prototipo CES.

**Nota Fuente:** Autor del proyecto.

**Interfaz Materia:** En esta pantalla se puede hacer la gestión de las materias: agregar, modificar y eliminar. Podemos seleccionar la carrera a la cual vamos a gestionar materias.

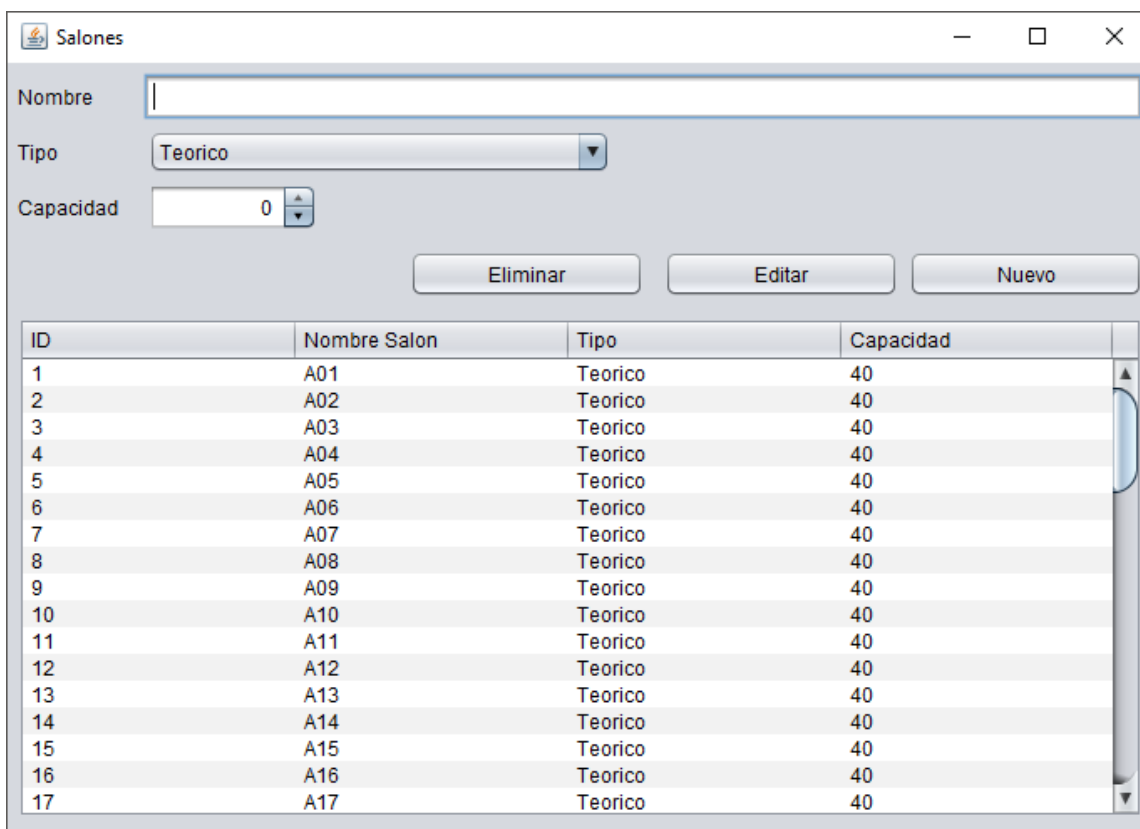
The screenshot shows a window titled 'Materia' with a form for managing subjects. The form includes a dropdown for 'Carrera' (set to 'Ing. SISTEMAS'), a text field for 'Materia', a dropdown for 'Tipo' (set to 'Teorico'), and numeric spinners for 'Horas' and 'Estudiantes' (both set to 0). There is also a text field for 'Codigo'. Below the form are three buttons: 'Eliminar', 'Modificar', and 'Nueva'.

Id	Nombre Materias	Tipo	N° Horas	N° Estudiantes	Codigo
1	FUNDAMENTOS DE PROGRAMACION	Computación	5	80	192101
2	ELECTIVA INSTITUCIONAL	Teorico	2	80	192102
3	INTRODUCCION A LA INGENIERIA	Computación	3	90	192103
16	CALCULO DIFERENCIAL	Teorico	4	120	192104
17	MATEMATICAS DISCRETAS	Teorico	3	80	192105
18	COMUNICACION	Teorico	2	80	192106
26	PROGRAMACION I	Computación	4	60	192201
27	TEORIA GENERAL DE SISTEMAS	Computación	3	30	192202
28	CALCULO INTEGRAL	Teorico	4	80	192203
29	FISICA MECANICA	Laboratorio	5	60	192204
30	ALGEBRA LINEAL	Teorico	3	80	192205
31	ELECTIVA I	Teorico	2	40	192206
32	PROGRAMACION II	Computación	4	30	192207

**Figura 30.** Interfaz materia del prototipo CES.

**Nota Fuente:** Autor del proyecto.

**Interfaz Salones:** En esta pantalla se puede hacer la gestión de los salones: agregar, modificar y eliminar.



The screenshot shows a window titled "Salones" with a form for adding a room and a table of existing rooms. The form includes fields for "Nombre", "Tipo" (set to "Teorico"), and "Capacidad" (set to 0). Below the form are buttons for "Eliminar", "Editar", and "Nuevo". The table below has columns for "ID", "Nombre Salon", "Tipo", and "Capacidad".

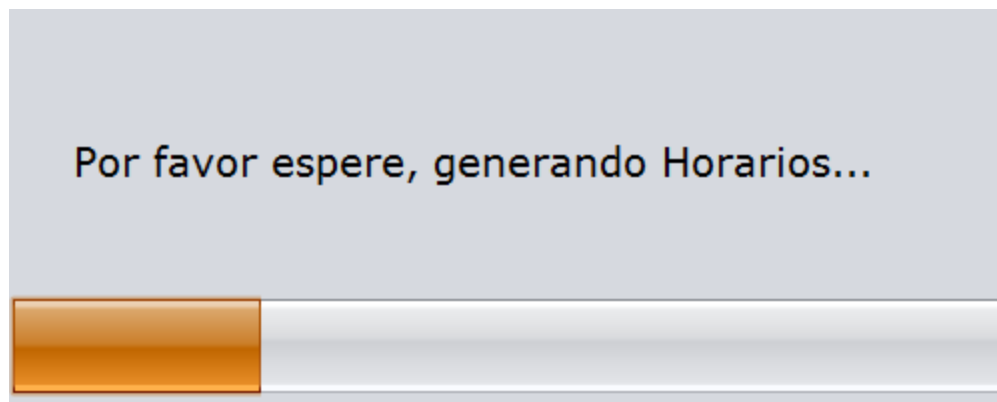
ID	Nombre Salon	Tipo	Capacidad
1	A01	Teorico	40
2	A02	Teorico	40
3	A03	Teorico	40
4	A04	Teorico	40
5	A05	Teorico	40
6	A06	Teorico	40
7	A07	Teorico	40
8	A08	Teorico	40
9	A09	Teorico	40
10	A10	Teorico	40
11	A11	Teorico	40
12	A12	Teorico	40
13	A13	Teorico	40
14	A14	Teorico	40
15	A15	Teorico	40
16	A16	Teorico	40
17	A17	Teorico	40

**Figura 31.** Interfaz salones del prototipo ces.

**Nota Fuente.** Autor del proyecto.

**Interfaz de ejecución:** En esta pantalla podemos ver los resultados de la ejecución del algoritmo, también podemos fijar los horarios seleccionando uno de la tabla de horarios y la opción fijar, luego podemos generar los reportes a Excel mediante la opción reportes.





**Figura 32. Generando horarios en el prototipo CES.**

**Nota Fuente:** Autor del proyecto.

### **Pruebas**

En esta sección, se procede a realizar las pruebas necesarias para ver el funcionamiento del prototipo CES Educativo.

Se ejecuta el archivo CESv1.0.jar, se muestra la pantalla de carga de los componentes y paso seguido queda la interfaz principal.

Podemos consultar los docentes, los salones, las carreras y sus materias. Para generar los horarios damos clic en el botón generar de la parte inferior derecha.

Cuando el algoritmo termina de generar los horarios nos muestra lo siguiente

Id	Lista de Horarios	Fijado
101	Horario 101	No
100	Horario 100	No
99	Horario 99	No
98	Horario 98	No
97	Horario 97	No
96	Horario 96	No
95	Horario 95	No
94	Horario 94	No
93	Horario 93	No
92	Horario 92	No
91	Horario 91	No
90	Horario 90	No
89	Horario 89	No
88	Horario 88	No
87	Horario 87	No
86	Horario 86	No
85	Horario 85	No
84	Horario 84	No
83	Horario 83	No
82	Horario 82	No
81	Horario 81	No
80	Horario 80	No
79	Horario 79	No
78	Horario 78	No
77	Horario 77	No
76	Horario 76	No
75	Horario 75	No
74	Horario 74	No
73	Horario 73	No

**Figura 33. Ejecución del prototipo CES.**

**Nota Fuente:** Autor del proyecto.

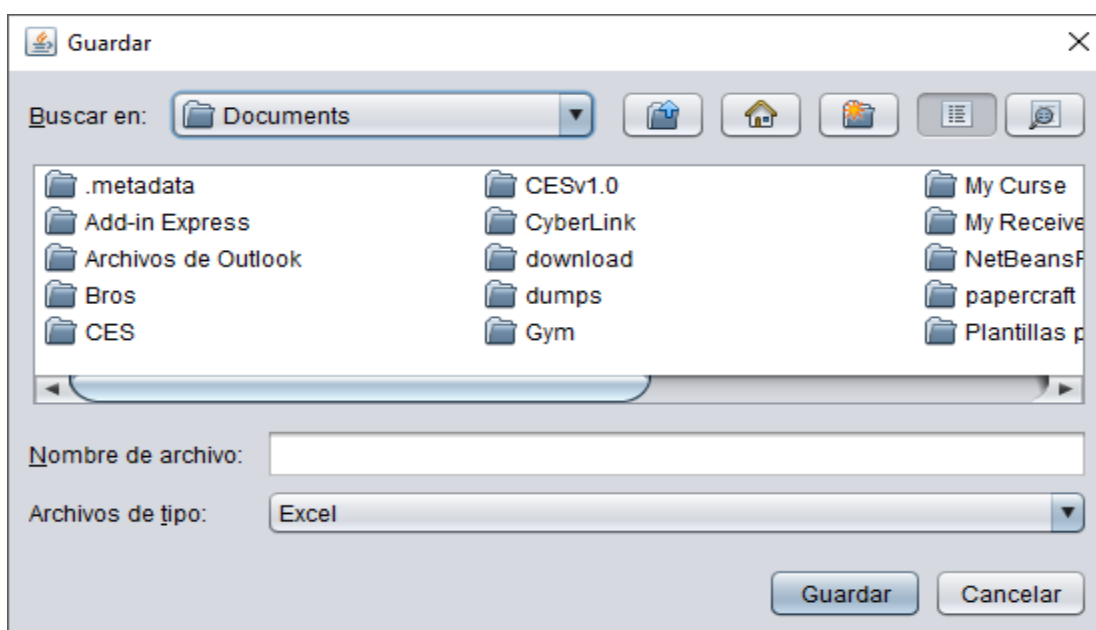
En la tabla de la derecha aparecen los horarios generados, un total de 100 combinaciones de horarios. Al momento de seleccionar un horario de la tabla, el semestre, y la carrera podemos dar visualizar, esto perímite cargar el horario.

Grupo	Materia	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Profesor
A	CALCULO DIFERENCIAL	10:00 12:00 B204			08:00 10:00 PRIMAVERA4			EDGAR
B	CALCULO DIFERENCIAL			16:00 18:00 B205	13:00 15:00 B195			LUIS CARLOS
C	CALCULO DIFERENCIAL	10:00 12:00 B208			07:00 09:00 B209			LUIS CARLOS
A	INTRO MEC			15:00 17:00 A08				JOHN
B	INTRO MEC				10:00 12:00 B210			JOHN
A	COM ORAL Y ESCRITA		06:00 08:00 B203					JAVIER ANTONIO
B	COM ORAL Y ESCRITA				07:00 09:00 B204			JAVIER ANTONIO
A	PROG MEC			18:00 20:00 COMPUTO...		06:00 08:00 COMPUTO...		DULMAR YESID
B	PROG MEC	06:00 08:00 COMPUTO...					09:00 11:00 COMPUTO...	GERARDO ALFONSO
A	EXPRESION GRAFICA						08:00 11:00 SALA DE DL...	GERMAN DAVID
B	EXPRESION GRAFICA		16:00 19:00 SALA DE DL...					GERMAN DAVID
C	EXPRESION GRAFICA	09:00 12:00 SALA DE DL...						GERMAN DAVID
A	QUIMICA Y LAB		19:00 21:00 LABORATO...				06:00 08:00 LAB TOPOG...	CARLOS FERNANDO
B	QUIMICA Y LAB	09:00 11:00 LABORATO...				15:00 17:00 LABORATO...		CARLOS FERNANDO
C	QUIMICA Y LAB	08:00 10:00 LAB TOPOG...	14:00 16:00 LABORATO...					CARLOS FERNANDO
A	GEOMETRIA Y TRIGONO...	09:00 11:00 A11					06:00 08:00 B208	JHRO

**Figura 34. Visualización de horarios en el prototipo CES.**

**Nota Fuente:** Autor del proyecto.

En la figura podemos observar la visualización del horario generado por el prototipo CES del primer semestre de la carrera ingeniería mecánica. Para realizar el reporte en Excel de este horario debemos dar en el botón fijar de la parte inferior derecha, una vez fijado el horario, en la tabla de horarios nos avisará que el horario ha sido fijado marcando un si en la casilla fijado. Este proceso debe hacerse para los horarios de todos los semestres y carreras. Una vez fijado el horario podemos generar el reporte en Excel seleccionando la opción reporte del menú superior.



**Figura 35. Generando reportes en Excel**

**Nota Fuente:** Autor del proyecto.

Damos un nombre para el archivo de reporte de horario y guardamos.

REPORTE DE HORARIOS FIJADOS								
SEDE:	Algodonal							
SEMESTRE:	1							
CARRERA:	Ing. MECANICA							
GRUPO	MATERIA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	PROFESOR
A	CALCULO DIFERENCIAL MC							EDGAR
B	CALCULO DIFERENCIAL MC			16:00-18:00 B205	13:00-15:00 B105			LUIS CARLOS
C	CALCULO DIFERENCIAL MC	10:00-12:00 B208			07:00-09:00 B209			LUIS CARLOS
A	INTRO MEC			15:00-17:00 A08				JOHN
B	INTRO MEC				10:00-12:00 B210			JOHN
A	COM ORAL Y ESCRITA		06:00-08:00 B203					JAVIER ANTONIO
B	COM ORAL Y ESCRITA				07:00-09:00 B204			JAVIER ANTONIO
B	PROG MEC	8:00 COMPUTO GEN					1:00 COMPUTO GEN	GERARDO ALFONSO
A	PROG MEC			0:00 COMPUTO GEN		0:08:00 COMPUTO C		DULMAR YESID
A	EXPRESION GRAFICA						-11:00 SALA DE DIB	GERMAN DAVID
B	EXPRESION GRAFICA		-19:00 SALA DE DIB					GERMAN DAVID
C	EXPRESION GRAFICA	-12:00 SALA DE DIB						GERMAN DAVID
A	QUIMICA Y LAB		LABORATORIO FLU				>08:00 LAB TOPOGR	CARLOS FERIANDO
B	QUIMICA Y LAB	ATORIO MATERIALE				LABORATORIO SUE		CARLOS FERIANDO
C	QUIMICA Y LAB	>10:00 LAB TOPOGR	LABORATORIO MAQ					CARLOS FERIANDO
D	GEOMETRIA Y TRIGONOMETRIA					10:00-12:00 B210	07:00-09:00 B108	JAIRO
C	GEOMETRIA Y TRIGONOMETRIA		18:00-20:00 A04		08:00-10:00 A16			ELIECER ENRIQUE
B	GEOMETRIA Y TRIGONOMETRIA					14:00-16:00 A04		JAIRO
A	GEOMETRIA Y TRIGONOMETRIA	09:00-11:00 A11					06:00-08:00 B208	JAIRO

Figura 36. Reporte en Excel de horarios

Nota Fuente: Autor del proyecto.

## **Capítulo 5. Recolección de Información**

La información necesaria para el desarrollo de esta investigación fue recolectada a través de entrevistas con los directores de programa de cada carrera perteneciente a la facultad de ingenierías, entrevistas al jefe de planeación de la UFPSO, entrevistas a profesores de la facultad y estudiantes. Esto permitió evidenciar el problema presentado, mejorar el entendimiento del problema, identificar las variables inmersas en el mismo y la puesta en marcha de la investigación que dio lugar al diseño y la implementación de un prototipo CES Educativo. En el estado del arte realizado evidenciamos las técnicas utilizadas para dar solución de optimización a nuestro problema, seleccionamos la técnica de inteligencia artificial a aplicar, en este caso algoritmos genéticos.

## Conclusiones

Al terminar el presente trabajo, se ha llegado a las siguientes conclusiones:

El problema de asignación de horarios académicos ha sido tema de investigación en los últimos 60 años debido a la complejidad que estos representan. A pesar de que la inteligencia artificial ha logrado la aplicación de avanzados algoritmos de búsqueda y optimización para resolver este tipo de problema, no se ha llegado a una solución óptima del mismo. Los algoritmos genéticos aplicados en el presente trabajo son una de esas técnicas, utilizadas en la mayoría de los trabajos citados porque han demostrado tener gran ventaja en rendimiento y ejecución. Se evidenció en el desarrollo de la presente investigación que la aplicación de algoritmos genéticos en la solución de problemas de *timetabling* es una muy buena opción, sin embargo era fundamental hacer un análisis de las condiciones del dominio para cada caso, debido a que las condiciones de operación varían de una institución a otra; es por esto que la primera fase de la investigación implicaba obtener una gran cantidad de información y de conocimiento sobre el dominio, recurriendo a las fuentes principales (directores de plan de estudio y profesores) se pudo conceptualizar este dominio de manera acertada a partir de la comprensión de todas las variables: cantidad de salones, horas disponibles de los docentes, materias. El modelamiento que se haga de las variables y los ajustes posteriores son fundamentales para desarrollar un algoritmo que pueda alcanzar una solución aceptable al problema.

Como paso intermedio entre el modelado del conocimiento y la programación de un sistema basado en conocimiento siempre se debe desarrollar una buena formalización del

problema convirtiendo el discurso del problema en reglas, restricciones y hechos. Esto permite entender de manera adecuada las condiciones y variables que intervienen en el problema. La UFPSO presenta un gran número de situaciones especiales (restricciones) en cuanto a generación de horarios se refiere, se hizo necesario seleccionar aquellas restricciones consideradas las más importantes para las condiciones particulares de la facultad de ingenierías y de la Universidad, y unir las con una serie de condiciones globales, que aplican a la mayoría de instituciones de educación superior. Los algoritmos genéticos resultaron ser una alternativa efectiva debido a que genera soluciones de manera rápida y confiable en problemas que requieren optimización.

La presente investigación demostró las ventajas del uso de técnicas como los algoritmos genéticos en problemas cotidianos como el *timetabling*; para poder entender las bondades y el alcance de la solución se construyó un prototipo de software donde se implementaron las reglas y las restricciones analizadas para este problema en particular, evidenciando las ventajas en la disminución del tiempo destinado por profesores y directores de plan de estudios al momento de generar horarios de clase, facilitar la organización y sincronización de las personas encargadas de los horarios y optimizar la asignación de los recursos de clases de la Universidad, ahora se tiene la posibilidad de delegarle a una herramienta computacional el desarrollo automático de una tarea que tardaba días o semanas en el peor de los casos, en minutos u horas, también se garantiza que los salones sean asignados a materias que si los necesitan y como se generan todos los horarios de la facultad en el mismo tiempo, se puede unificar esta tarea entre los tres directores de programa.

El prototipo desarrollado: CES Educativo implementa la técnica de los algoritmos genéticos para generar horarios de clases de la facultad de ingenierías de la UFPSO, permitiendo de forma automática e innovadora solucionar o mitigar de cierta forma problemas de optimización presentados cada semestre.



## Recomendaciones

Al terminar el presente trabajo, se ha llegado a las siguientes recomendaciones:

De acuerdo al primer prototipo de generación de horarios académicos utilizando la técnica de los algoritmos genéticos es recomendable continuar con la investigación y desarrollo que permita la construcción de una herramienta que esté completamente integrada a los sistemas de información académica de la UFPSO y que permita asignar horarios para toda la Universidad.

## Referencias

- Aguilar, L., Baena, L., & Azuela, M. (1996). *Fundamentos de Programacion*. Madrid: McGraw-hill/interamericana de España.
- Ahumada, J. (2014). *Generacion de Horarios Academicos en INACAP utilizando Algoritmos Geneticos*. Santiago: Universidad de Chile.
- Burke, E., & Erben, W. (2000). *Practice and Theory of Automated Timetabling III*. Konstanz.
- Castilla, M., Jacinto, P., & Joel, T. (2008). *Algoritmo genetico para el problema de la calendarizacion de horarios: "una nueva representacion del individuo"*. Universidad del Caribe.
- Castro, E., & Medaglia, A. (s.f.). *Heurística basada en programación entera binaria para el problema de asignación de salones en una universidad*. Bogota: Universidad de los Andes.
- Colombia, C. d. (1991). *Constitucion Politica de Colombia*. Obtenido de [https://books.google.com.co/books?id=OE9i6M0-V4AC&pg=PA14&dq=constitucion+politica+de+colombia&hl=es-419&sa=X&redir\\_esc=y#v=onepage&q=constitucion%20politica%20de%20colombia&f=false](https://books.google.com.co/books?id=OE9i6M0-V4AC&pg=PA14&dq=constitucion+politica+de+colombia&hl=es-419&sa=X&redir_esc=y#v=onepage&q=constitucion%20politica%20de%20colombia&f=false)
- Colorni, A., Dorigo, M., & Maniezo, V. (2013). *A Genetic Algorithm to solve the Timetabling Problem*. Milan: Politecnico di Milano.
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction to Algorithms*. Londres: The MIT Press.

- Cortez, A., Rosales, G., Naupuri, R., & Vega, H. (2010). Sistema de apoyo a la generación de horarios basado en algoritmos genéticos. *Revista de Investigación de Sistemas e Informatica*, 37-55.
- Costabel, S. (2005). *Meta heurísticas aplicadas a un problema de asignación de salones y horarios a asignaturas*. Montevideo: Universidad de la Republica.
- De Jong, K. (1975). *Analysis of the behavior of a Class og Genetic Adaptive Systems*. Hampton: The University of Michigan.
- Fiallos, J., Garcia, J., & Perez, J. (2008). Diseño de un sistema de asignación automática de horario de clases: Caso UNITEC. *Latin American and Caribbean Conference for Engineering and Trchnology*, 1-10.
- Flores, P., Brau, E., Monteverde, J., Salazar, N., Figueroa, J., Cadena, E., & Lizarraga, C. (2004). Experimentos con Algoritmos Genéticos para resolver un problema real de Programacion Maestros-Horarios-Cursos. *Sistemas, Cibernetica e Informatica Volumen1-Numero2*, 42-46.
- Gamo, D. (2006). *Computacion Evolutiva: Algoritmos Geneticos*.
- Gestal, M., Rivero, D., Rabuñal, J., Dorado, j., & Pazos, A. (2010). *Introduccion a los Algoritmos Geneticos y la Programacion Genetica*. Coruña: Digitalia.
- Glover, F. (1989). Tabu Search Part I. *ORSA Journal on Computing Vol. 1 núm 3*, 190-206.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston: Addison-Wesley Longman Publishing Co.
- Gomez, V., Artaza, F., Irigoyen, E., & Iriondo, N. (2005). *Un metodo para la asignacion de la docencia en el ambito universitario mediante algoritmos geneticos*. Bilbao: Universidad del Pais Vasco.

- Gonzales, L., & Trujillo, L. (2011). *Modelo de Gestion del Conocimiento para la Programacion de Horarios en universidades*. Cali: Universidad ICESI.
- Gonzalez, B., & Quintero, J. (2010). *Heurística basada en programación entera binaria para el problema de asignación de salones en una universidad*. Pereira: Universidad tecnologica de Pereira.
- Granada, M., Toro, E., & Franco, J. (2006). Programación Óptima de Horarios de Clase usando un Algoritmo Memético. *Scientia et Technica Vol XII núm 30*, 255-260.
- Heredero, C. (2004). *Informatica y Comunicaciones en la Empresa*. Madrid: ESIC.
- Hernandez, R., Miranda, J., & Rey, P. (2008). Programacion de Horarios de Clases y Asignacion de Salas para la Facultad de Ingenieria de la Universidad Diego Portales Mediante un Enfoque de Programacion Entera. *Revista Ingenieria de Sistemas*, 121-141.
- Holland, J. (1992). Genetic Algorithms. *Scientific American* , 66-72.
- Johnston, J. (2000). *Aplicacion de Algoritmos Geneticos para la asignacion de carga academica en instituciones de educacion superior*.
- Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science, New Series, Vol. 220, núm 4598*, 671-680.
- Mejia, J. (2008). *Asignacion de Horarios de Clases Universitarias Mediante Algoritmos Evolutivos*. Barranquilla: Universidad del Norte.
- Russell, S. j., & Norvig, P. (2004). *Inteligencia Artificial un enfoque moderno*. Madrid: Pearson Educacion.
- Schaerf, A. (1999). *A Survey of Automated Timetabling*. Roma: Universidad di Roma "la Sapienza".
- Schaerf, A. (1999). *Artificial Intelligence Review*. Springer Netherlands.

Smith, C. (2006). *The History of Artificial Intelligence*. Washington.

Takeyas, B., & johnston, j. (2001). *Modelo de Asignacion de Carga Academica Usando*

*Algoritmos Geneticos*. Mexico: Instituto Tecnologico de Nuevo Laredo.

Weber, R. (2000). *Mathematics for operacional research*.

## Apéndice

## Diagramas de flujo del algoritmo genético.

### GENERACIÓN DE LA POBLACIÓN INICIAL

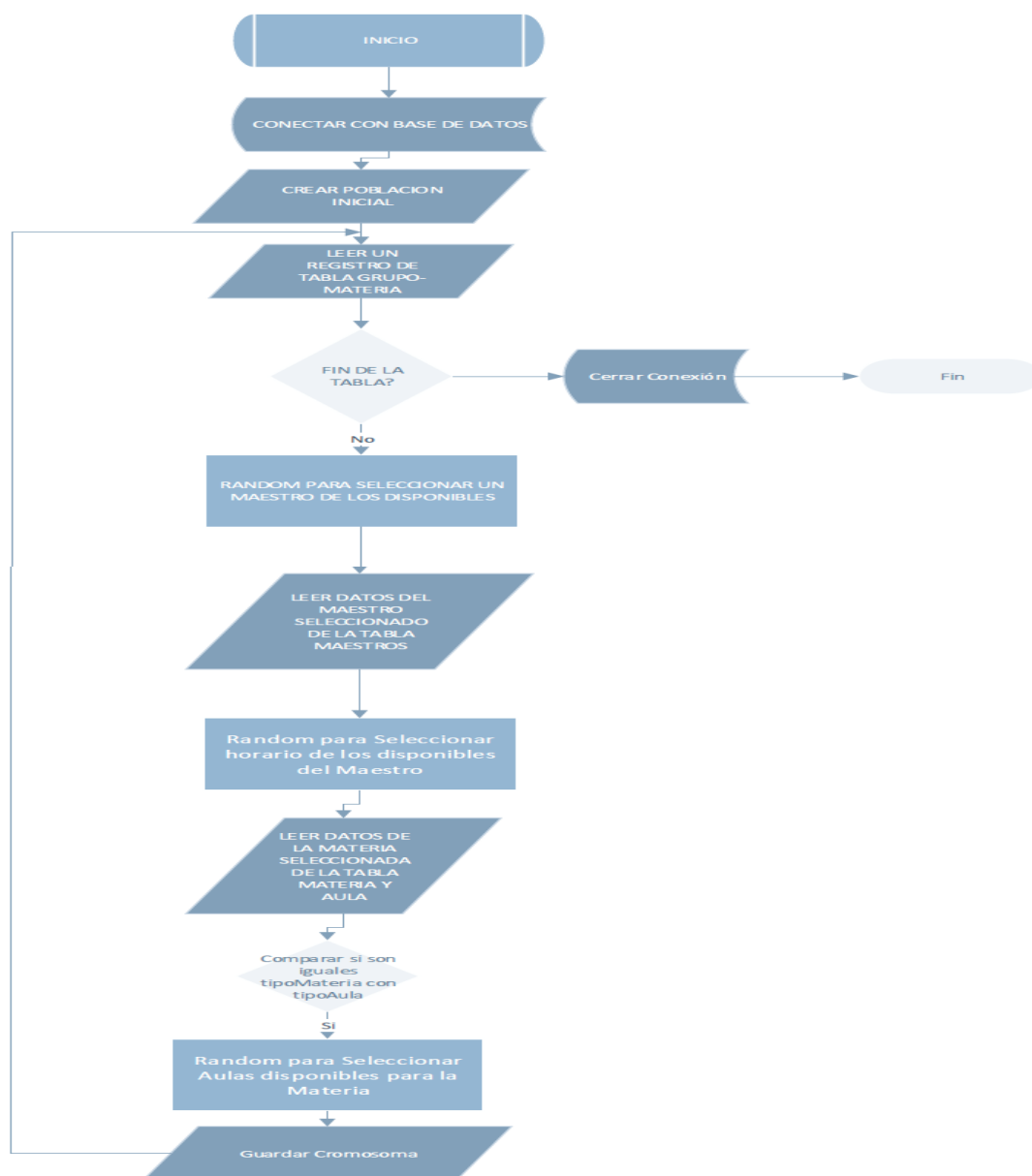
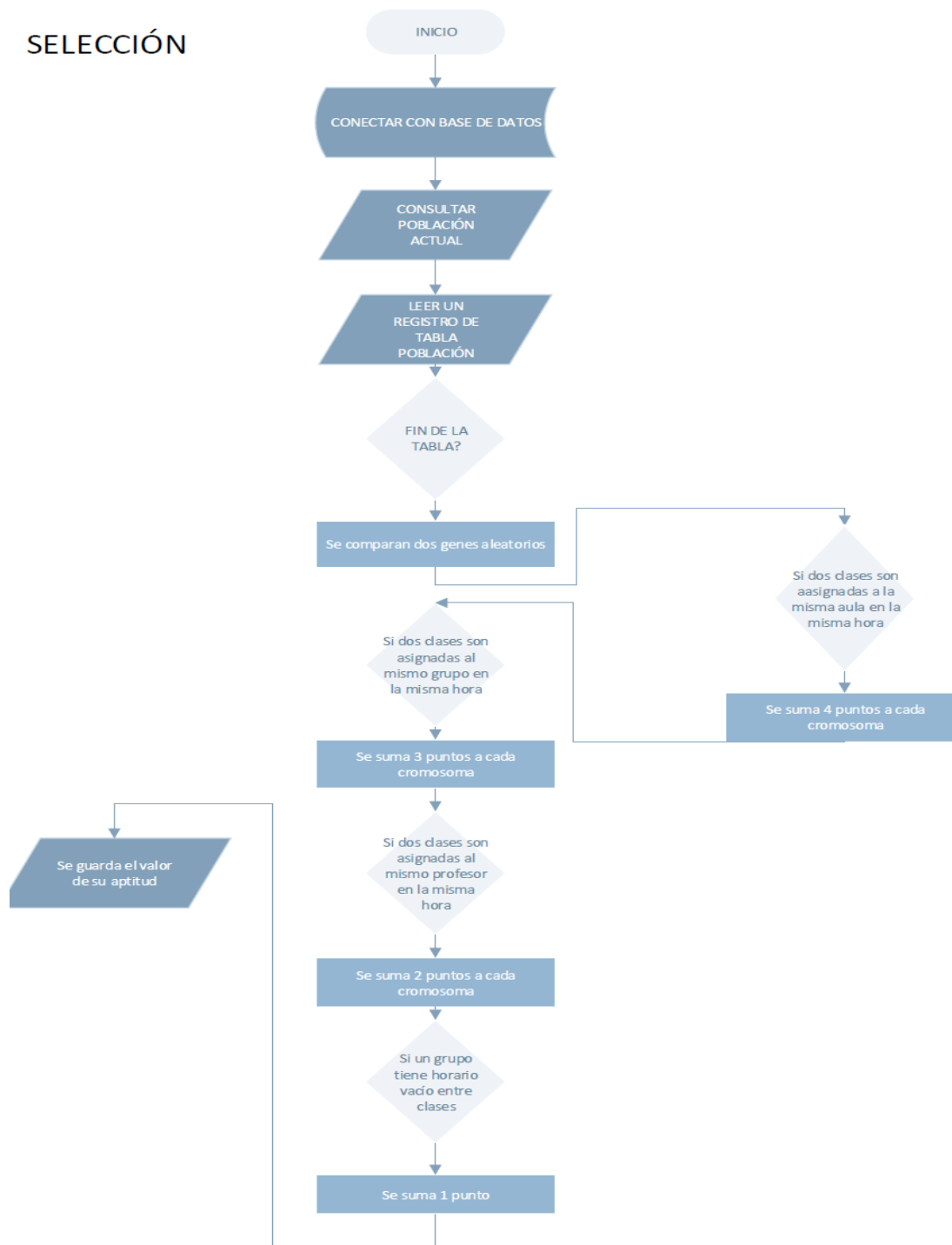


Figura 37. Diagrama de flujo población inicial.

Nota Fuente: Autor del proyecto.

## SELECCIÓN

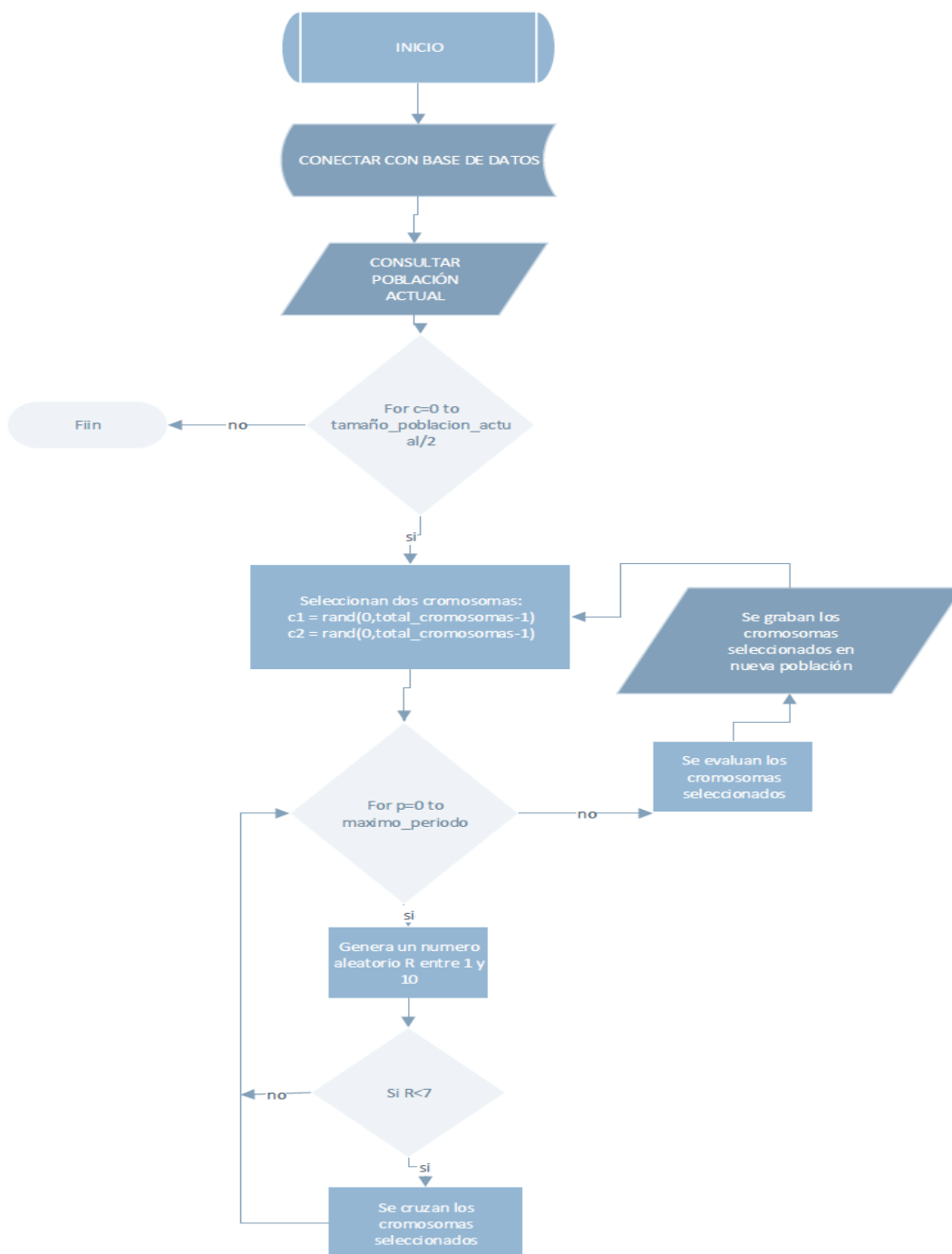


**Figura 38. Diagrama de flujo Operador selección.**

Nota Fuente: Autor del proyecto.

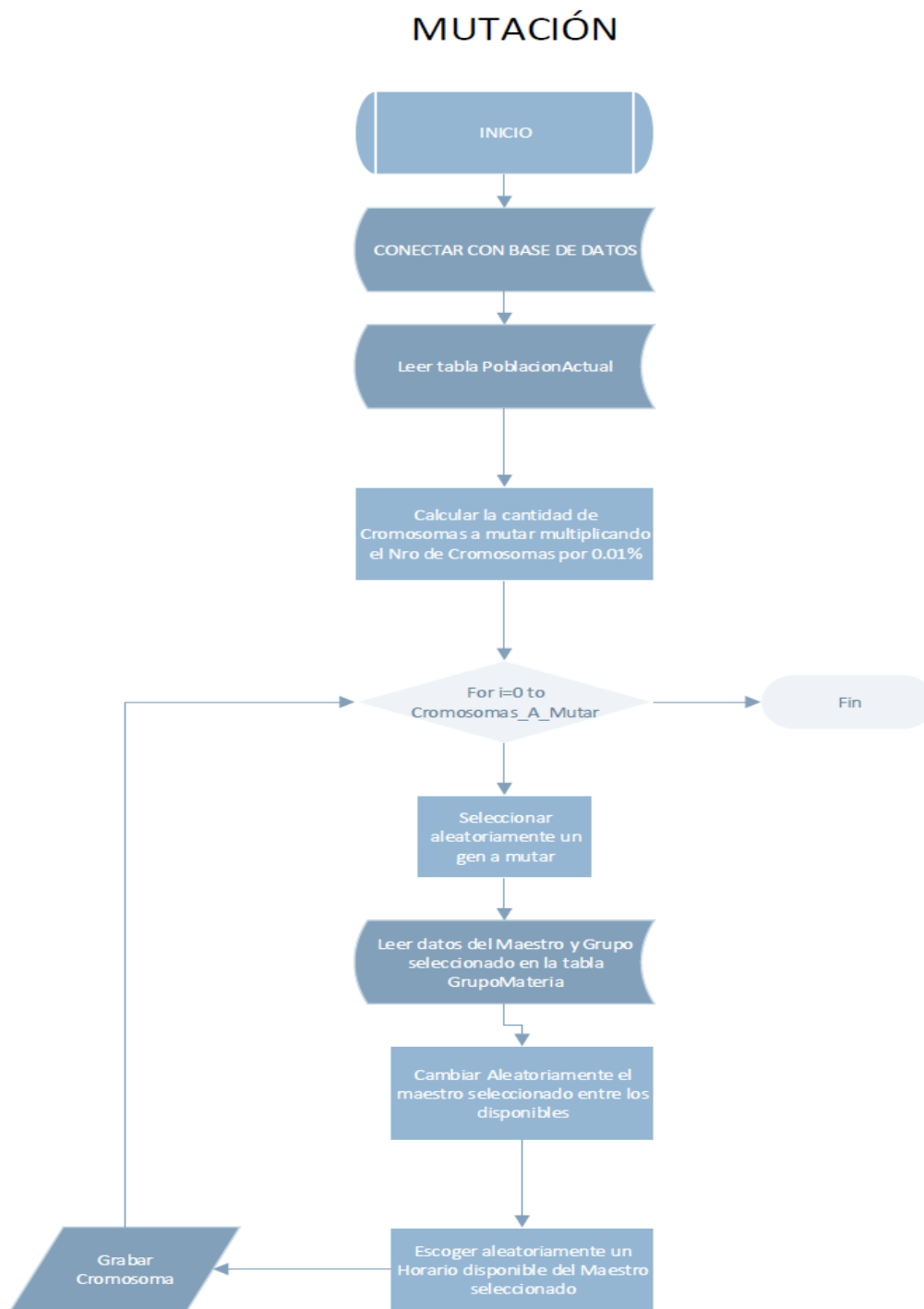


## CRUZA UNIFORME



**Figura 39. Diagrama de flujo operador cruce uniforme.**

Nota Fuente: Autor del proyecto.



**Figura 40. Diagrama de flujo operador mutación.**

Nota Fuente: Autor del proyecto.