

## Diseño de metaheurística y modelo matemático para asignación de horarios en el programa de ingeniería industrial en la PUJ Cali

Felipe Hernández Rendón <sup>1a,c</sup>, Nicolás Vidal Duque <sup>2a,c</sup>

Daniel Morillo Torres <sup>b,c</sup>, Jorge Enrique Alvarez Patiño <sup>c</sup>

<sup>a</sup>Estudiante de Ingeniería Industrial

<sup>b</sup>Profesor, Director del Proyecto de Grado, Departamento de Ingeniería Civil e Industrial

<sup>c</sup>Pontificia Universidad Javeriana, Cali, Colombia

---

### Resumen en español

La asignación de horarios (*Timetabling*) es un problema combinatorio, desafiante, laborioso y repetitivo en distintos sectores; este trata de ordenar una cantidad finita de personal, espacios o recursos a bloques horarios. En el University Course Timetabling (UCT) los cursos deben tomar lugar y distribuir las sesiones académicas evitando cruces lógicos como: un profesor no puede dictar dos materias a la misma hora, un estudiante no puede asistir a dos clases simultáneamente, entre otros. Las universidades e instituciones educativas de nivel superior en Colombia optan por resolver estos problemas de manera manual mediante la prueba y error. Debido a la volátil calidad de las soluciones y complejidad de tipo *NP-Hard*, la metodología actual en las instituciones usualmente resulta en soluciones lejanas al óptimo.

Para abordar este problema en el Programa de Ingeniería industrial en la PUJ como caso aplicado, se diseñó un algoritmo de dos etapas: una metaheurística de tipo búsqueda local iterativa y un método de inicialización a partir de un modelo de programación lineal entera-mixta. Mediante un diseño de experimentos Taguchi se establecieron los parámetros para optimizar el desempeño del algoritmo. Como objetivo se maximizaron las preferencias cumplidas por profesor y la reducción de cruces entre materias que se pueden adelantar o atrasar, a su vez que se eliminaron conflictos entre asignaturas que comparten semestre, profesor y salón. Como consecuencia de la implementación, se redujo en un 99,37% el tiempo asociado al proceso de asignación de horarios semestral, como también un aumento del 14,8% respecto a las preferencias de los profesores. De esta forma la institución obtiene una mayor eficiencia implementando herramientas de automatización y optimización al mismo tiempo que se consideran factores sociales como las preferencias.

*Palabras claves: Timetabling, Horarios, Asignación, Metaheurística, Optimización*

---

### Abstract

Timetabling is a combinatorial, challenging, laborious and repetitive problem in different industries. It consists of ordering a finite amount of personnel, rooms, or resources to hourly blocks. In University Course Timetabling (UCT) courses must take a location and distribute the academic sessions avoiding lecture conflicts such as: a professor cannot lecture two classes at the same time, a student cannot attend two classes simultaneously, among others. Universities and higher education institutions in Colombia opt to solve these problems manually through trial and error. Due to the problem's NP-Hard complexity and the volatile quality of solutions, current university methodologies usually result in infeasible solutions.

In the PUJ's Industrial Engineering Program a two-stage algorithm was designed to address this problem: an iterative local search type metaheuristic with an initialization method based on a mixed-integer linear programming model. Through a Taguchi design of experiments, the parameters were established to optimize the performance of the algorithm. As an objective the preferences fulfilled by the teacher were maximized and the crossovers between subjects that can be viewed in advanced or delayed were reduced. Situations that include conflicts between subjects that share the same semester, teacher and classroom were eliminated. Because of the implementation the time associated with the semester scheduling process was reduced by 99.37%, as well as an increase of 14.8% regarding the preferences of the professors. In such a way the institution obtains greater efficiency by implementing automation and optimization tools while considering social factors such as preferences.

*Key Words: Timetabling, Schedules, Assignment, Metaheuristic, Optimization*

---

**Tabla de contenido**

I. PROJECT CHARTER ..... 4

II. DEFINIR ..... 6

    A. Contexto y Justificación ..... 6

    B. Grupos de interés ..... 9

    C. Requerimientos..... 10

III. MEDIR ..... 13

    A. Aplicabilidad de la problemática..... 13

    B. Casos de prueba..... 14

IV. ANALIZAR ..... 17

    A. Análisis de Oportunidad..... 17

    B. Revisión de literatura ..... 19

    C. Exploración de ideas y selección de alternativa ..... 20

    D. Objetivos ..... 22

    E. Plan de trabajo (PdT)..... 22

V. DISEÑAR..... 23

    A. Desarrollo del diseño de la solución..... 23

    B. Modelo de programación lineal entera-mixta propuesto ..... 24

    C. Diseño de la metaheurística ..... 28

    D. Validación del modelo matemático ..... 37

    E. Validación de la metaheurística ..... 39

VI. VERIFICAR..... 46

    A. Medición de Impactos y Resultados..... 46

    B. Estandarización de la solución ..... 48

    C. Conclusiones..... 49

    D. Recomendaciones ..... 50

VII. GLOSARIO..... 51

VIII. REFERENCIAS ..... 51

IX. ANEXOS..... 54

---

## Índice de Tablas

TABLA I IDENTIFICACIÓN DE REQUERIMIENTOS .....	10
TABLA II CASOS APLICADOS DE REFERENCIA .....	13
TABLA III. RESULTADOS DE INDICADORES DE DESEMPEÑO .....	17
TABLA IV COMPARACIÓN DE RESULTADOS POR CASOS. ....	19
TABLA V. PLAN DE TRABAJO .....	22
TABLA VI. COMBINACIONES DE DIAS .....	30
TABLA VII. BLOQUES HORARIOS .....	31
TABLA VIII. KPIs EJECUCIÓN DEL MODELO MATEMÁTICO CON TIEMPO LÍMITE 60 SEGUNDOS .....	38
TABLA IX. FACTORES CONTROLABLES ESTRUCTURAS DE VECINDARIOS .....	39
TABLA X. FACTORES DE RUIDO ESTRUCTURAS DE VECINDARIOS .....	39
TABLA XI. DATOS PARA EL DISEÑO TAGUCHI, ESTRUCTURAS DE VECINDARIOS .....	40
TABLA XII. RELACION ENTRE NUMERO DE VECINDARIOS POR ESTRUCTURA CON NIVEL DE SIGNIFICANCIA.....	42
TABLA XIII. FACTORES CONTROLABLES CRITERIOS DE PERTURBACIÓN .....	43
TABLA XIV. DATOS PARA EL DISEÑO TAGUCHI, CRITERIOS DE PERTURBACIÓN .....	44
TABLA XV. COMPLEJIDAD DE LOS ESCENARIOS.....	47
TABLA XVI. COMPARACIÓN KPIs ENTRE ESCENARIOS Y ALTERNATIVAS .....	47
TABLA XVII. TABLA DE ANEXOS .....	54

## Índice de Figuras

Fig. 1. Número de artículo relacionados a Timetabling publicados en revistas científicas por año [7].....	7
Fig. 2 Diagrama de flujo de la preparación de la matricula .....	8
Fig. 3. <i>Matriz de Poder vs Interés</i> .....	10
Fig. 4. Diagrama SIPOC del proyecto .....	14
Fig. 5. Categorización de restricciones y función objetivo .....	15
Fig. 6. Exploración de literatura, características de cada metodología.....	15
Fig. 7. <i>Comparación entre resultados horarios del modelo de programación lineal preliminar</i> .....	18
Fig. 8. Categorización de tamaño según características del problema [22].....	21
Fig. 9. Resultados de la tercera instancia pequeña [22] .....	21
Fig. 10. Distribución horaria factible. ....	24
Fig. 11. Descripción gráfica Búsqueda Local [27].....	28
Fig. 12. Secuencia gráfica ILS [28] .....	29
Fig. 13. Colored Conflict Graph .....	32
Fig. 14. Vector solución.....	32
Fig. 15. Diagrama de flujo del algoritmo genérico Branch and Bound [35] .....	34
Fig. 16. Consola AMPL post ejecución del modelo de programación lineal entera-mixta. ....	37
Fig. 17. Ejemplo resultado gráfico .....	38
Fig. 18. Resultados diseño Taguchi para estructuras de vecindarios, efectos principales para medias .....	40
Fig. 19. Resultados diseño Taguchi para estructuras de vecindarios, efectos principales para Relaciones SN .....	41
Fig. 20. R-cuadrado del diseño Taguchi para las estructuras de vecindarios .....	41
Fig. 21. Coeficientes de modelos estimados del diseño Taguchi para las estructuras de vecindarios .....	41
Fig. 22. Prueba de diferencia de medias entre la combinación (1,1,1) y (2,1,1) .....	42
Fig. 23. Relación entre número de vecinos y valor de la función objetivo.....	43
Fig. 24. Resultados diseño Taguchi para criterios de perturbación, efectos principales para medias.....	45
Fig. 25. Resultados diseño Taguchi para criterios de perturbación, efecto principales para Relaciones SN.....	45
Fig. 26. R-cuadrado del diseño Taguchi para los criterios de perturbación .....	46
Fig. 27. Coeficientes del diseño Taguchi para los criterios de perturbación .....	46

## Índice de Algoritmos

ALGORITMO I. BÚSQUEDA LOCAL .....	28
ALGORITMO II. ITERATED LOCAL SEARCH (ILS) .....	29
ALGORITMO III. ESTRUCTURA DE VECINDARIO – LECTURE SWAP .....	35
ALGORITMO IV. ESTRUCTURA DE VECINDARIO – EVENT SWAP .....	35
ALGORITMO V. ESTRUCTURA DE VECINDARIO – INSERT TS.....	36
ALGORITMO VI. CRITERIO DE PERTURBACIÓN – RANDOM ASSIGNMENT PERTURBATION.....	36
ALGORITMO VII. CRITERIO DE PERTURBACIÓN – RANDOM SWAP PERTURBATION.....	37

## I.PROJECT CHARTER

<b>Breve resumen del proyecto (Business case)</b>	
<p>En el <i>University Course Timetabling</i> (UCT) los cursos deben tomar lugar y distribuir las sesiones académicas evitando cruces lógicos como: un profesor no puede dictar dos materias a la misma hora, un estudiante no puede asistir a dos clases simultáneamente, entre otros. El propósito es diseñar una agenda de 5 días hábiles asignando la hora, el aula y la materia, respondiendo a las preguntas ¿Dónde? ¿Cuándo? y ¿Qué materia? Las universidades e instituciones educativas de nivel superior en Colombia optan por resolver estos problemas de manera manual mediante la prueba y error. Debido a la volátil calidad de las soluciones y complejidad de tipo <i>NP-Hard</i>. Este es el caso del programa de ingeniería industrial en la PUJ, donde se buscará implementar un modelo matemático y una metaheurística para acercarse a una solución óptima, buscando mejoras en cuanto a reducción de cruces entre materias, la satisfacción de preferencias (tanto de estudiantes y profesores) y tiempo de ejecución</p>	
<b>Oportunidad (Problem statement)</b>	<b>Impacto en los actores (Business Need—Stakeholders)</b>
Carece de un método automatizado y efectivo que considere las variables actuales (método actual) y, adicionalmente, la preferencia horaria de los profesores y las materias que se suelen adelantar o atrasar.	Mejorar en cuanto a la reducción de cruces entre materias, la satisfacción de preferencias (tanto de estudiantes y profesores) y tiempo de ejecución. Se espera que los métodos e información recopilados puedan ser aplicados en otros programas dentro de la universidad o en otras instituciones.
<b>Objetivo general (Goal statement)</b>	
Diseñar un modelo matemático y una metaheurística para realizar una asignación de horarios eficiente del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali, con el fin de maximizar el porcentaje del cumplimiento de preferencias de los profesores y reducir los cruces entre materias.	
<b>Objetivos específicos (Project Scope)</b>	
Caracterizar y cuantificar los requerimientos y características propias del sistema actual, determinando los parámetros respectivos como las preferencias con base en la información recolectada a partir de censos hacia los profesores y reuniones con el director de carrera.	
Formular un modelo de programación lineal entera-mixta considerando las restricciones y consideraciones del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.	
Diseñar el algoritmo metaheurístico seleccionado considerando las restricciones y requerimientos del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.	
Contrastar el desempeño de las soluciones obtenidas por el algoritmo y modelo matemático al ejecutarlo tanto en el programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali	

<b>Actividad</b>	<b>Fecha de inicio</b>	<b>Fecha de entrega</b>	<b>Encargado</b>
Caracterizar y cuantificar los requerimientos y características propias del sistema actual	01/03/23	10/03/23	Nicolas Vidal y Felipe Hernández
Formular un modelo de programación lineal entera-mixta considerando las restricciones y consideraciones del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.	11/03/23	31/03/23	Nicolas Vidal y Felipe Hernández
Diseñar el algoritmo metaheurístico seleccionado considerando las restricciones y requerimientos del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali	11/03/23	07/04/23	Nicolas Vidal y Felipe Hernández
Contrastar el desempeño de las soluciones obtenidas por el algoritmo y modelo matemático al ejecutarlo tanto en el programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.	08/04/23	22/04/23	Nicolas Vidal y Felipe Hernández

## II. DEFINIR

### A. Contexto y Justificación (¿por qué?)

El problema de asignación de horarios o *Timetabling Problem* (TTP) consiste en programar los cursos o profesores que se imparten en un periodo académico determinado dentro de una institución a franjas horarias. Esta asignación debe considerar los profesores designados para cada asignatura, franjas horarias disponibles, salones requeridos, entre otras características [1]. Debido a que la mayoría de los casos de asignación de horarios son similares, de acuerdo con la literatura se han clasificado en tres problemas principales: *High-School Timetabling* (HTT), *University Course Timetabling* (UCT), y *University Examination Timetabling* (ETT)[2].

En primer lugar, el HTT se centra en la programación de los cursos de los colegios, donde se previene el cruce de horarios de los profesores. Por otra parte, el UCT busca organizar un grupo de asignaturas que comparten estudiantes de tal forma que no se crucen los cursos en el horario. Por último, el (ETT) que busca programar los exámenes de un conjunto de asignaturas con el fin de evitar el cruce de estos, al mismo tiempo que intenta dispersarlos lo más posible para disminuir la carga académica en los estudiantes y así mejorar su desempeño [2]. A pesar de la aparente simplicidad del problema, la programación de estos cursos es un proceso catalogado con una complejidad *NP-Hard* [3], lo que implica que, en el peor de los casos, estos problemas no pueden ser resueltos de manera óptima en un tiempo polinomial (tiempo razonable), por este motivo el TTP es aún de gran interés para la comunidad académica. Además, teniendo en cuenta que cada colegio o universidad tiene sus propias reglas, requisitos y características, la modelación general que abarque todos los problemas es inviable, haciendo cada caso un sistema único.

Todos los problemas de TTP constan con tres conjuntos básicos los cuales son la cantidad de cursos o asignaturas a asignar, número total de horarios disponibles y los salones. Sus dinámicas ocasionan que existan restricciones básicas, las cuales pueden ser duras o suaves. Dentro de las primeras se encuentra que todos los cursos deben ser asignados una vez, cada asignatura debe ser asignada a un solo horario, no puede haber una asignación en el mismo horario si comparten salón o profesor, y que no debe haber superposición de cursos en el mismo horario en caso de que tengan estudiantes en común [4], estas restricciones deben cumplirse de forma obligatoria, pues de lo contrario la solución sería no factible para el sistema. En las restricciones suaves se encuentran las preferencias de los estudiantes o profesores en cuanto a su horario, estas son consideraciones deseables, pero no obligatorias, muchas veces relacionadas con la calidad de la solución, no con su factibilidad. La función objetivo general de los problemas de *Timetabling* es la siguiente:

$$\text{Max } \sum_i^I \sum_t^T \sum_c^C P_{c,t,i} X_{c,t,i} \quad (1)$$

Donde se maximiza la sumatoria de  $P_{c,t,i}$  siendo el peso correspondiente a la clase  $X_{c,t,i}$  sobre  $t$  horas,  $c$  código de clase e  $i$  salones.

Con la finalidad de maximizar las asignaciones de acuerdo con las preferencias establecidas ya sea por los profesores, estudiantes o Institución educativa [4].

La importancia de este problema se ha acentuado más en los últimos años, por ejemplo, en Estados Unidos actualmente la logística educacional posee un reto debido a que los estudiantes optan por llevar una educación flexible, lo que significa que organizan su plan de estudios de acuerdo con sus preferencias, dejando de lado la tradicional ubicación semestral de las asignaturas. Esto ha provocado que se muestre inconformidad entre los estudiantes, profesores y el área administrativa de las universidades debido a que la programación de los horarios genera insatisfacción (además de complejidad) [5]. En un estudio realizado por la Fundación Universitaria Konrad Lorenz, se evaluó el estrés académico de los estudiantes y los factores que los afectan, donde se evidenció que el 41,7% de los estudiantes poseen estrés elevado por “Falta de tiempo para cumplir con las actividades académicas” y el 26,6% por “Acabar la carrera en los plazos estipulados” [6].

El rendimiento de un estudiante se determina parcialmente por el manejo de tiempos o plazos disponibles al realizar la asignación de los horarios, ya que un horario mal distribuido impide un buen manejo del tiempo durante el semestre. En caso de que los horarios no estén correctamente diseñados, al avanzar en el plan de estudios, se pueden presentar cruces entre asignaturas que potencialmente afecten negativamente el avance de los estudiantes provocando así estrés por no poder acabar la carrera en los plazos estipulados.

El incremento del interés por nuevas técnicas de solución al TTP, también se puede evidenciar mediante el número de artículos publicados en revistas científicas relacionadas con este tema como se refleja en la Fig. 1 [5].

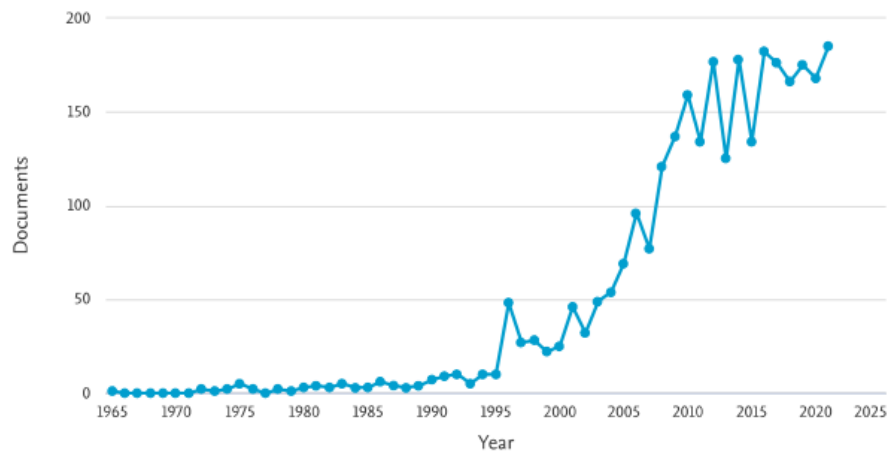


Fig. 1. Número de artículo relacionados a Timetabling publicados en revistas científicas por año [7]

Esta problemática puede ser abordada por la investigación de operaciones, debido a que el reto de organizar correctamente los horarios representa dificultades logísticas y administrativas para las Instituciones Educativas a nivel global que afrontan cada ciclo académico, por lo que buscan resolver los problemas de asignación de horarios a partir de diferentes técnicas clasificadas en dos grupos: técnicas exactas y aproximadas. Las Técnicas exactas (tradicionales) son métodos completos que recorren todo el espacio de soluciones, computando cada una de las posibles combinaciones a un determinado problema, algunos algoritmos logran eliminar regiones de búsqueda donde se garantiza que no se encuentra la mejor solución. Luego de evaluar el espacio factible, estos métodos encuentran una solución óptima. Sin embargo, el éxito de este modelo depende del número de variables en el problema [8]. En este grupo se encuentra la programación entera, *Backtracking*, métodos basados en ramificación y acotamiento, búsqueda exhaustiva, entre otras. Las Técnicas aproximadas se diferencian a las exactas debido a que no exploran todo el espacio de búsqueda, sino que acotan el espacio (regiones no promisorias), por lo que también son llamados métodos incompletos [9], así reducen drásticamente el tiempo de cómputo, sacrificando precisión pues no garantizan encontrar una solución óptima. En este grupo se encuentra recocido simulado, algoritmos evolutivos, algoritmos voraces (GRASP), redes neuronales, heurísticas constructivas, búsqueda local, entre otros [10].

Dentro de las técnicas mencionadas, una de las más utilizadas son la programación lineal o también conocido como algoritmos de optimización matemática, en donde se consideran una serie de restricciones y función objetivo lineales. Esta técnica se implementa principalmente en *High-School Timetabling* (HTT) ya que son problemas que contienen menos variables por lo que se podrá encontrar el valor óptimo en un menor tiempo. Por otro lado, se encuentran los Algoritmos metaheurísticos, los cuales solucionan problemas de propósito general con un tamaño mayor como *University Course Timetabling* (UCT), en estos casos se buscará encontrar una solución aceptable en un tiempo computacional razonable. Por último, se encuentra el enfoque hiperheurístico, los cuales utilizan el mismo principio de las heurísticas, pero implementa un método de selección automático para poder aplicar la heurística [11].

Las universidades e instituciones educativas de nivel superior colombianas optan por resolver estos problemas de manera manual debido a su complejidad matemática. Debido a su complejidad *NP-Hard* es necesario utilizar técnicas no tradicionales para encontrar soluciones cercanas al óptimo. Por lo que las universidades, tanto colombianas como internacionales, tienden a aceptar la ineficiencia de llevar el proceso a mano y basarse en la distribución de clases del periodo anterior, intentando realizar el menor número de modificaciones posibles [12].

Esta investigación se centrará en la Facultad de Ingeniería de la Pontificia Universidad Javeriana de Cali la cual es una institución de educación superior. Este problema estaría catalogado dentro del UCT, la cual está enfocada en brindar una experiencia académica de calidad. Actualmente, la Facultad se encarga de planificar la disposición y organización de las asignaturas que se ofertan, dicha programación se realiza de manera semestral y considera principalmente tres factores, los cuales son la demanda de los distintos cursos, la asignación de horarios y la disponibilidad de salones. Un diagrama que representa este proceso se puede apreciar en la Fig. 2.

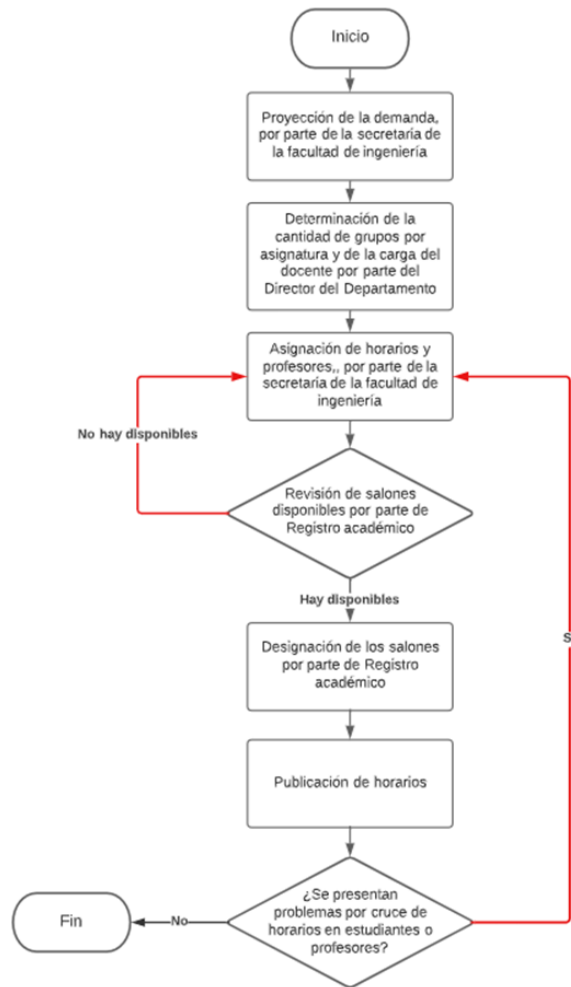


Fig. 2 Diagrama de flujo de la preparación de la matricula

Al observar la Fig. 2 se puede observar que el proceso actual es iterativo, esto debido a que los horarios propuestos son rechazados por falta de salones, esta situación se genera cuando dos o más materias son asignadas en el mismo horario y resultan compartiendo el mismo espacio. En el semestre 2022-1 se presentó la situación en donde el curso de Simulación y Análisis sistémico fueron programadas al mismo tiempo con la necesidad del mismo salón, por lo que se tuvieron que realizar cambios una vez inició el semestre, en consecuencia, se retrasaron los itinerarios de las materias. Por otro lado, asignaturas que se cruzan para los estudiantes o por incompatibilidad con la disponibilidad de los profesores por responsabilidades dentro o fuera de la universidad, lo que demuestra un problema no solo en la programación de horarios, sino en el sistema de información, lo que conlleva a un desperdicio de tiempo y recursos, ya que la planificación actual puede durar semanas.

Por lo anterior, el proyecto busca plantear un método estandarizado para la asignación de horarios en la Facultad de Ingeniería de la Pontificia Universidad Javeriana. La propuesta debe encontrar soluciones factibles en un menor tiempo respecto al método actual, lo que conlleva a una reducción o eliminación de los reprocesos que surgen a partir de las malas asignaciones de los horarios, aliviando la carga administrativa para la Secretaria de la Facultad, Director de Carrera y área de Registro Académico. En cuanto a los estudiantes, se espera que se reduzcan los cruces de materias tanto en el mismo semestre como en los casos donde se presenten atrasos o adelantos de asignaturas.

De obtener el rendimiento esperado el método y herramientas involucradas serán implementadas en el resto de las facultades dentro de la Universidad Javeriana, brindando así una formación académica de calidad a través de un servicio con excelencia a todos los estudiantes.

## B. Grupos de interés

La realización del proyecto conlleva una serie de consecuencias e implicaciones, lo que determina un conjunto de interesados quienes influyen o son afectados por este. Los interesados tienen papeles diferentes dentro de la elaboración y resultado; algunos suministran recursos, otros pueden tener decisiones y voz sobre el proyecto, otros aportan conocimiento o tienen influencia legal y normativa. Estos se organizan basados en su peso sobre el proceso para diferenciar y darle prioridad a los requerimientos de los de mayor influencia, como se muestra en la Fig. 3.

El primer grupo de interés es el equipo 201, cuyos integrantes están desarrollando el proyecto. Estos tienen una influencia directa en el proceso y resultado final; son quienes suministran la mayor parte de los recursos y son quienes aportan la mayor cantidad de conocimiento.

El segundo interesado es el Director de la Carrera de Ingeniería Industrial de la Javeriana Cali, quien fue la persona encargada de identificar el problema y contratar al grupo de personas para resolverlo y diseñar una solución. Este interesado tiene voz y poder de cambio dentro de la elaboración del proyecto debido a que el resultado final se implementa en la carrera bajo su dirección. Debido a esto, es la persona que administra los recursos y tiene la mayor influencia normativa.

De la mano del director de carrera, el tercer interesado es la Asistente de Facultad, quien es la encargada del proceso actual (manual) de asignación de horarios. Tiene una gran influencia debido a que es la fuente de conocimiento más valiosa debido a su experiencia en el proceso; asimismo es la persona más afectada por los resultados del proyecto debido a que su carga laboral, en lo que respecta a la asignación de horarios, será reducida por la automatización del proceso. Adicionalmente tiene altas influencias normativas en cuanto a la elaboración y resultado del proyecto.

Simultáneamente, la automatización de organización de horarios para la facultad reducirá la cantidad de tiempo que consume esta tarea, por lo que se podrán establecer varias posibilidades en cuestión de horas, lo cual implica una mayor flexibilidad y organización dentro de la facultad. La facultad tiene influencia dentro del proyecto, también aporta recursos y tiene influencias normativas para la implementación.

Los profesores de ingeniería son el quinto grupo de interés debido a que el proyecto afecta directamente su horario laboral como su calendario personal. La disponibilidad horaria del profesor se ve influenciada por la asignación horaria de las asignaturas que dictan, por lo que la correcta asignación de horarios permite al docente enseñar una serie de asignaturas de semestres diferentes, en horarios diferentes sin cruzarse con otras asignaturas o su disponibilidad personal. Adicionalmente los profesores afectan directamente la realización del proyecto debido a las restricciones horarias personales y profesionales de cada uno.

Los departamentos externos, como los son el de matemáticas y estadística, el de ciencias naturales, de ingeniería de sistemas, entre otros, influyen directamente en el proyecto debido a que en la carrera de ingeniería industrial se dictan asignaturas de estos departamentos. Por ejemplo, las clases de introducción a la programación, economía, cálculos, físicas, químicas, estadística y probabilidad, entre otras son clases que ofrecen estos departamentos externos. Debido a esta procedencia externa el programa de ingeniería debe acomodar su asignación de horarios después de que cada departamento ya establece un horario para sus materias, por lo que la asignación propia de la carrera debe girar en torno a decisiones externas.

El séptimo grupo de interés son los estudiantes de ingeniería, quienes serán afectados directamente del resultado final. La asignación de horarios influye diariamente a los estudiantes debido a que su semestre y disponibilidad por asignatura será determinados por el resultado del proyecto. Sin una correcta asignación de horarios, los estudiantes se verán enfrentados a problemas de compatibilidad horaria, retrasos en su plan de estudios y la imposibilidad de adelantar algunas asignaturas de semestres superiores.

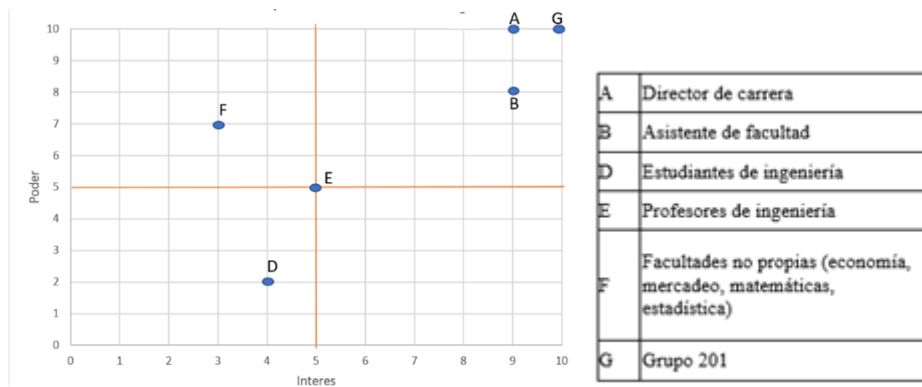


Fig. 3. Matriz de Poder vs Interés

### C. Requerimientos

Con el propósito de obtener resultados concluyentes se establecen una serie de requerimientos por parte de los interesados o la naturaleza de la problemática los cuales se sitúan en la TABLA I. Estos lineamientos se obtienen mediante el análisis y comprensión profunda de los métodos e implicaciones relacionadas con el proceso actual de organización de horarios. Categorizando cada condición de la siguiente forma: restricciones de diseño, especificaciones de diseño, leyes, normas y estándares.

Los requerimientos y restricciones de factibilidad controlan recursos como el tiempo de algunos involucrados, disponibilidad de salones, el tiempo requerido por cada clase, entre otros. Es importante resaltar que las siguientes restricciones limitan la creación o implementación del diseño, ya sean factores físicos (materiales) o conceptuales (abstractos).

TABLA I

#### IDENTIFICACIÓN DE REQUERIMIENTOS

Grupos de interés	VoC (Requisitos grupos de interés)	Restricciones de diseño	Especificaciones de diseño	Leyes, normas y estándares	
				Legislación y Requisitos Aplicables	Importancia o Efecto
Director de carrera	El director de carrera requiere un sistema automatizado que organice la distribución de horarios en la carrera de ingeniería industrial. Este sistema debe asignar y administrar los recursos disponibles como lo son el personal, los espacios de la universidad y el tiempo. Este sistema debe ser rápido en comparación con un ser humano para que se evidencie una mejora del proceso actual (manual).	Disponibilidad del director de carrera para atender las reuniones y resolver dudas. Los tiempos se tienen que cuadrar con mucho tiempo de anticipación y en ocasiones la carga laboral complica la dinámica de trabajo.  Actualmente se considera la licencia de uso de un software especializado para la organización de horarios, sin embargo, el presupuesto no ha sido aprobado.	Establecer un tiempo máximo de respuesta de 7 días.  Reservar espacios de reunión dentro de la facultad.  Seleccionar un método de comunicación rápido, con registro, flexible y con todas las herramientas de trabajo. ( <i>teams</i> )  El diseño debe cumplir con una intensidad estándar de horas semanales.  La facultad de ingeniería requiere que las asignaturas dicten únicamente una vez al día. No es correcto que un mismo grupo reciba clase de la misma asignatura el mismo día en horarios diferentes; las clases por asignatura diarias son máximo uno.  El diseño final debe asegurar que las clases no se puedan dar dos días seguidos. Una misma clase no se puede dictar lunes y martes, o jueves y viernes, tiene que pasar un día de por medio para que la carga del estudiante no sea tan alta y alcancen a realizar trabajos o lecturas de tarea.	Cronograma de avances	Definición de fechas, avances, entregas o cumplimiento de objetivos. Revisiones constantes acerca del avance  El diseño mejorará la calidad del día académico de los estudiantes.  Con la implementación del diseño se establecen estándares de calidad e indicadores que evalúan las alternativas de horarios.

			<p>Es necesario que las clases que están diseñadas, según el plan de estudios estándar, para cursarse en el mismo semestre no se crucen. Si “cálculo I” y “álgebra lineal” son materias de tercer semestre el resultado del proyecto debe evitar un cruce de horarios entre esas dos asignaturas.</p> <p>La facultad de ingeniería necesita que el diseño final pueda evitar cruces entre materias que usualmente se pierden o se adelantan. Por ejemplo: un estudiante está en 4to semestre y desea adelantar una materia de 5 semestre; también quiere matricular una materia que canceló en tercer semestre.</p> <p>Algunas materias ya tienen una categorización como “propensas a repetir” (en la cual están las materias que usualmente los estudiantes pierden o cancelan, por lo que tienen que repetir las el siguiente semestre) o como “propensas a ser adelantadas” (materias que usualmente se matriculan en semestres inferiores a los que se pretenden matricular). De esta forma se espera que el diseño pueda minimizar la cantidad de problemas de cruces que se presentan por estas materias con tendencia identificada.</p> <p>La facultad de Ingeniería y Ciencias establece que todas las clases que se dicten a los estudiantes deberán ser dictadas por un profesor. Lo que se traduce en asegurar que cada clase debe ser dictada por un profesor y solo uno. No puede haber clases sin un profesor asignado.</p> <p>Por la capacidad física de los salones la facultad establece que cada asignatura debe tener habilitados una cantidad de grupos puntual. Por ejemplo: la clase “Cálculo I” debe tener dos grupos: “A” y “B”, cada uno con una capacidad de 20 personas, 40 en total. De esta forma la facultad requiere el aseguramiento de todos los cupos por asignatura.</p> <p>Algunas clases requieren espacios especiales como los laboratorios de cómputo, el laboratorio de ingeniería industrial, entre otras opciones. Las asignaturas que requieren un mismo espacio no se pueden dictar al mismo tiempo para que ambas tengan acceso a las herramientas.</p>		
Asistente de facultad	Requiere un sistema que, como herramienta, le ayude a reducir la cantidad de tiempo que le toma la tarea de asignación de horarios sin reducir la calidad del resultado.	Disponibilidad para reuniones y revisiones. Al ser la persona encargada de la metodología actual de asignación de horarios se debe mantener un contacto cercano para comprender los pasos del proceso actual.	<p>El diseño debe asignar un horario a <b>todas</b> las materias del plan de estudios.</p> <p>No puede bajar la calidad de la asignación final.</p> <p>El diseño debe demorarse menos que el método actual.</p>	N/A	<p>Revisiones acerca de los pasos del proceso.</p> <p>Actualizaciones sobre el rendimiento y velocidad del diseño.</p> <p>Con el diseño la asistente de facultad reducirá el tiempo que tiene que invertir en la organización de horarios semestralmente.</p>

Grupo 201	<p>El grupo requiere acceso a repositorios, bases de datos y libros relacionados con las herramientas y conocimientos implicados.</p> <p>Es necesario el acceso a información propia de la facultad para acomodar parámetros y restricciones.</p> <p>Se requiere disponibilidad de personal de facultad para realizar pruebas y control con información real.</p>	<p>Disponibilidad de recursos digitales del programa de ingeniería industrial.</p> <p>Habilitación de acceso a sistemas de almacenamiento de información (Oracle) para la obtención de la información.</p>	<p>El diseño debe reducir considerablemente el tiempo y recursos empleados para la elaboración de horarios en comparación con la metodología actual.</p> <p>Alta flexibilidad en cuanto a cambios repentinos. El diseño debe adaptarse y reevaluar alternativas rápidamente en el momento en que se presenten cambios.</p>	N/A	<p>Diseñar el modelo de asignación de horarios con el fin de satisfacer a los otros interesados.</p> <p>Se ven afectados académicamente de manera directa</p>
Profesores de ingeniería industrial	<p>El diseño debe organizar las clases considerando el tiempo disponible y preferible de los profesores.</p>	<p>Los profesores son personas susceptibles a cambios o preferencias de disponibilidad. Lo que representa dificultades en la organización fija de un horario.</p>	<p>Se tiene que considerar que un profesor puede dar una clase por periodo de tiempo. En otras palabras, significa que un profesor no puede dar dos clases al mismo tiempo, tiene que dedicarle el 100% de su ocupación únicamente a una clase.</p> <p>Puede que el profesor "A" no esté dictando clase, pero si está ocupado en actividades no laborales, por lo que se considera que "A" no está disponible. El diseño debe considerar las horas en las que los profesores están disponibles, pero <b>prefieren</b> no dictar clase.</p>	Decreto 80 de 1980 – Organización del sistema de educación postsecundaria. [13]	
Estudiantes de ingeniería industrial	<p>Requieren una asignación de horarios que no tenga cruces entre materias del mismo semestre y minimice la cantidad de cruces entre asignaturas propensas a repetir y propensas adelantar.</p>	<p>Los estudiantes no siguen al pie de la letra el plan de estudio. Existen múltiples casos de estudiantes que matriculan materias pertenecientes a diferentes semestres, por lo que se presentan problemas de cruces.</p>	N/A	<p>Las horas dedicadas a estudiar por semestre se determinan por la cantidad de créditos matriculados y asignados por el programa.</p>	<p>Se reduce la carga académica diaria de los estudiantes.</p> <p>Aumenta la satisfacción de los estudiantes con su horario al realizar el proceso de matrícula.</p>
Facultades y Departamentos externos (economía, mercadeo, matemáticas, estadística)	<p>Las materias que no son propias de la carrera y las proporcionan departamentos externos se consideran materias externas. Estas facultades tienen la independencia de establecer sus horarios de acuerdo con sus propios criterios. Por lo tanto, es necesario que el diseño no intente asignar recursos de tiempo esas estas materias; se mantiene la asignación que se hizo por su departamento respectivo.</p>	<p>Administrar el flujo de información acerca de la asignación semestral de materias externas para facilitar la recopilación de información y la integración dentro del diseño.</p>	<p>El diseño debe tomar la asignación horaria ya establecida por cada programa como input y trabajar el resto de la distribución alrededor de las materias externas. Ej: La materia "Constitución política" no es ofrecida por la facultad de ingeniería ni el programa de ingeniería industrial, por lo tanto, el programa de derecho debe, <b>individualmente</b>, asignar una franja horaria, salón y profesor a esta asignatura. Posteriormente el diseño realiza la asignación del resto de materias alrededor de esta, sin sugerirle cambios.</p>	N/A	N/A

### III.MEDIR

#### A. Aplicabilidad de la problemática

Con el fin de demostrar la aplicabilidad de la problemática abordada se consideraron casos reales encontrados en artículos científicos que han sido publicados en revistas académicas indexadas. En este apartado se busca identificar las características del problema abordado, qué solución se planteó y finalmente un análisis de los resultados. A partir de esto se recolecta información útil para el caso en la Pontificia Universidad Javeriana en la TABLA II, para definir la problemática, restricciones o requerimientos, y alternativas de solución.

TABLA II

CASOS APLICADOS DE REFERENCIA

ARTICULO	PROBLEMA	RESULTADO
High quality timetables for Italian schools [14]2022Claudio Crobu, Massimo Di Francesco, Enrico Gorgone	<p>Se plantea una variante del <i>Timetabling</i>, a partir de un caso de estudio en escuelas italianas. Los nuevos requisitos obligan a:</p> <ul style="list-style-type: none"> <li>(i) Proporcionar a los maestros los mismos tiempos de inactividad</li> <li>(ii) Evitar días consecutivos con una gran carga de trabajo</li> <li>(iii) Limitar múltiples lecciones diarias para cada clase</li> </ul> <p>Se presenta un modelo de programación entera el cual es nombrado <i>Italian High School Timetabling Problem</i>(IHSTP), el cual se llevará a cabo en 2 pasos. El primer paso asigna los profesores a los distintos horarios, mientras que el segundo asigna los cursos a los profesores [14].</p>	<p>Al ejecutar el modelo se utilizaron 2 configuraciones distintas, en la primera se consideran todas las restricciones, en cambio en la segunda se trabaja con las restricciones relajadas (simplificadas).</p> <p>En la primera configuración, se obtuvo que el 100% de los casos se resolvieron de manera óptima utilizando en promedio el 68% del tiempo límite en los casos reales y el 80% en los casos generados. En la segunda configuración se realizaron experimentos tanto con los 2 pasos propuestos como en 1 solo paso (directa).</p> <p>Cuando se implementó en un solo paso, se obtuvo el valor óptimo en el 67% de los casos reales, sin embargo, en los casos generados el gap fue desde el 15% a 27%.</p> <p>Finalmente, la segunda configuración utilizando los 2 pasos demuestra ser mejor en el 56% de los casos reales, al compararse con el modelo de programación lineal KSS y la heurística KHE, estos últimos son modelos ya existentes que fueron obtenidos de la literatura para evaluar el desempeño de la formulación propuesta [14].</p>
A General Mathematical Model for University Courses Timetabling: Implementation to a Public University in Malaysia [15] 2022 Zahidah Mohd Zaulira, Nurul Liyana Abdul Azizb, Nur Aidya Hanum Aizam <a href="https://doi.org/10.11113/mjfas.v18n1.2408">https://doi.org/10.11113/mjfas.v18n1.2408</a>	<p>Esta investigación busca centrarse en las preferencias humanas en la programación de los cursos universitarios, por lo que se propone un modelo matemático basado en datos reales de una universidad en Malasia, donde se realizaron encuestas para conocer las preferencias de los usuarios (estudiantes y profesores).</p> <p>Las preferencias para horarios y lugares para cada asignatura son representadas con números enteros que van de 1 a 5, donde 5 representa el horario más preferido y 1 el menos preferido [15].</p>	<p>El modelo obtiene la solución óptima en 1340 segundos, es decir, aproximadamente 22 minutos. Esta solución representa el 97,5% del valor ideal, que se conseguiría en el caso que todas las asignaciones se realizaran en el horario y lugar de más preferencia (5). El 91,8% de los cursos fueron asignados en el horario y lugar de máxima preferencia, mientras que el 0,4% de los cursos fueron programados en el horario y lugar de menos preferencia [15].</p>
A Genetic Algorithm for the Real-world University Course Timetabling Problem [16] 2022	<p>Se propone un algoritmo genético para la asignación de horarios en University Malaysia Sabah-Labuan International Campus, el cual será comparado con los horarios elaborados de manera manual que actualmente son usados en la universidad.</p>	<p>Al comparar el desempeño del algoritmo propuesto con el método manual se obtuvo una mejora en cuanto a las restricciones duras, ya que el algoritmo presenta 0 violaciones mientras que actualmente 35 estudiantes</p>

<p>Chee Hung Wong, Say Leng Goh, Jonathan Likoh</p> <p>DOI:10.1109/CSPA55076.2022.9781907</p>	<p>El objetivo de la propuesta es cumplir con todas las restricciones duras y minimizar las violaciones de las restricciones suaves [16].</p>	<p>presentan cruces entre cursos y 2 materias están asignadas a salones o lugares incompatibles. En cuanto a las violaciones de restricciones suaves, se redujeron en un 46% al implementar el algoritmo [16].</p>
<p>An integer programming formulation for a case study in university timetabling [17] 2004 S. Daskalaki, T. Birbas, E. Housos</p> <p>DOI:10.1016/S0377-2217(03)00103-6</p>	<p>Se presenta una formulación de programación entera, donde se proporcionan restricciones generales para la mayoría de las instituciones educativas. En este artículo se diferencian las clases normales de los laboratorios con el fin de evitar el cruce de asignaturas que requieran salones o espacios especiales. A su vez hay 2 tipos de asignaturas, las cuales son las obligatorias y las electivas, esta distinción se hace para construir las restricciones duras y suaves. Por último, se considera la disponibilidad de recursos humanos, ya que la disposición de los profesores puede estar sujeta a deberes administrativos. El caso aplicado se llevó a cabo en 3 instituciones en Grecia, las cuales son una escuela, Liceo y finalmente la Universidad de Patras [17].</p>	<p>En todas las instituciones se logró obtener el resultado óptimo, en la escuela se requirieron 2,5 minutos, en el liceo 18,5 minutos y finalmente 95 minutos para la Universidad. Además, se concluyó que el problema puede ser dividido en dos partes de acuerdo con la ubicación semestral de los cursos, por lo que primero se ejecutaría el programa para las asignaturas de primeros semestres y posteriormente para las materias con mayor ubicación semestral[17].</p>
<p>A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university[18] 2005 Ricardo Santiago-Mozos, Sancho Salcedo Sanz, Mario De Prado Cumplido, Carlos Bousoño-Calzón</p> <p>DOI:10.1016/j.cor.2003.11.030</p>	<p>En este artículo se presenta la aplicación de un algoritmo heurístico en la Universidad de Vigo en España, a diferencia de los otros artículos, en este se busca realizar la asignación de horarios basados en la preferencia de cada estudiante, para esto se abordó el algoritmo en dos fases. En la primera etapa, se asignarán los alumnos a los distintos grupos que ellos consideren prioritarios, teniendo como factor decisivo la preferencia de cada alumno. Posteriormente, se asignarán a los grupos de asignaturas que no consideren prioritarias [18].</p>	<p>El algoritmo es capaz de asignar el 99% de las solicitudes hechas por los estudiantes, siendo el 90% de estas en los grupos deseados por los alumnos. Además, minimizó la compacidad de los horarios ya que se considera un factor positivo en cuanto a la calidad de los horarios según los estudiantes de la Universidad de Vigo en España [18].</p>

### B. Casos de prueba

El proceso de agendamiento de asignaturas fue descrito en la Fig. 4, no obstante, no se describen las necesidades ni resultados del proceso. El diagrama SIPOC, Fig. 4, se encarga de ilustrar el flujo de información previo y resultante del proceso de asignación horaria.

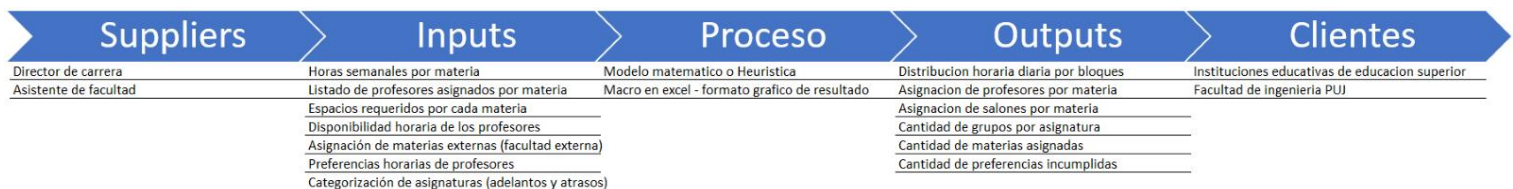


Fig. 4. Diagrama SIPOC del proyecto

Como se observa en la TABLA II las diferentes estrategias empleadas en cada uno de los documentos estudiados varían de acuerdo con el enfoque y resultado deseado. No obstante, la naturaleza de los problemas es la misma, por lo que las propuestas comparten algunas de las condiciones y restricciones que acotan el espacio de solución. En la Fig. 5. se establece la identificación

(ID) y descripción de las restricciones empleadas en los artículos estudiados. Complementariamente la Fig. 6. indica con una X la restricción y función objetivo implicada en la revisión de literatura y el proyecto de diseño.

RESTRICCIONES DURAS		RESTRICCIONES SUAVES		FUNCION OBJETIVO	
ID	DESCRIPCION	ID	DESCRIPCION	ID	DESCRIPCION
R1	Cumplir con el numero de sesiones semanales por asignatura	R17	Considerar preferencias de profesores	FO1	Minimizar violaciones a condiciones (difiere de restricciones)
R2	Un profesor no puede dictar dos clases al mismo tiempo	R18	Compactar el horario (Minimizar huecos entre clases)	FO2	Maximizar preferencias de asignacion
R3	Dictar maximo dos asignaturas por seccion del dia (manana o tarde) por profesor	R19	Evitar dar clases en horarios de la noche	FO3	Minimizar violaciones a restricciones suaves
R4	Dictar clases dia de por medio	R20	Considerar preferencias horarias por tipo de materia (teoricas o practicas)	FO4	Minimizar asignaturas no asignadas
R5	Todos los profesores tienen el mismo tiempo de inactividad	R21	Evitar que el estudiante reciba mas de dos clases seguidas	FO5	Minimizar requerimientos de asignaturas de baja prioridad
R6	Todas las materias deben tener asignacion			FO6	Minimizar la compacidad de la asignacion
R7	Respetar capacidad por salon				
R8	Respetar horarios especiales (hora de almuerzo)				
R9	Cumplir con requerimientos de salon por asignatura (laboratorios, salas de computo, ect)				
R10	No cruzar asignaturas del mismo semestre				
R11	No cruzar asignaturas en un mismo salon				
R12	No cruzar asignaturas dictadas por el mismo profesor				
R13	No exceder horas laborales maximas de los profesores				
R14	Algunas materias no pueden dictarse consecutivamente				
R15	Las clases se asignan de lunes a viernes				
R16	Respetar disponibilidad de profesores				

Fig. 5. Categorización de restricciones y función objetivo

Titulo del artículo	Restriccion																										Heuristica	Modelo matematico			
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	FO1	FO2	FO3	FO4	FO5			FO6		
High quality timetables for Italian schools	X	X	X	X	X											X	X					X									X
A General Mathematical Model for University Courses Timetabling: Implementation to a Public University in		X	X	X		X	X	X	X	X	X		X	X		X			X	X			X								X
A Genetic Algorithm for the Real-world University Course Timetabling Problem	X						X			X	X				X	X		X	X		X			X							X
An integer programming formulation for a case study in university timetabling	X					X		X		X	X	X				X	X	X								X					X
A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a							X			X										X				X		X	X			X	X
Aplicacion de timetabling en la universidad Javeriana Cali	X	X		X		X		X	X	X	X	X		X	X	X	X			X				X	X					X	X

Fig. 6. Exploración de literatura, características de cada metodología

Para la correcta implementación del *Timetabling* dentro de la Pontificia Universidad Javeriana Cali es necesario recopilar información local relacionada con las restricciones y funciones objetivo analizadas en los artículos anteriormente citados. Mediante reuniones con los interesados y una vasta exploración dentro de la base de datos de la universidad se obtuvo la siguiente información (Ver Anexo 9):

- **Número de sesiones semanales por cada materia.**

La intensidad semanal de las asignaturas depende de los créditos asociados en el plan de estudios. Dentro de la Pontificia Universidad Javeriana Cali las materias tienen intensidad de una, dos o tres veces a la semana. Dentro del programa de ingeniería industrial el 21% de las asignaturas se ven una vez a la semana, el 59% dos veces a la semana y el 20% restante se ven tres veces a la semana.

- **Duración de la sesión de clase por asignatura**

La duración de las sesiones se determina por la facultad. Esto depende de la intensidad teórica o práctica propia a la asignatura. El 74% de las asignaturas del programa de ingeniería industrial tienen una duración de dos horas, mientras que el 26% restante tiene una duración de tres horas. No obstante, existe la modalidad de sesiones de una sola hora.

- **Requisitos espaciales que requiere cada asignatura.**

Cada asignatura tiene requisitos espaciales de acuerdo con los temas abordados, puede que se requiera el laboratorio de ingeniería industrial, una sala de cómputo, un salón normal o la sala 3.6 (laboratorio computacional de ingeniería industrial). Dentro de las asignaturas del programa de ingeniería industrial el 22% puede realizarse en la sala 3.6, el 4% puede realizarse en el laboratorio, el 14% en una sala de cómputo y el 36% en un salón normal. La sumatoria de los porcentajes resulta mayor al 100% debido a que las asignaturas tienen la opción de dictarse en más de un tipo de salón, dependiendo de la actividad a desarrollan en aquella sesión.

- **Preferencias horarias por cada tipo de asignatura.**

Las preferencias horarias de las asignaturas dependen de su intensidad teórica o práctica, y también de las preferencias de los profesores que las dictan. Adicionalmente los profesores encargados de impartir la clase pueden tener preferencias respecto a la hora. Las preferencias se establecen por días de la semana y por hora del día. Ejemplo: La clase de Cálculo I tiene un componente teórico fuerte, por lo que se prefiere dictar en el bloque de 9:00 am – 11:00 am; sin embargo, el profesor prefiere que se dicte los lunes y no los viernes. De esta forma hay una preferencia positiva los lunes de 9 a 11, y una preferencia negativa los viernes en ese mismo horario.

- **Conjunto de asignaturas por semestre.**

Las asignaturas propias de cada semestre se determinan por la facultad de ingeniería. Estos son conjuntos de materias que se pueden matricular de acuerdo con la malla curricular, sin embargo, no es estrictamente necesario apearse a este plan de estudios. Un estudiante tiene la opción de matricular asignaturas de semestres superiores o inferiores si se considera necesario.

- **Conjunto de asignaturas propensas a ser repetidas.**

En caso de reprobado o cancelar una asignatura el estudiante debe volverla a matricular en un semestre diferente al indicado por el plan de estudios para completar esos créditos. Basándose en la dificultad de la materia se identifican tendencias a reprobado o cancelar ciertas asignaturas más que otras. Estas asignaturas con un índice de repetición alto son consideradas como propensas a ser repetidas.

- **Conjunto de asignaturas propensas a ser adelantadas.**

Los estudiantes tienen la oportunidad de adelantar las asignaturas cuyos prerrequisitos sean cumplidos. Por lo que se evidencia una tendencia a matricular, en un semestre inferior al indicado por el plan de estudios, ciertas asignaturas.

- **Disponibilidad horaria de profesores por asignatura.**

A diferencia de las preferencias horarias por asignatura, las cuales pueden ser violadas, la disponibilidad determina los días y horarios en las que pueden o no dictar la clase. Esto depende de la disponibilidad del profesor, disponibilidad de espacios y horarios especiales (almuerzo, reuniones de facultad). Ejemplo: la preferencia horaria de la asignatura Cálculo I indica que se prefiere dar la clase el lunes de 9 – 11 am, no obstante, la universidad establece que ese horario no está disponible debido a rutinas de limpieza. Por lo tanto, se tiene que buscar otra alternativa y violar la preferencia.

Con el fin de contrastar los diseños propuestos se modificaron los datos suministrados por la Facultad de Ingeniería y Ciencias en la PUJ con el fin de generar casos de prueba adicionales. Mediante estos se evidencia el desempeño de las soluciones en diferentes escenarios y distintas características. Las alteraciones realizadas al caso principal son detalladas en la etapa VERIFICAR.

### C. *KPI's*

Los KPI's son aquellos indicadores que determinan la categorización de una solución, es decir, diferencian mediante un puntaje el espacio solución. En un modelo matemático se conoce como la función objetivo; en este proyecto de diseño se establece una maximización de las preferencias de asignación y una minimización de los cruces con materias propensas a ser adelantadas o a ser repetidas. De esta forma la solución con mayor cantidad de preferencias satisfechas y menor cantidad de cruces se considera la óptima, la de mejor puntaje. La TABLA III detalla los indicadores de las variables como su situación actual y la meta deseada. La situación actual se encuentra detallada en el Anexo 18. Esta se basa en la distribución horaria que usó el programa de ingeniería industrial durante el periodo académico 2023-1. Para la estimación de los KPI's se programó en el lenguaje de programación C++ un algoritmo para calcular la suma de las preferencias cumplidas, dichas preferencias se estimaron a partir de encuestas a los profesores considerados por el programa de ingeniería industrial, los cuales en una escala de 1 a 5, siendo 5 el valor de mayor preferencia. Para los profesores no considerados por el programa se les asignó una preferencia de valor 3 (neutra) en todos los

horarios. El algoritmo también calcula el número de cruces entre materias del mismo semestre y entre materias que son propensas a atrasar o adelantar.

TABLA III.  
RESULTADOS DE INDICADORES DE DESEMPEÑO

Variable	Indicador	Actualidad	Meta
Preferencias cumplidas	Suma de las preferencias cumplidas	290	>300
Cruces entre materias propensas a adelantar o propensas a repetir	Cantidad total de cruces	57	<50
Cruce entre materias para todos los grupos del mismo semestre	Cantidad total de cruces	19	0

#### IV. ANALIZAR

##### A. Análisis de Oportunidad

Con el fin de demostrar que el *Timetabling* es un problema de decisión, primero se mostrará su clasificación y análisis de complejidad según la literatura científica, luego se llevará a cabo un ejercicio donde a dos asignaciones horarias distintas se les medirá la cantidad total de cruces y el porcentaje de cumplimiento de preferencias, ya que estas son las variables que se consideraron al establecer los KPI's o indicadores de desempeño.

La complejidad de este problema se ha categorizado de acuerdo con la literatura como *NP-Hard* [19], es decir que es un problema no determinístico y no polinomial, por lo que no existe un algoritmo conocido que resuelva el problema, en el peor de los casos, dentro de un tiempo razonable (polinomial). Si bien esta complejidad yace en las características inherentes del problema, existe una relación entre el tamaño del problema y el número de variables y restricciones que se necesitan para representarlo. Así, al ejecutar inicialmente nuestro problema en AMPL el número de variables y restricciones asociados a nuestro proyecto fueron 1'256,090 variables y 3'819,640 restricciones, esto permite evidenciar el tamaño del problema abordado que crece de forma exponencial [20].

Para ejemplificar el problema de decisión, se tomó el modelo de programación lineal preliminar mencionado en el apartado de los indicadores de desempeño, en el que se establecieron tres tiempos de ejecución para evaluar la calidad de diferentes soluciones. Las tres propuestas están sujetas a los mismos parámetros y bajo los mismos escenarios. Los resultados muestran los cruces entre materias matriculadas del semestre actual con materias de semestres superiores e inferiores que pertenezcan a los conjuntos de "materias propensas a adelantar" o "materias propensas a repetir".

La Fig. 7. muestra los tres casos. El caso A tuvo un tiempo de ejecución de 1 segundo, el caso B de tres segundos, y el caso C de 30 segundos. En particular nos centraremos en la asignación en séptimo semestre. La columna izquierda expone su distribución horaria mediante una tabla que asocia las horas del día con el día de la semana; las celdas color amarillo son contenedores de las clases, representadas mediante su código secuencial respectivo. Por otra parte, la columna derecha contiene el mismo tipo de tabla, pero, con la distribución de las asignaturas de los conjuntos  $k1 \cup k2$  (materias con tendencia a ser repetidas o adelantadas). De la misma forma las celdas coloreadas representan la ubicación de las asignaturas, las de color naranja indican un cruce con alguna asignatura del séptimo semestre. Por ejemplo, en el caso A, el jueves, en las horas 7 y 8 se presenta un cruce entre la materia 36 (propia del séptimo semestre) y la materia 28 (perteneciente al conjunto  $k1 \cup k2$ ).

<b>A</b>	<b>7mo SEMESTRE</b>	<b>K1 &amp; K2</b>																																																																																																																																																																
	<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td>36</td><td>20</td><td>36</td><td></td></tr> <tr><td>8</td><td>36</td><td>20</td><td>36</td><td></td></tr> <tr><td>9</td><td></td><td>20</td><td></td><td></td></tr> <tr><td>10</td><td></td><td></td><td></td><td></td></tr> <tr><td>11</td><td>43</td><td></td><td>43</td><td></td></tr> <tr><td>12</td><td>43</td><td></td><td>43</td><td></td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td></td><td>24</td></tr> <tr><td>15</td><td></td><td></td><td></td><td>24</td></tr> <tr><td>16</td><td></td><td>35</td><td>35</td><td>24</td></tr> <tr><td>17</td><td></td><td>35</td><td>3</td><td>35</td></tr> <tr><td>18</td><td>47</td><td></td><td>3</td><td></td></tr> <tr><td>19</td><td>47</td><td></td><td>3</td><td></td></tr> <tr><td>20</td><td>47</td><td></td><td></td><td></td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7	36	20	36		8	36	20	36		9		20			10					11	43		43		12	43		43		13					14				24	15				24	16		35	35	24	17		35	3	35	18	47		3		19	47		3		20	47				<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td>34</td><td></td><td>28</td><td>34</td></tr> <tr><td>8</td><td>34</td><td></td><td>28</td><td>34</td></tr> <tr><td>9</td><td></td><td>39</td><td>28</td><td>39</td></tr> <tr><td>10</td><td></td><td>39</td><td></td><td>39</td></tr> <tr><td>11</td><td>8</td><td></td><td>8</td><td></td></tr> <tr><td>12</td><td>8</td><td></td><td>8</td><td></td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td></td><td></td></tr> <tr><td>15</td><td></td><td></td><td></td><td></td></tr> <tr><td>16</td><td></td><td></td><td></td><td></td></tr> <tr><td>17</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>18</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>19</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>20</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7	34		28	34	8	34		28	34	9		39	28	39	10		39		39	11	8		8		12	8		8		13					14					15					16					17			4		18			4		19			4		20				
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7	36	20	36																																																																																																																																																														
	8	36	20	36																																																																																																																																																														
	9		20																																																																																																																																																															
	10																																																																																																																																																																	
	11	43		43																																																																																																																																																														
	12	43		43																																																																																																																																																														
	13																																																																																																																																																																	
	14				24																																																																																																																																																													
	15				24																																																																																																																																																													
	16		35	35	24																																																																																																																																																													
	17		35	3	35																																																																																																																																																													
	18	47		3																																																																																																																																																														
	19	47		3																																																																																																																																																														
	20	47																																																																																																																																																																
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7	34		28	34																																																																																																																																																													
8	34		28	34																																																																																																																																																														
9		39	28	39																																																																																																																																																														
10		39		39																																																																																																																																																														
11	8		8																																																																																																																																																															
12	8		8																																																																																																																																																															
13																																																																																																																																																																		
14																																																																																																																																																																		
15																																																																																																																																																																		
16																																																																																																																																																																		
17			4																																																																																																																																																															
18			4																																																																																																																																																															
19			4																																																																																																																																																															
20																																																																																																																																																																		
<b>B</b>	<b>7mo SEMESTRE</b>	<b>K1 &amp; K2</b>																																																																																																																																																																
	<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td>36</td><td>24</td><td>3</td><td>36</td></tr> <tr><td>8</td><td>36</td><td>24</td><td>3</td><td>36</td></tr> <tr><td>9</td><td></td><td>24</td><td>3</td><td></td></tr> <tr><td>10</td><td></td><td></td><td></td><td></td></tr> <tr><td>11</td><td>43</td><td></td><td>43</td><td></td></tr> <tr><td>12</td><td>43</td><td></td><td>43</td><td></td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td></td><td>20</td></tr> <tr><td>15</td><td></td><td></td><td></td><td>20</td></tr> <tr><td>16</td><td></td><td>35</td><td>35</td><td>20</td></tr> <tr><td>17</td><td></td><td>35</td><td>35</td><td></td></tr> <tr><td>18</td><td></td><td></td><td></td><td>47</td></tr> <tr><td>19</td><td></td><td></td><td></td><td>47</td></tr> <tr><td>20</td><td></td><td></td><td></td><td>47</td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7	36	24	3	36	8	36	24	3	36	9		24	3		10					11	43		43		12	43		43		13					14				20	15				20	16		35	35	20	17		35	35		18				47	19				47	20				47	<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td>34</td><td></td><td></td><td>34</td></tr> <tr><td>8</td><td>34</td><td></td><td></td><td>34</td></tr> <tr><td>9</td><td></td><td></td><td>39</td><td>39</td></tr> <tr><td>10</td><td></td><td>28</td><td>39</td><td>39</td></tr> <tr><td>11</td><td>8</td><td>28</td><td></td><td>8</td></tr> <tr><td>12</td><td>8</td><td>28</td><td></td><td>8</td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td></td><td></td></tr> <tr><td>15</td><td></td><td></td><td></td><td></td></tr> <tr><td>16</td><td></td><td></td><td></td><td></td></tr> <tr><td>17</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>18</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>19</td><td></td><td></td><td>4</td><td></td></tr> <tr><td>20</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7	34			34	8	34			34	9			39	39	10		28	39	39	11	8	28		8	12	8	28		8	13					14					15					16					17			4		18			4		19			4		20				
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7	36	24	3	36																																																																																																																																																													
	8	36	24	3	36																																																																																																																																																													
	9		24	3																																																																																																																																																														
	10																																																																																																																																																																	
	11	43		43																																																																																																																																																														
	12	43		43																																																																																																																																																														
	13																																																																																																																																																																	
	14				20																																																																																																																																																													
	15				20																																																																																																																																																													
	16		35	35	20																																																																																																																																																													
	17		35	35																																																																																																																																																														
	18				47																																																																																																																																																													
	19				47																																																																																																																																																													
	20				47																																																																																																																																																													
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7	34			34																																																																																																																																																													
8	34			34																																																																																																																																																														
9			39	39																																																																																																																																																														
10		28	39	39																																																																																																																																																														
11	8	28		8																																																																																																																																																														
12	8	28		8																																																																																																																																																														
13																																																																																																																																																																		
14																																																																																																																																																																		
15																																																																																																																																																																		
16																																																																																																																																																																		
17			4																																																																																																																																																															
18			4																																																																																																																																																															
19			4																																																																																																																																																															
20																																																																																																																																																																		
<b>C</b>	<b>7mo SEMESTRE</b>	<b>K1 &amp; K2</b>																																																																																																																																																																
	<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td>3</td><td>35</td><td></td><td>35</td></tr> <tr><td>8</td><td>3</td><td>35</td><td></td><td>35</td></tr> <tr><td>9</td><td>3</td><td></td><td></td><td></td></tr> <tr><td>10</td><td>24</td><td></td><td>20</td><td></td></tr> <tr><td>11</td><td>24</td><td>43</td><td>20</td><td>34</td></tr> <tr><td>12</td><td>24</td><td>43</td><td>20</td><td>34</td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td></td><td></td></tr> <tr><td>15</td><td></td><td></td><td></td><td></td></tr> <tr><td>16</td><td>36</td><td></td><td>36</td><td></td></tr> <tr><td>17</td><td>36</td><td></td><td>36</td><td></td></tr> <tr><td>18</td><td></td><td></td><td></td><td>47</td></tr> <tr><td>19</td><td></td><td></td><td></td><td>47</td></tr> <tr><td>20</td><td></td><td></td><td></td><td>47</td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7	3	35		35	8	3	35		35	9	3				10	24		20		11	24	43	20	34	12	24	43	20	34	13					14					15					16	36		36		17	36		36		18				47	19				47	20				47	<table border="1"> <thead> <tr><th colspan="5">DIAS</th></tr> <tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th></tr> </thead> <tbody> <tr><td>7</td><td></td><td></td><td>34</td><td>34</td></tr> <tr><td>8</td><td></td><td></td><td>34</td><td>34</td></tr> <tr><td>9</td><td></td><td>39</td><td></td><td>39</td></tr> <tr><td>10</td><td></td><td>39</td><td></td><td>39</td></tr> <tr><td>11</td><td>8</td><td></td><td></td><td>8</td></tr> <tr><td>12</td><td>8</td><td></td><td></td><td>8</td></tr> <tr><td>13</td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td></td><td></td><td>28</td><td></td></tr> <tr><td>15</td><td></td><td></td><td>28</td><td></td></tr> <tr><td>16</td><td></td><td></td><td>28</td><td></td></tr> <tr><td>17</td><td></td><td></td><td></td><td></td></tr> <tr><td>18</td><td></td><td></td><td></td><td></td></tr> <tr><td>19</td><td></td><td></td><td></td><td></td></tr> <tr><td>20</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	DIAS					L	M	M	J	V	7			34	34	8			34	34	9		39		39	10		39		39	11	8			8	12	8			8	13					14			28		15			28		16			28		17					18					19					20				
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7	3	35		35																																																																																																																																																													
	8	3	35		35																																																																																																																																																													
	9	3																																																																																																																																																																
	10	24		20																																																																																																																																																														
	11	24	43	20	34																																																																																																																																																													
	12	24	43	20	34																																																																																																																																																													
	13																																																																																																																																																																	
	14																																																																																																																																																																	
	15																																																																																																																																																																	
	16	36		36																																																																																																																																																														
	17	36		36																																																																																																																																																														
	18				47																																																																																																																																																													
	19				47																																																																																																																																																													
	20				47																																																																																																																																																													
	DIAS																																																																																																																																																																	
	L	M	M	J	V																																																																																																																																																													
	7			34	34																																																																																																																																																													
8			34	34																																																																																																																																																														
9		39		39																																																																																																																																																														
10		39		39																																																																																																																																																														
11	8			8																																																																																																																																																														
12	8			8																																																																																																																																																														
13																																																																																																																																																																		
14			28																																																																																																																																																															
15			28																																																																																																																																																															
16			28																																																																																																																																																															
17																																																																																																																																																																		
18																																																																																																																																																																		
19																																																																																																																																																																		
20																																																																																																																																																																		

Fig. 7. Comparación entre resultados horarios del modelo de programación lineal preliminar

Los resultados de la comparación anterior se pueden sintetizar en la TABLA IV. Se identifica que el número de cruces entre materias disminuye con el aumento del tiempo de ejecución (análisis computacional). De la misma forma se evidencia una relación proporcional entre este tiempo y la función objetivo, que representa una mejor asignación de acuerdo con los indicadores de desempeño. Simultáneamente la columna de AbsMiGap, que representa la diferencia porcentual entre la solución actual y la solución óptima, presenta una relación inversamente proporcional, lo que implica que entre más tiempo corre el algoritmo, el resultado se acerca más a la solución óptima.

TABLA IV.

COMPARACIÓN DE RESULTADOS POR CASOS

ID	Numero de cruces	Función objetivo	AbsMiGap [%]	Tiempo de ejecución [s]
A	9	51,6	163,462	1
B	8	82,9	133,855	3
C	4	202,8	9,316	30

De acuerdo con lo anterior se ve que los resultados del proceso de *Timetabling* mejoran conforme al tiempo que se analizan y se evalúan las combinaciones. Debido a esto, bajo las mismas condiciones y parámetros, existen pseudo-infinitas soluciones factibles, donde la diferencia entre ellas es cuantificable y comparable, por lo que se concluye que es un problema de decisión complejo.

### B. Revisión de literatura

Los problemas de *Timetabling* se pueden resolver mediante dos tipos de métodos: los métodos exactos y los métodos iterativos de aproximación. Los métodos exactos se basan en "...el uso de técnicas analíticas o matemáticas, que aseguran la convergencia a una solución óptima, si ésta existe. Estos métodos son diseñados bajo supuestos y características específicas tales como continuidad, diferenciabilidad, espacio de búsqueda pequeño o linealidad, entre otros. Con base en teoremas matemáticos desarrollan procedimientos que garantizan una solución óptima" [21]. Por otro lado, los métodos iterativos de aproximación "incorporan un proceso iterativo, donde, en cada iteración, mediante algún razonamiento, se realiza una búsqueda que permite la construcción o mejora de la solución actual" [21].

Dentro de los métodos exactos se encuentran alternativas como la búsqueda exhaustiva, la programación lineal entera-mixta, divide y vencerás, la programación dinámica, *Branch & Bound*, entre otros. No obstante, la programación lineal entera-mixta es el método ideal para resolver problemas de *Timetabling* debido a la calidad de resultados obtenidos en tiempos viables [4]. Mientras tanto, dentro de los métodos iterativos de aproximación se encuentran las heurísticas, metaheurísticas, e hiperheurísticas de las cuales las metaheurísticas son las que usualmente se emplean para la resolución de estos problemas debido a su rendimiento y calidad de soluciones. A continuación, se hace una breve descripción de cada uno:

- **Modelos de programación lineal entera-mixta**

Son modelos matemáticos que buscan minimizar o maximizar una función objetivo lineal, considerando una serie de restricciones que también deben ser lineales. Uno de los puntos a destacar es que estos modelos siempre buscan el valor óptimo, sin embargo, en problemas de tipo *NP-Hard* como el UCT, el tiempo de ejecución del modelo en el peor de los casos sería demasiado alto [22].

- **Algoritmos metaheurísticos**

Son algoritmos que solucionan problemas encontrando inicialmente una solución aceptable o factible, y a partir de esta se realiza una búsqueda de la mejor solución, los algoritmos heurísticos no garantizan un resultado óptimo, sin embargo, pueden obtener buenas soluciones en un tiempo computacional razonable a diferencia de un modelo de programación lineal [23]. A continuación, se describirán cinco de las metaheurísticas más relevantes en la literatura, estas son: *Evolutionary Algorithm*, *Ant Colony Optimization*, *Iterated Local Search*, *Simulated Annealing* y *Tabu Search*. Estos algoritmos se desempeñan especialmente bien en problemas combinatorios que son computacionalmente difíciles (NP-Hard) [24].

Los modelos de programación entera-mixta difieren de acuerdo con el problema a resolver, por lo que hay tantos tipos de modelos como problemas. Esto implica que no hay subcategorías para este tipo de metodología. Sin embargo, este no es el caso de los algoritmos metaheurísticos, los que se componen de diferentes subcategorías con características propias como: reglas para recorrer el espacio solución, criterios para alterar las soluciones y razonamientos de selección. Los principales algoritmos metaheurísticos empleados en problemas de *Timetabling* son [22]:

- **Genetic Algorithm**

Los algoritmos evolutivos se basan en los mecanismos de la evolución natural, este algoritmo opera sobre una población de soluciones potenciales. Las generaciones se llevan a cabo cuatro etapas: selección, mutación, reproducción y reemplazo. En la selección, se eligen las soluciones más aptas para hacer la función de “padres” para la siguiente generación, ya que en la etapa de reproducción se recombinan partes de cada una de las soluciones progenitoras para formar una nueva solución, la mutación aporta diversidad a la búsqueda, al finalizar esta etapa la población de soluciones iniciales es reemplazada por las nuevas manteniendo a las mejores [22].

- **Ant Colony Optimization**

La optimización de colonias de hormigas consiste en que en cada iteración del algoritmo cada una de las hormigas construye evento por evento la asignación completa de las asignaturas, en cada evento la hormiga elige de forma probabilística la franja horaria para este, tomando en cuenta primero la información heurística, la cual comprende la evaluación de las restricciones del sistema teniendo en cuenta las asignaciones previas. En segundo lugar, considera la información estigmérgica en forma de un tipo de “feromona” la cual está dada por la estimación del beneficio relacionado con las funciones objetivo establecidas, según lo juzgado por iteraciones anteriores del algoritmo. Al finalizar todas estas asignaciones se construye la solución candidata, la cual se buscará mejorar a través de búsqueda local [22].

- **Iterated Local Search**

La búsqueda local iterativa proporciona nuevas soluciones iniciales obtenidas de las perturbaciones de una solución actual, para realizar este algoritmo se deben tener en cuenta cuatro componentes, en primer lugar un procedimiento para generar una solución inicial, un procedimiento de perturbación el cual consiste en modificar la solución actual para generar alternativas, un procedimiento de búsqueda local donde se retorna la mejor solución encontrada y finalmente un procedimiento para decidir a qué solución se le aplica la siguiente perturbación [22].

- **Simulated Annealing**

Es similar a la búsqueda local iterativa, sin embargo, evita quedar atascado en óptimos locales cuando se utilizan métodos de búsqueda locales simples. Lo que se hace en diferencia a la búsqueda local tradicional es que se acepta un movimiento de empeoramiento de acuerdo con una probabilidad en función del deterioro a nivel de la función objetivo, de tal forma que cuanto peor es un movimiento, es menos probable que se acepte [22].

- **Tabú Search**

Es una metaheurística de búsqueda local que utiliza estructuras de memoria para evitar quedar atascada en óptimos locales y de esta forma lograr un equilibrio de intensificación y diversificación. La búsqueda Tabú permite buscar alternativas que no disminuyan el valor de la función objetivo en los casos que no se violen las restricciones, cuando se realiza un movimiento, este se considera “tabú” para las siguientes iteraciones, una solución está prohibida si se obtiene aplicando un movimiento tabú a la solución actual [22]

### C. *Exploración de ideas y selección de alternativa*

La calidad de los métodos descritos depende del tiempo de computación y cumplimiento de restricciones duras y suaves; esto difiere de acuerdo con el algoritmo empleado. Por lo que la selección del método a emplear en el proyecto depende de los requerimientos del problema, el tamaño del problema y el tiempo en el que se desea disponer de la solución.

Se seleccionaron dos alternativas, la primera corresponde a un método exacto, que corresponde a un modelo de programación lineal, debido a que este tipo de método busca la solución óptima. Sin embargo, debido al tiempo de cómputo requerido y con el fin de comparar la eficacia, se seleccionará también una de las metaheurísticas, de la cual se espera que encuentre buenas soluciones en un tiempo computacional razonable. Además, el uso de un modelo de programación lineal está sujeto a la compra de licencias que incluyen un IDE como AMPL y un solucionador (Solver) como Gurobi o Cplex, entre otros. Por lo tanto, con la intención de no depender de este costo, se decide implementar una metaheurística, método sin costo.

En [22] se hace la comparación de rendimiento entre los métodos mencionados anteriormente basados en dos características de las soluciones calculadas: cantidad de violaciones de las restricciones suaves. Dicha comparación se basó de acuerdo con el tamaño del problema.

Los problemas de *Timetabling* se categorizan por su tamaño, este se define de acuerdo con el número de eventos (materias) a asignar, número salones de clase disponibles, número de tipos de salón (si es un laboratorio, sala de cómputo, entre otros), número

de características simultaneas por asignatura (salón de cómputo y laboratorio, o solo un laboratorio), porcentaje de utilización de estas características, número de estudiantes, máximo número de clases por estudiante, y, máximo número de estudiantes por clase.

De acuerdo con la Fig. 8. el problema del programa de ingeniería industrial se categoriza como pequeño al tener 50 asignaturas, no considerar número de salones, tener cuatro tipos de salón, máximo dos características por asignatura, un porcentaje de utilización del 100%, no considerar número de estudiantes, considerar menos de 20 asignaturas por estudiante, y no considerar un máximo de estudiantes por asignatura.

<i>Class</i>	small	medium	large
<i>Num_events</i>	100	400	400
<i>Num_rooms</i>	5	10	10
<i>Num_features</i>	5	5	10
<i>Approx_features_per_room</i>	3	3	5
<i>Percent_feature_use</i>	70	80	90
<i>Num_students</i>	80	200	400
<i>Max_events_per_student</i>	20	20	20
<i>Max_students_per_event</i>	20	50	100

Fig. 8. Categorización de tamaño según características del problema [22]

La metodología de comparación empleada en [22] se basa en la resolución de cinco instancias de tamaño pequeño, donde cada instancia se compone de 500 diferentes intentos. Lo que implica que se registraron los cálculos y soluciones de 2500 problemas por cada metaheurística. Es importante resaltar que se limitó el tiempo de computación a 90 segundos por intento. Los resultados estadísticos de la tercera instancia se presentan en la Fig. 9.

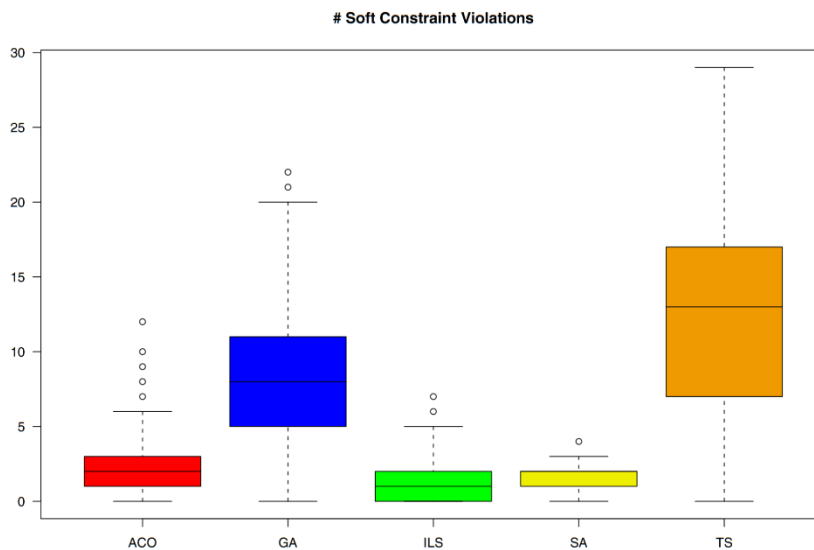


Fig. 9. Resultados de la tercera instancia pequeña [22]

La Fig. 9. se compone de dos gráficas: la gráfica del lado izquierdo mide la cantidad de violaciones a restricciones suaves por tipo de metaheurística. Se evidencia que para esta instancia la metaheurística con menor cantidad de violaciones a restricciones suaves promedio es la ILS (*Iterative Local Search*). Este resultado se mantiene al correr todas las tres instancias, por lo que se concluye que este algoritmo, para resolver un problema de asignación horaria pequeño, es el que presenta menor cantidad de

violaciones a restricciones suaves, por ende, mejor calidad de la solución. De esta forma se determina que el algoritmo a emplear para el caso de la Pontificia Universidad Javeriana Cali es el: *Iterated Local Search* (ILS).

#### D. Objetivos

##### Objetivo general

Diseñar un modelo matemático y una metaheurística para realizar una asignación de horarios eficiente del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali, con el fin de maximizar el cumplimiento de preferencias de los profesores y reducir los cruces entre materias.

##### Objetivos específicos

- Caracterizar y cuantificar los requerimientos y características propias del sistema actual, determinando los parámetros respectivos como las preferencias con base en la información recolectada a partir de censos hacia los profesores y reuniones con el director de carrera.
- Formular un modelo de programación lineal entera-mixta considerando las restricciones y consideraciones del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.
- Diseñar el algoritmo metaheurístico seleccionado considerando las restricciones y requerimientos del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.
- Contrastar el desempeño de las soluciones obtenidas por el algoritmo y modelo matemático al ejecutarlo en el programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.

#### E. Plan de trabajo (PdT)

TABLA V.  
PLAN DE TRABAJO

Objetivo	Área IISE[25]	Herramientas de Ingeniería Industrial	Actividad	Entregable (alcance)	Fecha entrega
Caracterizar y cuantificar los requerimientos y características propias del sistema actual, determinando los parámetros respectivos como las preferencias con base en la información recolectada a partir de censos hacia los profesores y reuniones con el director de carrera	Information Engineering	C. Information Requirements for Organizations 1. Classification of information	Definir una escala para cuantificar las preferencias de los profesores	Vector de escala para las preferencias	27-febrero-2023
	Information Engineering	N. Forms, Programs, and Procedures 1. Forms/reports design	Realizar el formulario haciendo uso de Google Forms, el cual estará vinculado a una hoja de Excel para recopilar las respuestas	Archivo en Excel, con los resultados del censo	03-marzo-2023
	Information Engineering	R. Data Analytics 2. Data preparation	Formatear las respuestas para que puedan ser leídas como parámetros tanto por el modelo matemático como el algoritmo	Archivo txt. donde estén escritos los parámetros	10-marzo-2023
Formular un modelo de programación lineal entera-mixta considerando las restricciones y consideraciones del programa de ingeniería industrial en la Pontificia Universidad	Operations Research and Analysis	B. Linear Programming (LP)	Definir los conjuntos, y parámetros del modelo	Formulación matemática en Word, .dat, .mod y .run	17-marzo-2023
	Operations Research and Analysis	B. Linear Programming (LP)	Sintetizar las restricciones y función objetivo que considerara el modelo		24-marzo-2023
	Operations Research and Analysis	B. Linear Programming (LP)	Definición de las variables de decisión y formulación del modelo matemático		31-marzo-2023

Javeriana de Cali.					
Diseñar el algoritmo metaheurístico seleccionado considerando las restricciones y requerimientos del programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali	Information Engineering	K. System Design 3. Data design	Comprender el encode y adaptarlo al problema actual	Script, donde se editan los datos a un formato propio del algoritmo	07-abril-2023
	Operations Research and Analysis	I. Metaheuristics	Establecer la metodología de generación de vecindario	Seudocódigo de la metodología	14-abril-2023
	Operations Research and Analysis	I. Metaheuristics	Inicialización de una solución factible (verificación de restricciones duras)	Script e información codificada	21-abril-2023
	Operations Research and Analysis	I. Metaheuristics	Exploración del espacio de solución mediante cambios en la solución propuesta (acoplamiento de las restricciones suaves)	Seudocódigo y Script del algoritmo	28-abril-2023
	Information Engineering	.D. Designing Information Outputs 8. Data visualization concepts	Comprender el decode y adaptarlo al problema actual	Solución codificada y decodificada	07-abril-2023
Contrastar el desempeño de las soluciones obtenidas por el algoritmo y modelo matemático al ejecutarlo tanto en el programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali.	Operations Research and Analysis	B. Linear Programming (LP)	Ejecutar el modelo matemático en AMPL	Archivos txt. donde se encuentran los resultados	15-abril-2023
	Information Engineering	D. Designing Information Outputs 8. Data visualization concepts	Hacer un decode a través de una macro en Excel para visualizar los resultados	Excel con los resultados gráficos	16-abril-2023
	Operations Research and Analysis	I. Metaheuristics	Ejecutar el algoritmo metaheurístico	Solución codificada	17-abril-2023
	Information Engineering	D. Designing Information Outputs 8. Data visualization concepts	Utilizar el decode para visualizar los resultados	Solución decodificada, con gráficos	18-abril-2023
	Operations Research and Analysis Information Engineering	A. Operations Research 2. Heuristic versus optimization procedures L. System Evaluation & Justification 3. Evaluation of proposals	Comparar los resultados de ambas metodologías	Tabla comparativa con sus respectivos diagramas	20-abril-2023
	Information Engineering	R. Data Analytics 5. Descriptive analysis	Analizar los resultados	Archivo en Word	22-abril-2023

## V.DISEÑAR

### A. Desarrollo del diseño de la solución

Teniendo en cuenta la revisión de la literatura se seleccionaron un modelo matemático y una metaheurística como metodología que permita la asignación asertiva y oportuna de todas las materias sin generar ningún cruce duro, al mismo tiempo que maximiza el cumplimiento de preferencias de los profesores y se reducen cruces suaves. De esta forma, ambas alternativas consideran los mismos supuestos. Esta selección se justifica en la complejidad del problema abordado, pues si bien existen problemas pequeños que se pueden resolver de forma exacta, la naturaleza *NP-Hard* del problema hace inviable la resolución de modelos a escalas mayores, es ahí donde las metaheurísticas cobran protagonismo, sacrificando un poco de precisión (no garantizan la optimalidad) en favor de eficiencia computacional.

En esta sección primero se describirán las características del diseño junto con los supuestos definidos en el caso, luego se detallan los componentes del modelo de programación lineal entera-mixta como parámetros, variables, conjuntos y la función objetivo. Posterior a esto, se definirán las técnicas o metodologías para definir la generación de soluciones iniciales, estructuras de vecindarios y criterios de perturbación para la *Búsqueda local Iterativa*. Para finalmente validar el funcionamiento de los diseños y que cumplan con los objetivos establecidos.

### 1. Características del diseño

Retomando los requerimientos y especificaciones mencionados en apartados anteriores y considerando las especificidades propias del caso aplicado en el programa de ingeniería industrial en la Pontificia Universidad Javeriana de Cali, se definen los siguientes supuestos y características:

- i. La programación tiene como horizonte de planeación cinco días durante una semana, que corresponden a los días de lunes a viernes.
- ii. En el programa de ingeniería industrial existen materias con intensidad horaria de dos o tres horas, considerando 11 bloques horarios como se observa en la Fig. 10.

Hora	Bloques		
	2 horas	3 horas	
7am	1	7 7-10	
8am	7-9		
9am	2	8 10-1	
10am	9-11		
11am	3		
12am	11-1		
1pm			
2pm	4	9 2-5	
3pm	2-4		
4pm	5	10 5-8	
5pm	4-6		
6pm	6		11 6-9
7pm	6-8		
8pm			
9pm			

Fig. 10. Distribución horaria factible.

- iii. Las asignaturas están sujetas a una intensidad semanal definida previamente por la Facultad. En caso de que presenten una intensidad semanal de dos o más días, la asignación se debe hacer a la misma hora.
- iv. Las materias se pueden ver como máximo una vez al día.
- v. No se pueden programar clases de la misma asignatura dos días seguidos.
- vi. Se considera que todas las asignaturas poseen dos grupos a petición del director de carrera, esto a pesar de que en realidad existen asignaturas que solo poseen un grupo.
- vii. Todas las materias tienen un único profesor asociado, sin embargo, un profesor puede impartir distintas clases.
- viii. Para el programa de ingeniería industrial se consideran cuatro tipos de salones: salas de cómputo, salones regulares, el laboratorio de ingeniería industrial y, de forma especial, el salón Lagos 3,6 que se considera de uso exclusivo para el programa. Las soluciones solo deben evitar cruces en el laboratorio y salón Lagos 3,6 ya que, al ser únicos y que existen asignaturas en un mismo semestre que pueden requerirlos, es donde se pueden presentar conflictos.
- ix. De acuerdo con el plan de estudios del programa, cada asignatura pertenece a un semestre en específico, las materias del mismo semestre no se pueden cruzar (en horarios). A pesar de ser 10 semestres, dado que en el último solo se imparte una o ninguna materia, se consideran nueve semestres.
- x. De acuerdo con la tasa de cancelación y pérdida suministrada por la Facultad de Ingeniería y Ciencias, se consideró un conjunto de asignaturas que se pueden atrasar o adelantar dependiente del semestre.
- xi. Todas las asignaturas deben ser asignadas.
- xii. Cada asignatura está ligada a un único profesor, en los casos de las materias que solo cuentan con un grupo, al añadir el segundo grupo por la petición realizada por el director del programa, se considera que esta segunda asignatura es impartida por el mismo docente del primer grupo.

### B. Modelo de programación lineal entera-mixta propuesto

Considerando los supuestos del diseño, se definieron los conjuntos, parámetros y variables implementadas en el modelo, seguido de la función objetivo y las restricciones correspondientes. Todos estos elementos se detallan a continuación.

### 1. Definición de conjuntos

- $D := \{1, \dots, d, \dots, 5\}$  conjunto de días hábiles para la asignación de clases semanales.
- $B := \{1, \dots, b, \dots, 11\}$  conjunto de los bloques horarios disponibles para asignar materias, estos se corresponden a los presentados en la Fig. 10.
- $J_B$  conjunto que depende de  $B$ , el cual contiene los bloques con los que es incompatible cada bloque  $b \in B$ , por ejemplo, al observar la Fig. 10. se ve que el bloque  $b = 6$  es incompatible con el bloque 10 y 11.
- $INC := \{(1,7), \dots, (b1, b2), \dots, (10,11)\}$  conjunto de parejas de bloques incompatibles.
- $S := \{1, \dots, s, \dots, 9\}$  conjunto de semestres considerados.
- $L := \{1, \dots, l, \dots, 4\}$  conjunto de salones disponibles, donde 1: Salón 3,6; 2: Laboratorio de ingeniería industrial; 3: Sala de cómputo; 4: Salón regular.
- $G := \{1, 2\}$  conjunto de posibles grupos que surjan de una misma asignatura.
- $P := \{1, \dots, p, \dots, 57\}$  conjunto de los profesores relacionados a las asignaturas del programa.
- $K := \{1, 2\}$  conjunto sobre tipos de asignaturas, estas pueden ser: 1: aquellas que pertenecen a un semestre en el plan de estudios, 2: aquellas que son atrasadas o adelantadas y que se pueden cruzar.
- $M_{s,k}$  materias en el semestre  $s \in S$  de tipo  $k \in K$ . Por ejemplo, en  $M_{1,1}$  se encuentran las materias del primer semestre y que pertenecen a ese mismo semestre en el plan de estudios, mientras que en  $M_{3,2}$  se encuentran las materias que en el tercer semestre se pueden ver atrasadas o adelantadas.
- $MS_s$  conjunto de las materias que se pueden llegar a tomar en un semestre  $s \in S$ .
- $MT := \{1, \dots, m, \dots, 50\}$  conjunto de todas las materias que se encuentran en todos los semestres.

### 2. Definición de parámetros

Para el funcionamiento del modelo se necesitaron los siguientes datos de entrada:

- $IS_m$ : intensidad semanal que debe tener cada materia  $m \in MT$ .
- $HB_{m,b}$  indica si la materia  $m \in MT$  se puede impartir en las horas que corresponden al bloque  $b \in B$ .
- $SA_{m,l}$ : señala que salón o lugar  $m \in MT$  está habilitado para cada  $m \in MT$ .
- $PR_{m,g,p}$ : denota que para cada materia  $m \in MT$ , del grupo  $g \in G$ , le corresponde el profesor  $p \in P$ .
- $Preferencia_{m,b,d,g}$ : contiene las preferencias existentes para cada materia  $m \in MT$  del grupo  $g \in G$ , en el bloque  $b \in B$  del día  $d \in D$ . Esta información se obtuvo a partir de encuestas directas con los profesores del programa de ingeniería industrial, posteriormente las preferencias de cada profesor fueron vinculadas con sus respectivas materias. Por lo que las preferencias están ligadas directamente a la asignatura y el grupo que imparte cada profesor encuestado. Las preferencias se midieron en una escala de 1 a 5, donde cinco es el horario que más se prefiere. Los profesores de los cuales no se obtuvo respuesta o no fueron considerados por el programa de ingeniería industrial, se les asignó una preferencia neutra igual a 3.

### 3. Definición de variables de decisión

Variables principales

- $X_{d,b,m,l,g,p}$ : variable binaria que toma el valor de 1 si la materia  $m \in MT$  del grupo  $g \in G$ , es asignada el día  $d \in D$  en el lugar  $l \in L$  al profesor  $p \in P$  en el bloque  $b \in B$ . Es 0 en el caso que no sea asignada.
- $Y_{d,s,b}$ : número de cruces entre materias que se pueden atrasar o adelantar  $k = 2$  en el día  $d \in D$  con ubicación semestral  $s \in S$  en el mismo bloque  $b \in B$ .

VARIABLES DE APOYO

- $N_{b,m,g}$ : variable binaria que toma el valor de 1 si la materia  $m \in MT$  del grupo  $g \in G$ , es asignada en el bloque  $b \in B$ . Esta variable se usa para asegurar que una misma asignatura con una intensidad semanal mayor o igual a 2, se imparta en el mismo bloque horario.

#### 4. Función Objetivo

El valor por maximizar es el número de preferencias menos la cantidad de cruces suaves entre materias que se pueden adelantar o atrasar. Generando así una Función multi-objetivo, ya que contiene dos criterios diferentes en la evaluación del desempeño de la solución. De acuerdo a la literatura, para establecer los pesos o importancia de cada criterio se deben categorizar en cuanto a la relevancia que tienen para los grupos de interés [26], siendo en este caso principalmente el Director del programa de Ingeniería Industrial que en conjunto con la Secretaria de la Facultad establecieron que ambos componentes debían compartir categoría, es decir, poseer el mismo peso en la función objetivo, por lo que no se asignaron coeficientes relacionados a la importancia de estos.

$$\text{Maximize } z: \sum_{p \in P} \sum_{d \in D} \sum_{b \in B} \sum_{m \in MT} \sum_{g \in G} \sum_{l \in L} (X_{d,b,m,l,g,p} * \text{Preferencia}_{m,b,d,g}) - \sum_{d \in D} \sum_{s \in S} \sum_{b \in B} (Y_{d,s,b}) \quad (2)$$

#### 5. Restricciones

##### a) Intensidad diaria

De acuerdo con las especificaciones del diseño, una materia considerando todos los bloques y lugares en un mismo día no puede ser asignada más de una vez.

$$\sum_{l \in L} \sum_{b \in B} X_{d,b,m,l,g,p} \leq 1, \quad \forall d \in D, g \in G, p \in P, m \in MT \quad (3)$$

##### b) Días consecutivos

$$\sum_{l \in L} \sum_{b \in B} X_{d,b,m,l,g,p} + X_{(d+1),b,m,l,g,p} \leq 1, \quad \forall d \in \{1..4\}, g \in G, p \in P, m \in MT \quad (4)$$

De acuerdo con las especificaciones del diseño, al igual como se nombra en la característica v. en el apartado anterior, no se deben programar asignaturas en días consecutivos, ya que es una de las políticas dentro de los horarios para el programa de Ingeniería industrial.

##### c) Intensidad horaria

De acuerdo con la Fig. 10, a cada asignatura le corresponden bloques de 2 o 3 horas, para diferenciar la intensidad horaria de cada materia se utilizó el parámetro  $HB$ , que habilita a la variable principal a tomar valores solo en los bloques que le corresponden. El parámetro  $HB$  también permite obligar o prohibir a una asignatura a ser vista en un bloque en específico, esto puede ser de utilidad en caso de que haya materias que por fuerza mayor solo puedan ser vistas en un horario en concreto.

$$X_{d,b,m,l,g,p} \leq HB_{m,b}, \quad \forall d \in D, g \in G, p \in P, m \in MT, l \in L, b \in B \quad (5)$$

##### d) Intensidad semanal

Para garantizar que las asignaciones se realicen solo el número de veces que corresponden en la semana, se utilizó el parámetro  $IS_m$  acompañado de una variable binaria de apoyo, esta última por cada materia y grupo se habilitará en un único bloque horario.

$$\sum_{l \in L} \sum_{p \in P} \sum_{d \in D} X_{d,b,m,l,g,p} = IS_m * N_{b,m,g}, \quad \forall g \in G, b \in B, m \in MT \quad (6)$$

$$\sum_{b \in B} N_{b,m,g} = 1, \quad \forall g \in G, m \in MT \quad (7)$$

e) *Cruces por salón*

De manera similar que en (11), la restricción (9) habilita a la variable principal a tomar valor solamente en los lugares que le corresponden a partir del uso del parámetro  $SA_{m,l}$ , en (8) se restringe la variable a no realizar más de una asignación en el mismo bloque del mismo día en los lugares que tienen conflicto, que son el 1: Salón 3,6 y en el 2: Laboratorio de ingeniería industrial. Nótese que para los bloques se utiliza el conjunto  $J_B$  el cual permite considerar los bloques que son incompatibles en esta restricción.

$$\sum_{p \in P} \sum_{g \in G} \sum_{m \in MT} \sum_{j \in J_b} X_{d,j,m,l,g,p} \leq 1, \quad \forall d \in D, b \in B, l \in \{1,2\} \quad (8)$$

$$X_{d,b,m,l,g,p} \leq SA_{m,l}, \quad \forall d \in D, g \in G, p \in P, m \in MT, l \in L, b \in B \quad (9)$$

f) *Cruces por profesor*

En cuanto a los profesores, la restricción (10) habilita la asignación de las materias solamente con el profesor que le corresponde a partir del parámetro  $PR_{m,g,p}$  mientras que en (11) se evitan asignaciones que compartan profesor, bloque y día.

$$X_{d,b,m,l,g,p} \leq PR_{m,g,p}, \quad \forall d \in D, g \in G, p \in P, m \in MT, l \in L, b \in B \quad (10)$$

$$\sum_{l \in L} \sum_{g \in G} \sum_{m \in MT} \sum_{j \in J_b} X_{d,j,m,l,g,p} \leq 1, \quad \forall d \in D, p \in P, b \in B \quad (11)$$

g) *Cruces por semestre – mismo bloque (Restricción dura)*

La restricción (12) obliga al modelo de encontrar soluciones sin cruces de materias del mismo semestre, en caso de que no existieran soluciones que permitan evitar todos estos cruces duros, sería necesario relajar la restricción añadiéndole una variable que habilite los cruces, sin embargo, dicha variable también debería ir en la función objetivo penalizada.

$$\sum_{p \in P} \sum_{g \in G} \sum_{l \in L} \sum_{m \in M_{s,1}} X_{d,b,m,l,g,p} \leq 1, \quad \forall d \in D, s \in S, b \in B \quad (12)$$

h) *Cruces por adelanto o atraso – mismo bloque (Restricción suave)*

Debido a la cantidad de asignaturas que se pueden adelantar o atrasar, no es posible eliminarlos a todos, por lo que se implementó la restricción (13) cuya variable de penalización se encuentra también en la función objetivo. Esta restricción abarca los cruces que se dan en un mismo bloque horario.

$$\sum_{p \in P} \sum_{g \in G} \sum_{l \in L} \sum_{m \in MS_s} X_{d,b,m,l,g,p} \leq 1 + Y_{d,s,b}, \quad \forall d \in D, s \in S, b \in B \quad (13)$$

i) *Cruces por semestre – bloques incompatibles (Restricción dura)*

Para complementar la restricción (12), es necesario considerar los cruces entre los bloques incompatibles, para lograrlo se utilizó el conjunto de parejas  $(b1, b2) \in INC$ , que contiene las combinaciones de bloques que se cruzan.

$$\sum_{p \in P} \sum_{g \in G} \sum_{l \in L} \sum_{m \in M_{s,1}} X_{d,b1,m,l,g,p} + X_{d,b2,m,l,g,p} \leq 1, \quad \forall d \in D, s \in S, (b1, b2) \in INC \quad (14)$$

j) Cruces por adelanto o atraso – bloques incompatibles (Restricción suave)

Finalmente, debido a la imposibilidad de eliminar los cruces suaves, se propuso la restricción (15) que considera los bloques incompatibles al igual que en (14) pero con las asignaturas de adelanto o atraso por semestre.

$$\sum_{p \in P} \sum_{g \in G} \sum_{l \in L} \sum_{m \in M_s} X_{d,b1,m,l,g,p} + X_{d,b2,m,l,g,p} \leq 1 + Y_{d,s,b1} + Y_{d,s,b2}, \quad \forall d \in D, s \in S, (b1, b2) \in INC \quad (15)$$

### C. Diseño de la metaheurística

En esta sección se describe a detalle la metaheurística propuesta para resolver el problema, cuya base metodológica es la *Iterated Local Search* (ILS), que parte del principio de una búsqueda local. Esta consiste en, a partir de una solución inicial, mejorar iterativamente las soluciones encontradas mediante la exploración de soluciones vecinas. El objetivo es encontrar la mejor solución dentro de una vecindad específica de la solución actual. Una vez se encuentra la mejor solución del vecindario, esta pasa a ser el nuevo punto de partida para recorrer más vecindarios y continuar la búsqueda de mejores soluciones, la búsqueda local se detiene cuando en el vecindario no se encuentran soluciones que mejoren la actual (Fig. 11 (tomada de [27]), ALGORITMO I). A esta última solución se le denomina óptimo local. Así, unos de los componentes más importantes es definir la estructura de vecindario que permitirá generar soluciones vecinas a partir de la actual.

ALGORITMO I

BÚSQUEDA LOCAL

---

#### Algorithm 1: Local Search

---

**Input** :  $S_0 \leftarrow$  Starting point solution

**Output**: Best solution  $S$  found so far

$S \leftarrow S_0$

**while**  $S$  is not locally optimal **do**

    Find  $S' \in N(S)$  with  $f(S') > f(S)$

$S \leftarrow S'$

**return**  $S$

---

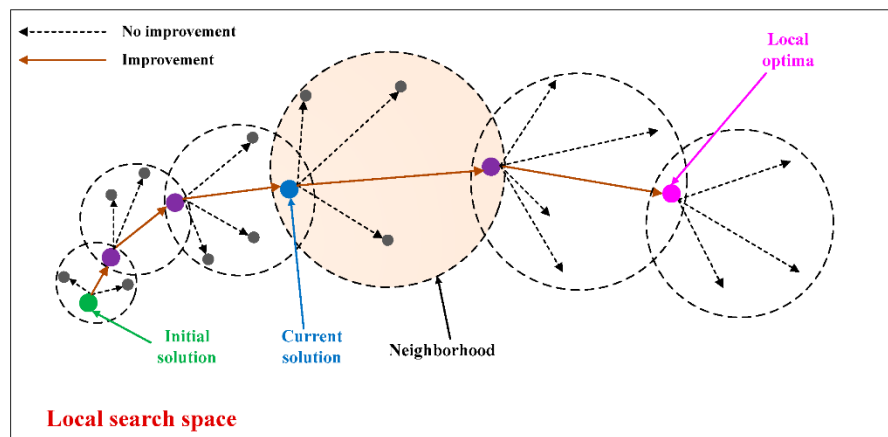


Fig. 11. Descripción gráfica Búsqueda Local [27]

Por su parte la ILS, una vez la búsqueda local encuentra un óptimo local, realiza una perturbación a la mejor solución obtenida, esta solución alterada sirve como una nueva solución inicial que es sometida a una búsqueda local para obtener mejores resultados. El propósito de esto es evitar que el algoritmo se quede estancado en óptimos locales, estos mecanismos se denominan estrategia de exploración en la búsqueda. En la Fig. 12 se puede ver un ejemplo del funcionamiento de la metaheurística y la estructura del algoritmo en el ALGORITMO II.

ALGORITMO II  
ITERATED LOCAL SEARCH (ILS)

---

**Algorithm 2:** Iterated Local Search

---

**Input :**  $I \leftarrow$  Instance FROM PUJ  
**Output:** Best solution  $S$  found so far

$S_0 \leftarrow$  GenerateInitialSolution( $I$ )  
 $S \leftarrow$  LocalSearch( $S_0$ )  
**while** *Stopping criteria not met* **do**  
     $S' \leftarrow$  Perturbation( $S$ )  
     $S'' \leftarrow$  LocalSearch( $S'$ )  
     $S \leftarrow$  AcceptanceCriterion( $S, S''$ )  
**return**  $S$

---

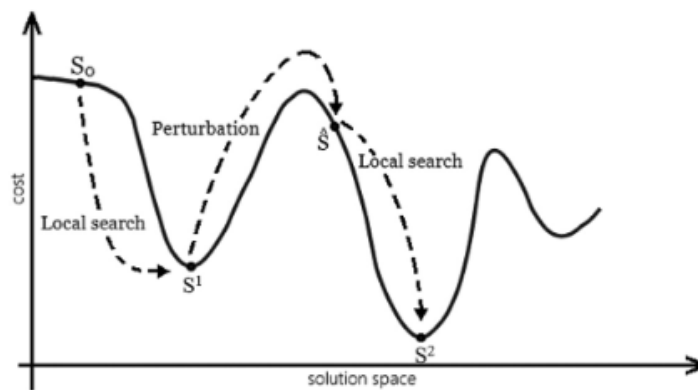


Fig. 12. Secuencia gráfica ILS [28]

A continuación, se describen los componentes del diseño del algoritmo, que fue implementado en el lenguaje de programación C++.

### 1. Codificación del problema

La codificación juega un papel esencial en las metaheurísticas, al referirse al proceso de representar soluciones en problemas de optimización o búsqueda mediante estructuras de datos congruentes. En este contexto, la codificación desempeña un papel crucial ya que define cómo se representan y manipulan las soluciones, lo que impacta directamente en la capacidad del algoritmo para encontrar soluciones óptimas o de alta calidad en un tiempo razonable.

La importancia de una codificación adecuada radica en varios aspectos fundamentales. En primer lugar, la codificación determina la representación de las soluciones, definiendo qué variables o componentes deben incluirse y cómo se relacionan entre sí. Por lo tanto, una codificación eficiente debe ser capaz de representar todas las soluciones del espacio solución.

Además, la codificación también tiene un impacto en el espacio de búsqueda. Cada codificación debe ser biyectiva: cada combinación en la representación computacional referencia únicamente una solución del problema. Por lo tanto, la codificación representa una solución única en dicho espacio. Asimismo, la codificación influye en los operadores de búsqueda, como mutación, recombinación y búsqueda local. Estos operadores se aplican a las soluciones codificadas para explorar y explotar el espacio de búsqueda en busca de soluciones óptimas. Por lo tanto, una codificación inadecuada puede converger en operadores de búsqueda ineficientes o incluso a soluciones infactibles.

Una codificación eficiente también puede mejorar la eficiencia computacional de la metaheurística. Si la codificación es simple y permite operaciones rápidas y eficientes, la metaheurística puede explorar y evaluar un mayor número de soluciones en un tiempo menor. Además, la codificación puede influir en la capacidad de la metaheurística para converger hacia soluciones óptimas o de alta calidad. Una codificación bien diseñada puede facilitar el descubrimiento de soluciones prometedoras y acelerar el proceso de búsqueda, mientras que una codificación deficiente puede dificultar el progreso y limitar la calidad de las soluciones encontradas [29].

Para la representación del problema se definieron las siguientes estructuras:

a) *Combinaciones de días (CD)*

Las *combinaciones de días* se definieron con el propósito de identificar, mediante un número secuencial, los días en los que una asignatura debe tomar lugar. Esta estructura tiene tres variaciones:

i) *Día singular:*

La clase se dicta no más de una vez a la semana, por lo que puede tomar un valor del uno al cinco [1 – 5] representando los 5 días de la semana en los que se da clase: lunes, martes, miércoles, jueves o viernes.

ii) *Día dual*

La clase se dicta exactamente dos veces a la semana, por lo que puede tomar valores del uno al seis [7 – 11] representando las combinaciones de los cinco días de la semana en muestras de a dos días, con un espacio mínimo entre clases de un día. Ejemplos de esta categoría son: lunes y miércoles, martes y jueves, lunes y jueves, lunes y viernes.

iii) *Día triple*

La clase se dicta exactamente tres veces a la semana, por lo que toma el valor de 12, representando la combinación de días mediante la separación entre clases por un día: lunes, miércoles y viernes.

La TABLA VI. define los números secuenciales con los que se identifica cada bloque horario:

TABLA VI.

COMBINACIONES DE DÍAS

Categorización	Combinación de Días	Identificación
Singular	Lunes	1
	Martes	2
	Miércoles	3
	Jueves	4
	Viernes	5
Dual	Lunes – Miércoles	6
	Lunes – Jueves	7
	Lunes – Viernes	8
	Martes – Jueves	9
	Martes – Viernes	10
	Miércoles – Viernes	11
Triple	Lunes – Miércoles – Viernes	12

b) *Bloques horarios (BH)*

Los *bloques horarios* agrupan las doce horas de clase, desde las 7:00h hasta las 21:00h (excluyendo la hora de almuerzo), de dos formas diferentes:

i) *Bloques de dos horas*

Las clases que tienen una duración de dos horas se pueden asignar en seis bloques paralelos. Ejemplos de estos bloques son: 7:00 – 9:00 y 9:00-11:00.

ii) *Bloques de tres horas*

Las clases que tienen una duración de tres horas se pueden asignar en cinco bloques. Ejemplos de los bloques son: 7:00-10:00 y 10:00-13:00.

La TABLA VII. muestra las horas del día con el numero identificador del *BH* correspondiente. La representación gráfica, distribución en el día, se encuentra en la Fig. 10.

TABLA VII.

BLOQUES HORARIOS

Categorización	Rango horario	Identificador
Bloque de dos horas	7:00 – 9:00	1
	9:00 – 11:00	2
	11:00 – 13:00	3
	14:00 – 16:00	4
	16:00 – 18:00	5
	18:00 – 20:00	6
Bloque de tres horas	7:00 – 10:00	7
	10:00 – 13:00	8
	14:00 – 17:00	9
	17:00 – 20:00	10
	18:00 – 21:00	11

c) *Time Slots (TS)*

Esta estructura se compone por un par de datos que determinan los días de la semana y las horas del día, respectivamente. Estos usan los números de identificación de los *BH* y *CD*:  $TS = (CD, BH)$ . El número total de *TS* se obtiene mediante producto entre la cantidad total de *bloques horarios* y *combinaciones de días* para un total de 132 *TS*.

Por ejemplo: una asignatura con el  $TS = (8,2)$  implica que se imparte los lunes y viernes desde las 9:00 hasta las 11:00.

d) *Lectures*

Un *Lecture* es la estructura que describe una asignatura de forma que define si una actividad está asignada a un horario determinado. Esta se compone por un número de identificación *id* y un *TS*.

Por ejemplo: el *Lecture* con un  $id = 6$  y un  $TS = (8,2)$  corresponde a que la materia Calculo Integral es asignada los lunes y viernes de 9:00 a 11:00.

e) *Representación gráfica*

El problema de *Curriculum Based Course Timetabling* (CB-CTT) tiene una representación generalizada mediante un grafo no direccionado  $G(V, E)$  conformado por un conjunto de vértices (nodos)  $V$  y un conjunto de aristas (conexiones)  $E$  entre los vértices. La aplicación de esta estructura matemática en el área de *Timetabling* se interpreta de la siguiente manera [30]:

- **Nodos:**

Los nodos representan un *Lecture*, por lo tanto, cada nodo tiene dos atributos: un *id* y un *TS*. El *Time Slot* se interpreta como el color del nodo, por lo tanto, hay 132 colores diferentes.

- **Aristas:**

Las aristas representan un conflicto entre dos *Lectures*. Los conflictos son violaciones a las restricciones duras (*hard constraints*) [31], por lo tanto, involucran situaciones en las que dos *Lectures* no pueden asignarse al mismo tiempo por alguna de las siguientes situaciones:

- Ambas asignaturas las dicta el mismo profesor.
- Ambas asignaturas son propias del mismo semestre dentro del plan de estudios.
- Ambas asignaturas requieren alguno de los salones especiales (sala de cómputo 3.6 o el laboratorio de ingeniería industrial).

La Fig. 13. contiene un ejemplo del problema simplificado representado en un grafo no direccionado con vértices coloreados. Nótese que no hay conexión entre dos nodos del mismo color, por lo tanto, no hay un par *Lectures* que tengan conflictos (violaciones a restricciones duras) y estén asignados al mismo *TS*; asegurando la factibilidad de la solución [31].

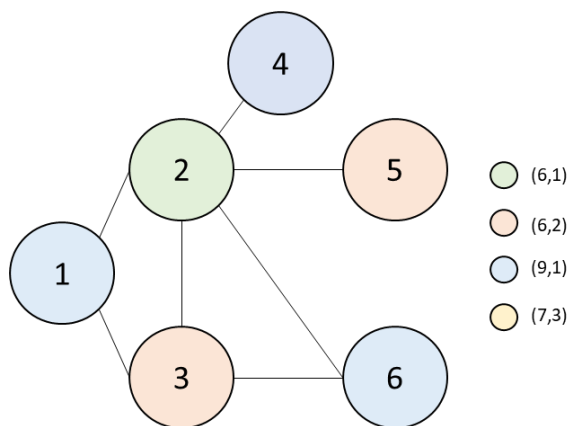


Fig. 13. Colored Conflict Graph

La codificación de la solución empleada es un vector que contiene todas las *Lectures* (*id* y *TS*) [32]. Este formato permite ejecutar reglas de movimiento con facilidad, recorrido de toda la solución en tiempo lineal y acceso a elementos de la solución en tiempo constante. La Fig. 14. muestra el vector solución de la asignación.

ID	1	2	3	4	...	97	98	99	100
TS	(3,8)	(4,11)	(9,1)	(3,8)	...	(10,1)	(11,11)	(7,1)	(3,4)

Fig. 14. Vector solución

En los problemas de *Timetabling* la función objetivo es una suma ponderada donde el costo asociado a cada una de las violaciones a restricciones varía de acuerdo con si es una restricción dura o suave. Las penalidades asociadas a restricciones duras de acuerdo a la literatura deben tener un peso superior relativo a las demás variables, para garantizar que, a la hora de realizar los movimientos en nuestra búsqueda local, priorice la eliminación de los denominados cruces duros [26]. La búsqueda tiene como objetivo maximizar las preferencias de los profesores, por lo que, se suman los valores correspondientes de los *TS* asignados a los *Lectures* de cada profesor. En este proyecto, se consideró que es igual de relevante la satisfacción de las preferencias de los profesores que reducir los cruces suaves por lo que en la función de aptitud no se le asignaron coeficientes que representen un nivel de importancia diferente. Teniendo en cuenta lo anterior, la función de aptitud es calculada de acuerdo con la ecuación (16).

$$Z = \text{SumaPreferencias} - 1000 * \text{CrucesDuros} - \text{CrucesSuaves} \tag{16}$$

## 2. Solución inicial

Para generar un punto de partida como solución inicial se consideraron únicamente soluciones factibles debido a que en cuanto mejor sea la solución inicial mejor será el resultado de la búsqueda, de esta forma se evita que se lleguen a óptimos locales previos a alcanzar factibilidad. El espacio solución es el resultado de todas las combinaciones posibles que se pueden asignar al vector solución; en este caso se le pueden asignar 132 *TS* diferentes a 100 *Lectures* permitiendo repetición, por lo que la cantidad de soluciones en el espacio solución se estima de la siguiente forma:

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!} \quad (17)$$

$$C(132, 100) = \frac{132!}{100!(132-100)!} = 4.55 \times 10^{30} \quad (18)$$

La cantidad de soluciones calculada incluye, en su mayoría, soluciones no factibles, un ejemplo es agendar todas las clases en el mismo bloque horario. Aun así, la cantidad de soluciones válidas sigue siendo lo suficiente grande como para ser inviable estimar la asignación óptima mediante una búsqueda exhaustiva.

En [33] se emplea un método de inicializaron aleatorio, donde se le asignan a todos los *Lectures* un *TS* correspondiente de forma aleatoria. Bajo esta metodología se puede empezar la búsqueda desde cualquier solución, factible o infactible, dejando a la búsqueda como único mecanismo para asignar *TS* que no generan violaciones a las restricciones duras. No obstante, el algoritmo puede iniciar dese un punto alejado de un conjunto de soluciones factibles y quedarse dentro de múltiples óptimos locales infactibles.

Para evitar esta situación se propone una metodología compuesta de dos etapas que asegura factibilidad desde la solución inicial como se realiza en [34] (metodología *B&B* y metaheurística), en nuestro caso:

1. Generación de la solución inicial mediante el modelo matemático por medio de AMPL; empleando un algoritmo de *Branch and Bound (B&B)* y el *solver* de *Cplex*. Esto es posible puesto que *Cplex* es capaz de encontrar una solución factible en un tiempo corto.
2. Búsqueda Local Iterativa (*ILS*) partiendo de la solución resultado de la etapa anterior.

El *Branch and Bound* es una técnica utilizada en optimización para resolver problemas de gran tamaño dividiéndolos en subproblemas más pequeños y abordándolos de manera eficiente. Este enfoque se basa en la idea de encontrar límites superiores e inferiores para las soluciones, lo que permite descartar subproblemas que no pueden contener una solución óptima, y así enfocarse en aquellos que tienen el potencial de mejorar la solución actual. La estrategia se implementa mediante la construcción de un árbol de búsqueda, donde cada nodo representa un subproblema y las ramas del árbol se generan mediante ramificación. Este proceso puede ser de bifurcación (dos ramas) o de ramificación múltiple, dependiendo de las características del problema. A medida que se exploran los nodos del árbol, se actualiza la mejor solución encontrada hasta el momento. El *Branch and Bound* es ampliamente utilizado en problemas de optimización combinatoria, como la planificación de tareas, la asignación de recursos y la programación lineal entera, donde se busca encontrar la solución óptima dentro de un conjunto finito de posibilidades [35].

La Fig. 15. expone la lógica del *B&B*; cabe resaltar que la factibilidad de las soluciones se conserva a lo largo de las ramificaciones puesto que, en caso de infactibilidad se corta la rama y prosigue a un nodo hermano. Esto implica que los caminos creados a hasta los nodos hoja (nodos finales que no tienen hijos) son factibles.

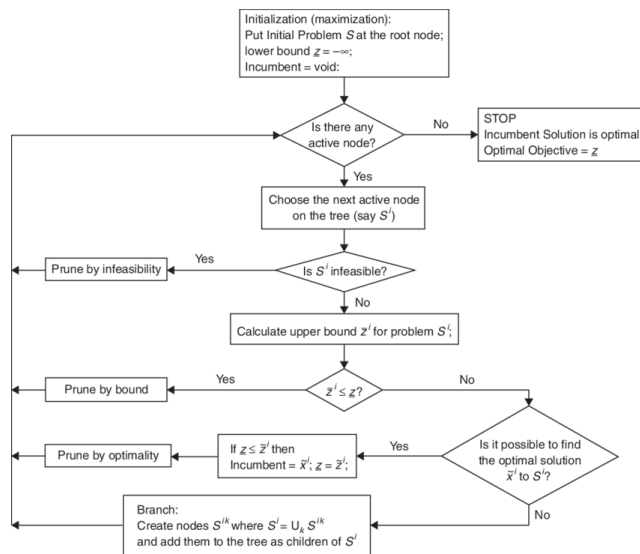


Fig. 15. Diagrama de flujo del algoritmo genérico Branch and Bound [35]

Como consecuencia, el algoritmo *ILS* posterior mantiene la factibilidad desde la solución inicial como su *input* hasta encontrar el óptimo local. Esto es invariante dentro de la búsqueda local al no permitir que una nueva solución contenga violaciones a las restricciones duras, puesto que esto implicaría una penalización en la función de acoplamiento de peso 1.000 unidades.

### 3. Estructuras de vecindario

Como se describe en el apartado C, la búsqueda local es un algoritmo que se basa en la generación de vecindarios a partir de una regla de alteración [33]. El algoritmo inicia partiendo de una solución base para que después, iterativamente, se genere un vecindario mediante la alteración del vector solución (la codificación). De este nuevo vecindario se determina el mejor vecino, como aquel que tiene la función de aptitud (*fitness function*) más alta. Posteriormente se adopta este nuevo vecino como la solución actual, reiniciando el proceso para generar un nuevo vecindario alrededor de esta nueva solución.

Las reglas de alteración determinan el ritmo en que la búsqueda converge a un óptimo local, por lo que diferentes reglas generan diferentes ritmos y caminos en el algoritmo [33]. En primer lugar, se consideraron las estructuras de vecindario que siguen el principio de **explotación**, estas buscan encontrar mejores soluciones utilizando distintas combinaciones con los recursos disponibles actualmente [36]. Al implementar estas estructuras, siempre se estarán considerando los mismos TS actuales, con la intención de encontrar la mejor combinación de estos con los Lectures. A continuación, se describen las estructuras de vecindario usadas en la propuesta.

#### a) Lecture Swap

En esta estructura de vecindario, se analizan todas las combinaciones de parejas posibles entre los Lectures. En cada una de estas combinaciones se intercambian los TS asignados respectivamente y se recalculan los KPIs de la nueva solución. De forma que al recorrer la estructura de vecindarios se selecciona la solución cuya función de aptitud sea mayor, asegurando que, en caso de presentar soluciones infactibles, elija la de menor penalización (ver ALGORITMO III.). El vecindario de esta estructura está conformado por 519 soluciones. Por ejemplo, las asignaturas de Distribución de Plantas y Operaciones II tienen una intensidad de dos veces a la semana, y la duración de sus clases es de dos horas, por lo que a nivel de combinación de días y bloques horarios son compatibles para intercambiar sus TS. Al realizar el intercambio, Operaciones II va a adoptar el TS de Distribución de Plantas y viceversa.

ALGORITMO III

ESTRUCTURA DE VECINDARIO – LECTURE SWAP

---

**Algorithm 4:** Lecture Swap

---

```

Input :  $S \leftarrow$  Current solution
Output: Best solution  $S''$  found so far

InitializeSolution ( $S''$ ) // Initialize as the worst solution
foreach  $L_{first} \in$  TotalLectures do
    // Generate a set of compatible Lectures
    LectureSet  $\leftarrow$  GetCompatibleLectures ( $L_{first}$ )
    foreach  $L_{second} \in$  LectureSet do
        // Swap TimeSlots in the current solution
         $S' \leftarrow$  SwapTS( $S, L_{first}, L_{second}$ )
        if  $f(S') > f(S'')$  then
             $S'' \leftarrow S'$ 
return  $S''$ 

```

---

b) *Event Swap*

Para cada uno de los TS existentes se itera dentro de los TS compatibles, para así seleccionar todos los eventos (*Lectures*) que tienen asignado dicho TS y se intercambian con otro *Lecture* que esté asignado en el TS compatible, una vez realizado el cambio, se vuelven a calcular los KPI's de la solución encontrada (ver ALGORITMO IV). El vecindario de esta estructura está conformado por 1 184 soluciones. Por ejemplo, se selecciona el TS que corresponde al día lunes de 7:00 a 9:00, y se selecciona otro TS compatible, es decir con intensidad de una vez a la semana y con duración de dos horas. Posteriormente se realiza una búsqueda sobre la solución y se identifican los *Lectures* que están asignados a ambos TS por separado. De tal forma que se intercambian todos los eventos (*Lectures*) entre estos dos TS.

ALGORITMO IV

ESTRUCTURA DE VECINDARIO – EVENT SWAP

---

**Algorithm 5:** Event Swap

---

```

Input :  $S \leftarrow$  Current solution
Output: Best solution  $S''$  found so far

InitializeSolution ( $S''$ ) // Initialize as the worst solution
foreach  $TS_1 \in$  TotalTimeSlots do
    // Generate a set of compatible TimeSlots
    TimeSlotSet  $\leftarrow$  GetCompatibleTimeSlots ( $TS_1$ )
    foreach  $TS_2 \in$  TimeSlotSet do
        foreach  $L_{first} \in$  GetSetOfEventsIn( $TS_1$ ) do
            foreach  $L_{second} \in$  GetSetOfEventsIn( $TS_2$ ) do
                // Swap TimeSlots in the current solution
                 $S' \leftarrow$  SwapTS( $S, L_{first}, L_{second}$ )
                if  $f(S') > f(S'')$  then
                     $S'' \leftarrow S'$ 
return  $S''$ 

```

---

c) *Insert TS*

En este caso, a cada uno de los *Lectures* se les asignan todos los TS que sean compatibles y estén disponibles; posteriormente se reevalúan los KPI's de la nueva solución con la inserción (ver ALGORITMO V). Esta estructura explora 2 486 soluciones. Por ejemplo, la materia Sistemas avanzados de Producción, posee una intensidad de una vez a la semana y una duración de tres horas por sesión, en esta estructura de vecindario se explorarían todos los TS que tengan la misma intensidad semanal y horaria.

**Algorithm 6:** Insert TS

---

```

Input :  $S \leftarrow$  Current solution
Output: Best solution  $S''$  found so far

InitializeSolution ( $S''$ ) // Initialize as the worst solution
foreach Lecture  $\in$  TotalLectures do
    // Generate a set of compatible TimeSlots
    TimeSlotSet  $\leftarrow$  GetCompatibleTimeSlots (Lecture)
    foreach TimeSlot  $\in$  TimeSlotSet do
        // Assign TimeSlot to the Lecture in the current solution
         $S' \leftarrow$  AssignTS( $S$ , Lecture, TimeSlot)
        if  $f(S') > f(S'')$  then
             $S'' \leftarrow S'$ 
return  $S''$ 

```

---

## 4. Criterio de perturbación

Con el fin de generar nuevas soluciones que se alejen de la solución actual y permitan explorar diferentes sectores del espacio de búsqueda. En el algoritmo ILS se plantea realizar perturbaciones, estas consisten en introducir cambios aleatorios en la solución de la búsqueda local. Al hacer esto, se aumenta la probabilidad de encontrar nuevas soluciones con estructuras diferentes que pueden conducir a una mejor solución. El criterio de parada se estableció al ejecutar cinco perturbaciones seguidas sin obtener una mejora en la función de aptitud. De acuerdo con la literatura, existen distintos tipos de criterios para hacer perturbaciones [33]:

a) *Random Assignment Perturbation*

En esta perturbación se selecciona un *Lecture* de forma uniformemente aleatoria, para posteriormente asignarle un TS del conjunto con el que el *Lecture* es compatible de forma aleatoria [22] (ver ALGORITMO VI). Cuando se realiza una única perturbación se retorna al punto de partida o bien alcanza un óptimo local ya visitado, lo que hace que el criterio no sea útil, es por esto que se implementó la misma perturbación entre dos y ocho veces, incrementando la probabilidad de cambiar lo suficiente la estructura actual como para explorar otra zona de búsqueda.

**Algorithm 7:** Random Assignment Perturbation

---

```

Input :  $S \leftarrow$  Current solution
Output: Altered solution  $S'$ 

 $U \leftarrow$  UniformDistribution(1, |TotalLectures|)
Lecture  $\leftarrow$  SelectRandomLecture(TotalLectures,  $U$ )

CompatibleTS  $\leftarrow$  GetCompatibleTimeSlots(Lecture)
 $U \leftarrow$  UniformDistribution(1, |CompatibleTS|)
TimeSlot  $\leftarrow$  SelectRandomTS(CompatibleTS,  $U$ )

 $S' \leftarrow$  AssignTS( $S$ , Lecture, TimeSlot)
return  $S'$ 

```

---

b) *Random Swap Perturbation*

Este tipo de perturbación comparte principio con la estructura de vecindario Intercambiar TS, ya que una pareja de *Lectures* elegidos aleatoriamente cambiaran sus respectivos TS [22] (ver ALGORITMO VII), para evitar que en la exploración de vecindarios se retorne al mismo óptimo local, se realizaron entre dos y ocho perturbaciones.

**Algorithm 8:** Random Swap Perturbation

---

**Input** :  $S \leftarrow$  Current solution  
**Output**: Altered solution  $S'$

$U \leftarrow \text{UniformDistribution}(1, |\text{TotalLectures}|)$   
 $L_{first} \leftarrow \text{SelectRandomLecture}(\text{TotalLectures}, U)$

$\text{CompatibleLectures} \leftarrow \text{GetCompatibleTimeSlots}(L_{first})$   
 $U \leftarrow \text{UniformDistribution}(1, |\text{CompatibleLectures}|)$   
 $L_{second} \leftarrow \text{SelectRandomLecture}(\text{CompatibleLectures}, U)$

$S' \leftarrow \text{SwapTS}(S, L_{first}, L_{second})$   
**return**  $S'$

---

c) *Mixed Perturbation*

Finalmente, con la intención explorar vecindarios significativamente diferentes en el espacio de solución, se propone realizar perturbaciones de tipo *a* y *b* de manera conjunta. Para evitar que en la búsqueda local se retorne al óptimo local, se probó realizar de 1 a 10 perturbaciones para evaluar el desempeño del criterio.

D. *Validación del modelo matemático*

Con el fin de corroborar el funcionamiento del modelo previamente descrito, se implementó en AMPL utilizando CPLEX como software de optimización. Inicialmente se ejecutó el modelo sin un tiempo límite, por lo que pasadas aproximadamente 12 horas desde que se efectuó, se tuvo que interrumpir el programa, esto en primer lugar dejó en evidencia no sería posible encontrar una solución exacta u óptima en un tiempo razonable. Por lo que se tuvo que recurrir a la opción de CPLEX ‘timelimit’ que permite correr el modelo con un tiempo límite preestablecido en segundos. Teniendo en cuenta lo anterior, se ejecutó el modelo con un tiempo límite de 60 segundos, al culminar el tiempo, fue posible encontrar una solución factible. (Ver Fig. 16.).

```

602937 MIP simplex iterations
274 branch-and-bound nodes
absmipgap = 71.2564, relmipgap = 0.24828
512 integer variables rounded (maxerr = 9.76439e-09).
Assigning integrality = 5e-09 might help.
Currently integrality = 1e-05.
_nvars = 1256090
_ncons = 3819640

_solve_elapsed_time = 60.141

solve_result_num = 422

```

Fig. 16. Consola AMPL post ejecución del modelo de programación lineal entera-mixta.

La solución fue condensada en 2 nuevas variables para el grupo A y Blas cuales consisten en la variable X pero expresada solo en términos de días, bloques y materias como se ve en las ecuaciones (19) y (20). Para validar el resultado, las nuevas variables se almacenaron en archivos de texto plano y a través del uso de MACROS de Excel fueron transcritos al formato de las soluciones utilizado por la metaheurística como se ve en la Fig. 14. Esto se hizo para verificar que las asignaciones se hicieron de manera correcta verificando los resultados de forma empírica.

$$\sum_{l \in L} \sum_{p \in P} X_{d,b,m,l,p} = G1X_{d,b,m}, \quad \forall d \in D, b \in B, m \in MT \quad (19)$$

$$\sum_{l \in L} \sum_{p \in P} X_{d,b,m,l,2,p} = G2X_{d,b,m}, \quad \forall d \in D, b \in B, m \in MT \quad (20)$$

La solución encontrada es factible, ya que no presentó cruces de profesores, salones ni entre asignaturas del mismo semestre, sin embargo, al observar las preferencias no llegó a la meta de ser mayor a 300, ni a la meta de reducir los cruces suaves a menos de 50, no obstante, esto puede estar asociado al tiempo límite establecido (ver **Error! Reference source not found.**)

TABLA VIII.  
KPI's EJECUCIÓN DEL MODELO MATEMÁTICO CON TIEMPO LÍMITE 60 SEGUNDOS

Z	Preferencias	Número de cruces duros	Número de cruces suaves
198	294	0	96

En segundo lugar, también se ratificó la solución de forma gráfica, como se observa en el Anexo 19. Para lograr obtener los horarios en este formato se realizó una Macro en Excel que recibe la solución utilizada en la metaheurística y organiza los horarios por semestre y grupo. A continuación, en la Fig. 17. se muestra el semestre 4 para el grupo A de la solución encontrada por el modelo matemático. Este método gráfico permite no solo corrobora que se validen las restricciones, sino que cumple como entregable para los *stakeholders*.

	Lunes	Martes	Miércoles	Jueves	Viernes
7			Ingeniería de Métodos LUIS ALONSO VELASCO ROLDAN A Laboratorio		Ingeniería de Métodos LUIS ALONSO VELASCO ROLDAN A Laboratorio
8			Ingeniería de Métodos LUIS ALONSO VELASCO ROLDAN A Laboratorio		Ingeniería de Métodos LUIS ALONSO VELASCO ROLDAN A Laboratorio
9		Cálculo Multivariable Cálculo Multivariable A A Salon regular		Cálculo Multivariable Cálculo Multivariable A A Salon regular	
10		Cálculo Multivariable Cálculo Multivariable A A Salon regular		Cálculo Multivariable Cálculo Multivariable A A Salon regular	
11	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular	Estática y Resistencia de Mate Estática y Resistencia de Mate A A Salon regular	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular	Estática y Resistencia de Mate Estática y Resistencia de Mate A A Salon regular	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular
12	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular	Estática y Resistencia de Mate Estática y Resistencia de Mate A A Salon regular	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular	Estática y Resistencia de Mate Estática y Resistencia de Mate A A Salon regular	Teoría de Probabilidades Teoría de Probabilidades A A Salon regular
13					
14	Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular		Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular		Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular
15	Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular		Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular		Física Térmica y Ondulatoria Física Térmica y Ondulatoria A A Salon regular
16					
17					
18					
19					
20					
21					

Fig. 17. Ejemplo resultado gráfico

E. Validación de la metaheurística

De igual forma, para corroborar el funcionamiento del algoritmo metaheurístico descrito, se analizó el desempeño de las estructuras de vecindarios y la evaluación de los criterios de perturbación.

1. Estructuras de Vecindarios

Para realizar la validación de la estructura de vecindarios se plantea un diseño de experimentos robusto basado en el método Taguchi, este analiza cómo los diferentes parámetros afectan la media y la varianza de una característica de rendimiento. El diseño experimental propuesto por Taguchi implica el uso de arreglos ortogonales para organizar los parámetros que afectan el proceso y los niveles en los que deben variar. Esto permite hacer pruebas para distintas combinaciones y con base a esto, determinar qué factores afectan más a la calidad del producto con una cantidad mínima de experimentación, ahorrando así tiempo y recursos [37].

Para la implementación del método se utilizó el software estadístico Minitab, se ingresaron primeramente los factores controlables (Ver TABLA IX), estos indican las características del diseño que se están evaluando y pueden ser combinadas entre estas, en este caso los factores poseen dos niveles que expresan si el factor es aplicado o no.

TABLA IX

FACTORES CONTROLABLES ESTRUCTURAS DE VECINDARIOS

Factores controlables	Niveles	
	1	2
Lecture Swap	Se aplica	No se aplica
Event Swap	Se aplica	No se aplica
Insert TS	Se aplica	No se aplica

En segundo lugar, los factores de ruido son aquellas condiciones que no hacen parte del diseño, pero sirven para probar las estructuras de vecindario en condiciones distintas, en este caso, los factores de ruido están dados por seis soluciones iniciales que varían en su tiempo de ejecución como se ve en la TABLA X.

TABLA X

FACTORES DE RUIDO ESTRUCTURAS DE VECINDARIOS

Factores de ruido	Niveles					
	1	2	3	4	5	6
Tiempo ejecución AMPL para solución inicial	10	20	30	40	50	60

Por último, la función de aptitud formulada en (16), resultante del óptimo local encontrado al utilizar las combinaciones de estructuras de vecindario, en la TABLA XI se observan en las filas las combinaciones generadas por el método Taguchi, que, al ser solo tres factores con 2 niveles, se probaron todas las combinaciones posibles. Además, en el análisis se considera un nivel de significancia del 0,05 el cual indica un riesgo del 5% de concluir que existe una asociación cuando no hay una relación real entre los factores controlables y los resultados.

TABLA XI

DATOS PARA EL DISEÑO TAGUCHI, ESTRUCTURAS DE VECINDARIOS

Factores controlables			Factores de ruido					
			1	2	3	4	5	6
Lecture Swap	Event Swap	Insert TS	10	20	30	40	50	60
1	1	1	271	275	276	276	275	275
1	1	2	218	225	215	215	220	220
1	2	1	276	272	271	271	266	266
1	2	2	216	223	208	208	208	208
2	1	1	269	272	273	277	271	271
2	1	2	218	226	216	216	222	222
2	2	1	261	261	261	261	260	260
2	2	2	197	200	185	185	186	186

Al analizar el diseño de *Taguchi*, como se ve en la Fig. 18. y Fig. 19., para cada factor controlable se elige el nivel 1, es decir que el diseño final contará con las tres estructuras de vecindarios.

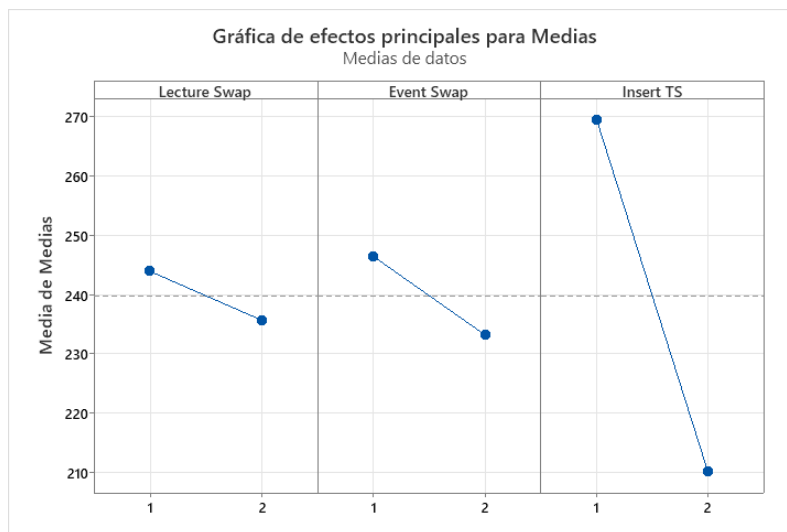


Fig. 18. Resultados diseño Taguchi para estructuras de vecindarios, efectos principales para medias

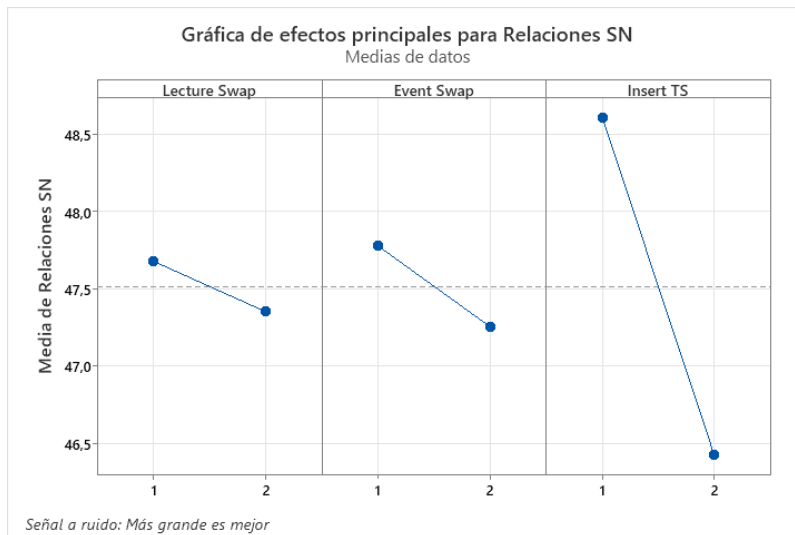


Fig. 19. Resultados diseño Taguchi para estructuras de vecindarios, efectos principales para Relaciones SN

Para determinar en qué porcentaje el modelo compuesto por los tres factores controlables explican la variabilidad de los datos de respuesta, se analiza la medida estadística R-cuadrado, el cual revela que tan cerca están los datos de la línea de regresión ajustada. Como se ve en la Fig. 20, el 92,80% de la variabilidad observada en la respuesta deseada puede ser explicada por los factores incluidos en el diseño experimental. Además, esto implica que el modelo captura la mayoría de los patrones y tendencias en los datos, lo que lo hace útil para predecir la respuesta deseada dentro del rango de los factores evaluados en el diseño experimental.

### Resumen del modelo

S	R-cuadrado	R-cuadrado(ajustado)
0,3317	95,89%	92,80%

Fig. 20. R-cuadrado del diseño Taguchi para las estructuras de vecindarios

Al observar el nivel de significancia de las distintas estructuras de vecindarios, como se ve en la Fig. 21., *Insert TS* es la estructura más significativa, seguida por *Event Swap*, sin embargo, *Lecture Swap* no es considerada significativa, por lo que de acuerdo con su valor P, podría ser excluida del diseño, no obstante, debido a que en la gráfica de efectos principales (Ver Fig. 18.) se determinó el nivel 1, *Lecture Swap* debe ser elegido, es decir, que se considera mejor incluirla en el diseño.

### Coefficientes de modelos estimados para Relaciones SN

Término	Coef	EE del coef.	T	P
Constante	47,5172	0,1173	405,130	0,000
Lecture 1	0,1631	0,1173	1,391	0,237
Event Sw 1	0,2617	0,1173	2,231	0,090
Insert T 1	1,0900	0,1173	9,294	0,001

Fig. 21. Coeficientes de modelos estimados del diseño Taguchi para las estructuras de vecindarios

Para analizar la relevancia de incluir la estructura de vecindarios *Insert TS*, se realizó la prueba de diferencia de medias de la combinación ganadora, que incluye las tres estructuras (1,1,1) con la combinación que no incluye *Insert TS* pero sin las otras dos estructuras (2,1,1). Como se ve en la Fig. 22., se puede observar que las medias de los resultados obtenidos con ambas combinaciones son significativamente diferentes con un 10% de riesgo de que sean iguales. Además, la gráfica muestra como la combinación (1,1,1) presenta un mejor promedio y menor desviación estándar, por lo que refuerza el resultado obtenido en el diseño de *Taguchi*.

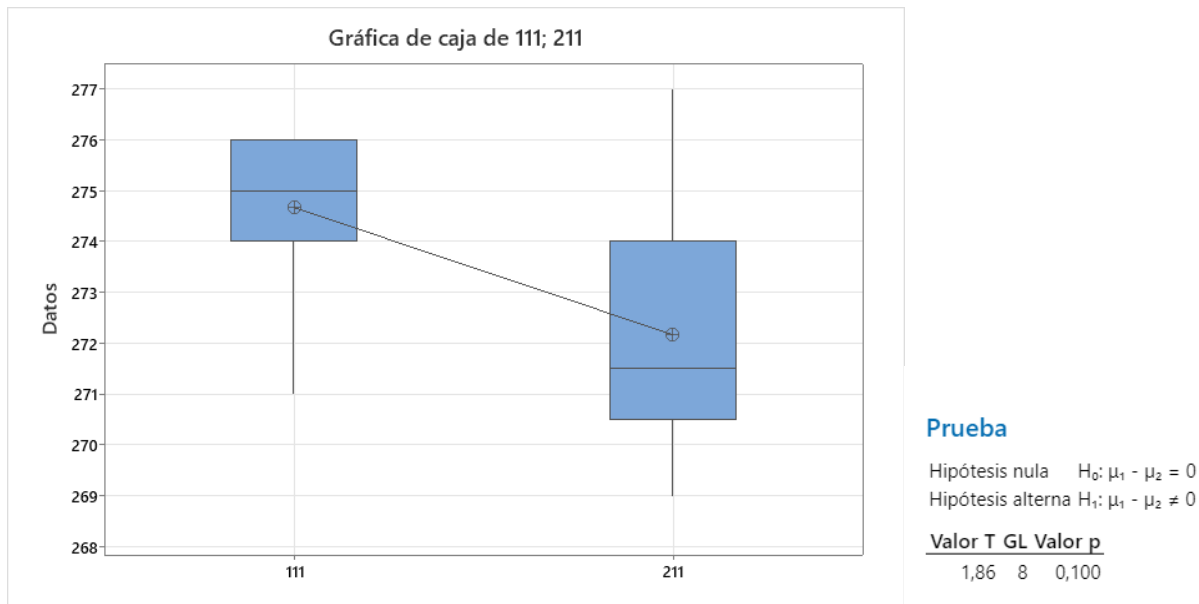


Fig. 22. Prueba de diferencia de medias entre la combinación (1,1,1) y (2,1,1)

Es pertinente destacar, tal como se ve en la TABLA XII, que entre más soluciones explora cada estructura, mayor fue el nivel de significancia de esta, demostrando así que en cuanto mayor sea el número de vecinos en una estructura de vecindarios, mayor será la capacidad de exploración del espacio de búsqueda. Esto permite examinar un conjunto más amplio de soluciones cercanas en cada iteración de la búsqueda local, lo que aumenta las posibilidades de encontrar una solución óptima o de alta calidad. Además, al tener un vecindario más grande, se aumenta la probabilidad de encontrar soluciones fuera de los óptimos locales.

TABLA XII

RELACIÓN ENTRE NUMERO DE VECINOS POR ESTRUCTURA CON SU NIVEL DE SIGNIFICANCIA

Estructura	Número de vecinos	Valor P
Lecture Swap	519	0,237
Event Swap	1 184	0,090
Insert TS	2 486	0,001

De acuerdo con lo anterior, se analizó la relación entre número total de vecinos explorados en cada combinación con los resultados obtenidos (Z), como se ve en la Fig. 23, al realizar la gráfica de línea ajustada por regresión se obtuvo que la función de aptitud final tiene una relación directa con el número total de vecinos, esta relación es explicada a un 90,9% a partir de la ecuación de una recta. Demostrando así que entre mayor sea la cantidad de vecinos distintos en una estructura de vecindarios, mejor será el desempeño de la búsqueda local.

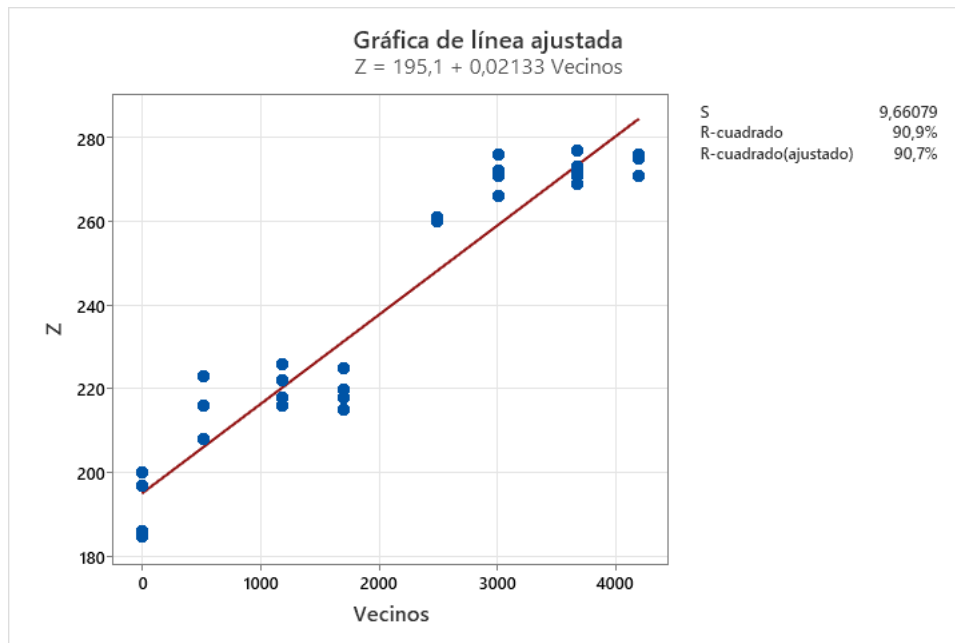


Fig. 23. Relación entre número de vecinos y valor de la función objetivo

## 2. Criterios de perturbación

Para realizar la validación de los criterios de perturbación se plantea nuevamente un diseño de experimentos robusto basado en el método *Taguchi*. De igual forma que en las estructuras de vecindarios, en primer lugar, se definieron los factores controlables (Ver TABLA XIII), los cuales indican las características del diseño que se están evaluando y pueden ser combinadas entre estas, en esto caso los factores poseen cinco niveles que representan las veces que serán aplicados los criterios de perturbación.

TABLA XIII

FACTORES CONTROLABLES CRITERIOS DE PERTURBACIÓN

Factores controlables	Niveles				
	1	2	3	4	5
Random Assignment Perturbacion	0	2	4	6	8
Random Swap Perturbation	0	2	4	6	8

En segundo lugar, los factores de ruido son aquellas condiciones que no hacen parte del diseño y que nos sirven para probar las estructuras de vecindario en condiciones distintas, en este caso, los factores de ruido están dados por las 6 soluciones iniciales utilizadas en la validación de las estructuras de vecindarios (Ver TABLA X).

Por último, la característica de rendimiento es la función de aptitud formulada en (16), en la TABLA XIV se observan en las filas las combinaciones generadas por el método *Taguchi*, que, al ser solo dos factores con cinco niveles, se probaron todas las combinaciones posibles. Además, en el análisis se considera un nivel de significancia del 0,05 el cual indica un riesgo del 5% de concluir que existe una asociación cuando no hay una asociación real.

TABLA XIV

DATOS PARA EL DISEÑO TAGUCHI, CRITERIOS DE PERTURBACIÓN

Factores controlables		Factores de ruido					
		1	2	3	4	5	6
Random Assignment Peturbation	Random Swap Perturbation	10	20	30	40	50	60
1	1	271	275	276	276	275	275
1	2	271	276	276	278	275	275
1	3	271	275	277	276	275	275
1	4	271	276	276	276	279	275
1	5	271	280	276	276	278	275
2	1	279	275	276	276	275	278
2	2	279	275	276	276	275	275
2	3	272	278	276	277	278	275
2	4	278	276	276	276	279	276
2	5	271	276	278	276	281	275
3	1	271	276	276	276	279	275
3	2	271	275	276	276	275	275
3	3	271	276	276	276	275	275
3	4	271	280	278	275	275	276
3	5	271	277	282	276	279	275
4	1	272	276	276	276	278	275
4	2	271	277	276	279	278	275
4	3	271	275	276	276	275	275
4	4	271	275	277	276	278	275
4	5	271	275	277	278	279	275
5	1	272	278	276	276	275	275
5	2	271	276	276	280	275	275
5	3	271	277	275	277	275	275
5	4	271	275	276	276	275	278
5	5	271	276	276	276	275	278

Al analizar el diseño de *Taguchi*, como se ve en la Fig. 24 y Fig. 25, para cada factor controlable se elige de la siguiente manera: el nivel dos para el *Random Assignment* y el nivel cinco para el *Random Swap*, es decir que al momento de hacer las perturbaciones una vez se llegue a un óptimo local, se deben hacer dos y ocho respectivamente.

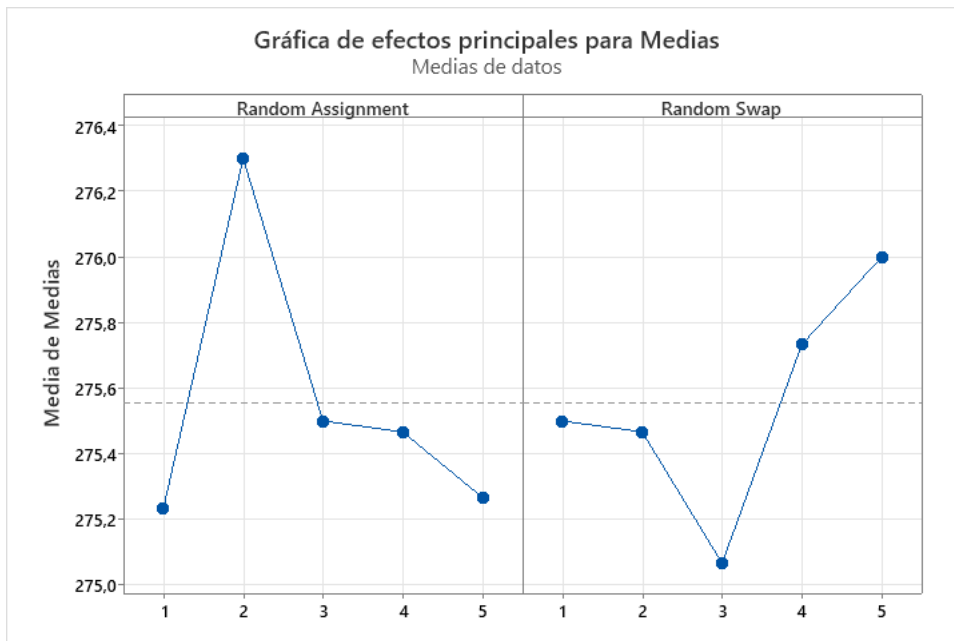


Fig. 24. Resultados diseño Taguchi para criterios de perturbación, efectos principales para medias

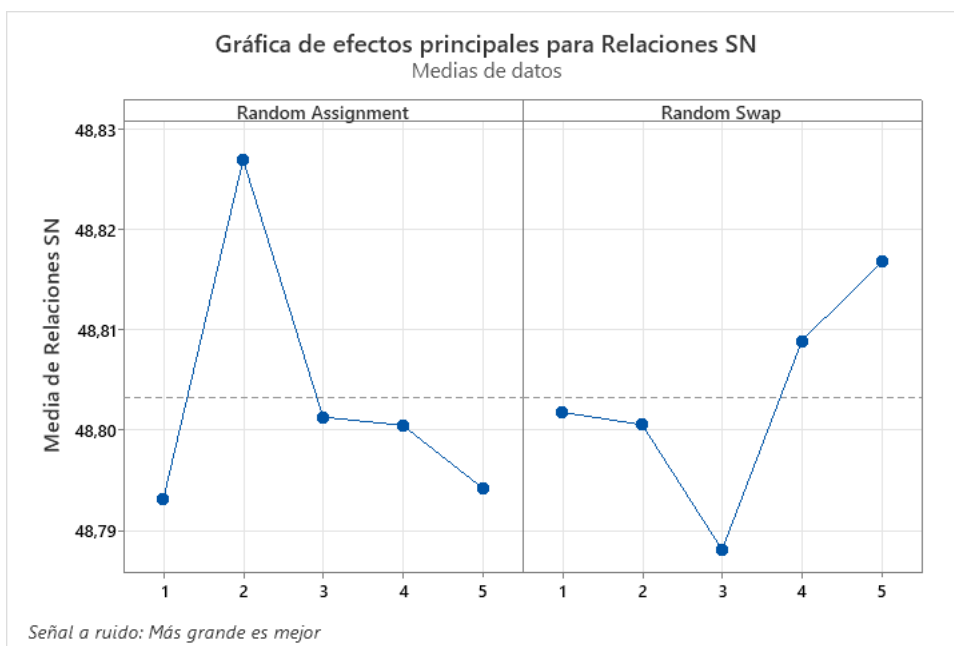


Fig. 25. Resultados diseño Taguchi para criterios de perturbación, efecto principales para Relaciones SN

Para determinar en qué porcentaje el modelo compuesto por los dos factores controlables explican la variabilidad de los datos de respuesta, se analiza la medida estadística R-cuadrado, el cual deja ver que tan cerca están los datos de la línea de regresión ajustada. Como se ve en la Fig. 26, el 50,14% de la variabilidad observada en la respuesta deseada puede ser explicada por los factores incluidos en el diseño experimental. Por lo que aproximadamente la mitad de la variabilidad observada en la respuesta deseada puede ser explicada por los factores incluidos en el diseño experimental. Esto indica que aún existe una proporción significativa de variabilidad no explicada por los factores incluidos en el diseño. Esto sugiere que puede haber otros factores no

evaluados o interacciones complejas que también influyen en la respuesta y deben ser considerados en futuros análisis o diseños experimentales. Este valor revela que las perturbaciones no generan cambios considerables en la función objetivo, sin embargo, teniendo en cuenta el contexto en que se hacen las perturbaciones, estas se realizan una vez se encuentra un óptimo local, por lo que, a pesar de que no se presenten cambios en todas las ocasiones, cualquier mejora, aunque sea mínima debe ser tenida en cuenta. En la experimentación, la mayor variación con respecto al óptimo local alcanzado al culminar la búsqueda local implementando las tres estructuras de vecindarios descritas anteriormente, fue un aumento del 2,87%, lo cual desde el punto de vista de la búsqueda local iterativa es un acercamiento al óptimo global.

### Resumen del modelo

S	R-cuadrado	R-cuadrado(ajustado)
0,0137	66,76%	50,14%

Fig. 26. R-cuadrado del diseño Taguchi para los criterios de perturbación

Al observar el nivel de significancia de los distintos criterios de perturbación, como se ve en la Fig. 27., ambos criterios son considerados significativos, por lo que impactan la variable de respuesta.

### Análisis de Varianza de Relaciones SN

Fuente	GL	SC Sec.	SC Ajust.	MC Ajust.	F	P
Random Assignment	4	0,003776	0,003776	0,000944	5,02	0,008
Random Swap	4	0,002270	0,002270	0,000567	3,02	0,050
Error residual	16	0,003010	0,003010	0,000188		
Total	24	0,009056				

Fig. 27. Coeficientes del diseño Taguchi para los criterios de perturbación

El hecho de que todos los factores sean estadísticamente significativos sugiere que cada factor individualmente tiene un impacto en la respuesta deseada. Esto implica que los cambios en los niveles de los factores pueden afectar la respuesta y que se deben considerar en el proceso. Sin embargo, dado que solo se ha explicado la mitad de la variabilidad observada en la respuesta deseada, existen oportunidades para mejorar aún más la calidad o el rendimiento de los criterios de perturbación. Es posible que se requieran ajustes adicionales en los niveles de los factores o la inclusión de nuevos factores para lograr una mayor explicación de la variabilidad y una optimización más precisa de la respuesta.

## VI. VERIFICAR

### A. Medición de Impactos y Resultados

#### 1. Resultados

##### a) Casos de prueba

Para el proyecto, es fundamental respaldar los enfoques propuestos en la etapa DISEÑAR: un modelo matemático y una metaheurística. Para la verificación es importante probar y validar el desempeño de las alternativas considerando diferentes casos de prueba, esto con el fin de medir y evaluar el rendimiento en una variedad de escenarios, y determinar el comportamiento de ambos enfoques bajo distintas condiciones. Otro punto que considerar es la eficacia, ya que los casos de prueba proporcionan una oportunidad para verificar si los diseños son eficaces para resolver el problema de asignación, al probar los distintos escenarios se pretende identificar las fortalezas y debilidades de cada alternativa. Por último, los casos tienen como finalidad evidenciar la robustez de los diseños, garantizando así su utilidad bajo otras condiciones.

Para la elaboración de los casos de prueba, se partió del caso real del programa de ingeniería industrial y se modificaron aspectos relacionado a los cruces duros específicamente por salones, ya que se pretende analizar el desempeño de las alternativas al abordar casos con un número distinto de restricciones duras que afectan directamente la factibilidad al alterar el espacio de solución. No se alteraron características como el número de asignaturas o de profesores, porque dado que se trata de un problema de asignación de tipo *CTT*, al cambiar esto dificultaría la comparación de los KPI's como las preferencias que dependen directamente de los profesores actuales o de los cruces suaves por materias que se pueden adelantar o atrasar en el plan de estudios.

Como se mencionó en la descripción del diseño, el problema se puede ver representado a partir de un grafo no direccionado (Ver Fig. 13.), donde las aristas representan los conflictos entre materias, a continuación, en la TABLA XV se observa el nivel de restricción de cada caso haciendo referencia al número de conflictos existentes en cada problema.

TABLA XV.

COMPLEJIDAD DE LOS ESCENARIOS

Escenario	Número de cruces duros o arcos entre nodos
a. Suavizado	642
b. Actual	676
c. Restringido	708

b) Comparación de alternativas

Con el propósito de verificar el rendimiento computacional del algoritmo ILS como método de búsqueda aproximado se realizó una comparación entre este y la implementación del AMPL (*Branch and Bound*). Posteriormente se inició la búsqueda simultáneamente hasta que el ILS encontrara el óptimo local; en ese momento ambos procedimientos fueron detenidos. Se registraron los *KPI's* junto con el tiempo que estuvieron buscando en segundos. Los resultados de la comparación para cada uno de los casos de prueba se ven a continuación en la TABLA XVI:

TABLA XVI

COMPARACIÓN *KPI's* ENTRE ALTERNATIVAS Y ESCENARIOS

AMPL					
Escenario	Tiempo [s]	Z	Preferencias	Curces duros	Cruces suaves
a	12	187	286	0	99
b	13	186	281	0	95
c	11	209	296	0	87

ILS					
Escenario	Tiempo [s]	Z	Preferencias	Curces duros	Cruces suaves
a	12	281	336	0	55
b	13	276	333	0	57
c	11	281	331	0	50

Se observa que el algoritmo de ILS tiene un mejor rendimiento en el tiempo de ejecución que el APML al converger a un óptimo local con una función de acoplamiento mayor en los tres casos de prueba. En promedio la metaheurística tiene una función de acoplamiento 85 unidades sobre el segundo método, resultante de una mejora del 15,9% en cuanto a la suma de preferencias cumplidas y una disminución del 42,3% en cuanto a los cruces suaves. De acuerdo con esto se concluye que la metaheurística genera soluciones en el mismo tiempo que el método de AMPL.

2. Medición de impactos

A pesar de que el proyecto es desarrollado bajo la metodología de investigación aplicada, y de que el objetivo se centra en diseñar y comparar el desempeño del modelo matemático junto con la metaheurística, se deben tener en cuenta las consecuencias ya sean positivas o negativas al implementar el diseño en el contexto real de una universidad como es el caso de la Pontificia Universidad Javeriana, en específico en la Facultad de Ingeniería y Ciencias para el programa de ingeniería industrial. Para analizar los impactos se considera el enfoque financiero y social.

a) Impacto financiero

En cuanto a los impactos financieros del proyecto, en primer lugar, se tiene que dado que el tiempo requerido desde organizar la información, hasta obtener la asignación horaria final se redujo de 80 horas al semestre a tan solo 30 minutos considerando los ajustes a los parámetros que puedan variar de un semestre a otro, representando una reducción del 99,37%. Considerando las horas de trabajo como un recurso que le cuesta a la universidad, es un logro significativo en términos de eficiencia y productividad.

Al requerir menos tiempo para llevar a cabo la asignación de recursos, se libera una cantidad significativa de tiempo que puede ser utilizado en otras tareas o proyectos. Esto aumenta la productividad general del área administrativa de la Facultad de Ingeniería

y Ciencias, ya que los recursos previamente asignados a la tarea de organizar los horarios ahora pueden enfocarse en otras actividades de mayor valor añadido. Al reducir drásticamente el tiempo, se mejora la capacidad de respuesta de la Facultad ante cambios o solicitudes de asignación inmediatas. Esto permite una mejor optimización de los recursos disponibles, evitando la subutilización o sobreutilización de estos como se ve en la Fig. 2., lo que a su vez puede generar ahorros adicionales y una mayor eficiencia en la utilización de los recursos.

Al automatizar y agilizar el proceso, se facilita la obtención de información actualizada y precisa sobre la disponibilidad y asignación de horarios. Esto proporciona una base sólida para la toma de decisiones informadas y estratégicas, lo que puede tener un impacto positivo en la planificación y ejecución de proyectos, evitando posibles retrasos, conflictos o ineficiencias.

Por otro lado, la implementación del lenguaje de programación para ejecutar el modelo matemático tiene un costo asociado, a continuación, se presentan las opciones para adquirir estos lenguajes.

- **Opción 1:** Licencias de los *solvers* que utiliza *AMPL* como *Cplex* o *Gurobi*. Ambos *solvers* tienen como objetivo encontrar soluciones óptimas, sin embargo, como se observó en el desarrollo del proyecto, se ve que el modelo lineal converge rápidamente a soluciones factibles, sin embargo, no alcanza optimalidad. Los tipos de licencia que se pueden adquirir son los siguientes:

**Licencia de usuario único:** \$4 750 dólares.

**Licencia de Sesión Única:** \$7 250 dólares.

**Licencia multiusuario:** \$12 750 dólares [38].

- **Opción 2:** Se puede utilizar el *solver* de manera gratuita, el cual es llamado *AMPL Community Edition*. Esta versión caduca a los 30 días de ser instalada además debe ser utilizada con conexión a internet. En esta opción se pueden utilizar *solvers* como *Cplex* o *Gurobi* [38].
- **Opción 3:** En lugar de adquirir alguna licencia para ejecutar el modelo matemático, se puede recurrir a la página web *NEOS (Network-Enabled Optimization System)* la cual es una plataforma en línea que proporciona acceso a una amplia gama de recursos y herramientas para la optimización matemática. Dentro de *NEOS* se pueden encontrar *solvers* como *Cplex* o *Gurobi*. El uso de la página web no tiene ningún costo asociado. Otra alternativa sin costo puede ser el *solver* gratuito *Gusek* el cual permite abordar modelos matemáticos de distintas categorías [39].

Por último, es pertinente considerar en la capacitación de las personas encargadas para el correcto uso tanto de los archivos ejecutables como de la adaptación de los parámetros en caso de ser necesarios. Esto con el fin de garantizar el uso continuo tanto del modelo como de la metaheurística.

#### *b) Impacto social*

El impacto social que conlleva la implementación del proyecto se concentra en dos aspectos propios de los profesores y estudiantes al aumentar su conformidad en la institución.

El KPI de cumplimiento de preferencias de los profesores indica la conformidad que tienen los maestros al dar clase en los horarios que se les asignaron, por lo tanto, se está dando prioridad a su satisfacción en el horario laboral. Es importante volver a mencionar que cada profesor dio una calificación a todas las horas de la semana en las que se puede dictar clase, lo cual el algoritmo asigna mediante la maximización del KPI. De acuerdo con [40] la satisfacción de los empleados en el espacio laboral tiene relación directa con su calidad de vida fuera y dentro del trabajo como también su productividad, de manera que el proyecto influencia positivamente la vida laboral y personal de los profesores.

Similarmente, uno de los requerimientos de los interesados, como se menciona anteriormente, es la minimización de los cruces suaves. Estos son cruces entre materias específicas pertenecientes al semestre superior e inferior del semestre cursado actualmente por un estudiante, cuyo índice de repetición y tendencia a ser adelantadas es alto. Debido a esto, los estudiantes tendrán la facilidad de matricular estas asignaturas con menor probabilidad de cruzarse con las asignaturas del semestre propio, facilitando su recorrido por la universidad al minimizar los atrasos por conflictos en la matrícula.

#### *B. Estandarización de la solución*

Al estandarizar el diseño propuesto, se crea una estructura coherente y consistente que permite a cualquier usuario comprender y utilizar la propuesta de manera más eficiente. Al seguir un conjunto de estándares predefinidos, se reduce la curva de aprendizaje

y se evita la necesidad de capacitaciones intensivas. A continuación, se establece en un lenguaje común las indicaciones para operar los entregables del proyecto.

Para garantizar el funcionamiento del diseño, se hará entrega del documento en Excel llamado “Parámetros” (Anexo 23), en este archivo se encuentra la información relevante para ejecutar tanto el modelo matemático como la metaheurística. Dentro de la información se tiene:

- Nombre de la asignatura
- Número que identifica la asignatura
- Intensidad semanal
- Duración de cada sesión de clase (horas)
- Tipo de salón requerido
- Profesor que dicta la materia en el grupo A
- Profesor que dicta la materia en el grupo B
- Semestre en que se encuentra la asignatura en el plan de estudios
- Semestre en que se puede atrasar o adelantar la materia

El usuario o persona encargada de actualizar la información, solo debe realizar los cambios en esta hoja, una vez terminada la etapa de actualización de datos, se debe ejecutar la macro principal con la cual se obtendrán los datos de manera organizada para sustituir los anteriores en el archivo .dat en el caso del modelo matemático, y en el caso de la metaheurística en las demás hojas del Excel se encontrarán los mapas y matrices utilizadas para el algoritmo.

En cuanto a los programas necesarios para ejecutar el diseño propuesto, para el modelo matemático se debe tener una de las opciones mencionadas en los impactos financieros, ya sea alguna de las versiones de AMPL descargada o si se va a usar la página web de NEOS. Para la metaheurística, el algoritmo diseñado en C++ puede ser implementado en cualquier sistema operativo, ya sea Windows, Mac (Apple) o Linux, y se utilizarían compiladores como g++, Xcode y Code:Blocks respectivamente para cada sistema operativo.

Es necesario recalcar que para ejecutar tanto el modelo matemático como la metaheurística, la persona encargada no necesariamente debe tener conocimientos en el uso de estos Softwares o entornos de desarrollo integrado (IDE), por lo que solo se le debe capacitar para ejecutar los programas.

Una vez terminada la ejecución, los resultados serán impresos en archivos txt, los cuales utilizando las macros del documento de Excel ubicado en el Anexo 29, podrán convertir la solución de forma vectorial a la solución grafica como se ve en la Fig. 17.

### *C. Conclusiones*

En este proyecto se abordó la problemática conocida como asignación de horarios o Timetabling, este es un problema combinatorio, desafiante, laborioso y repetitivo en distintos sectores; este trata de ordenar una cantidad finita de personal, espacios o recursos a bloques horarios. En el University Course Timetabling (UCT) los cursos deben tomar lugar y distribuir las sesiones académicas evitando cruces lógicos como: un profesor no puede dictar dos materias a la misma hora, un estudiante no puede asistir a dos clases simultáneamente, entre otros. Las universidades e instituciones educativas de nivel superior en Colombia optan por resolver estos problemas de manera manual mediante la prueba y error. Debido a la volátil calidad de las soluciones y complejidad de tipo NP-Hard, la metodología actual en las instituciones usualmente resulta en soluciones lejanas al óptimo. Para abordar este problema en el Programa de Ingeniería industrial en la PUJ como caso aplicado, se diseñó un algoritmo de dos etapas: una metaheurística de tipo búsqueda local iterativa y un método de inicialización a partir de un modelo de programación lineal entera-mixta.

La comprensión completa de los requerimientos y características del sistema actual es fundamental para el desarrollo de estrategias efectivas de mejora, esto se evidenció en la identificación de las asignaturas que podían atrasarse o adelantarse al avanzar en el plan de estudios, ya que inicialmente estos datos no fueron brindados por parte de la Facultad, por lo que se realizó la solicitud para conocer la tasa de cancelación y pérdida de las asignaturas. Por lo que, al obtener una visión clara de las necesidades y características existentes, es posible identificar áreas de oportunidad y diseñar soluciones específicas y adaptadas

a los desafíos particulares del sistema. Además, la cuantificación precisa de los parámetros relevantes proporciona una base sólida para el análisis y la toma de decisiones informadas. Al contar con datos concretos y confiables, es posible evaluar el rendimiento del sistema, identificar tendencias y patrones, y establecer objetivos realistas y medibles para la mejora continua.

Al culminar los diseños propuestos y realizar su respectiva comparación, el resultado final fue la implementación de un sistema compuesto por una metaheurística de búsqueda local iterativa que toma como input una solución inicial desarrollada por un algoritmo de *Branch and Bound* mediante AMPL. Este sistema maximiza las preferencias de los profesores y minimiza los cruces ente asignaturas que tienen un alto índice de repetición o adelantamiento. Al comparar los KPI's del estado actual y al implementar la metaheurística se obtiene una mejora del 14,8% en cuanto a la suma de preferencias y una eliminación total de los cruces duros ocasionados por conflictos de profesores, salones o misma ubicación semestral. En cuanto a los cruces suaves, debido a que en la medición de los KPI's resultantes de la aplicación de la metaheurística se realizó bajo el supuesto de contar con dos grupos por asignatura a pesar de que en realidad pudiera tener un solo grupo, no hubo cambios con respecto a los cruces suaves a nivel numérico, sin embargo, el hecho de que con el supuesto obtenga el mismo número de cruces suaves a pesar de que desde el punto de vista del gráfico no direccionado en el diseño haya un mayor número de nodos y conflictos, deja ver que el algoritmo posee un buen desempeño en este aspecto.

Al observar la eficacia de cada una de las alternativas planteadas, se comprueba que el modelo matemático como técnica exacta su éxito depende del número de variables, restricciones y características propias del modelo[8] por lo que al tener 1'256.090 variables y 3'819.640 restricciones, siendo un problema combinatorio, se demostró que no era el método indicado en términos de convergencia, sin embargo, a pesar de que el modelo matemático no converge a una solución óptima en un tiempo razonable, si logra obtener resultados factibles en un promedio de diez segundos. Por otro lado, como técnica aproximada, la búsqueda local iterativa converge a mejores soluciones en un menor tiempo computacional.

Por último, la integración de estas herramientas fue fundamental para el desarrollo exitoso de este proyecto. La combinación del diseño de experimentos Taguchi, la modelación matemática, la programación y el análisis estadístico permitió abordar los desafíos desde diferentes perspectivas y obtener resultados sólidos y confiables. Esta integración proporcionó una metodología sólida y coherente para el desarrollo del proyecto, permitiendo la toma de decisiones informadas y la obtención de soluciones eficientes.

#### *D. Recomendaciones*

Teniendo en cuenta la delimitación del problema inicial, al abordar solamente el programa de ingeniería industrial se constató que la complejidad computacional llevó a usar herramientas de optimización no exactas, sin embargo, con el fin de lograr un mayor impacto en la Pontificia Universidad Javeriana, se sugiere adaptar el diseño para abordar la asignación de horarios a nivel de Facultades, lo cual aunque aumenta la complejidad del problema, tendría un impacto mayor en la organización además de que se podrían gestionar de manera más precisa el uso de espacios compartidos entre distintas carreras o el hecho de hayan profesores que dictan asignaturas en más de un programa a la vez.

Con el fin de independizar la metaheurística del modelo matemático, se propone la implementación en el apartado de la generación de la solución inicial un algoritmo de coloreo de grafos usando el concepto de *Back tracking*, el cual sigue el principio de búsqueda exhaustiva del *Branch and Bound* que utilizan los *solvers* de AMPL como Cplex o Gurobi. La razón de ser de la propuesta es simplificar el uso del diseño ganador, evitando la etapa de ejecución en AMPL con el fin de obtener la solución inicial para ser usada en la metaheurística. Además, al eliminar el uso de AMPL no se tendrían que adquirir las licencias presentadas en el apartado del Impacto económico.

En cuanto a los supuestos usados en la etapa de diseño, para la implementación en el contexto real, podrían utilizarse el número de grupos por asignatura definido para el periodo académico actual, ya que al estar considerando dos grupos por asignatura sin importar que la materia real solo disponga de uno, se está restringiendo el espacio de solución, ya que desde el punto de vista de la representación gráfica del problema como el grafo no direccionado, se están añadiendo nodos y aristas que complejizan el problema, por lo que se no se están explorando combinaciones que podrían llevar a una mejor solución.

Los problemas de asignación ya sea de recursos, personal o materiales; están presentes en todos los niveles de la industria, los diseños propuestos a pesar de que se elaboraron específicamente para *Curriculum Based Course Timetabling* pueden ser interpolados a otros problemas de asignación que cuenten con características similares al caso desarrollado, los puntos clave que aporta el diseño a los problemas de asignación son el uso de preferencias y cruces suaves relacionados a atrasos y adelantos, esta última característica podría resultar de utilidad para contextos donde se trabajen con ventanas de tiempo y haya una probabilidad asociada a que la asignación pueda variar en su ubicación ya sea temporal o en el espacio.

## VII.GLOSARIO

**Timetabling (TTP):** Los problemas de programación educativa (Timetabling) consisten en la asignación de tiempos y recursos a los eventos o franjas horarias de las instituciones educativas, teniendo en cuenta varias restricciones duras y blandas. En ocasiones, este problema puede ser resuelto manualmente. Sin embargo, esta tarea se vuelve prácticamente imposible cuando hay que gestionar cientos de asignaturas, recursos y restricciones.[41]

**NP-Complete:** Son problemas de decisión que no pueden ser resueltos de manera óptima en un tiempo polinomial (tiempo razonable), por lo que se deben emplear técnicas no exactas para buscar soluciones factibles, pero no óptimas debido a su complejidad [3].

**NP-Hard:** Son problemas que deben ser reducidos a NP-Complete para poder ser resueltos, ya que encontrar la solución óptima no es posible, por lo que se debe simplificar para encontrar soluciones factibles.[3]

## VIII.REFERENCIAS

- [1] N. Pillay, "A review of hyper-heuristics for educational timetabling," *Ann Oper Res*, vol. 239, no. 1, pp. 3–38, Apr. 2016, doi: 10.1007/s10479-014-1688-1.
- [2] S. Ceschia, L. di Gaspero, and A. Schaerf, "Educational timetabling: Problems, benchmarks, and state-of-the-art results," *European Journal of Operational Research*. Elsevier B.V., 2022. doi: 10.1016/j.ejor.2022.07.011.
- [3] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Comput Ind Eng*, vol. 86, pp. 43–59, 2015, doi: 10.1016/j.cie.2014.11.010.
- [4] Aizam N. A. H. and Uvaraja V., "Generic Model for Timetabling Problems by Integer Linear Programming Approach," *World Academy of Science, Engineering and Technology International Journal of Mathematical and Computational Sciences*, vol. 9, no. 12, 2015.
- [5] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," *Ann Oper Res*, vol. 275, no. 1, pp. 145–160, Apr. 2019, doi: 10.1007/s10479-017-2688-8.
- [6] Rafael Garcia-Ros, Francisco Pérez-González, Josefa Pérez-Blasco, and Luis A. Natividad, "Evaluación del estrés académico en estudiantes de nueva incorporación a la universidad," *Rev Latinoam Psicol*, vol. 44, no. 2, pp. 143–154, 2012, [Online]. Available: <http://www.redalyc.org/articulo.oa?id=80524058011>
- [7] "Scopus - Analyze search results." <https://www-scopus-com.bdbib.javerianacali.edu.co/term/analyzer.uri?sid=fb5036cc52356cf519b0de8d624111b4&origin=resultslist&src=s&s=TITLE-ABS-KEY%28timetabling%29&sort=plf-f&sdt=b&sot=b&sl=26&count=3046&analyzeResults=Analyze+results&txGid=527a19c2ae4840f82f762f6543547e45> (accessed Oct. 13, 2022).
- [8] D. de Werra, "An introduction to timetabling," *Eur J Oper Res*, vol. 19, no. 2, pp. 151–162, Feb. 1985, doi: 10.1016/0377-2217(85)90167-5.

- [9] Edmund Burke, Kirk Jackson, Jeffrey Kingston, and Rupert Weare, "Automated University Timetabling: The State of the Art," 1997.
- [10] Vasugi Musaliar, Rohini, and Manoj Kumar Mishra, "An Overview of the Heuristic Approaches for University Course Timetabling System," *International Journal of Engineering Research & Technology*, 2016.
- [11] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Syst Appl*, vol. 165, Mar. 2021, doi: 10.1016/j.eswa.2020.113943.
- [12] J. C. Marín Ángel and P. A. Maya Duque, "Modelo lineal para la programación de clases en una institución educativa," *Ing Cienc*, vol. 12, no. 23, pp. 47–71, Feb. 2016, doi: 10.17230/ingciencia.12.23.3.
- [13] MINISTERIO DE EDUCACION NACIONAL, "DECRETO NÚMERO 80 DE 1980," 1980.
- [14] C. Crobu, M. di Francesco, and E. Gorgone, "High quality timetables for Italian schools," *Comput Oper Res*, vol. 148, Dec. 2022, doi: 10.1016/j.cor.2022.105980.
- [15] Z. Mohd Zaulir, N. Liyana, A. Aziz, N. Aidya, and H. Aizam, "A General Mathematical Model for University Courses Timetabling: Implementation to a Public University in Malaysia," 2022.
- [16] C. H. Wong, S. L. Goh, and J. Likoh, "A Genetic Algorithm for the Real-world University Course Timetabling Problem," in *2022 IEEE 18th International Colloquium on Signal Processing and Applications, CSPA 2022 - Proceeding*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 46–50. doi: 10.1109/CSPA55076.2022.9781907.
- [17] S. Daskalaki, T. Birbas, and E. Housos, "An integer programming formulation for a case study in university timetabling," in *European Journal of Operational Research*, Feb. 2004, pp. 117–135. doi: 10.1016/S0377-2217(03)00103-6.
- [18] R. Santiago-Mozos, S. Salcedo-Sanz, M. Deprado-Cumplido, and C. Bousoño-Calzón, "A two-phase heuristic evolutionary algorithm for personalizing course timetables: A case study in a Spanish university," *Comput Oper Res*, vol. 32, no. 7, pp. 1761–1776, Jul. 2005, doi: 10.1016/j.cor.2003.11.030.
- [19] A. Muklason, R. G. Irianti, and A. Marom, "Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 656–664. doi: 10.1016/j.procs.2019.11.169.
- [20] S. Even, A. Itai, and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 691–703, Dec. 1976, doi: 10.1137/0205048.
- [21] D. Morillo, L. Moreno, and J. Díaz, "Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos (RCPS): una revisión Parte 1," *Ing Cienc*, vol. 10, no. 19, pp. 247–271, Jan. 2014, doi: 10.17230/ingciencia.10.19.12.

- [22] O. Rossi-Doria *et al.*, "A comparison of the performance of different metaheuristics on the timetabling problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2740, pp. 329–351, 2003, doi: 10.1007/978-3-540-45157-0\_22/COVER.
- [23] Abeer Bashab, Ashraf Osman Ibrahim, Ali Ahmed, and Ajith Abraham, "A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms," *Neural Comput Appl*, 2020, doi: 10.1007/s00521-020-05110-3.
- [24] Edmund Burke, Kirk Jackson, and Rupert Weare, "Automated University Timetabling: The State of the Art," *Comput J*, vol. 40, no. 9, 1997.
- [25] "THE INDUSTRIAL ENGINEERING BODY OF KNOWLEDGE," *Institute of Industrial Engineers*, 2019.
- [26] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf, and T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," *Comput Oper Res*, vol. 65, pp. 83–92, Jul. 2016, doi: 10.1016/j.cor.2015.07.002.
- [27] S. Liu, Z. K. Feng, W. jing Niu, H. rong Zhang, and Z. guo Song, "Peak operation problem solving for hydropower reservoirs by elite-guide sine cosine algorithm with Gaussian local search and random mutation," *Energies (Basel)*, vol. 12, no. 11, Jun. 2019, doi: 10.3390/en12112189.
- [28] G. L. Cravo and A. R. S. Amaral, "Adaptive iterated local search for the parallel row ordering problem," *Expert Syst Appl*, vol. 208, Dec. 2022, doi: 10.1016/j.eswa.2022.118033.
- [29] K. Sörensen and F. W. Glover, "Metaheuristics," in *Encyclopedia of Operations Research and Management Science*, Boston, MA: Springer US, 2013, pp. 960–970. doi: 10.1007/978-1-4419-1153-7\_1167.
- [30] R. Ganguli and S. Roy, "A Study on Course Timetable Scheduling using Graph Coloring Approach," 2017. [Online]. Available: <http://www.ripublication.com>
- [31] V. Nandhini, "A Study on Course Timetable Scheduling and Exam Timetable Scheduling using Graph Coloring Approach," *Int J Res Appl Sci Eng Technol*, vol. 7, no. 3, pp. 1999–2006, Mar. 2019, doi: 10.22214/ijraset.2019.3368.
- [32] M. Assi, B. Halawi, and R. A. Haraty, "Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 899–906. doi: 10.1016/j.procs.2018.08.024.
- [33] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems," *Inf Sci (N Y)*, vol. 191, pp. 146–168, May 2012, doi: 10.1016/j.ins.2011.12.018.
- [34] S. L. Goh, G. Kendall, and N. R. Sabar, "Improved local search approaches to solve the post enrolment course timetabling problem," *Eur J Oper Res*, vol. 261, no. 1, pp. 17–29, Aug. 2017, doi: 10.1016/j.ejor.2017.01.040.

- [35] K. Kianfar, “Branch-and-Bound Algorithms,” in *Wiley Encyclopedia of Operations Research and Management Science*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011. doi: 10.1002/9780470400531.eorms0116.
- [36] J. Dréo, P. Siarry, A. Pétrowski, and E. Taillard, *Metaheuristics for Hard Optimization*. Berlin/Heidelberg: Springer-Verlag, 2006. doi: 10.1007/3-540-30966-7.
- [37] “Diseños de Taguchi - Minitab.” <https://support.minitab.com/es-mx/minitab/21/help-and-how-to/statistical-modeling/doe/supporting-topics/taguchi-designs/taguchi-designs/> (accessed May 19, 2023).
- [38] “Standard Pricing – AMPL.” <https://ampl.com/standard-pricing/> (accessed May 19, 2023).
- [39] “GUSEK.” <https://gusek.sourceforge.net/gusek.html> (accessed May 21, 2023).
- [40] T. van der Voordt and P. A. Jensen, “The impact of healthy workplaces on employee satisfaction, productivity and costs,” *Journal of Corporate Real Estate*, vol. 25, no. 1, pp. 29–49, Feb. 2023, doi: 10.1108/JCRE-03-2021-0012.
- [41] G. H. G. Fonseca, H. G. Santos, E. G. Carrano, and T. J. R. Stidsen, “Integer programming techniques for educational timetabling,” *Eur J Oper Res*, vol. 262, no. 1, pp. 28–39, Oct. 2017, doi: 10.1016/j.ejor.2017.03.020.

## IX. ANEXOS

TABLA XVII.  
TABLA DE ANEXOS

No. Anexo	Nombre	Desarrollo (propio o terceros)	Tipo de Archivo (PDF, HTML, Excel, Word...)
1	2022201 - Anexo 1. Entrevista Director de Carrera	Propio	Word
2	2022201 - Anexo 2. Entrevista secretaria Director Ingeniería Industrial Universidad del Valle	Propio	Word

3	2022201 - Anexo 3. Hoja de trabajo contexto y justificación	Propio	Word
4	2022201 - Anexo 4. Hoja de trabajo identificación de grupo de interés	Propio	Word
5	2022201 - Anexo 5. Hoja de trabajo identificación de los requerimientos de grupos de interés	Propio	Word
6	2022201 - Anexo 6. Lista de chequeo de revisión documental	Propio	Word
7	2022201 - Anexo 7. Resultados encuesta sobre calidad del proceso de matrícula	Propio	Excel
8	2022201 - Anexo 8. Hoja de trabajo exploración del mercado	Propio	Word
9	2022201 - Anexo 9. Datos horarios facultado Ingeniería Javeriana Cali	Tercero	Excel
10	2022201 - Anexo 10. Datos caso de prueba ITC07	Tercero	Zip
11	2022201 - Anexo 11. Datos caso de prueba XHSTT	Tercero	Zip
12	2022201 - Anexo 12. Ejecución 1 segundo	Propio	Excel
13	2022201 - Anexo 13. Ejecución 3 segundos	Propio	Excel
14	2022201 - Anexo 14. Ejecución 30 segundos	Propio	Excel
15	2022201 - Anexo 15. Comparación ejecuciones	Propio	Excel
16	2022201 - Anexo 16. Plan de trabajo (Project)	Propio	Project
17	2022201 - Anexo 17. Project Charter	Propio	Excel
18	2022201 - Anexo 18. Distribución horaria actual	Propio	Excel
19	2022201 - Anexo 19. Solución AMPL 60seg	Propio	Excel
20	2022201 - Anexo 20. Vector solución AMPL 60seg	Propio	Txt
21	2022201 - Anexo 21. Modelo matemático	Propio	Zip
22	2022201 - Anexo 22. Código fuente	Propio	Zip
23	2022201 - Anexo 23. Parámetros para el programa	Propio	Zip
24	2022201 - Anexo 24. Solución semilla 1	Propio	Txt
25	2022201 - Anexo 25. Solución semilla 2	Propio	Txt
26	2022201 - Anexo 26. Solución semilla 3	Propio	Txt
27	2022201 - Anexo 27. Benchmarking de perturbaciones	Propio	Excel
28	2022201 - Anexo 28. Benchmarking de reglas de movimiento	Propio	Excel
29	2022201 - Anexo 29. Macro resultado grafico	Propio	Excel

**Nota de Aceptación:**

**Proyecto de Diseño Aprobado.** en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali para optar el título de Ingeniero Industrial.



---

HERNAN CAMILO ROCHA NIÑO  
Decano de la Facultad de Ingeniería y Ciencias



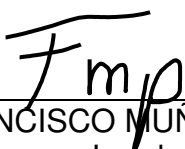
---

JORGE ENRIQUE ÁLVAREZ PATIÑO  
Director Carrera Ingeniería Industrial



---

DANIEL MORILLO TORRES  
Director Proyecto de Diseño



---

FRANCISCO MUÑOZ PRADO  
Jurado 1



---

ALVARO FIGUEROA CABRERA  
Jurado 2

Santiago de Cali, mayo 22 de 2023

**Ingeniero Jorge Enrique Álvarez Patiño**

Director de Carrera de Ingeniería Industrial  
Pontificia Universidad Javeriana  
Cali

Asunto: Carta de aval para la entrega definitiva del documento Proyecto de Diseño 2022201

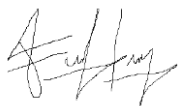
Por medio de la presente doy aval para la entrega final a la carrera, del documento que soporta el Proyecto de Diseño 2022202, bajo mi dirección, desarrollado por los estudiantes Felipe Hernández Rendón y Nicolas Vidal Duque titulado *“Diseño de metaheurística y modelo matemático para asignación de horarios en el programa de ingeniería industrial en la PUJ Cali”*.

Cordialmente



---

Daniel Morillo Torres  
Director de Proyecto de Diseño 2022201



---

Felipe Hernández Rendón



---

Nicolás Vidal Duque