INSTITUTO TECNOLÓGICO DE ZITÁCUARO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

MAESTRÍA EN SISTEMAS COMPUTACIONALES

“GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES DE EMPLEADOS DE CINES MEDIANTE EL USO DE ALGORITMOS GENÉTICOS”

T E S I S

PARA OBTENER EL GRADO DE:

MAESTRÍA EN SISTEMAS COMPUTACIONALES

PRESENTA:

Eric César Guzmán Malvaez

DIRECTOR DE TESIS

Noel Enrique Rodríguez Maya

CO-DIRECTOR DE TESIS

Jhacer Kharen Ruíz Garduño

LUGAR Y FECHA

Zitácuaro, Michoacán a 01 de julio de 2026.

Carta de Autorización	Revisión:	002	Fecha de emisión:	20-ene-2025
ITZ-AC-TL-FO-003	Elaborado por:		División de Estudios de Posgrado e Investigación	
	Revisado por:		Sistema de Gestión Integrado	
	Autorizado por:		Subdirección académica	

H. Zitácuaro, Mich., 01 de julio de 2026

CARTA DE AUTORIZACIÓN

DR. SAMUEL EFREN VIÑAS ALVAREZ
COORDINADOR DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE

De acuerdo con los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México, en donde se establecen los requisitos para la obtención del grado de Maestría; el H. Comité Tutorial **del C. ERIC CESAR GUZMÁN MALVAEZ,** estudiante del programa de **MAESTRÍA EN SISTEMAS COMPUTACIONALES**, con número de control: **M24650516,** después de haber realizado la revisión del contenido y formato de tesis "**GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES DE EMPLEADOS DE CINES MEDIANTE EL USO DE ALGORITMOS GENÉTICOS**" emite su consentimiento para continuar con el proceso de obtención de grado académico correspondiente.

Por ese motivo se solicita a usted Autorizar al **C. ERIC CESAR GUZMÁN MALVAEZ** la impresión y reproducción de la tesis *in comento.*

A T E N T A M E N T E
Excelencia en Educación Tecnológica®

H. Comité Tutorial

Director de tesis

Dr. Noel Enrique Rodríguez Maya

Codirectora de tesis

Dra. Jhacer Kharen Ruiz Garduño

Dr. Rafael Salas Zárate
Revisor

Mtra. Claudia Teresa González Ramírez
Revisora

Declaración de Originalidad	Revisión:	002	Fecha de emisión:	20-ene-2025
ITZ-AC-TL-FO-005	Elaborado por:		División de Estudios de Posgrado e Investigación	
	Revisado por:		Sistema de Gestión Integrado	
	Autorizado por:		Subdirección académica	

DECLARACIÓN DE ORIGINALIDAD DE LA TESIS

En H. Zitácuaro, Michoacán, a **17** de **agosto** de **2026**, el que suscribe, **Eric César Guzmán Malvaez**, estudiante del programa de **Maestría en Sistemas Computacionales** del Instituto Tecnológico de Zitácuaro, con número de control: **M24650516**, manifiesto que soy autor intelectual de la presente tesis, la cual fue dirigida por el **Dr. Noel Enrique Rodríguez Maya** con nombre: **"GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES DE EMPLEADOS DE CINES MEDIANTE EL USO DE ALGORITMOS GENÉTICOS"**.

Declaro que la tesis es una obra original, que es de mi autoría y que toda la información y materiales extraídos de otras fuentes han sido debidamente referenciados. Que a la obra no ha sido previamente publicada y que, en caso de violación de derechos de autor, me hago responsable y exonero de toda responsabilidad al Tecnológico Nacional de México / Instituto Tecnológico de Zitácuaro.

ATENTAMENTE

Eric César Guzmán Malvaez

Carta Cesión de Derechos	Revisión:	002	Fecha de emisión:	20-ene-2025
ITZ-AC-TL-FO-004	Elaborado por:		División de Estudios de Posgrado e Investigación	
	Revisado por:		Sistema de Gestión Integrado	
	Autorizado por:		Subdirección académica	

CARTA DE CESIÓN DE DERECHOS

En H. Zitácuaro, Michoacán, a **17** de **Agosto** de **2026**, el que suscribe, **Eric César Guzmán Malvaez**, estudiante del programa de **Maestría en Sistemas Computacionales** del Instituto Tecnológico de Zitácuaro, con número de control: **M24650516**, manifiesto que soy autor intelectual de la presente tesis, la cual fue dirigida por el **Dr. Noel Enrique Rodríguez Maya** y cedo íntegramente los derechos de trabajo de tesis titulado: **"GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES DE EMPLEADOS DE CINES MEDIANTE EL USO DE ALGORITMOS GENÉTICOS"** al Tecnológico Nacional de México / Instituto Tecnológico de Zitácuaro para su uso con fines académicos y de investigación.

Los usuarios pueden consultar y reproducir el contenido para todos los usos que tengan finalidad académica siempre y cuando sea citada la fuente información.

ATENTAMENTE

Eric César Guzmán Malvaez

Autorización de Impresión de Tesis	Revisión:	002	Fecha de emisión:	20-ene-2025
ITZ-AC-TL-FO-006	Elaborado por:		División de Estudios de Posgrado e Investigación	
	Revisado por:		Sistema de Gestión Integrado	
	Autorizado por:		Subdirección académica	

H. Zitácuaro, Mich.,a 17 de agosto de 2026

AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. ERIC CÉSAR GUZMÁN MALVAEZ
NO. DE CONTROL: M24650516
MAESTRÍA EN SISTEMAS COMPUTACIONALES
PRESENTE

Conforme los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México y por recomendación del H. Comité Tutorial, esta División le autoriza imprimir y reproducir la tesis: **"GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES DE EMPLEADOS DE CINES MEDIANTE EL USO DE ALGORITMOS GENÉTICOS".**

Ruego a Usted dar puntual seguimiento al formato en vigor que, para tal caso, indica las características de diseño que deberá contener tan importante documento.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®

DR. SAMUEL EFREN VIÑAS ALVAREZ
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
INSTITUTO TECNOLÓGICO DE ZITÁCUARO

SECRETARIA DE EDUCACION PUBLICA
INSTITUTO TECNOLOGICO DE ZITACUARO
DEPTO DE DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION

Recibí 20/08/26

1 GENERALIDADES 1

1.1 INTRODUCCIÓN 2
1.2. ANÁLISIS DEL PROBLEMA 4
1.3. JUSTIFICACIÓN 7
1.4. OBJETIVOS 9
1.4.1. Objetivo General 9
1.4.2. Objetivos Específicos 9
1.5. HIPÓTESIS 11
1.6. CONTRIBUCIONES 12
1.7. PRODUCTOS ACADÉMICOS GENERADOS 13
1.7.1. Participación en Congresos 13
1.7.2. Registro de Derecho de Autor 13
1.7.3. Publicación de artículos 14

2. ESTADO DEL ARTE Y DE LA TÉCNICA 15

2.1. ESTADO DEL ARTE 16
2.2. ESTADO DE LA TÉCNICA 25

3. MARCO TEÓRICO 31

3.1. FUNDAMENTOS DE LA PROGRAMACIÓN DE TURNOS Y HORARIOS 32
3.1.1. Métodos tradicionales de programación de turnos 32
3.1.2. Metaheurísticas en programación de horarios 34
3.1.3. Algoritmos Genéticos 35
3.1.4. Modelos de optimización multiobjetivo 38

4. METODOLOGÍA 40

4.1. ENFOQUE 41
4.2. DISEÑO DE INVESTIGACIÓN 41
4.3. VARIABLES 42
4.3.1. Variables independientes 42
4.3.2. Variables dependientes 42
4.4. MÉTRICAS DE EVALUACIÓN 43
4.4.1. Función objetivo (fitness) 43
4.4.2. Penalización por restricciones fuertes 44
4.4.3. Penalización por restricciones débiles 44
4.4.4. Métrica de tiempo de ejecución 45
4.4.5. Análisis de convergencia 45
4.4.6. Criterios de evaluación 45
4.5. PROCESO EXPERIMENTAL 46
4.6. HERRAMIENTAS Y ENTORNO DE DESARROLLO 47
4.7. POBLACIÓN Y MUESTRA 48
4.8. VALIDACIÓN DEL MODELO 49

5. GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES 50

5.1. RESTRICCIONES DEL MODELO 51
5.2. MODELO MATEMÁTICO (FUNCIÓN OBJETIVO) 53
5.3. REPRESENTACIÓN DEL CROMOSOMA 54

5.3.1. Estructura del cromosoma *55*
5.4. INTERPRETACIÓN DE LA SOLUCIÓN 55
5.5. DISEÑO DEL ALGORITMO GENÉTICO 56
5.5.1. Estructura general del algoritmo *57*
5.5.2. Inicialización de la población *57*
5.5.3. Evaluación del fitness *58*
5.5.4. Selección *58*
5.5.5. Cruza (Crossover) *59*
5.5.6. Mutación *59*
5.5.7. Elitismo *60*
5.5.8. Criterio de paro *61*
5.5.9. Parámetros del algoritmo *61*
5.5.10. Procedimiento general del algoritmo *61*
5.6. ARQUITECTURA DEL SISTEMA 63
5.6.1. Estructura general del sistema *63*
5.6.2. Capa de presentación *64*
5.6.3. Capa de lógica de negocio *64*
5.6.4. Capa de datos *65*
5.6.5. Flujo de ejecución del sistema *66*
5.6.6. Ventajas de la arquitectura propuesta *66*
5.7. INTERFAZ GRÁFICA DE USUARIO 66
5.8. DISEÑO DE LA INTERFAZ 67
5.8.1. Funcionalidades principales *67*
5.9. REPRESENTACIÓN DE LA INFORMACIÓN 68
5.10. INTERACCIÓN CON EL SISTEMA 68

6. RESULTADOS Y ANÁLISIS 69

6.1. MÉTRICAS DE EVALUACIÓN 70
6.2. IMPACTO DEL TAMAÑO DE LA POBLACIÓN 72
6.3. IMPACTO DEL NÚMERO DE GENERACIONES 73
6.4. ANÁLISIS GLOBAL DEL DESEMPEÑO DEL ALGORITMO 73
6.5. EVIDENCIA DE CONVERGENCIA 74
6.6. RELACIÓN ENTRE CALIDAD Y TIEMPO DE EJECUCIÓN 76
6.7. ANÁLISIS DE FACTIBILIDAD 77
6.8. PARÁMETROS ÓPTIMOS DE POBLACIÓN Y GENERACIONES 77
6.9. AJUSTE DEL OPERADOR DE CRUZA EN EL ALGORITMO GENÉTICO 79
6.10. *CONFIGURACIÓN DE PARÁMETROS SELECCIONADA* 80
6.11. COMPARATIVO DE DESEMPEÑO FRENTE AL MÉTODO MANUAL 81
6.12. VISUALIZACIÓN DEL SISTEMA 83

7. CONCLUSIONES Y RECOMENDACIONES 87

7.1. CONCLUSIONES 88
7.2. RECOMENDACIONES 90
7.2.1. Recomendaciones para futuros trabajos de investigación *90*
7.2.2. Recomendaciones para la institución educativa *90*
7.3. TRABAJO FUTURO 91

8. REFERENCIAS 93

Índice de tablas

Tabla 1. *Comparación entre las principales técnicas utilizadas para la generación automática de horarios.* 26
Tabla 2. *Comparación de softwares para la gestión de turnos.* 28
Tabla 3. *Requerimientos funcionales asociados a restricciones fuertes* 51
Tabla 4. *Requerimientos funcionales asociados a restricciones débiles* 52
Tabla 5. *Ejemplo horario* 55
Tabla 6. *Estadísticas descriptivas de configuraciones representativas del Algoritmo Genético* .71
Tabla 7. *Configuración óptima global del algoritmo* 80
Tabla 8. *Comparativo de desempeño entre método manual y sistema propuesto.* 82

Índice de Figuras

Figura 1. *Mapa de calor del desempeño del algoritmo.* 74
Figura 2. *Gráfica de convergencia.* 75
Figura 3. *Tiempo de ejecución promedio por configuración del algoritmo.* 76
Figura 4. *Gráfica de mejores configuraciones encontradas.* 78
Figura 5. *Sensibilidad del algoritmo a la probabilidad de cruzamiento.* 80
Figura 6. *Configuración de sucursal.* 83
Figura 7. *Configuración de empleados.* 84
Figura 8. *Generación de horarios laborales.* 85
Figura 9. *Resultado final de horarios laborales.* 85

Resumen

La generación manual de horarios laborales en cadenas de cines representa una tarea compleja dentro de la gestión operativa del personal. La alta rotación de empleados, los cambios constantes en la demanda de asistentes y las restricciones legales y operativas dificultan la organización eficiente de los turnos.

En este trabajo se desarrolló un modelo automatizado mediante el uso de Algoritmos Genéticos, con el objetivo de optimizar la asignación de turnos laborales. Cada horario se codificó como un cromosoma, integrando diversas dimensiones fundamentales: la disponibilidad y capacitación de los empleados, los horarios de operación del cine, los niveles de demanda y los criterios legales, operativos y organizacionales considerados en el modelo.

El proceso evolutivo del algoritmo se realizó mediante operadores genéticos como la selección por torneo, el cruce uniforme y la mutación adaptativa. Posteriormente, la calidad de cada horario se evaluó a través de una función objetivo basada en penalizaciones, la cual permitió medir el cumplimiento de las restricciones y criterios definidos para la operación.

La validación del modelo se realizó con base en escenarios simulados y datos de prueba representativos del contexto operativo de las cadenas de cines. De esta forma, fue posible evaluar el comportamiento de la metodología en condiciones cercanas a la operación real, sin comprometer la confidencialidad ni la disponibilidad de datos específicos.

Los resultados obtenidos en las pruebas realizadas mostraron que el modelo desarrollado contribuye a reducir el tiempo de generación de horarios laborales, mejorar la asignación del personal y favorecer el cumplimiento de las restricciones legales y operativas. Asimismo, se identificó su utilidad para apoyar el control del presupuesto de nómina, disminuir riesgos administrativos y fortalecer la eficiencia operativa de la organización.

Abstract

The manual generation of work schedules in cinema chains represents a complex task within personnel operations management. High employee turnover, constant changes in customer demand, and legal and operational restrictions make the efficient organization of shifts difficult.

This study developed an automated model based on Genetic Algorithms with the objective of optimizing the assignment of work shifts. Each schedule was encoded as a chromosome, integrating several key dimensions: employee availability and training, cinema operating hours, demand levels, and the legal, operational, and organizational criteria considered in the model.

The evolutionary process of the algorithm was carried out using genetic operators such as tournament selection, uniform crossover, and adaptive mutation. Subsequently, the quality of each schedule was evaluated through a penalty-based objective function, which made it possible to measure compliance with the restrictions and criteria defined for the operation.

The model was validated using simulated scenarios and test data representative of the operational context of cinema chains. This made it possible to evaluate the behavior of the methodology under conditions close to real operation, without compromising the confidentiality or availability of specific data.

The results obtained from the tests showed that the developed model contributes to reducing the time required to generate work schedules, improving staff assignment, and supporting compliance with legal and operational restrictions. Likewise, its usefulness was identified in supporting payroll budget control, reducing administrative risks, and strengthening the operational efficiency of the organization.

1 Generalidades

1.1 Introducción

La generación de horarios laborales en cadenas de cines ha sido históricamente una tarea compleja, especialmente cuando se enfrenta a la rotación constante de personal, los cambios en la demanda de los asistentes y las estrictas normativas laborales. Esta actividad, tradicionalmente realizada de manera manual, consume una cantidad considerable de tiempo y puede estar sujeta a errores humanos que afectan la eficiencia operativa, el control administrativo y el aprovechamiento de los recursos disponibles.

La problemática de la generación de horarios laborales en cadenas de cines no solo radica en la combinación de múltiples restricciones para asignar turnos, sino también en la variabilidad de la demanda. A diferencia de otras industrias, la afluencia de clientes en un cine está sujeta a factores externos inmediatos, como estrenos nacionales, condiciones climáticas, temporadas vacacionales o promociones comerciales. Esta situación puede generar una brecha entre el personal programado y las necesidades reales de operación, lo que deriva en sobrecostos, asignaciones ineficientes o falta de cobertura en periodos de alta actividad.

La implementación de un sistema automatizado para la generación de horarios laborales puede ser considerada una alternativa para mejorar la administración de la fuerza laboral en cadenas de cines. A través de este enfoque, la planeación de horarios es realizada con base en criterios operativos, legales y administrativos, tales como la demanda estimada, la disponibilidad del personal, la capacitación requerida y los recursos presupuestales disponibles. Con ello, se busca que la asignación de turnos sea efectuada de manera más eficiente, reduciendo errores derivados de la programación manual y favoreciendo una mejor distribución del personal conforme a las necesidades reales de operación. Además, se contribuye al control de los costos laborales y al fortalecimiento de la continuidad del servicio durante los periodos de mayor demanda.

La Inteligencia Artificial ha sido aplicada en distintos procesos organizacionales como una herramienta de apoyo para la toma de decisiones y la optimización de recursos. Dentro de este campo, los Algoritmos Evolutivos representan una alternativa adecuada para abordar problemas complejos de búsqueda y optimización, ya que permiten explorar diferentes combinaciones de solución mediante procesos inspirados en la selección natural. Su aplicación resulta pertinente cuando el número de posibles asignaciones es amplio y no puede ser evaluado de forma eficiente mediante métodos manuales o convencionales.

En el presente trabajo se desarrolló un modelo basado en Algoritmos Genéticos para automatizar la generación de horarios laborales en cadenas de cines. Mediante este enfoque, cada posible horario fue representado como una solución candidata y evaluado de acuerdo con su nivel de cumplimiento respecto a los criterios definidos para la operación. Con ello, se buscó reducir la carga administrativa asociada a la elaboración manual de horarios, mejorar la consistencia del proceso de asignación y fortalecer la capacidad de planeación de la empresa ante escenarios operativos variables.

La principal contribución de esta investigación consistió en el desarrollo de un modelo automatizado para la asignación de turnos laborales en cadenas de cines, orientado a optimizar la asignación de turnos, garantizar el cumplimiento de las disposiciones laborales y mejorar el desempeño operativo de la organización.

1.2. Análisis del problema

La planificación y asignación de turnos en cadenas de cines constituye una función crítica dentro de la gestión operativa del personal. Este proceso impacta directamente en la continuidad del servicio, el control de los costos laborales, la productividad de las áreas operativas y la capacidad de respuesta ante cambios en la demanda. En un entorno con horarios extendidos, múltiples puestos de trabajo y una alta rotación de personal, la generación manual de horarios laborales representa una actividad compleja que consume tiempo gerencial y aumenta la posibilidad de errores en la programación.

En primer lugar, la asignación de turnos debe responder a las condiciones reales de cada puesto. En áreas como taquilla, dulcería, salas, limpieza, mantenimiento o puntos específicos de venta, el personal requiere distintos niveles de capacitación y experiencia operativa. Por ello, una asignación inadecuada no solo afecta la cobertura de las actividades, sino que también puede generar baja productividad, errores en la ejecución de tareas, uso ineficiente del personal disponible y necesidad de ajustes posteriores. Para la gerencia, este problema se traduce en menor control operativo y en una administración menos precisa de los recursos humanos.

Otro factor determinante es la variabilidad de la demanda. La afluencia de asistentes en los cines cambia de acuerdo con el día de la semana, los horarios de función, los estrenos de películas, las temporadas vacacionales, las promociones comerciales y los eventos especiales. Esta variación impide mantener una plantilla fija para todos los periodos, ya que el número de empleados requeridos debe ajustarse a las necesidades reales de operación. Una planeación deficiente puede provocar sobreasignación de personal, incrementando el costo de nómina, o falta de cobertura en horarios críticos, afectando la continuidad y eficiencia del servicio. De acuerdo con la revisión del proceso operativo y la experiencia documentada durante el análisis del caso de estudio, una mala programación puede representar sobrecostos en horas laborales estimadas o requerir varias horas semanales de ajustes administrativos por sucursal.

Además, la programación de turnos debe integrar restricciones laborales, operativas y políticas internas, como límites de jornada, descansos obligatorios, tipos de horario, días laborados, disponibilidad registrada, capacitación por puesto y condiciones contractuales. Para efectos del modelo propuesto, estas condiciones pueden distinguirse en restricciones fuertes y restricciones débiles. Las restricciones fuertes corresponden a aquellas que deben cumplirse obligatoriamente para que un horario sea factible, como no asignar más de un turno por día al mismo empleado, respetar la disponibilidad registrada, considerar la capacitación requerida para cada puesto, cumplir con los límites de jornada y mantener los descansos establecidos. Por su parte, las restricciones débiles representan criterios deseables de mejora, como equilibrar la distribución de horas, favorecer una mejor cobertura de la demanda, reducir ajustes posteriores y utilizar de manera más eficiente el presupuesto disponible. Cuando estos elementos se administran de forma manual o mediante herramientas semiautomáticas, el proceso depende en gran medida de la experiencia del encargado y de revisiones repetitivas en hojas de cálculo u otros medios operativos. Esta situación incrementa la posibilidad de omisiones, inconsistencias, pagos no previstos, incumplimientos normativos y decisiones poco estandarizadas entre sucursales.

La problemática se intensifica porque los responsables de elaborar los horarios deben evaluar simultáneamente disponibilidad, capacitación, perfiles de puesto, demanda estimada, restricciones legales, restricciones operativas y presupuesto de nómina. Esta combinación genera un número elevado de posibles asignaciones, lo que vuelve difícil identificar manualmente una distribución eficiente. Conforme a la revisión del proceso actual de programación, apoyada en la observación del caso de estudio y en la información proporcionada por personal involucrado en la elaboración de horarios, el tiempo destinado a esta tarea puede alcanzar entre 6 y 12 horas por semana, dependiendo del tamaño de la sucursal, la cantidad de empleados, los cambios de disponibilidad y los ajustes operativos requeridos.

Desde una perspectiva técnica, la asignación de turnos no corresponde únicamente a una tarea administrativa, sino a un problema de optimización combinatoria de alta complejidad. La cantidad de empleados, puestos, días, horarios, restricciones e interacciones posibles genera un espacio de soluciones demasiado amplio para evaluarse de manera exhaustiva mediante hojas de cálculo, reglas manuales o métodos tradicionales de programación lineal aplicados de forma convencional. Esta característica lo aproxima a problemas clasificados como NP-hard, en los que encontrar una solución óptima exacta puede requerir un costo computacional elevado conforme aumenta el tamaño del problema.

Por lo anterior, surge la necesidad de implementar una solución automatizada que apoye la generación de horarios laborales en cadenas de cines. Bajo este enfoque, el uso de metaheurísticas como los Algoritmos Genéticos representa una alternativa viable para explorar múltiples combinaciones de asignación y obtener soluciones de buena calidad en tiempos razonables. Su aplicación permite reducir el tiempo de planeación, mejorar la utilización del personal, fortalecer el control del presupuesto de nómina, disminuir riesgos por errores de programación y apoyar la toma de decisiones estratégicas de la gerencia y los dueños de la organización.

1.3. Justificación

La asignación de turnos laborales en una cadena de cines constituye un problema de optimización complejo debido a la necesidad de satisfacer simultáneamente restricciones legales, operativas y organizacionales. La generación manual de horarios implica analizar múltiples variables, como los horarios de operación del cine, la capacitación de los empleados para cada puesto, los límites de jornada establecidos por la legislación laboral y la cobertura necesaria para atender la variabilidad de la demanda de clientes. Conforme aumenta el número de empleados y restricciones, este proceso genera una gran cantidad de combinaciones de posibles soluciones que superan la capacidad analítica humana. Esto incrementa significativamente el tiempo invertido por los gerentes y eleva el riesgo de errores críticos, tales como la subcobertura en horas pico o el incumplimiento involuntario de disposiciones legales. En consecuencia, se ven afectados los costos operativos y la calidad del servicio, lo que hace evidente la necesidad de soluciones automatizadas basadas en técnicas de optimización.

Un sistema automatizado permite gestionar la variabilidad de la demanda de manera dinámica, ajustando los turnos de los empleados según las necesidades operativas y la disponibilidad del personal. Esto no solo reduce los costos operativos derivados de la sobrecarga de trabajo o la ociosidad, sino que también optimiza el uso de los recursos humanos disponibles, mejorando la eficiencia del cine.

Asimismo, la implementación de este tipo de soluciones contribuye a estandarizar el proceso de asignación de turnos, disminuyendo la probabilidad de errores y mejorando la consistencia de los horarios generados. Como consecuencia, la organización puede optimizar la utilización de sus recursos disponibles, facilitar el cumplimiento de la normativa laboral y mejorar su capacidad de respuesta ante cambios en las necesidades operativas. Por lo tanto, el desarrollo de un modelo automatizado basado en Algoritmos Genéticos representa una

alternativa viable para incrementar la eficiencia de los procesos de planificación y fortalecer la gestión operativa dentro de una cadena de cines.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar e implementar un sistema automatizado para la generación de horarios laborales en una cadena de cines mediante Algoritmos Genéticos, con el propósito de optimizar la asignación de turnos considerando restricciones legales, operativas y organizacionales.

1.4.2. Objetivos Específicos

1. Identificar y formalizar las restricciones legales, operativas y organizacionales involucradas en la asignación de turnos laborales en una cadena de cines, clasificándolas en restricciones fuertes y restricciones débiles para su incorporación dentro del modelo de optimización.
2. Diseñar un modelo de representación de soluciones basado en Algoritmos Genéticos, definiendo la estructura de los cromosomas y los operadores evolutivos necesarios para la generación de horarios laborales factibles.
3. Desarrollar una función objetivo que permita evaluar la calidad de las soluciones generadas, considerando el cumplimiento de restricciones laborales y operativas, así como criterios de optimización asociados a la asignación de personal.
4. Implementar un Algoritmo Genético para la generación automática de horarios laborales, incorporando mecanismos de selección, cruzamiento, mutación y elitismo que permitan la evolución progresiva de las soluciones.
5. Desarrollar una interfaz gráfica que facilite la configuración de parámetros, la generación de horarios y la visualización de los resultados obtenidos por el algoritmo.

6. Evaluar experimentalmente el desempeño del Algoritmo Genético mediante distintas configuraciones de población, generaciones y probabilidad de cruzamiento, analizando su impacto sobre la calidad de las soluciones y el tiempo de ejecución.

7. Determinar una configuración de parámetros adecuada para el Algoritmo Genético, con base en los resultados obtenidos durante la experimentación.

1.5. Hipótesis

La implementación de un sistema automatizado para la generación de horarios laborales, basado en Algoritmos Genéticos, permitirá optimizar la asignación de turnos en una cadena de cines mediante la generación de soluciones factibles que cumplan con las restricciones legales y operativas del problema. Asimismo, contribuirá a reducir el esfuerzo requerido para la planificación manual y a mejorar la eficiencia del proceso de generación de horarios laborales.

1.6. Contribuciones

Las principales contribuciones de esta investigación aplicada son:

- Desarrollo de un sistema automatizado para la generación de horarios laborales en cadenas de cines.
- Aplicación y adaptación de un algoritmo genético a un contexto poco explorado, como la generación de horarios laborales en cines.
- Modelado del problema considerando variables relevantes del entorno, tales como empleados, turnos, puestos, disponibilidad, capacitación, horarios de operación, presupuesto semanal de horas, normativa laboral y políticas internas.
- Diseño de una función objetivo basada en penalizaciones para evaluar y comparar la calidad de las soluciones generadas.
- Separación entre restricciones fuertes y débiles, priorizando la factibilidad y optimizando posteriormente la calidad de los horarios.
- Validación experimental del modelo mediante diferentes configuraciones de parámetros del algoritmo.
- Identificación de una configuración eficiente del algoritmo genético.
- Contribución al apoyo en la toma de decisiones operativas.
- Reducción de tiempo y costos en la elaboración de horarios laborales.
- Propuesta de una herramienta con potencial de integración en sistemas reales de gestión laboral.
- Referencia para estudiantes e investigadores que trabajen temas de optimización y generación de horarios laborales.

1.7. Productos académicos generados

1.7.1. Participación en Congresos

Como producto académico derivado de la presente investigación, se participó con la exposición de la ponencia titulada *"Generación automática de horarios laborales de empleados de cines mediante el uso de algoritmos genéticos"*, presentada en el marco del 20º Congreso Estatal de Ciencia, Tecnología e Innovación y 14º Encuentro de Jóvenes Investigadores del Estado de Michoacán, llevado a cabo los días 8, 9 y 10 de octubre de 2025 en el Centro de Convenciones y Exposiciones de Morelia (CECONEXPO), en la ciudad de Morelia, Michoacán.

1.7.2. Registro de Derecho de Autor

Como producto tecnológico derivado de la presente investigación, se obtuvo el registro de derecho de autor ante el Instituto Nacional del Derecho de Autor (INDAUTOR). Este registro respalda la originalidad del trabajo y su potencial de aplicación en sistemas reales de gestión laboral.

Tipo: Registro de Derecho de Autor.

Autores: Eric César Guzmán Malvaez, Noel Enrique Rodríguez Maya y Jhacer Kharen Ruíz Garduño.

Título: Generación automática de horarios laborales de empleados de cines mediante el uso de Algoritmos Genéticos.

Rama: Programas de computación.

Descripción: Interfaz gráfica que implementa un algoritmo genético para la generación automática de horarios laborales de empleados de cines.

Registro de obra: INDAUTOR 03-2026-012609263100-01.

Fecha: 23 de febrero de 2026.

Estado: Terminado.

1.7.3. Publicación de artículos

Estado: En proceso.

Título: An Evolutionary Approach to Solving the Employee Timetabling Problem in a Cinema Chain.

Congreso internacional: Mexican International Conference on Artificial Intelligence.

Indexado: Libro Lecture Notes in Artificial Intelligence, DBLP, EI Compendex, INSPEC, SCImago, SCOPUS

ISSN: Electronic ISSN 2945-9141, Print ISSN 2945-9133

Fecha: del 02 al 06 de noviembre de 2026.

2. Estado del arte y de la técnica

2.1. Estado del Arte

La asignación de turnos laborales es un problema recurrente en diversas industrias del sector servicios, especialmente en aquellas con alta rotación de personal y horarios variables, como hospitales, aeropuertos y centros comerciales. Aunque tradicionalmente esta tarea se ha realizado de manera manual o mediante sistemas semiautomatizados, dichas alternativas presentan limitaciones de flexibilidad, escalabilidad y precisión cuando se aplican en escenarios dinámicos y complejos.

Diversas investigaciones han explorado métodos de Inteligencia Artificial para abordar este problema. En particular, los algoritmos genéticos han demostrado ser eficaces en tareas de optimización con múltiples restricciones, como es el caso de la programación de horarios. Estos algoritmos se inspiran en la evolución natural y son capaces de encontrar soluciones cercanas al óptimo en problemas donde el espacio de búsqueda es amplio y no lineal.

La generación de horarios para empleados es un problema clásico en la investigación con distintos métodos propuestos encaminados a mejorar la asignación de recursos y la eficiencia operacional. Soriano et al. (2020) definen este problema, conocido como Employee Scheduling Problem (ESP) o Workforce Scheduling, como la asignación óptima de turnos a empleados considerando restricciones legales, operativas y personales. En la industria del entretenimiento como los cines, este problema se vuelve particularmente complejo debido a la variabilidad en la demanda de clientes, la diversidad de roles (taquilla, limpieza, atención al cliente) y las preferencias de los empleados.

Cuartas y Muñoz (2024) abordaron los desafíos en la creación de horarios académicos en la Universidad EIA de Colombia, como la gestión eficiente de recursos, la satisfacción de preferencias individuales y la adaptación a restricciones cambiantes. Utilizando Algoritmos Genéticos, se desarrolló un sistema que optimiza la generación de horarios académicos. El enfoque incluye entrevistas con responsables del diseño de horarios, el desarrollo de una base

de datos para almacenar información clave, la construcción de un modelo matemático para representar el problema como una cuestión combinatoria y la implementación de un Algoritmo Genético con operadores de selección, cruce y mutación. Los resultados muestran mejoras significativas en la eficiencia de los espacios, la satisfacción de estudiantes y profesores, y la calidad de la gestión académica, validando los horarios generados frente a los métodos tradicionales.

Terán-Pozo et al. (2022) analizaron la influencia de los Algoritmos Genéticos en la generación de horarios académicos para una unidad educativa con el propósito de resolver las limitaciones asociadas a los métodos manuales de planificación tales como errores y pérdida de tiempo. Para ello, desarrollaron un sistema basado en Python siguiendo un proceso estructurado compuesto por las fases de planificación, diseño, implementación y pruebas. En su propuesta implementaron técnicas como selección, cruce y mutación, evaluando el "fitness" de las soluciones generadas. Los resultados evidenciaron una reducción significativa en el tiempo necesario para generar horarios, así como una mayor eficiencia en comparación con los métodos manuales. Aunque los autores identificaron áreas de mejora como la continuidad de los horarios, el modelo automatizado ofreció soluciones óptimas, validando su utilidad en entornos educativos.

De et al. (2024) desarrollaron un sistema automático para la generación de horarios académicos en la Unidad Educativa "Carlos Pomerio Zambrano" ubicada en la provincia de Manabí, Canuto, Ecuador. El objetivo principal fue mejorar la eficiencia del proceso administrativo relacionado con la planificación de horarios. Para ello, emplearon métodos como el histórico-lógico, la deducción y el análisis estadístico, además de una revisión bibliográfica exhaustiva. El sistema desarrollado permite a los docentes generar automáticamente la carga horaria para los periodos académicos, optimizando la gestión de recursos y mejorando la eficiencia administrativa. Los resultados destacan un avance significativo en la organización y gestión educativa, beneficiando tanto a la institución como al personal docente.

Acuña-Galván et al. (2022) presentaron un Algoritmo Genético diseñado en Java para optimizar la generación de roles académicos en la Licenciatura en Tecnologías de la Información de la Escuela Superior de Tizayuca. El modelo aborda restricciones fuertes y débiles, utiliza procesos de selección por torneo, cruzamiento uniforme y mutación, y se apoya en una base de datos centralizada para su operación. Los resultados destacan una notable reducción en el tiempo de creación de horarios y una alta adaptabilidad a las restricciones, mejorando la eficiencia administrativa y reduciendo el desgaste físico y mental del personal docente, aunque se identificaron oportunidades para optimizar aún más asignaciones como las de laboratorios y optativas.

Xu y Wang (2022) propusieron un modelo híbrido para la programación de turnos en centros de llamadas, combinando la colonia de abejas artificiales (ABC) con programación entera. La primera etapa de su modelo generaba soluciones preliminares que cumplían con restricciones fuertes (como descansos, horarios máximos y disponibilidad), mientras que la segunda refinaba las soluciones para atender preferencias del personal y equilibrar la carga laboral. En sus resultados mostraron que el enfoque híbrido lograba soluciones de alta calidad incluso en escenarios de gran escala, mejorando la satisfacción de los empleados y reduciendo costos operativos, lo que valida la pertinencia de aplicar técnicas bioinspiradas en problemas complejos de asignación de personal.

Kool (2022) abordó el problema de la programación de turnos en supermercados, destacando su complejidad y las limitaciones existentes en la literatura especializada. Como alternativa de solución, propuso un modelo basado en redes neuronales que trata la programación de turnos como un problema de clasificación multi-etiqueta, debido a que un mismo empleado puede ser asignado a múltiples intervalos de tiempo dentro de un horizonte de planificación. En este enfoque, cada intervalo de trabajo se representa como una etiqueta, permitiendo predecir de manera simultánea las asignaciones correspondientes a los distintos

horarios de un empleado. Esta formulación resulta adecuada para problemas de programación de personal, ya que considera no solo la asignación o no asignación de un empleado, sino también la combinación de intervalos de trabajo que deben cumplir con restricciones operativas específicas, tales como la disponibilidad del personal y la carga laboral esperada. El modelo emplea capas lineales y el algoritmo de descenso de gradiente estocástico para el entrenamiento y evaluación de los datos, obteniendo resultados prometedores a pesar de utilizar un conjunto limitado de características. Los resultados mostraron que la propuesta fue capaz de generar horarios que satisfacen restricciones relevantes del problema, particularmente aquellas relacionadas con la disponibilidad de los empleados y la cobertura de la demanda esperada.

Guan et al. (2021) estudiaron el problema de la programación de turnos en hospitales, considerando aspectos como el equilibrio de la carga laboral, las preferencias del personal y las restricciones operativas. Para abordar este problema, propusieron una alternativa híbrida que combina programación matemática con dos algoritmos de optimización: optimización por enjambre de partículas (Particle Swarm Optimization, PSO) y optimización por colonia de hormigas (Ant Colony Optimization, ACO). El algoritmo ACO, inspirado en el comportamiento de las hormigas al buscar rutas óptimas hacia fuentes de alimento, emplea feromonas simuladas para guiar la búsqueda hacia soluciones más eficientes. La integración de estas técnicas permitió modelar el problema como una optimización multiobjetivo, buscando equilibrar simultáneamente la eficiencia operativa y la calidad de las asignaciones de personal. Los resultados obtenidos mostraron mejoras significativas en la generación de horarios, evidenciando además la capacidad de escalabilidad y adaptación del modelo ante diferentes escenarios de planificación.

Sahraeian (2024) realizó una revisión detallada de la literatura sobre la programación de turnos, destacando su creciente complejidad y relevancia en sectores como la salud, el transporte y los servicios. Se categorizan los trabajos existentes en objetivos, dominios de aplicación, restricciones, metodologías de solución y métricas de evaluación. Entre los métodos utilizados

se encuentran algoritmos matemáticos exactos, heurísticos y metaheurísticos, abordando problemas de alta complejidad computacional (NP-hard). Brechas identificadas: se destacan la falta de modelos que consideren flexibilidad en descansos, la asignación de personal con habilidades múltiples, la integración insuficiente de tecnologías emergentes como la inteligencia artificial, y un enfoque limitado en escenarios de alta incertidumbre. Propuestas para futuras investigaciones: se sugiere desarrollar modelos que incorporen mayor flexibilidad en descansos y asignaciones, herramientas basadas en Inteligencia Artificial y aprendizaje automático, enfoques que optimicen la asignación de personal multicompetencia, y la exploración de restricciones emergentes como el bienestar laboral y la sostenibilidad ambiental. Además, se destaca la necesidad de metodologías robustas para gestionar incertidumbre en la demanda.

Gómez et al. (2020) abordaron la baja eficiencia en la generación de horarios académicos en instituciones educativas, proponiendo un sistema basado en Algoritmos Genéticos, seleccionados tras un análisis comparativo con otras técnicas metaheurísticas como GRASP, búsqueda tabú y colonia de hormigas. Definen qué Algoritmos Genéticos destacan por no requerir conocimientos previos del problema, su sencilla adecuación de parámetros, y la capacidad de manejar múltiples soluciones con tiempos reducidos. En comparación, técnicas como búsqueda tabú y colonia de hormigas muestran mayor complejidad en la configuración y tiempos más altos para resolver problemas similares. Los resultados de su sistema implementado con Algoritmos Genéticos evidencian una reducción del 99.12% en el tiempo de generación de horarios respecto a métodos manuales, alcanzando un nivel de eficiencia del 93.75%. Este enfoque no solo mejora la gestión administrativa, sino que optimiza los recursos académicos.

Henry Nelson et al. (2021) analizaron los desafíos asociados a la elaboración de horarios para actividades académicas, destacando el tiempo considerable que requiere su construcción manual y la complejidad de realizar modificaciones sin afectar las asignaciones de los profesores.

Para abordar esta problemática, propusieron el uso de Algoritmos Genéticos, los cuales optimizan las soluciones mediante operadores evolutivos como el cruzamiento y la mutación, priorizando el cumplimiento de restricciones fuertes relacionadas con la validez del horario sobre restricciones débiles orientadas a mejorar su calidad. Los resultados mostraron que esta técnica permite automatizar y acelerar significativamente el proceso de planificación, además de facilitar la reprogramación dinámica de actividades respetando las restricciones previamente establecidas, contribuyendo así a una mayor eficiencia en la gestión de horarios académicos.

Li et al. (2024) examinaron la problemática asociada a la programación en talleres de producción, considerando la complejidad de asignar recursos limitados, priorizar trabajos y organizar turnos de trabajadores. Para abordar este problema, propusieron un modelo de programación de enteros mixtos combinado con un algoritmo de evolución diferencial mejorado en dos etapas. Este algoritmo emplea estrategias heurísticas para generar soluciones iniciales de alta calidad y ajustar dinámicamente los turnos de acuerdo con la urgencia de las tareas. Los resultados experimentales demostraron que el enfoque propuesto reduce significativamente los días de retraso y optimiza las horas extras de los trabajadores, mejorando la eficiencia y adaptabilidad en entornos de producción complejos.

Burduk et al. (2022) analizaron los desafíos en la programación de producción en empresas manufactureras, destacando cómo la dependencia del humano genera errores y costos innecesarios. Para abordar este problema, presentaron el algoritmo Neuro-Tabu Search, que combina dos enfoques: la búsqueda Tabú, que evita soluciones repetitivas y ayuda a encontrar opciones óptimas, y redes neuronales auto-organizativas, que permiten ajustar las decisiones en tiempo real y mejorar la eficiencia del proceso. Este enfoque no solo optimiza la asignación de recursos y reduce costos, sino que también mejora la eficiencia, sostenibilidad y adaptabilidad del proceso de producción, ofreciendo una solución innovadora para las empresas del sector.

Bożejko et al. (2021) abordaron el problema en la industria automotriz, donde la asignación de recursos y la disponibilidad de operadores son limitadas. Presentaron un algoritmo llamado Neuro-Tabu Search, que combina estrategias de búsqueda inteligente con redes neuronales para evitar repetir errores y mejorar las decisiones. En sus resultados mostraron que este enfoque supera a otros métodos tradicionales, como el recocido simulado y la búsqueda tabú clásica, al reducir significativamente los tiempos de producción y mejorar la eficiencia en la gestión de recursos.

Soriano et al. (2020) abordaron el problema de la programación de horarios de empleados considerando simultáneamente la demanda de personal, las pausas laborales, las horas extras y las preferencias individuales de los trabajadores. Para resolver esta problemática, emplearon una combinación del Proceso de Jerarquía Analítica (Analytic Hierarchy Process, AHP) para priorizar criterios de decisión y Programación Lineal Entera Mixta (Mixed Integer Linear Programming, MILP) para generar soluciones factibles. Los resultados mostraron que el enfoque que propusieron permite generar horarios eficientes que respetan las preferencias de los empleados, optimizan la utilización de los recursos disponibles y cumplen con las restricciones laborales establecidas. Asimismo, los autores concluyeron que la metodología presenta características de escalabilidad y adaptabilidad, siendo adecuada para entornos con alta variabilidad en la demanda de personal.

Kletzander y Musliu (2018) abordaron el problema general de programación de horarios de empleados mediante el desarrollo de un framework modular diseñado para la implementación de diversos algoritmos heurísticos. La propuesta permite incorporar y modificar restricciones específicas sin necesidad de rediseñar completamente el sistema, facilitando su adaptación a distintos escenarios de planificación. El framework integra mecanismos para la generación de movimientos y la evaluación dinámica de restricciones, así como un algoritmo basado en recocido simulado, técnica de optimización inspirada en el proceso físico de enfriamiento gradual

de materiales que permite explorar y refinar progresivamente soluciones candidatas. Los resultados experimentales demostraron la eficacia del enfoque propuesto, obteniendo soluciones robustas y escalables en diferentes instancias del problema de programación de personal.

Viñas y Rodríguez (2018) desarrollaron un *software* para la generación automática de horarios académicos en instituciones educativas, abordando el problema desde una perspectiva metaheurística. Su propuesta utilizó Algoritmos Genéticos con operadores de selección, cruce uniforme y mutación adaptativa, logrando reducir significativamente el tiempo de programación y mejorar la flexibilidad frente a cambios imprevistos. Además, incluyeron una comparación con el recocido simulado, demostrando que, aunque esta técnica producía soluciones viables, presentaba mayores tiempos de cómputo. En sus resultados demostraron que los Algoritmos Genéticos no solo ofrecían soluciones más eficientes, sino también mayor escalabilidad y adaptabilidad a diferentes contextos, lo que refuerza su pertinencia como alternativa para problemas complejos de asignación de turnos.

Uribe Pérez (2016) implementó un modelo de asignación de turnos para conductores en un sistema de transporte masivo. El Algoritmo Genético diseñado incorporó la variabilidad de la demanda y las restricciones de descanso. A partir de su aplicación, se logró reducir significativamente el tiempo de programación y mejorar la equidad entre trabajadores, evitando la asignación repetitiva de los turnos menos deseados. Asimismo, se observaron mejoras en la cobertura operativa sin exceder los límites laborales establecidos por la normativa.

Velásquez-Restrepo et al. (2016) realizaron una revisión de metodologías cuantitativas para la optimización de servicios de urgencias hospitalarias. Entre las principales técnicas identificadas destacan la simulación de eventos discretos, la programación matemática, la dinámica de sistemas, la simulación Monte Carlo, la simulación basada en agentes, así como métodos heurísticos y metaheurísticos, entre ellos los Algoritmos Genéticos. Estas metodologías han sido aplicadas al análisis de la capacidad de atención, la planeación de recursos, la

generación de calendarios y la rotación de turnos del personal asistencial. Los resultados revisados muestran que este tipo de técnicas contribuye a mejorar la cobertura del servicio, reducir la sobrecarga laboral, disminuir los tiempos de espera y apoyar una atención más eficiente en situaciones de alta demanda.

Ochoa Buitrago (2012) analizó, en el sector hospitalario, la variabilidad en la asignación de personal en unidades de urgencias, un entorno caracterizado por la alta rotación de personal, la imprevisibilidad de la demanda y la necesidad de cumplir normativas estrictas. El estudio evidenció que la falta de herramientas automatizadas puede generar sobrecarga laboral y reducir la eficiencia operativa. Mediante la aplicación de técnicas de optimización, se propuso un modelo orientado a equilibrar la distribución de turnos, incrementar la eficiencia del servicio y contribuir al bienestar del personal. Esto evidencia la aplicabilidad de estas metodologías en contextos donde las fluctuaciones de la demanda son críticas.

Rodríguez et al. (2014) proponen un Algoritmo Genético para resolver un problema de generación automática de horarios académicos, tomando como caso de estudio el Instituto Tecnológico de Zitácuaro. El problema se plantea como la asignación de cursos, profesores, aulas y horarios disponibles, considerando restricciones fuertes y débiles. Los autores señalan que la generación manual de horarios en instituciones educativas requiere una cantidad considerable de tiempo y esfuerzo, debido a la combinación de múltiples variables y restricciones. Un aporte principal de su propuesta es la incorporación de un mecanismo de reparación, el cual permite corregir soluciones que inicialmente no cumplen con las restricciones fuertes y transformarlas en soluciones factibles. En los resultados obtenidos, se identifica que una población amplia, junto con tasas altas de cruzamiento y mutación, favorece el desempeño del algoritmo. Con esta configuración, el modelo logra eliminar las violaciones de restricciones fuertes, aunque permanecen algunas penalizaciones asociadas a restricciones débiles.

A partir de los trabajos revisados, se observa que los Algoritmos Genéticos y otras metaheurísticas han sido aplicados con éxito en la generación automática de horarios en sectores como educación, salud, manufactura, centros de llamadas, supermercados y transporte. Sin embargo, estos estudios se desarrollan bajo condiciones distintas a las de una cadena de cines, donde intervienen factores como demanda variable, diversidad de puestos, capacitación diferenciada, horarios extendidos y presupuesto semanal de horas.

En este sentido, la presente investigación atiende una brecha específica al adaptar un Algoritmo Genético al contexto operativo de cadenas de cines, integrando restricciones legales, operativas y organizacionales en un modelo de optimización con validación experimental y una interfaz funcional para la generación y visualización de horarios.

2.2. Estado de la Técnica

Actualmente, la programación de turnos y horarios cuenta con múltiples desarrollos tecnológicos que van desde soluciones comerciales hasta implementaciones académicas basadas en código abierto. Estas herramientas utilizan diferentes enfoques, como la programación lineal, heurísticas clásicas, algoritmos evolutivos y técnicas de inteligencia artificial.

El desarrollo de soluciones para la programación de horarios ha dado lugar a la aplicación de diversas técnicas, entre las que destacan:

- Algoritmos Genéticos (AG).
- Programación Lineal Entera Mixta (MILP).
- Optimización por Colonia de Hormigas (ACO).
- Enjambre de Partículas (PSO).
- Redes Neuronales.
- Sistemas Expertos.

- Métodos Heurísticos y Metaheurísticos.
- Modelos Basados en Simulación.

Además de las técnicas y herramientas mencionadas, existen librerías y frameworks que han tomado relevancia en los últimos años:

DEAP (Distributed Evolutionary Algorithms in Python): una de las librerías más usadas para implementar algoritmos evolutivos, con soporte para Algoritmos Genéticos, evolución diferencial y estrategias evolutivas (Fortin et al., 2012).

PyGAD: biblioteca de código abierto desarrollada en Python para la implementación de Algoritmos Genéticos. Permite definir funciones de aptitud personalizadas y configurar parámetros como tamaño de población, selección de padres, cruzamiento y mutación (Gad, 2021).

Google OR-Tools: ampliamente utilizado en la industria, especialmente su CP-SAT Solver, que combina programación por restricciones (CP) con satisfacción de restricciones (SAT) (Perron & Furnon, 2025).

OptaPlanner: herramienta open source desarrollada en Java aplicada en empresas de transporte, retail y salud, enfocada en problemas de Workforce Scheduling (OptaPlanner, 2023).

A continuación, se presenta una tabla comparativa (Tabla 1) con las principales técnicas utilizadas para la generación automática de horarios, destacando sus ventajas y desventajas:

Tabla 1. *Comparación entre las principales técnicas utilizadas para la generación automática de horarios.*

Técnica	Ventajas	Desventajas	Referencia

Algoritmos Genéticos	Adaptabilidad a restricciones. Manejo de múltiples soluciones. Buena eficiencia en problemas grandes.	Puede requerir ajustes manuales en parámetros. No siempre garantiza el óptimo global.	Cuartas & Muñoz (2024); Terán-Pozo et al. (2022); De et al. (2024); Acuña-Galván et al. (2022); Viñas & Rodríguez (2018); Gómez et al. (2020); Uribe Pérez (2016); Velásquez-Restrepo et al. (2016).
Programación Lineal Entera Mixta	Genera soluciones óptimas. Precisión en manejo de restricciones laborales.	Computacionalmente costosa en problemas grandes. Difícil ajuste a demandas dinámicas.	Soriano et al. (2020).
Optimización por Colonia de Hormigas (ACO)	Encuentra rutas óptimas en asignación de turnos. Flexible ante restricciones cambiantes.	Requiere muchos parámetros y tiempo para converger.	Guan et al. (2021).
Enjambre de Partículas (PSO)	Buen rendimiento en optimización multiobjetivo. Escalable a problemas grandes.	Puede estancarse en óptimos locales. Ajuste fino de parámetros.	Guan et al. (2021).
Redes Neuronales / Aprendizaje Automático	Aprende patrones históricos. Ajuste dinámico a la demanda en tiempo real.	Requiere grandes volúmenes de datos. Difícil interpretación de resultados.	Kool (2022).
Métodos Heurísticos y Metaheurísticos (GRASP, Búsqueda Tabú, Recocido Simulado, etc.)	Encuentran soluciones en menor tiempo. Se adaptan a restricciones específicas.	No garantizan soluciones óptimas. Necesitan calibración constante.	Gómez et al. (2020); Kletzander & Musliu (2018); Burduk et al. (2022); Bożejko et al. (2021).
Evolución Diferencial (DE)	Genera soluciones iniciales de calidad. Adaptable a problemas complejos.	Puede requerir ajustes complejos. Tiempo de cómputo alto en problemas grandes.	Li et al. (2024).
Neuro-Tabu Search (híbrido IA + Tabú)	Optimiza recursos. Mejora eficiencia y sostenibilidad.	Mayor complejidad de implementación. Requiere alta capacidad computacional.	Burduk et al. (2022); Bożejko et al. (2021).

Aprendizaje por Refuerzo (DRL)	Se adapta dinámicamente a la demanda. No requiere reglas explícitas.	Necesita gran cantidad de simulaciones. Difícil de interpretar.	Sahraeian (2024).
Programación por Restricciones (CP)	Excelente para restricciones fuertes (leyes laborales, descansos obligatorios). Proporciona soluciones factibles garantizadas.	Ineficiente en problemas de gran escala. No aprende de datos históricos.	Kletzander & Musliu (2018).

***Nota**: Comparación entre las principales técnicas utilizadas para la generación automática de horarios según artículos consultados.*

A continuación, se presenta la *Tabla 2* con los detalles de algunos de los principales softwares de gestión de turnos, incluyendo sus características, planes y precios, ventajas y desventajas.

Tabla 2. *Comparación de softwares para la gestión de turnos.*

Herramienta	Características	Planes y Precios	Ventajas	Desventajas	Referencia
Gurobi	Optimizador matemático avanzado (PL, MILP, QP, etc.)	Licencias académicas gratuitas. Comerciales: personalizadas.	Alta velocidad y precisión. Soporta múltiples lenguajes.	Requiere conocimientos técnicos avanzados. Costoso.	Gurobi Optimization, LLC (2022)
Odoo	ERP completo con módulo de RRHH y planificación.	Versión comunitaria gratuita. Empresarial: desde $24 USD/usuario/mes + módulos.	Integración total con otros módulos de negocio. Personalizable.	Puede volverse costoso con muchos módulos.	Odoo S.A. (2023)
OptaPlanner	Motor de optimización open source en Java.	Gratuito (open source).	Flexible, altamente personalizable, buena comunidad.	Requiere conocimientos en Java y desarrollo adicional.	OptaPlanner (2023)
Kronos (UKG)	Gestión de fuerza laboral:	Planes personalizados por empresa.	Muy robusto. Cumple con	Alto costo y complejidad en	UKG (2022)

	asistencia, turnos, análisis.		normativas laborales.	implementación.	
Workday Scheduling	Planificación basada en IA, cumplimiento legal, experiencia del empleado.	Planes personalizados (requiere contacto).	Funcionalidades modernas y centradas en el empleado.	Caro. Implementación compleja.	Workday (2022)
Shiftboard	Gestión de turnos y fuerza laboral, cambios en tiempo real.	Plan Enterprise y Enterprise Plus (precios bajo demanda).	Automatización, comunicación efectiva, adaptabilidad.	Curva de aprendizaje alta. Funciones limitadas en planes básicos.	Shiftboard (2021)
Woffu	Gestión de ausencias, vacaciones e integraciones.	Planes personalizados.	Fácil de usar. Ahorra tiempo en gestión de permisos.	Poca funcionalidad fuera de ausencias.	Woffu (2021)
Bizneo HR	Suite completa de RRHH (reclutamiento, desempeño, ausencias, formación).	Desde $69 MXN/empleado/mes. Hasta plan Enterprise.	Modular. Escalable. Muy completo.	Algunas funciones solo en planes altos. Implementación lenta.	Bizneo (2022)
PGPlanning	Gestión de proyectos y tareas, seguimiento de progreso.	Planes personalizados.	Visual y colaborativo. Adaptado a pymes y empresas.	Necesita capacitación. Algunas funciones sobran en equipos pequeños.	PGPlanning (2021)
Sesame HR	Control horario, evaluaciones, vacaciones, firma digital.	Desde $85 MXN/empleado/mes (Starter).	Enfoque en autogestión. Visual atractivo.	Plan gratuito limitado. Funciones avanzadas solo en planes superiores.	Sesame HR (2022)
Plano WFM	Gestión de turnos, previsión de demanda, control de personal.	Planes personalizados.	Eficiente para empresas con turnos rotativos.	Puede requerir integración externa.	Plano WFM (2021)
SDP Era Plan	Planificación estratégica y control horario para medianas y	Planes personalizados.	Muy detallado. Adaptado a organizaciones complejas.	Poca documentación pública.	SDP Era (2021)

	grandes empresas.				
aTurnos	Planificación de turnos, comunicación interna, control de tiempo, cambios y guardias.	Desde 1€/usuario/mes. Planes escalables.	Muy económico. Visual y fácil de implementar.	Integraciones limitadas.	aTurnos (2021)
Woxi Digital	Automatización de RRHH, control de horarios, tareas, permisos y fichajes.	Planes personalizados.	Ligero, rápido, enfocado en transformación digital.	No tan robusto como plataformas grandes.	Woxi Digital (2021)
ClickUp	Gestión de proyectos y tareas. Automatizaciones, integraciones y seguimiento de tiempo.	Gratis. Luego desde $5 USD/usuario/mes.	Muy versátil. Integración con otras apps.	Algunas funciones bloqueadas en planes superiores.	ClickUp (2022)

La programación de turnos ha evolucionado gracias a una variedad de enfoques, desde modelos matemáticos precisos hasta algoritmos inspirados en la naturaleza. Cada técnica tiene sus fortalezas: unas garantizan soluciones óptimas en contextos simples, otras se adaptan mejor a escenarios cambiantes y complejos. Sin embargo, estas técnicas presentan limitaciones, ya que algunas requieren calibración constante de parámetros, demandan mayores recursos computacionales o dependen de datos difíciles de obtener.

Existen herramientas comerciales y de código abierto que ofrecen funcionalidades útiles para la gestión de personal, pero no siempre se ajustan de manera directa a las condiciones particulares del problema estudiado. Por ello, la propuesta de esta investigación se orienta a integrar un modelo de optimización basado en Algoritmos Genéticos con una herramienta funcional que permita configurar datos operativos, generar horarios y visualizar los resultados de forma estructurada.

3. Marco teórico

3.1. Fundamentos de la programación de turnos y horarios

La programación de turnos, también conocida como Workforce Scheduling o Employee Scheduling Problem (ESP), consiste en la asignación de empleados a turnos laborales considerando restricciones legales, operativas y personales (Soriano et al., 2020). Se trata de un problema NP-hard, debido al gran número de combinaciones posibles y restricciones simultáneas, lo que hace inviable resolverlo de manera exhaustiva en contextos reales (Kletzander & Musliu, 2018).

La complejidad principal de este problema radica en su naturaleza combinatoria. A medida que aumenta el número de empleados, turnos y restricciones, el espacio de soluciones posibles crece de forma exponencial. En consecuencia, las soluciones tradicionales, como el diseño manual o los métodos deterministas, resultan insuficientes en entornos dinámicos donde los cambios pueden ocurrir en cuestión de horas (Velásquez-Restrepo et al., 2016).

La programación de turnos no debe entenderse únicamente como un problema técnico, sino también como un reto humano y organizacional. Una asignación mal diseñada puede impactar negativamente en la satisfacción laboral, incrementar la rotación de empleados y afectar la experiencia del cliente. Por ello, la búsqueda de soluciones automatizadas y flexibles, capaces de integrar tanto las restricciones normativas como las necesidades del personal, se ha convertido en una prioridad en la investigación aplicada y en el desarrollo de herramientas de Inteligencia Artificial (Sahraeian, 2024).

3.1.1. Métodos tradicionales de programación de turnos

Durante años, la asignación de turnos se ha abordado con métodos tradicionales que combinan fórmulas matemáticas y reglas prácticas. Estos enfoques buscan cumplir con las

normas laborales y operativas, mientras se intenta cubrir la demanda y mantener los costos bajo control.

Uno de los enfoques más utilizados es la Programación Lineal Entera Mixta (MILP, Mixed Integer Linear Programming), la cual modela el problema mediante variables binarias que representan la asignación de empleados a turnos específicos. Aunque este método puede proporcionar soluciones óptimas, el incremento en el número de empleados, turnos y restricciones provoca un crecimiento significativo en la complejidad computacional del problema, lo que limita su aplicación en entornos altamente dinámicos y de gran escala (Soriano et al., 2020; Sahraeian, 2024).

Ante estas limitaciones, surgieron las heurísticas: estrategias más ágiles que usan reglas sencillas para construir horarios funcionales en poco tiempo. Por ejemplo, algunos algoritmos asignan primero los turnos más críticos, mientras que otros ajustan progresivamente una solución inicial. Aunque estos métodos ofrecen ventajas en términos de eficiencia computacional, no siempre garantizan la mejor distribución, ya que pueden quedarse atrapados en soluciones poco óptimas sin explorar otras alternativas.

Si bien estos métodos tradicionales han demostrado ser eficaces en entornos relativamente estables, como instituciones educativas o centros de salud con niveles moderados de complejidad, su aplicación se vuelve más limitada cuando aumenta el número de empleados, turnos y restricciones a considerar. En este contexto, a partir de la década de 1990, las metaheurísticas comenzaron a consolidarse como una alternativa para abordar problemas de optimización de alta complejidad. Entre las técnicas más representativas se encuentran el recocido simulado, la búsqueda tabú y los algoritmos evolutivos, las cuales permiten explorar grandes espacios de búsqueda y obtener soluciones de alta calidad en tiempos computacionalmente razonables.

Estas metodologías ofrecen una mayor flexibilidad para adaptarse a escenarios dinámicos y a problemas caracterizados por múltiples restricciones, proporcionando un equilibrio adecuado entre calidad de solución y tiempo de procesamiento. Aunque los métodos tradicionales han contribuido significativamente al estudio de la programación de turnos, sus limitaciones en términos de flexibilidad y escalabilidad reducen su aplicabilidad en entornos altamente dinámicos, como las cadenas de cines. Por esta razón, resulta pertinente explorar enfoques basados en metaheurísticas, particularmente Algoritmos Genéticos, debido a su capacidad para generar soluciones factibles y eficientes en problemas de asignación de personal con alta complejidad combinatoria.

3.1.2. Metaheurísticas en programación de horarios

Ante el crecimiento de complejidad en el problema, los métodos heurísticos tradicionales comenzaron a verse limitados, por lo que surgieron los métodos metaheurísticos: un conjunto de algoritmos que no garantizan una solución óptima, pero que ofrecen soluciones de alta calidad en tiempos razonables, lo que las hace particularmente útiles en problemas de asignación de recursos humanos (Glover & Laguna, 1997; Talbi, 2009).

Una de las más utilizadas es el Recocido Simulado (Simulated Annealing), inspirado en el proceso físico de enfriamiento de los metales. Su fortaleza radica en la capacidad de escapar de los llamados “óptimos locales”, ya que de manera controlada permite aceptar temporalmente soluciones de peor calidad con el fin de explorar más ampliamente el espacio de búsqueda (Kirkpatrick, Gelatt & Vecchi, 1983). Esto lo convierte en una herramienta flexible en la generación de horarios donde abundan restricciones y combinaciones posibles.

Otra técnica relevante es la Búsqueda Tabú (Tabu Search), la cual utiliza mecanismos de memoria para evitar que el algoritmo regrese a soluciones previamente exploradas. Gracias a este enfoque, la búsqueda se orienta hacia regiones nuevas del espacio de soluciones, lo que

aumenta las probabilidades de encontrar combinaciones más eficientes (Glover, 1989). En el tema de asignación de turnos, esta técnica ha sido aplicada en hospitales y universidades con buenos resultados, reduciendo tiempos de cómputo y equilibrando cargas de trabajo.

El método GRASP (Greedy Randomized Adaptive Search Procedure) combina decisiones aleatorias con un poco de azar para construir soluciones iniciales de calidad, las cuales posteriormente se van puliendo mediante búsquedas locales. Este enfoque ha demostrado ser eficaz en contextos académicos y logísticos, donde las restricciones fuertes conviven con la necesidad de flexibilidad (Feo & Resende, 1995).

Aunque los resultados con estos métodos no son los óptimos sus características han permitido sus usos exitosos en sectores como la educación, salud y transporte. Por ejemplo, Gómez et al. (2020) compararon algoritmos de búsqueda tabú, GRASP y colonia de hormigas frente a los Algoritmos Genéticos en la generación de horarios académicos, demostrando que, aunque los primeros eran útiles en escenarios medianos, los Algoritmos Genéticos alcanzaban mejores resultados en eficiencia y escalabilidad.

Se puede afirmar que las metaheurísticas representan un punto intermedio entre las heurísticas y los modelos modernos de Inteligencia Artificial y que además son antecedentes claves que abrieron camino a los algoritmos evolutivos entre ellos los Algoritmos Genéticos, que son el eje central de esta investigación.

3.1.3. Algoritmos Genéticos

La evolución de los enfoques para la programación de turnos muestra un recorrido lógico: los métodos tradicionales aportaron rigor matemático, pero resultaron poco escalables; las metaheurísticas ofrecieron flexibilidad y rapidez, aunque sin garantías de alcanzar siempre soluciones óptimas. Este camino abrió la puerta a un enfoque más potente y versátil: los algoritmos evolutivos, y dentro de ellos, los Algoritmos Genéticos.

El cómputo evolutivo nació a mediados del siglo XX como un intento de trasladar a la computadora los principios de la evolución natural. En 1954, Nils Aall Barricelli realizó las primeras simulaciones de procesos evolutivos, demostrando que la selección y la variación podían dar lugar a comportamientos emergentes en entornos artificiales. Posteriormente, en las décadas de 1960 y 1970, diferentes investigadores consolidaron enfoques que hoy se consideran las tres ramas fundacionales: las estrategias evolutivas (Rechenberg y Schwefel), la programación evolutiva (Fogel y colaboradores) y los Algoritmos Genéticos (Holland).

La motivación de estos trabajos fue clara: los problemas de optimización en el mundo real —con espacios de búsqueda enormes, múltiples restricciones y cambios constantes— resultaban imposibles de resolver con métodos deterministas o exactos en un tiempo razonable. El cómputo evolutivo apareció entonces como una alternativa flexible, estocástica y adaptativa, capaz de encontrar soluciones de buena calidad en problemas donde no es factible garantizar el óptimo global (Holland, 1975).

Los Algoritmos Genéticos toman como base los principios de la evolución natural y la genética para diseñar mecanismos de búsqueda y optimización. La biología no solo provee la metáfora, sino también los procesos concretos que luego se traducen en operaciones computacionales.

La teoría de Charles Darwin (1859) planteó que los organismos mejor adaptados al ambiente tienen mayores probabilidades de sobrevivir y reproducirse, transmitiendo sus características a la siguiente generación. Este proceso de variación, selección y herencia es el núcleo de los Algoritmos Genéticos. Cada posible solución de un problema se representa como un “individuo” dentro de una población; los más aptos son seleccionados y generan descendencia, asegurando que las mejores características tiendan a prevalecer con el paso de las generaciones.

A los principios evolutivos de Darwin (1859) basados en la selección natural, se suman los descubrimientos de Mendel (1866) sobre herencia genética, que explican cómo los rasgos se transmiten entre generaciones. Estos fundamentos biológicos constituyen la base conceptual de los Algoritmos Genéticos, en los cuales las soluciones potenciales son representadas mediante cromosomas y genes que evolucionan a través de procesos análogos a la reproducción, el cruzamiento y la mutación (Goldberg, 1989).

La verdadera potencia de los Algoritmos Genéticos surge de la integración entre evolución y genética. En el diseño del modelo se considera lo siguiente:

- Cada solución se representa como un cromosoma (por ejemplo, una cadena binaria o un vector).
- Una función de aptitud (fitness function) evalúa la calidad de cada solución según los objetivos del problema.
- Los individuos con mayor aptitud tienen más probabilidades de ser seleccionados para reproducirse.
- Los operadores de cruce y mutación generan nuevas soluciones que combinan y varían la información genética.
- Con el paso de las generaciones, la población se adapta, encontrando soluciones cada vez más cercanas al óptimo.

John Holland formalizó estos principios en su libro Adaptation in Natural and Artificial Systems (1975), estableciendo la base teórica de los Algoritmos Genéticos e introduciendo el Teorema de los Esquemas. Este teorema explica cómo ciertos patrones parciales de solución presentes en los cromosomas, llamados esquemas, tienden a conservarse y propagarse dentro de la población cuando presentan un buen desempeño. En el contexto de la generación de

horarios laborales, esto puede interpretarse como la conservación de combinaciones favorables de asignación, por ejemplo, empleados capacitados, turnos factibles o soluciones con menor penalización, las cuales contribuyen a mejorar progresivamente la calidad de los horarios generados.

Los Algoritmos Genéticos son una traducción computacional de los procesos naturales de evolución y herencia. Su éxito radica en su capacidad de combinar exploración (gracias a la variabilidad introducida por mutación y cruce) con explotación (favoreciendo la reproducción de las soluciones más aptas), lo que los convierte en herramientas sumamente útiles para resolver problemas complejos de programación de horarios, logística y optimización en general.

3.1.4. Modelos de optimización multiobjetivo

En el ámbito de la programación de horarios, resulta insuficiente plantear el problema como un modelo mono objetivo. Aunque los enfoques tradicionales suelen centrarse en minimizar costos operativos o maximizar la cobertura de turnos, en la práctica se deben considerar múltiples criterios simultáneamente. Esto da lugar a los modelos de optimización multiobjetivo, que buscan un equilibrio entre distintos factores.

En el caso de la asignación de turnos, algunos de los objetivos más relevantes incluyen:

- Cumplimiento normativo: garantizar que no se excedan las horas máximas legales y que se respeten los descansos obligatorios.
- Eficiencia operativa: asegurar la cobertura adecuada de personal en horas de alta demanda, evitando tanto la sobrecarga como la ociosidad.
- Equidad en la distribución de turnos: lograr que los horarios asignados sean justos, evitando concentrar los turnos menos deseados en un mismo grupo de empleados.

- Reducción de costos: optimizar el uso de la mano de obra en función del presupuesto disponible.
- Satisfacción del personal: tomar en cuenta las preferencias individuales, la disponibilidad y la capacitación de los empleados.

La característica central de la optimización multiobjetivo es que no busca una única solución “óptima”, sino un conjunto de soluciones no dominadas. Una solución se considera no dominada cuando no existe otra que sea mejor en todos los objetivos simultáneamente. Esto permite al tomador de decisiones elegir entre diferentes alternativas según las prioridades del contexto (Coello Coello, Lamont & Van Veldhuizen, 2007).

Los algoritmos genéticos se adaptan de manera natural a este tipo de problemas, debido a que trabajan con poblaciones de soluciones y pueden explorar distintas regiones del espacio de búsqueda de manera paralela. En particular, modelos como NSGA-II han demostrado ser efectivos en problemas multiobjetivo complejos, al mantener diversidad entre las soluciones y favorecer la convergencia hacia el frente de Pareto (Deb et al., 2002).

Al llevar este enfoque a la programación de horarios laborales, se entiende que un buen horario no depende solamente de reducir costos o cubrir turnos disponibles. También es necesario considerar, al mismo tiempo, el cumplimiento de las restricciones laborales, operativas y organizacionales.

Por lo tanto, los modelos multiobjetivo representan un avance frente a los enfoques tradicionales, porque permiten analizar problemas en los que intervienen varios criterios de decisión. En esta investigación, dicha perspectiva sirve como base conceptual para el uso de una función objetivo basada en penalizaciones ponderadas. De esta manera, primero se prioriza el cumplimiento de las restricciones fuertes y, posteriormente, se optimizan criterios relacionados con la calidad operativa de los horarios generados.

4. Metodología

La presente metodología describe el enfoque experimental adoptado para el desarrollo y evaluación de un modelo de generación automática de horarios laborales basado en Algoritmos Genéticos.

4.1. Enfoque

La investigación adoptó un enfoque cuantitativo, basado en el análisis de variables numéricas que permitieron evaluar objetivamente el desempeño del modelo propuesto. Para ello, se emplearon métricas como el valor de la función objetivo (*fitness*), el cumplimiento de restricciones y el tiempo de ejecución del algoritmo, lo que facilitó la comparación sistemática entre distintas configuraciones y la obtención de resultados medibles y reproducibles (Creswell, 2014).

Asimismo, se consideró un enfoque experimental, ya que se modificaron de manera controlada parámetros del Algoritmo Genético, tales como el tamaño de la población, el número de generaciones y la probabilidad de cruzamiento, con el propósito de analizar su efecto sobre el desempeño del modelo bajo condiciones controladas (Hernández Sampieri et al., 2018).

El estudio fue de tipo aplicado, dado que se orientó a la solución de un problema real en el ámbito operativo de una cadena de cines, buscando desarrollar una herramienta funcional que contribuya a la mejora de los procesos de planificación de horarios laborales.

Desde el punto de vista metodológico, el problema corresponde a un caso de optimización combinatoria altamente restringido, cuya complejidad justifica el uso de técnicas metaheurísticas, específicamente Algoritmos Genéticos, como alternativa eficiente para la obtención de soluciones factibles y de buena calidad en tiempos computacionales razonables.

4.2. Diseño de investigación

El diseño de la investigación es experimental controlado, orientado a evaluar el desempeño del Algoritmo Genético mediante la variación sistemática de sus parámetros. Para

ello, se modifican de manera controlada variables como el tamaño de la población, el número de generaciones y la tasa o probabilidad de cruzamiento, las cuales influyen directamente en el equilibrio entre exploración y explotación durante el proceso evolutivo (Goldberg, 1989).

Debido a la naturaleza probabilística del Algoritmo Genético, cada configuración experimental se ejecutó 30 veces bajo las mismas condiciones, con el fin de reducir la variabilidad inherente al proceso y obtener resultados estadísticamente más representativos. A partir de estas ejecuciones se calcularon métricas como la penalización promedio y la variabilidad de los resultados obtenidos. Adicionalmente, se adoptó un enfoque comparativo que permitió identificar las configuraciones con mejor equilibrio entre calidad de solución y eficiencia computacional.

4.3. Variables

Se definieron variables independientes y dependientes con el propósito de analizar de manera estructurada el comportamiento del Algoritmo Genético y su impacto en la calidad de las soluciones generadas.

4.3.1. Variables independientes

- **Tamaño de la población (P):** número de individuos por generación, relacionado con la diversidad del espacio de búsqueda y el costo computacional.
- **Número de generaciones (G):** cantidad de iteraciones del proceso evolutivo, asociado al refinamiento progresivo de las soluciones.
- **Tasa o probabilidad de cruzamiento (Pc):** proporción con la que dos individuos seleccionados intercambian información genética para generar nuevas soluciones.

4.3.2. Variables dependientes

Las variables dependientes corresponden a los resultados obtenidos a partir de la ejecución del Algoritmo Genético y permiten evaluar su desempeño bajo distintas configuraciones.

- **Valor de la función objetivo (fitness):** mide la calidad global de la solución.
- **Penalización por restricciones fuertes:** indica el grado de incumplimiento de restricciones obligatorias.
- **Penalización por restricciones débiles**: mide la calidad operativa de las soluciones factibles.
- **Tiempo de ejecución**: evalúa el costo computacional del algoritmo.

La relación entre estas variables independientes y dependientes permite analizar el compromiso característico entre calidad de solución y eficiencia computacional en el algoritmo.

4.4. Métricas de evaluación

Para evaluar el desempeño del Algoritmo Genético propuesto, se definieron un conjunto de métricas que permiten medir de manera objetiva la calidad de las soluciones generadas, el grado de cumplimiento de las restricciones del problema y el costo computacional asociado al proceso. Estas métricas constituyen la base para el análisis comparativo entre diferentes configuraciones del algoritmo y para la validación del modelo propuesto. A continuación, se definen las métricas empleadas.

4.4.1. Función objetivo (*fitness*)

La métrica principal utilizada es el valor de la función objetivo, también conocido como fitness, la cual representa la calidad global de una solución. Esta función se define a partir de la combinación ponderada de penalizaciones asociadas a restricciones fuertes y débiles, estableciendo una jerarquía de prioridad entre ambas:

$$F(X) = \lambda_H \cdot Hard(X) + \lambda_S \cdot Soft(X)$$

Donde:

- $Hard(X)$: penalización por incumplimiento de restricciones fuertes.
- $Soft(X)$: penalización por incumplimiento de restricciones débiles.
- $\lambda_H \gg \lambda_S$: factor de ponderación que prioriza el cumplimiento de restricciones obligatorias.

El objetivo del algoritmo es minimizar el valor de $F(X)$, de tal forma que se obtengan soluciones factibles (sin violaciones de restricciones fuertes) y con alta calidad operativa (mínima penalización en restricciones débiles).

4.4.2. Penalización por restricciones fuertes

Esta métrica mide el nivel de incumplimiento de las restricciones obligatorias del problema. Dado que estas restricciones definen la factibilidad de la solución, su cumplimiento es prioritario dentro del proceso de optimización.

Un valor de penalización igual a cero indica que la solución cumple completamente con todas las restricciones fuertes, lo cual representa una condición necesaria para considerar la solución como válida dentro del contexto del problema.

4.4.3. Penalización por restricciones débiles

La penalización por restricciones débiles evalúa el grado en que una solución satisface criterios de calidad operativa, tales como la asignación basada en experiencia, preferencias de horario o distribución equilibrada del trabajo.

A diferencia de las restricciones fuertes, estas no invalidan la solución, pero permiten discriminar entre soluciones factibles, favoreciendo aquellas que optimizan el desempeño operativo.

4.4.4. Métrica de tiempo de ejecución

El tiempo de ejecución se considera como una métrica complementaria que permite evaluar el costo computacional del algoritmo. Esta métrica es particularmente relevante en aplicaciones prácticas, donde la generación de horarios debe realizarse en tiempos razonables.

El análisis del tiempo de ejecución permite identificar el impacto de las variables independientes sobre la eficiencia del algoritmo y establecer un balance entre calidad de solución y rendimiento.

4.4.5. Análisis de convergencia

Adicionalmente, se analiza el comportamiento del algoritmo en términos de convergencia, observando la evolución del valor de fitness a lo largo de las generaciones. Este análisis permite identificar patrones característicos del proceso evolutivo, tales como:

- Fase de exploración, donde se genera una alta diversidad de soluciones.
- Fase de explotación, donde el algoritmo se enfoca en mejorar soluciones prometedoras.
- Estabilización del fitness, indicando convergencia del algoritmo.

Este tipo de análisis es fundamental para comprender el comportamiento dinámico del Algoritmo Genético y validar su capacidad de encontrar soluciones de calidad.

4.4.6. Criterios de evaluación

Las métricas definidas permiten establecer los siguientes criterios de evaluación del algoritmo:

- Minimización de la penalización total (fitness).
- Eliminación de penalizaciones por restricciones fuertes.
- Reducción de penalizaciones por restricciones débiles.
- Obtención de tiempos de ejecución aceptables.
- Comportamiento de convergencia estable.

Estos criterios permiten realizar una evaluación integral del modelo, considerando tanto la calidad de las soluciones como la eficiencia del proceso.

4.5. Proceso experimental

El proceso experimental se llevó a cabo mediante una serie de etapas estructuradas que permiten evaluar el desempeño del Algoritmo Genético bajo distintas configuraciones:

1. **Definición del escenario de prueba**

   Se seleccionó un conjunto de datos representativos del entorno real, incluyendo información de empleados, disponibilidad, horarios del cine y requerimientos operativos.

2. **Configuración de parámetros del algoritmo**

   Se definieron diferentes combinaciones de parámetros, principalmente el tamaño de la población y el número de generaciones, con el fin de analizar su impacto en el desempeño del modelo.

3. **Ejecución del Algoritmo Genético**

   Para cada configuración, el algoritmo fue ejecutado múltiples veces, considerando su naturaleza aleatoria, con el objetivo de obtener resultados consistentes y representativos.

4. **Registro de resultados**

   Se almacenaron métricas relevantes de cada ejecución, incluyendo el valor del fitness, penalizaciones por restricciones fuertes y débiles, y tiempo de ejecución.

5. **Análisis de resultados**

   Los resultados obtenidos fueron analizados mediante técnicas estadísticas y visualización gráfica, permitiendo identificar patrones de comportamiento, convergencia del algoritmo y configuraciones óptimas.

4.6. Herramientas y entorno de desarrollo

El desarrollo e implementación del modelo propuesto se realizó en un entorno tecnológico basado en Ruby on Rails, utilizado para estructurar la lógica de negocio, la ejecución del Algoritmo Genético y la integración con la base de datos. Para el almacenamiento y consulta de la información se empleó una base de datos relacional MySQL, en la cual se gestionaron datos relacionados con empleados, disponibilidad, capacitación, horarios de operación, requerimientos de personal y resultados generados.

La interfaz gráfica fue desarrollada mediante tecnologías web específicamente con el framework ExtJS 7, integradas al sistema para permitir la configuración de parámetros, la ejecución del algoritmo, la visualización de horarios y la exportación de resultados en formatos XLSX y PDF.

Para el análisis experimental y la generación de gráficas se utilizó Python con librerías como pandas y matplotlib. Estas herramientas permitieron organizar los resultados obtenidos en las ejecuciones del algoritmo, calcular métricas de desempeño y generar gráficas de convergencia, mapas de calor, sensibilidad de parámetros y tiempo de ejecución.

El entorno de desarrollo utilizado permite la integración de todos los componentes del sistema en una solución funcional, escalable y adaptable, lo cual resulta fundamental para su aplicación en contextos reales de planificación de horarios laborales.

4.7. Población y muestra

La población de estudio está conformada por el conjunto total de empleados operativos del cine que pueden ser asignados a turnos en distintos puestos a lo largo de la semana, incluyendo personal de taquilla, acomodadores, operadores de proyección, limpieza y mantenimiento, supervisores de sala y personal de confitería. El tamaño de esta población varía según el cine, con un promedio estimado de entre 30 y 50 empleados en cines de tamaño medio.

La muestra corresponde a una semana representativa de operación, que incluye a todos los empleados disponibles durante dicho periodo y todos los días de la semana. Se consideran como criterios de inclusión aquellos empleados con disponibilidad activa, al menos un nivel mínimo de capacitación en algún puesto operativo y registro de su disponibilidad horaria. Se excluye al personal administrativo o gerencial, empleados con ausencias programadas y trabajadores sin registros válidos de disponibilidad o capacitación.

Cada empleado de la muestra cuenta con datos estructurados que incluyen identificador único, puestos y niveles de capacitación, disponibilidad diaria y horaria, preferencias personales y el historial de turnos asignados, con el fin de favorecer criterios de rotación y equidad. De forma complementaria, se recolectan datos del cine, tales como horarios de apertura y cierre, horarios operativos por puesto.

La información utilizada proviene de una base de datos actualizada, la cual es sometida a procesos de validación de integridad para garantizar la consistencia de la información. Para efectos de experimentación, los datos empleados son simulados, pero construidos a partir de patrones y estructuras observadas en datos reales del entorno operativo, lo que permite representar de manera fiel las condiciones del dominio de aplicación.

4.8. Validación del modelo

La validación del modelo se realizó mediante el análisis del cumplimiento de restricciones fuertes, la reducción de penalizaciones débiles, la estabilidad de los resultados en ejecuciones repetidas y la evaluación del tiempo de ejecución. Asimismo, se compararon distintas configuraciones del algoritmo permitiendo identificar configuraciones que optimizan el equilibrio entre calidad de solución y eficiencia computacional.

5. GENERACIÓN AUTOMÁTICA DE HORARIOS LABORALES

El problema de asignación de turnos laborales, descrito previamente en el planteamiento de la investigación, se formalizó en esta sección como un problema de optimización combinatoria altamente restringido.

5.1. Restricciones del modelo

Para la generación automática de horarios laborales, el modelo considera un conjunto de restricciones que representan las condiciones legales, operativas y organizacionales del problema. Estas restricciones se clasifican en dos grupos: restricciones fuertes y restricciones débiles.

Las restricciones fuertes corresponden a condiciones obligatorias que deben cumplirse para que un horario sea considerado factible. Su incumplimiento invalida la solución o genera una penalización alta dentro de la función objetivo. Por otro lado, las restricciones débiles representan criterios asociados al cumplimiento de políticas internas de la organización y a la mejora operativa del horario generado; estas pueden ser incumplidas parcialmente, pero su violación incrementa la penalización total de la solución. La Tabla 3 y Tabla 4 muestran el conjunto de Requerimientos funcionales asociados a las restricciones fuertes y restricciones débiles, respectivamente.

Tabla 3. *Requerimientos funcionales asociados a restricciones fuertes*

Código	Requerimiento funcional	Descripción	Valores o condición permitida
RF1	Un turno por día	El algoritmo debe evitar que un empleado sea asignado a más de un turno en el mismo día.	Máximo 1 turno por empleado por día.
RF2	Un puesto por intervalo de tiempo	El algoritmo debe impedir que un empleado sea asignado a dos puestos distintos en el mismo intervalo horario.	Máximo 1 puesto por empleado por intervalo de tiempo.
RF3	Duración legal de la jornada	El algoritmo debe validar que la duración del turno no exceda el límite legal según el tipo de jornada.	Diurna: máximo 8 h; mixta: máximo 7.5 h; nocturna: máximo 7 h.

RF4	Máximo de horas semanales	El algoritmo debe impedir que un empleado exceda el máximo legal de horas trabajadas por semana.	Máximo 48 horas semanales por empleado.
RF5	Día de descanso obligatorio	El algoritmo debe garantizar que el empleado tenga al menos un día de descanso por cada seis días trabajados.	Máximo 6 días consecutivos laborados.
RF6	Disponibilidad del empleado	El algoritmo debe asignar turnos únicamente dentro de los horarios en los que el empleado se encuentra disponible.	Disponible = 1; no disponible = 0.
RF7	Vacaciones o incidencias	El algoritmo debe evitar asignar turnos a empleados que presenten vacaciones o incidencias que impidan laborar.	Si existe incidencia, no se permite asignación.
RF8	Descanso entre turnos	El algoritmo debe validar que exista un periodo mínimo de descanso entre turnos consecutivos.	Mínimo 10 horas entre turnos.

Tabla 4. *Requerimientos funcionales asociados a restricciones débiles*

Código	Requerimiento funcional	Descripción	Valores o condición considerada
RD1	Prioridad por experiencia o capacitación	El algoritmo debe favorecer la asignación de empleados con mayor nivel de capacitación para cada puesto.	Mayor nivel de capacitación implica menor penalización.
RD2	Distribución equilibrada de carga	El algoritmo debe procurar una distribución equilibrada de horas entre los empleados disponibles.	Se penalizan desviaciones respecto al promedio de horas asignadas.
RD3	Cobertura eficiente de demanda	El algoritmo debe procurar que el número de empleados asignados se aproxime al personal requerido por puesto y horario.	Se penaliza la diferencia entre personal asignado y requerido.
RD4	Respeto a horarios operativos	El algoritmo debe favorecer asignaciones que se encuentren dentro de los horarios óptimos del cine y de cada puesto.	Las asignaciones dentro del horario operativo generan menor penalización.

A partir de esta clasificación, el modelo transforma los requerimientos funcionales en criterios cuantificables dentro de la función objetivo. De este modo, cada horario candidato se evalúa tanto por su factibilidad como por su nivel de ajuste a las condiciones operativas del cine. Las penalizaciones de cada restricción permiten comparar las soluciones generadas por el

Algoritmo Genético y priorizar aquellas con menor incumplimiento acumulado, manteniendo un equilibrio entre la viabilidad de la solución y el cumplimiento de las políticas de la organización. Con base en ello, a continuación, se presenta la formulación matemática del modelo propuesto.

5.2. Modelo matemático (Función objetivo)

Sea:

- $E = \{e_1, e_2, \dots, e_{|E|}\}$ el conjunto de empleados, donde cada elemento $e_i \in E$ representa a un empleado individual, para todo $i = 1,2, \dots, | E |$.
- $D = \{d_1, d_2, \dots, d_{|D|}\}$ el conjunto de días del horizonte de planificación, donde cada elemento $d_k \in D$ representa un día específico, para todo $k = 1,2, \dots, | D |$.
- $P = \{p_1, p_2, \dots, p_{|P|}\}$ el conjunto de puestos de trabajo, donde cada elemento $p_l \in P$ representa un puesto específico, para todo $l = 1,2, \dots, | P |$.
- $T = \{t_1, t_2, \dots, t_{|T|}\}$ el conjunto de intervalos de tiempo considerados dentro de la jornada operativa, donde cada elemento $t_m \in T$ representa un intervalo de tiempo específico, para todo $m = 1,2, \dots, | T |$.

Se define una variable de decisión:

$$x_{e,d,p,t} = \begin{cases} 1 & \text{si el empleado } e \text{ es asignado al puesto } p \text{ en el día } d \text{ en el intervalo } t \\ 0 & \text{en otro caso} \end{cases}$$

Una solución (X) representa una asignación completa de horarios laborales para el conjunto de empleados durante el horizonte de planificación. El objetivo del modelo es encontrar la solución (X) que minimice la siguiente función objetivo.

$$\min F(X) = \lambda_H \cdot \sum_{i=1}^{n} R\, F_i(X) + \lambda_S \cdot \sum_{j=1}^{m} R\, D_j(X) \qquad \text{(Ecuación 1)}$$

Donde:

- Las funciones $RF_i(X)$ representan penalizaciones asociadas a restricciones fuertes, las cuales determinan la factibilidad legal y laboral del horario generado.
- $RD_j(X)$ corresponden a penalizaciones relacionadas con las restricciones débiles orientadas al cumplimiento de políticas internas de la organización y a la calidad operativa del horario generado.
- λ_H y λ_S son coeficientes de ponderación asociados a las penalizaciones por restricciones fuertes y débiles, respectivamente. En particular, λ_H se define con un valor significativamente mayor que λ_S, es decir:

  $$\lambda_H \gg \lambda_S$$

  lo que garantiza que cualquier violación de las restricciones fuertes tenga un impacto dominante en la función objetivo, priorizando su cumplimiento sobre la optimización de restricciones débiles.

5.3. Representación del cromosoma

Para la implementación del Algoritmo Genético, es necesario definir una estructura de representación que permita codificar de manera eficiente las soluciones del problema de asignación de turnos. En este contexto, cada solución es representada como un cromosoma, el cual contiene la información completa de la asignación de turnos para todos los empleados durante el horizonte de planificación.

El cromosoma se define como un conjunto de genes, donde cada gen representa una asignación específica de un empleado a un turno en un día determinado. Formalmente, un cromosoma puede expresarse como:

$$C = \{g_1, g_2, \ldots, g_n\}$$

donde cada gen $g_i$contiene información estructurada de la forma:

$$g_i = (e, d, p, t_{inicio}, t_{fin})$$

donde:

e: empleado asignado.

d: día de trabajo.

p: puesto o estación.

t_{inicio}, t_{fin}: intervalo de tiempo del turno.

5.3.1. Estructura del cromosoma

La construcción del cromosoma cumple con las siguientes características:

- Se respete la granularidad temporal definida (por ejemplo, bloques de 15 minutos).
- Se considere la disponibilidad del empleado.
- Se permita la evaluación directa mediante la función de fitness.
- El cromosoma tiene una longitud fija, determinada por el horizonte de planificación, aunque el número de asignaciones activas puede variar según la demanda operativa.

5.4. Interpretación de la solución

Un cromosoma puede interpretarse como un conjunto de asignaciones que, en conjunto, forman el horario completo de operación; la Tabla 5 muestra un ejemplo parcial de horario.

Tabla 5. *Ejemplo horario*

Empleado	Día	Puesto	Inicio	Fin
e1	Lunes	Taquilla	10:00	16:00

e2	Lunes	Dulcería	12:00	18:00

Esta estructura permite representar de forma clara y directa las decisiones del algoritmo, facilitando tanto su evaluación como su posterior implementación en el sistema.

La representación propuesta presenta las siguientes ventajas:

- **Flexibilidad:** Permite modelar distintos tipos de turnos y configuraciones operativas
- **Escalabilidad:** Se adapta a diferentes tamaños de problema (más empleados, días o puestos)
- **Compatibilidad:** Se integra fácilmente con los operadores genéticos (cruza, mutación)
- **Evaluación eficiente:** Facilita el cálculo del fitness mediante la verificación directa de restricciones

5.5. Diseño del Algoritmo Genético

El Algoritmo Genético propuesto se diseñó como una estrategia metaheurística para resolver el problema de asignación de turnos laborales, alineándose con el proceso experimental descrito en la metodología. Gracias a este diseño, es posible analizar de manera ordenada cómo influyen los distintos parámetros del algoritmo en la calidad de los horarios generados, además de recorrer de forma eficiente las posibles soluciones para encontrar las mejores opciones según la función objetivo. Para lograrlo, se utilizan operadores como la selección por torneo, la cruza uniforme, la mutación adaptativa y el elitismo. Estos mecanismos se ajustaron al contexto del problema, buscando siempre generar soluciones que cumplan con las condiciones establecidas y que mejoren progresivamente.

Para el manejo de soluciones no factibles, el modelo utiliza un esquema de penalizaciones dentro de la función objetivo. Las soluciones que incumplen restricciones fuertes reciben una penalización mayor, con el propósito de orientar la evolución hacia horarios factibles. Una vez reducidas estas violaciones, las restricciones débiles permiten diferenciar la calidad operativa de las soluciones generadas. De esta manera, el algoritmo conserva capacidad de exploración, pero favorece progresivamente a los individuos con menor incumplimiento y mejor desempeño.

5.5.1. Estructura general del algoritmo

El proceso evolutivo sigue un esquema iterativo compuesto por las siguientes etapas:

- Inicialización de la población.
- Evaluación del fitness.
- Selección de individuos.
- Aplicación de cruza (crossover).
- Aplicación de mutación.
- Reemplazo y elitismo.
- Criterio de paro.

Este ciclo se repite durante un número determinado de generaciones, buscando mejorar progresivamente la calidad de las soluciones.

5.5.2. Inicialización de la población

La población inicial se genera mediante un proceso de inicialización aleatoria controlada, en el cual se construyen soluciones que exploran el espacio de búsqueda, incorporando al mismo tiempo restricciones básicas del problema, como los horarios operativos.

5.5.3. Evaluación del fitness

Cada individuo es evaluado mediante la función objetivo (Ecuación 1). Se prioriza el cumplimiento de restricciones fuertes, penalizando severamente las soluciones no factibles, lo que guía el proceso evolutivo hacia regiones válidas del espacio de búsqueda.

5.5.4. Selección

Para la selección se emplea el método de selección por torneo**.** Este procedimiento consiste en elegir aleatoriamente un subconjunto de (k) individuos de la población y comparar sus valores de aptitud. El individuo con mejor desempeño dentro del subconjunto es seleccionado como padre para participar en el proceso de reproducción.

En esta investigación se utiliza un tamaño de torneo ($k = 3$), lo que permite mantener un equilibrio entre presión selectiva y diversidad poblacional. Una mayor presión selectiva favorece la explotación de las mejores soluciones encontradas, mientras que la selección aleatoria de los participantes del torneo contribuye a conservar diversidad dentro de la población.

El procedimiento general de selección por torneo se describe de la siguiente manera:

1. Se seleccionan aleatoriamente (k) individuos de la población actual.
2. Se evalúa el valor de aptitud de cada individuo seleccionado.
3. Se elige como ganador del torneo el individuo con menor valor de penalización.
4. El individuo ganador se incorpora al conjunto de padres.
5. El proceso se repite hasta obtener el número requerido de padres para aplicar los operadores de cruzamiento y mutación.

Dado que el problema se formula como una minimización de penalizaciones, el individuo con menor valor de la función objetivo ($F(X)$) es considerado el más apto dentro de cada torneo.

5.5.5. Cruza (Crossover)

La cruza se aplica después de seleccionar dos individuos padres mediante la selección por torneo. En el modelo propuesto se utiliza cruzamiento uniforme, donde cada gen del cromosoma del descendiente puede heredarse de cualquiera de los dos padres.

Durante el proceso de cruzamiento, se recorre cada posición del cromosoma y se decide de forma aleatoria si el gen correspondiente se toma del primer padre o del segundo. De esta manera, el descendiente combina asignaciones de ambos horarios candidatos.

El cruzamiento se realiza de acuerdo con una probabilidad (p_c). Si el valor aleatorio generado es menor o igual a (p_c), se realiza la recombinación; en caso contrario, los descendientes conservan la estructura de los padres seleccionados. El análisis experimental permitió identificar $p_c = 0.80$ como el valor más adecuado.

Este operador favorece la exploración del espacio de soluciones al generar nuevas combinaciones de horarios, manteniendo al mismo tiempo información útil resultante de soluciones previamente evaluadas.

5.5.6. Mutación

El operador de mutación introduce variabilidad en la población mediante modificaciones aleatorias en los cromosomas. Esto consiste en seleccionar aleatoriamente uno o más genes del cromosoma alterando alguno de sus componentes, tales como el empleado asignado, ajuste del horario de inicio o fin, reasignación del puesto o modificación del día de trabajo, dependiendo de las restricciones de disponibilidad, capacitación y horarios operativos.

El procedimiento general de mutación se describe de la siguiente manera:

1. Se recorre cada individuo descendiente generado durante el proceso evolutivo.

2. Para cada gen del cromosoma, se evalúa si debe aplicarse mutación de acuerdo con la probabilidad de mutación (μ).
3. Si la mutación se aplica, se selecciona aleatoriamente el componente del gen que será modificado.
4. El nuevo valor se genera dentro del dominio permitido, considerando empleados disponibles, puestos válidos, horarios operativos e intervalos de tiempo definidos.
5. El individuo mutado se evalúa nuevamente mediante la función objetivo (Ecuación 1), incorporando las penalizaciones correspondientes en caso de incumplimiento de restricciones.

En esta investigación se utiliza una tasa de mutación base ($\mu = 0.12$), con un esquema adaptativo en el rango ($[0.04, 0.18]$). Esto permite aumentar la mutación cuando el algoritmo presenta señales de estancamiento y reducirla cuando la población muestra mejoras constantes.

5.5.7. Elitismo

Se incorpora un mecanismo de elitismo que garantiza la preservación de los mejores individuos de cada generación. Esto evita la pérdida de soluciones de alta calidad y acelera el proceso de convergencia del algoritmo.

En el modelo propuesto, al finalizar cada generación se ordenan los individuos de la población de acuerdo con el valor de la función objetivo. Debido a que el problema se formula como una minimización, los individuos con menor valor de penalización se consideran los mejores. De estos se seleccionan 5 individuos, los cuales se conservan directamente en la siguiente población. Este mecanismo permite preservar las soluciones de mayor calidad sin impedir que el resto de la población continúe explorando nuevas alternativas.

La incorporación del elitismo favorece la estabilidad del algoritmo, ya que garantiza que la mejor solución encontrada hasta una generación determinada no se degrade en generaciones posteriores. De esta manera, el proceso evolutivo mantiene la información de mayor calidad y continúa explorando nuevas soluciones a partir de los operadores genéticos.

5.5.8. Criterio de paro

El algoritmo finaliza cuando se cumple alguna de las siguientes condiciones:

- Se alcanza el número máximo de generaciones.
- No se observa mejora significativa en el fitness durante varias generaciones.

5.5.9. Parámetros del algoritmo

Los principales parámetros considerados son:

- Tamaño de la población (P).
- Número de generaciones (G).
- Tasa de cruza.
- Tasa de mutación.
- Tamaño del torneo.
- Número de individuos élite.

Estos parámetros fueron ajustados experimentalmente para encontrar un equilibrio entre calidad de solución y tiempo de ejecución.

5.5.10. Procedimiento general del algoritmo

```
Entrada:

    E  – conjunto de empleados
```

```
    D  - conjunto de días del horizonte de planificación

    P  - conjunto de posiciones

    R  - conjunto de restricciones fuertes y débiles

    N  - tamaño de población

    G  - número máximo de generaciones

    pc - probabilidad de cruza

    pm - probabilidad de mutación

Salida:

    X* - mejor horario encontrado


1:  Construir los datos de entrada del problema

2:  Generar aleatoriamente una población inicial Población(0) con N individuos

3:  Evaluar la aptitud de cada individuo de Población(0)

4:  X* - mejor individuo de Población(0)

5:  Para g = 1 hasta G hacer

6:     Seleccionar individuos padres desde Población(g-1)

7:     Aplicar operador de cruza a los padres con probabilidad pc

8:      Aplicar operador de mutación a los descendientes con probabilidad pm

9:    Evaluar la aptitud de cada descendiente

10:     Construir Población(g) mediante reemplazo y conservación de élite

11:    Si el mejor individuo de Población(g) mejora a X* entonces

12:         X* - mejor individuo de Población(g)

13:    Fin si

14: Fin para

15: Regresar X*
```

Evaluación de un individuo

```
Entrada:

    X   - individuo (horario candidato)

    WH  - pesos de restricciones fuertes

    WS  - pesos de restricciones débiles

    λH  - ponderación global de restricciones fuertes

    λS  - ponderación global de restricciones débiles

Salida:

    F(X) - valor de aptitud del individuo


1:  Calcular penalización por restricciones fuertes:

        Hard(X) = RF1(X) + RF2(X) +...+ RF8(X)

2:  Calcular penalización por restricciones débiles:

        Soft(X) = RD1(X) + RD2(X) +...+ RD4(X)

3:  Calcular el valor total de aptitud del individuo

        F(X) = λH·Hard(X) + λS·Soft(X)

4:  Regresar F(X)
```

5.6. Arquitectura del sistema

La propuesta de solución, no se limita únicamente al diseño e implementación de un modelo basado en Algoritmos Genéticos, sino que también se considera la implementación de una solución software que integra el Algoritmo Genético con una plataforma de gestión operativa. Esta arquitectura sigue el patrón Modelo-Vista-Controlador (MVC), lo cual permite una adecuada separación de responsabilidades.

5.6.1. Estructura general del sistema

La arquitectura del sistema se basa en un modelo de aplicación web de múltiples capas, compuesto por:

- Capa de presentación (frontend).
- Capa de lógica de negocio (backend).
- Capa de datos (base de datos).

Esta separación permite una mejor organización del sistema, facilitando su mantenimiento, escalabilidad y evolución.

5.6.2. Capa de presentación

La interfaz de usuario fue desarrollada utilizando tecnologías web, permitiendo la interacción con el sistema de manera intuitiva. A través de esta capa, los usuarios pueden:

- Ejecutar el Algoritmo Genético.
- Configurar parámetros del modelo.
- Visualizar los horarios generados.
- Analizar resultados de manera gráfica.
- Realizar ajustes finales agregando, modificando o eliminando horarios.

Esta capa actúa como punto de entrada al sistema y facilita la interpretación de las soluciones generadas.

5.6.3. Capa de lógica de negocio

El núcleo del sistema se implementó utilizando el framework Ruby on Rails, en donde se integró el Algoritmo Genético como parte de los servicios de negocio.

Dentro de esta capa se encuentran los componentes principales:

Módulo de generación de datos: Encargado de construir el escenario de entrada para el algoritmo, a partir de información real del sistema.

Motor del Algoritmo Genético: Implementa la lógica evolutiva, incluyendo inicialización, evaluación, selección, cruza y mutación.

Módulo de evaluación (fitness): Calcula la calidad de las soluciones con base en la función objetivo definida.

Módulo de persistencia de resultados: Guarda los horarios generados en la base de datos para su posterior consulta.

Esta capa constituye el núcleo funcional del sistema, donde se ejecuta la lógica de optimización.

5.6.4. Capa de datos

El sistema utiliza una base de datos relacional que almacena información relevante para el problema, incluyendo:

1. Empleados.
2. Disponibilidad.
3. Niveles de capacitación.
4. Horarios del cine.
5. Requerimientos de personal.
6. Históricos de programación.

Esta información es utilizada como entrada para el algoritmo y permite generar soluciones alineadas con el contexto real de operación.

5.6.5. Flujo de ejecución del sistema

El proceso de generación de horarios sigue el siguiente flujo:

1. El usuario solicita la generación del horario desde la interfaz.
2. El sistema construye el escenario de entrada.
3. Se ejecuta el Algoritmo Genético.
4. Se evalúan las soluciones generadas.
5. Se selecciona la mejor solución.
6. Se almacenan los resultados en la base de datos.
7. Los resultados se visualizan en la interfaz gráfica.

Este flujo permite integrar de manera eficiente todos los componentes del sistema en un proceso automatizado.

5.6.6. Ventajas de la arquitectura propuesta

- **Integración real:** El modelo puede ser utilizado en un entorno operativo.
- **Escalabilidad:** Permite adaptarse a diferentes tamaños de problema.
- **Modularidad:** Facilita la mejora o sustitución de componentes.
- **Usabilidad:** Permite a usuarios no técnicos interactuar con el sistema.

La arquitectura propuesta permite cerrar la brecha entre modelos teóricos de optimización y su aplicación práctica en entornos empresariales reales.

5.7. Interfaz gráfica de usuario

Con el objetivo de facilitar la interacción con el sistema y permitir la visualización de los resultados generados por el Algoritmo Genético, se desarrolló una interfaz gráfica de usuario

basada en tecnologías web. Esta interfaz constituye el punto de acceso principal para la ejecución del modelo y la interpretación de las soluciones obtenidas.

5.8. Diseño de la interfaz

El diseño permite representar de manera estructurada la información relacionada con la asignación de turnos, facilitando su comprensión y análisis. Se emplea una organización visual basada en componentes tipo panel y tablas dinámicas, lo que permite presentar información de forma ordenada y adaptable a diferentes volúmenes de datos.

5.8.1. Funcionalidades principales

La interfaz gráfica permite:

- **Ejecución del Algoritmo Genético:** El usuario puede iniciar el proceso de generación de horarios desde la interfaz, activando la ejecución del modelo propuesto.
- **Módulo de ajuste de sucursal:** Se incorpora un módulo que permite configurar parámetros operativos específicos de cada sucursal, tales como horarios de operación, requerimientos de personal, reglas de negocio y condiciones particulares del entorno. Este módulo permite adaptar el comportamiento del algoritmo a diferentes contextos operativos.
- **Módulo de ajuste de empleados:** Se dispone de un módulo para la gestión de información individual de los empleados, incluyendo disponibilidad horaria, niveles de capacitación y restricciones específicas. Esto permite personalizar las condiciones de asignación y mejorar la calidad de las soluciones generadas.
- **Visualización de resultados:** Los horarios generados se presentan en una tabla, permitiendo identificar fácilmente la asignación de empleados por día, horario y puesto.

- **Exportación de resultados:** El sistema permite la descarga de los horarios generados en formatos Excel (XLSX) y PDF, facilitando su distribución, análisis externo y uso operativo dentro de la organización. Esta funcionalidad permite integrar los resultados del modelo con procesos administrativos existentes.

5.9. Representación de la información

La información se presenta mediante estructuras visuales que permiten:

- Identificar rápidamente la asignación de turnos.
- Detectar posibles inconsistencias o áreas de mejora.
- Comparar resultados entre diferentes ejecuciones.

El uso de tablas dinámicas y componentes visuales facilita la interpretación de grandes volúmenes de datos.

5.10. Interacción con el sistema

La interfaz gráfica se integra con la lógica del sistema, lo que permite una comunicación directa con el *backend* para facilitar el envío de solicitudes destinadas a la ejecución del algoritmo, así como la recepción de resultados en tiempo real y la actualización dinámica de los horarios generados. De este modo, la inclusión de la interfaz gráfica permite conectar el modelo matemático con su aplicación práctica, ya que no solo cumple una función de visualización, sino que forma parte integral del sistema propuesto, consolidándolo como una herramienta funcional y aplicable en un entorno operativo real.

6. Resultados y análisis

El presente capítulo describe los resultados obtenidos a partir de la experimentación del Algoritmo Genético propuesto. Las pruebas se diseñaron con el objetivo de analizar su comportamiento bajo distintas configuraciones, principalmente variando el tamaño de la población (P) y el número de generaciones (G), así como observar su impacto en la calidad de las soluciones y en el tiempo de ejecución.

Para la evaluación, se definió un esquema de experimentación basado en múltiples combinaciones de parámetros. En particular, se consideraron valores de población en el rango de 40 a 600 individuos, y generaciones desde 80 hasta 500. Posteriormente, una vez identificada la configuración base con mejor desempeño, se realizó un análisis sobre la probabilidad de cruzamiento (Pc), considerando valores entre 0.50 y 0.90 con incrementos de 0.05, lo cual permitió valorar el efecto del operador de cruza sobre la calidad de las soluciones generadas.

Cada configuración fue ejecutada 30 veces de manera independiente, con el fin de reducir el efecto de la aleatoriedad del algoritmo y obtener resultados más representativos. En cada ejecución se registraron métricas como el valor de la función objetivo, la penalización por restricciones fuertes, la penalización por restricciones débiles y el tiempo de ejecución.

Los experimentos fueron realizados en un equipo de cómputo con procesador Intel Core i5 de 13.ª generación, 16 GB de memoria RAM, sistema operativo Windows 11 y un entorno de ejecución basado en Ruby on Rails.

En todos los casos se registraron ejecuciones exitosas, sin presencia de errores, lo cual indica estabilidad en la implementación.

6.1. Métricas de evaluación

La evaluación de las soluciones se realizó a partir de la función objetivo (Ecuación 1) definida en el modelo, la cual integra penalizaciones asociadas a restricciones fuertes y débiles. En este sentido, se analizaron tres componentes principales:

Por un lado, el valor total de la función objetivo, que representa la calidad global de la solución; entre menor sea este valor, mejor es el resultado obtenido. Adicionalmente, se consideró el componente de restricciones fuertes, el cual refleja el grado de incumplimiento y tiene un peso significativamente mayor dentro de la función objetivo. Finalmente, se evaluó el componente de restricciones débiles, relacionado con aspectos como preferencias y optimización operativa. Aunque su impacto es menor, permite refinar la calidad de los horarios generados.

También se registró el tiempo promedio de ejecución por configuración, lo cual resulta relevante para evaluar la viabilidad práctica del algoritmo.

La Tabla 6 presenta un resumen estadístico de configuraciones representativas del Algoritmo Genético, seleccionadas a partir de los límites inferiores, valores intermedios y configuraciones de mejor desempeño observadas en los experimentos. Se incluyen métricas como el promedio del fitness, desviación estándar, variabilidad, penalización promedio por restricciones fuertes y débiles, mejor caso y tiempo promedio de ejecución.

Tabla 6. *Estadísticas descriptivas de configuraciones representativas del Algoritmo Genético*

Configuración G×P	μ fitness	σ	CV(%)	μ RF	μ RD	Mejor caso	Tiempo (s)
80×40	1,152,952	N/D	N/D	850,667	302,286	656,025	10.4
80×300	281,523	N/D	N/D	32,000	249,523	228,706	74.0
100×40	1,045,711	N/D	N/D	748,000	297,711	599,831	**9.4**
100×300	275,913	N/D	N/D	29,333	246,579	225,406	150.6
320×40	597,171	N/D	N/D	326,667	270,504	316,894	23.8
320×200	233,324	30,684	13.15%	10,667	222,657	205,388	316.7
320×300	212,119	18,179	8.57%	**0**	212,119	168,350	596.5
400×40	537,031	122,967	22.90%	274,667	262,364	324,206	73.6
400×200	212,366	18,560	8.74%	**0**	212,366	179,688	208.0
400×300	204,179	17,728	8.68%	**0**	204,179	**139,319**	277.2
450×300	199,265	**12,964**	**6.51%**	**0**	199,265	176,000	274.7

500×40	477,775	147,710	30.92%	222,667	255,109	233,181	49.9
500×250	203,268	16,735	8.23%	**0**	203,268	161,569	224.0
500×300	**197,748**	18,470	9.34%	**0**	**197,748**	161,531	220.4

Nota: *G representa el número de generaciones y P el tamaño de población. La columna Var. (%) corresponde al coeficiente de variación, utilizado como medida relativa de variabilidad. En las configuraciones donde no se cuenta con desviación estándar, se indica como N/D. Es importante señalar que durante las pruebas de ejecución no se registró el peor caso y en algunas configuraciones, no se conservó la desviación estándar. Por esta razón, dichos campos se reportan como N/D. Sin embargo, los datos disponibles permiten analizar el comportamiento general del algoritmo, ya que se cuenta con el promedio del fitness, el mejor caso, las penalizaciones por restricciones fuertes y débiles, y el tiempo promedio de ejecución.*

La configuración 500×300 se considera la mejor en términos de desempeño promedio, ya que obtuvo el menor valor medio de fitness (197,748). Por otro lado, la configuración 450×300 mostró la mayor estabilidad estadística (σ = 12,964 y CV = 6.51 %), mientras que 400×300 produjo la mejor solución individual (139,319). En nuestro caso, se selecciona la configuración 500x300.

6.2. Impacto del tamaño de la población

Uno de los comportamientos más notorios se observa al analizar el efecto del tamaño de la población. A medida que el valor de p aumenta, la calidad de las soluciones mejora de forma consistente, especialmente en la reducción de las penalizaciones por restricciones fuertes.

Por ejemplo, para una configuración con g = 80, al incrementar la población de 40 a 200 individuos, el valor promedio de la función objetivo disminuye de aproximadamente 1,152,952 a 338,603, lo que representa una mejora significativa del 70%. Esta reducción está directamente relacionada con la disminución del componente de restricciones fuertes, el cual pasa de valores cercanos a 850,000 a menos de 80,000.

Sin embargo, este comportamiento no es completamente lineal. A partir de ciertos tamaños de población (alrededor de 200-240 individuos), la mejora comienza a estabilizarse, mostrando rendimientos decrecientes. En otras palabras, incrementar la población más allá de este punto genera mejoras marginales en la calidad, pero con un costo computacional considerablemente mayor.

Este patrón se repite en distintas configuraciones de generaciones, lo cual sugiere que existe un rango óptimo de población donde el algoritmo logra un buen balance entre exploración del espacio de soluciones y eficiencia computacional.

6.3. Impacto del número de generaciones

El número de generaciones también influye de manera importante en el desempeño del algoritmo. Al comparar configuraciones con diferentes valores de G, se observa que incrementar las generaciones permite una mejora progresiva en la calidad de las soluciones, particularmente en la reducción de restricciones fuertes.

Por ejemplo, manteniendo una población alta ($P = 300$), el valor promedio de la función objetivo disminuye de aproximadamente 281,522 para $G = 80$ a cerca de 217,117 para $G = 200$, y continúa reduciéndose en generaciones mayores.

No obstante, al igual que en el caso de la población, el beneficio de incrementar el número de generaciones presenta una tendencia a estabilizarse. A partir de cierto punto las mejoras adicionales son menos pronunciadas, lo que indica que el algoritmo ya ha alcanzado una zona cercana a la convergencia.

6.4. Análisis global del desempeño del algoritmo

Con el objetivo de analizar de manera integral el comportamiento del Algoritmo Genético, se construyó un mapa de calor que relaciona el tamaño de la población (P) y el número de generaciones (G) con el valor promedio de la función objetivo. Esta representación permite

visualizar de forma global el desempeño del algoritmo en todo el espacio de búsqueda evaluado, identificando regiones de alto y bajo rendimiento (Figura 1).

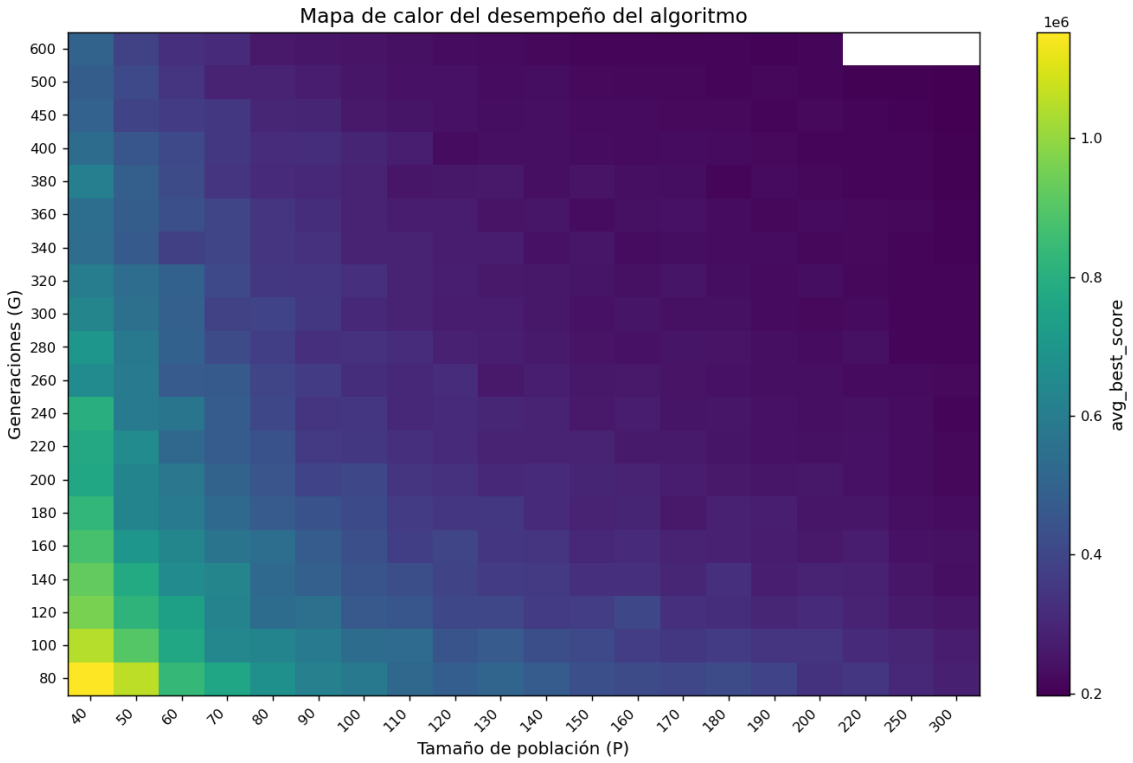

Figura 1. *Mapa de calor del desempeño del algoritmo.*

A partir del mapa de calor (Figura 1), se observa una tendencia clara a que el valor promedio de la función objetivo disminuya conforme aumentan el tamaño de la población y el número de generaciones. En contraste, las configuraciones con valores bajos de estos parámetros presentan un desempeño significativamente inferior, reflejado en altos niveles de penalización. Asimismo, se identifica que, a partir de cierto punto, las mejoras tienden a estabilizarse, lo cual indica la existencia de una zona de convergencia donde el algoritmo alcanza soluciones cercanas al óptimo. Este comportamiento confirma que el incremento simultáneo de población y generaciones favorece un equilibrio adecuado entre exploración y explotación del espacio de soluciones.

6.5. Evidencia de convergencia

El comportamiento observado en las distintas configuraciones permite identificar una tendencia de convergencia a encontrar la mejor solución. En las primeras generaciones, el algoritmo logra reducciones más pronunciadas del valor de la función objetivo, lo cual indica una exploración en diversas regiones del espacio de búsqueda y el descarte progresivo de soluciones con altos niveles de penalización. Posteriormente, el proceso evolutivo se enfoca en ajustes más finos, lo cual es consistente con el comportamiento de los algoritmos evolutivos, donde el proceso de búsqueda tiende a concentrarse en regiones cercanas a las soluciones previamente encontradas (Crepinsek et al., 2013).

Además, la consistencia en los resultados (30 ejecuciones exitosas por configuración) sugiere que el algoritmo converge de manera estable hacia soluciones similares, lo cual es deseable en este tipo de problemas.

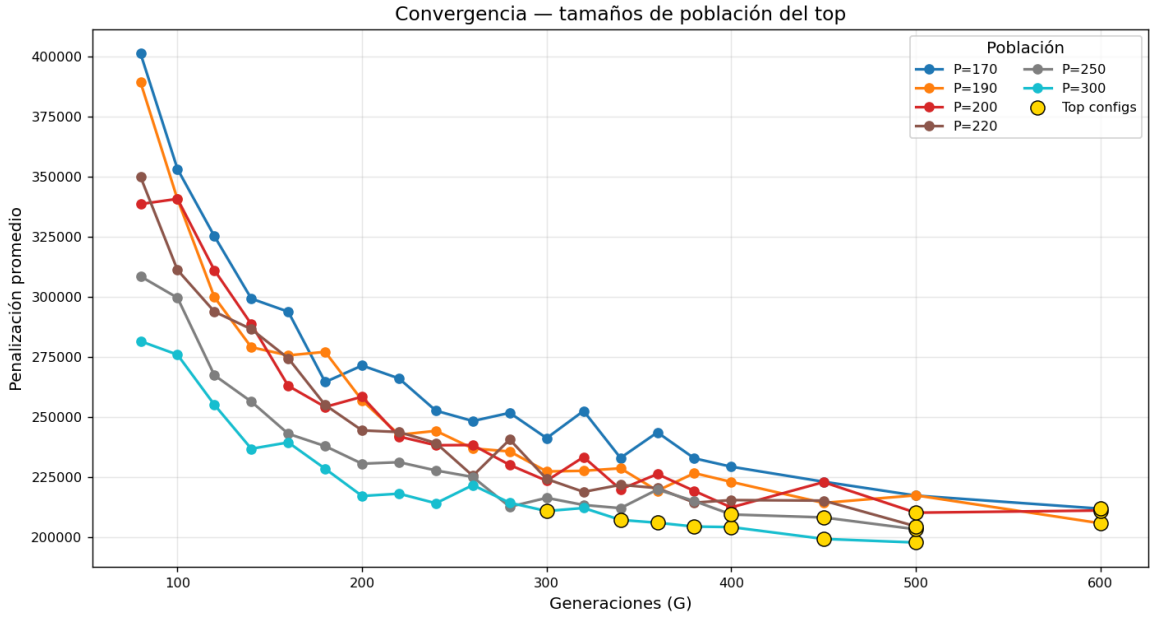

Figura 2. *Gráfica de convergencia.*

La Figura 2 muestra el comportamiento de convergencia del algoritmo para distintas configuraciones de población. Se observa una disminución progresiva de la penalización promedio conforme avanzan las generaciones, lo cual evidencia el proceso evolutivo del

algoritmo. Durante las primeras generaciones se presentan mejoras significativas, mientras que en etapas posteriores el algoritmo entra en una fase de refinamiento, donde las mejoras son más graduales.

6.6. Relación entre calidad y tiempo de ejecución

Uno de los aspectos más relevantes del análisis es la relación entre la calidad de las soluciones y el tiempo de ejecución. A medida que se incrementan los valores de *P* y *G*, el tiempo de procesamiento crece.

Por ejemplo, configuraciones con poblaciones pequeñas presentan tiempos promedios menores a 15 segundos, mientras que configuraciones con poblaciones superiores a 250 individuos pueden superar los 90 segundos, e incluso alcanzar varios minutos en algunos casos.

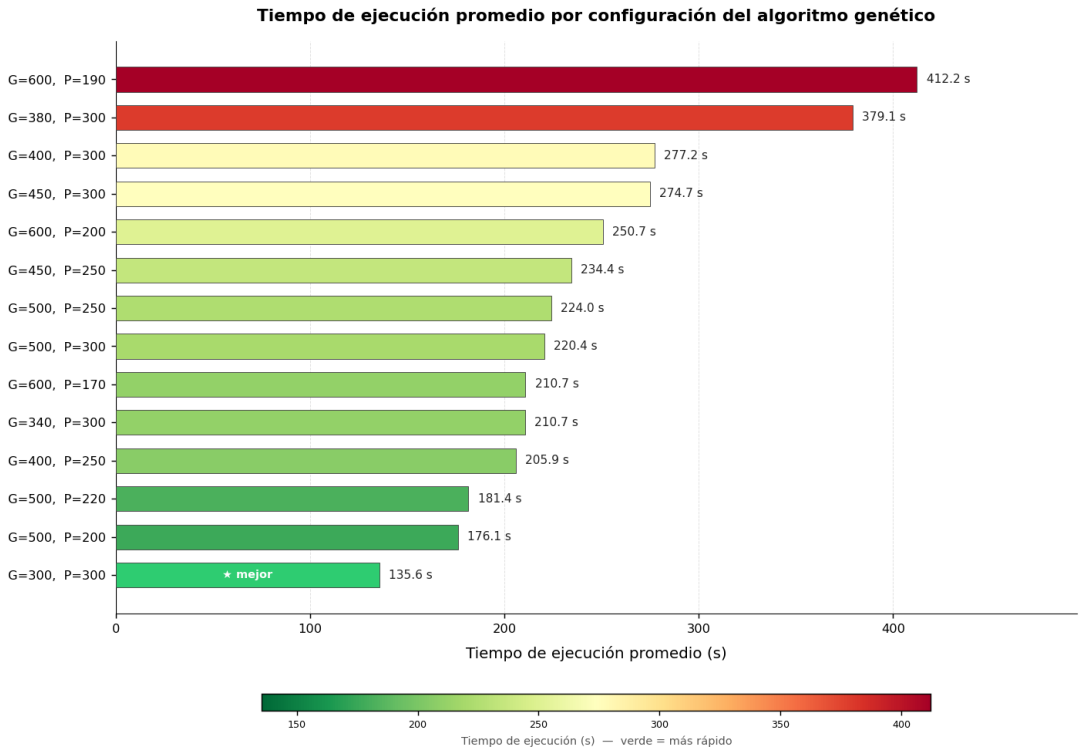

Figura 3. *Tiempo de ejecución promedio por configuración del algoritmo.*

Como se muestra en la Figura 3, las configuraciones con mejores resultados tienden a presentar mayores tiempos de ejecución, evidenciando un incremento considerable en el costo computacional conforme aumentan los parámetros del algoritmo. Existe un equilibrio importante: al aumentar el tamaño de la configuración se obtienen mejores resultados, pero también se incrementa el tiempo de cómputo. En el entorno real, donde los horarios deben generarse rápidamente, este factor cobra gran importancia.

6.7. Análisis de factibilidad

Un aspecto fundamental en la evaluación del algoritmo es el cumplimiento de las restricciones fuertes. Los resultados muestran una reducción significativa de este componente conforme se incrementan los parámetros del algoritmo.

En las configuraciones con mejor desempeño, las penalizaciones relacionadas con restricciones fuertes son mínimas o prácticamente nulas, lo que indica que el algoritmo logra aproximarse a soluciones factibles. No obstante, el valor restante de la función objetivo se asocia principalmente a restricciones débiles.

6.8. Parámetros óptimos de población y generaciones

La Figura 4 presenta las quince configuraciones con menor penalización promedio obtenidas durante la experimentación del Algoritmo Genético.

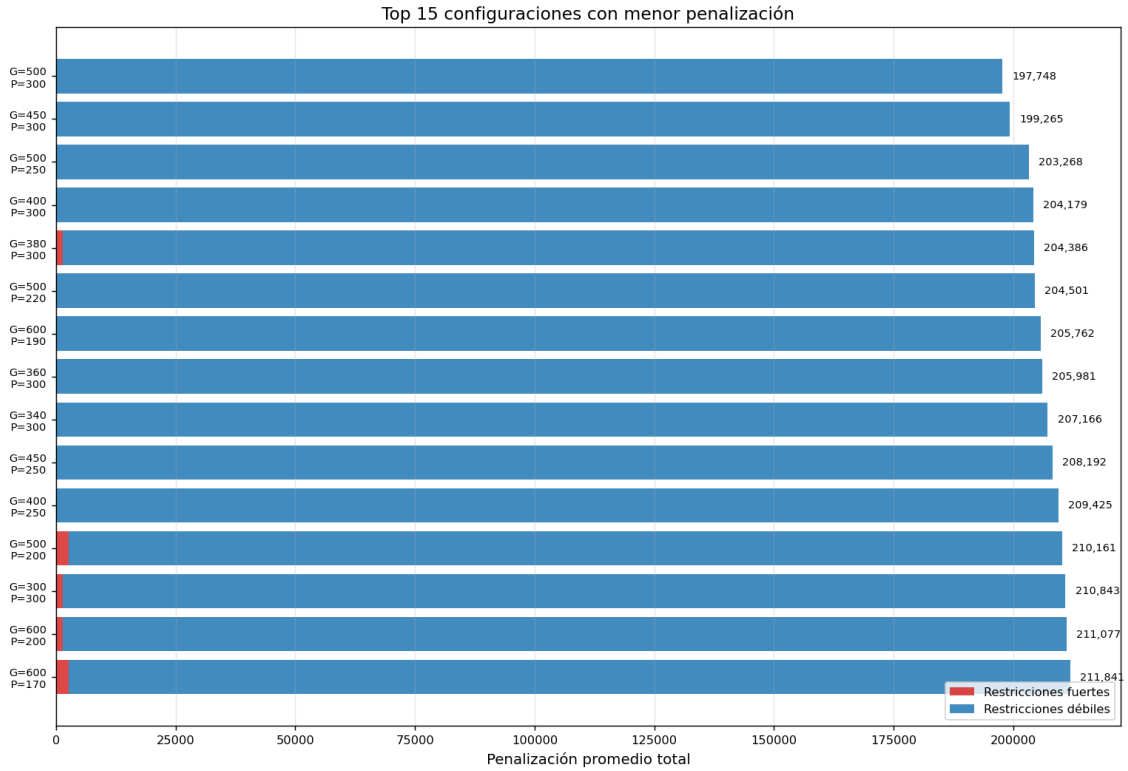

Figura 4. *Gráfica de mejores configuraciones encontradas.*

En general, se observa que las mejores soluciones se concentran en configuraciones con tamaños de población y números de generaciones relativamente altos, principalmente en combinaciones donde la población se encuentra entre 250 y 300 individuos, y el número de generaciones entre 400 y 500. La mejor configuración identificada corresponde a (G=500) y (P=300), con una penalización promedio de 197,748.

Asimismo, se observa que configuraciones cercanas, como (G=450, P=300), (G=500, P=250) y (G=400, P=300), presentan valores de penalización similares indicando que el algoritmo presenta una zona de desempeño estable alrededor de poblaciones altas y generaciones medias-altas, lo cual permite identificar no solo una configuración óptima, sino también un rango de parámetros con buenos resultados.

En las configuraciones con mejor desempeño, la penalización asociada a restricciones fuertes es mínima o prácticamente nula, mientras que la mayor parte del valor total proviene de restricciones débiles. La Figura 4 permite concluir que el incremento del tamaño de población y del número de generaciones mejora la calidad de las soluciones hasta cierto punto.

6.9. Ajuste del operador de cruza en el Algoritmo Genético

Una vez determinada la configuración base óptima del Algoritmo Genético, correspondiente a un tamaño de población de $P = 300$ y un número de generaciones de $G = 500$, se procedió a analizar el efecto de la probabilidad de cruzamiento (p_c) en la calidad de las soluciones obtenidas.

Para ello, se realizó un barrido de la probabilidad de cruzamiento que van desde *0.50* hasta *0.90*, utilizando incrementos de 0.05, lo que dio lugar a diez configuraciones distintas. Este rango fue seleccionado con base en valores comúnmente reportados en la literatura, donde se establece que probabilidades de cruzamiento moderadas a altas favorecen un equilibrio adecuado entre exploración y explotación del espacio de búsqueda (Goldberg, 1989; Mitchell, 1998). El resto de los parámetros del algoritmo se mantuvo constante conforme a la configuración base. Cada configuración de p_c fue evaluada mediante 30 ejecuciones independientes.

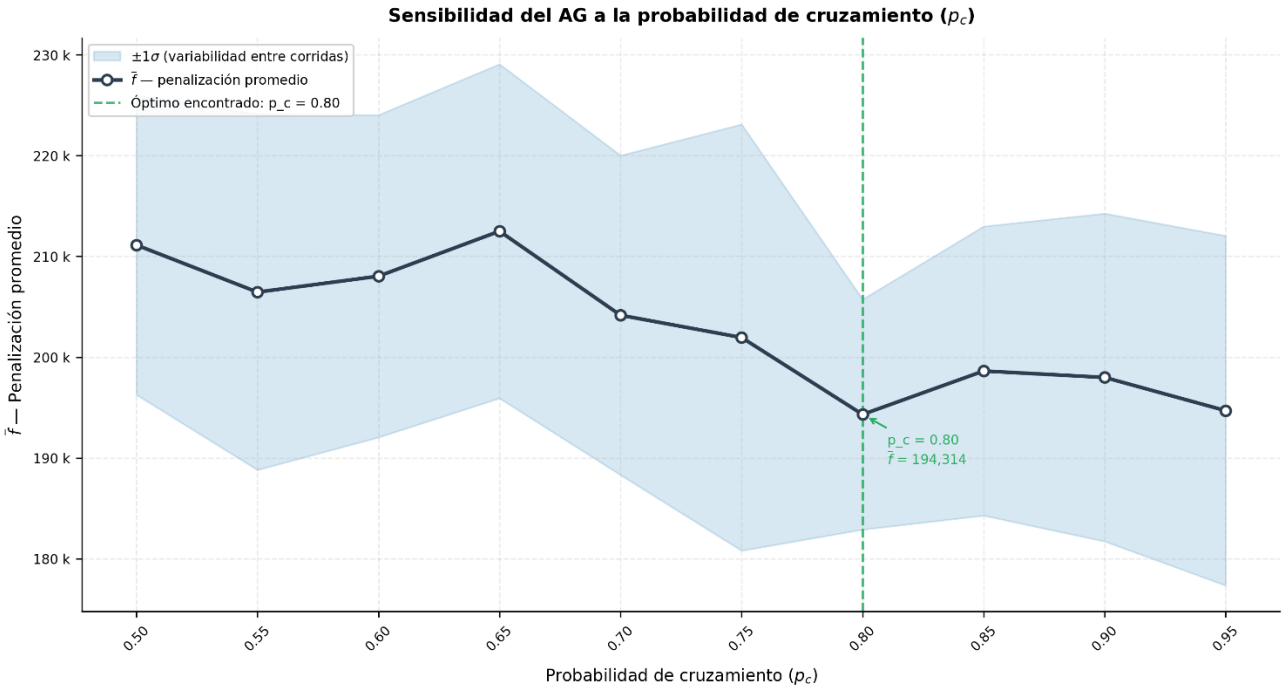

Figura 5. *Sensibilidad del algoritmo a la probabilidad de cruzamiento.*

La Figura 5 muestra la sensibilidad del Algoritmo Genético respecto a la probabilidad de cruzamiento. Se observa un comportamiento no lineal, con un mínimo claro en Pc = *0.80*, donde se obtiene el mejor valor de la función de penalización. Adicionalmente, se identifica que las configuraciones con mejor desempeño se concentran en el rango aproximado de $Pc \in$ *[0.75, 0.85]*, dentro del cual el algoritmo presenta resultados competitivos y una variabilidad relativamente baja.

6.10. *Configuración de parámetros seleccionada*

Con base en los resultados obtenidos en los distintos experimentos, se identificó una configuración recomendada para el Algoritmo Genético, la cual integra los valores de los parámetros que permiten alcanzar el mejor desempeño en términos de calidad de solución.

Tabla 7. *Configuración óptima global del algoritmo*

Parámetro	Valor

Tamaño de población (P)	300
Número de generaciones (G)	500
Probabilidad de cruzamiento (p_c)	0.80
Probabilidad de mutación (μ)	0.12 (adaptativa en rango [0.04–0.18])
Método de selección	Torneo (k): 3
Elitismo (e)	5 individuos
Criterio de estancamiento	10 generaciones
Mejor penalización promedio	≈ 197,748
Penalización por restricciones fuertes (RF)	≈ 0
Penalización por restricciones débiles (RD)	≈ 197,748
Tiempo promedio de ejecución	~200–347 segundos

***Nota**: Este resumen de parámetros considera tanto los parámetros principales del algoritmo como los parámetros internos de operación.*

6.11. Comparativo de desempeño frente al método manual

Para complementar la evaluación del sistema, se realizó un comparativo general entre el proceso manual de generación de horarios laborales y el método automatizado basado en la propuesta. Para ello, se analizan dos aspectos: el tiempo requerido para generarlos y su nivel de cumplimiento de restricciones.

En el método manual, la elaboración de horarios laborales depende directamente de la experiencia del gerente o responsable de programación, quien debe considerar la disponibilidad del personal, los horarios de operación del cine, la capacitación por puesto, los descansos, las restricciones laborales y el presupuesto semanal de horas. Este proceso puede requerir aproximadamente desde 6 hasta 12 horas de revisión y ajuste por semana, sobre todo cuando hay cambios o ajustes.

En cambio, con el sistema se genera automáticamente una propuesta de horario laboral a partir de los datos que ya están cargados, lo que reduce mucho el trabajo manual y permite hacer ajustes después si es necesario.

La Tabla 8 presenta un comparativo general entre ambos enfoques.

Tabla 8. *Comparativo de desempeño entre método manual y sistema propuesto.*

Criterio	Método Manual	Modelo propuesto
Tiempo de generación	Requiere varias horas de análisis, captura y ajuste por parte del responsable (aprox. 6 a 12 horas).	Genera una propuesta semanal en aproximadamente 200 a 300 segundos, según la configuración.
Dependencia del usuario	Alta; depende de la experiencia, criterio y disponibilidad del responsable.	Media; el usuario define parámetros y revisa resultados, pero la asignación inicial es automática.
Evaluación de restricciones	Se realiza manualmente, con riesgo de omitir disponibilidad, descansos, capacitación o límites de jornada.	Se evalúan automáticamente las restricciones fuertes y débiles descritas en la sección 6.1.
Factibilidad de la solución	Puede requerir múltiples correcciones para cumplir con las reglas.	Se obtienen soluciones factibles que cumplen con las restricciones fuertes (Tabla 3).
Capacidad de ajuste	Los cambios se realizan manualmente y pueden afectar otras asignaciones.	Permite generar una base y hacer ajustes posteriores fácilmente.
Escalabilidad	Limitada; el esfuerzo aumenta con más empleados y restricciones.	Mayor capacidad para manejar múltiples combinaciones en cada ejecución.
Capacitación del personal	Si cambia el responsable, puede requerirse capacitación nuevamente.	Menor impacto, ya que el proceso está estandarizado en el sistema.

En general el sistema propuesto representa una mejora significativa respecto al método manual, principalmente en tiempo y control de factibilidad. Mientras que el proceso manual requiere una revisión extensa por parte del responsable de programación, el enfoque automatizado permite obtener una propuesta semanal en pocos minutos.

Es importante señalar que el sistema no elimina por completo la participación del usuario, ya que la propuesta generada puede necesitar revisión o ajustes operativos específicos. Sin embargo, sí disminuye considerablemente la carga inicial de trabajo y proporciona una base funcional para la toma de decisiones. Por lo tanto, el método propuesto funciona como una herramienta de apoyo que mejora la eficiencia del proceso de generación de horarios laborales sin sustituir totalmente el criterio del responsable.

6.12. Visualización del sistema

Con el propósito de mostrar la aplicación práctica del modelo propuesto, se presenta una prueba de laboratorio con las principales interfaces desarrolladas para la configuración, ejecución y visualización de los horarios generados por el Algoritmo Genético.

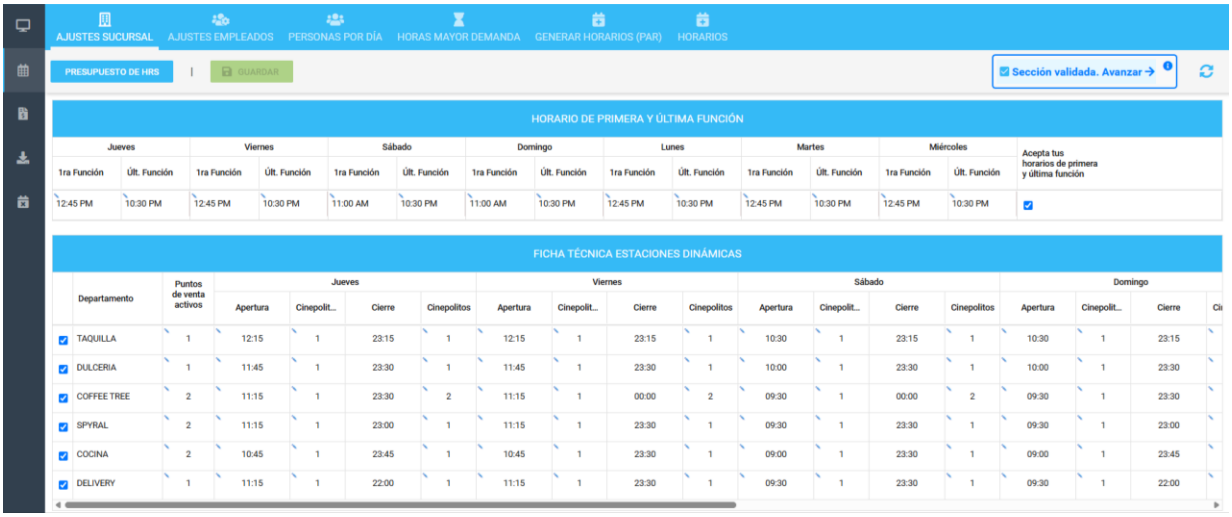

Figura 6. *Configuración de sucursal.*

La Figura 6 muestra la interfaz de configuración de la sucursal, donde se definen los parámetros operativos necesarios para la generación de horarios, tales como horarios de apertura y cierre, funciones por día y requerimientos por estación.

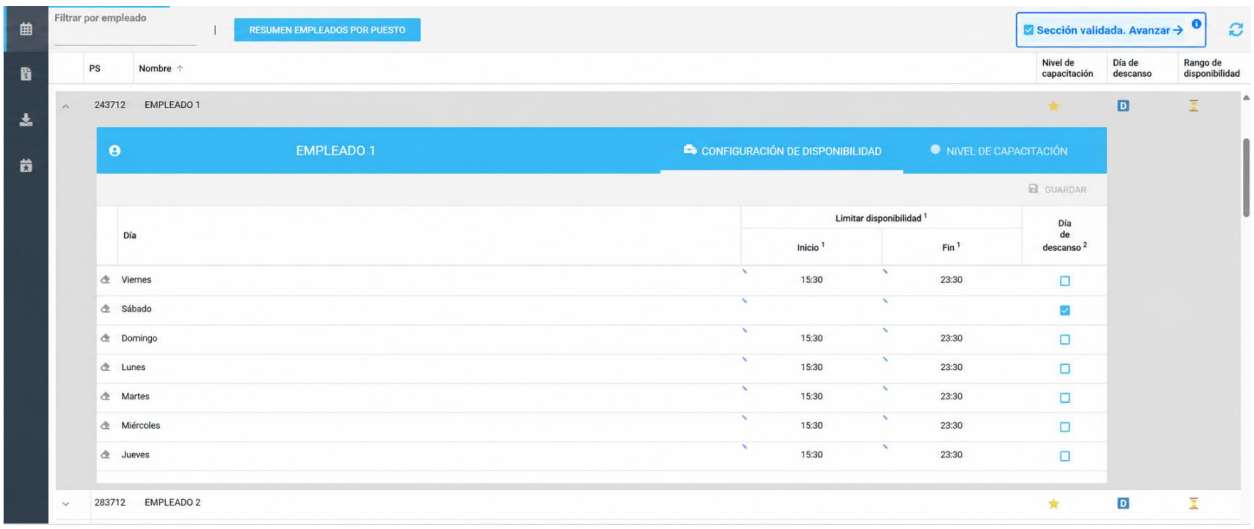

Figura 7. *Configuración de empleados.*

La Figura 7 presenta el módulo de configuración de empleados, donde se registran las disponibilidades, restricciones individuales y niveles de capacitación. Esta pantalla permite establecer las condiciones bajo las cuales cada empleado puede ser considerado dentro del proceso de asignación, favoreciendo que los horarios generados sean congruentes con la disponibilidad y competencias del personal.

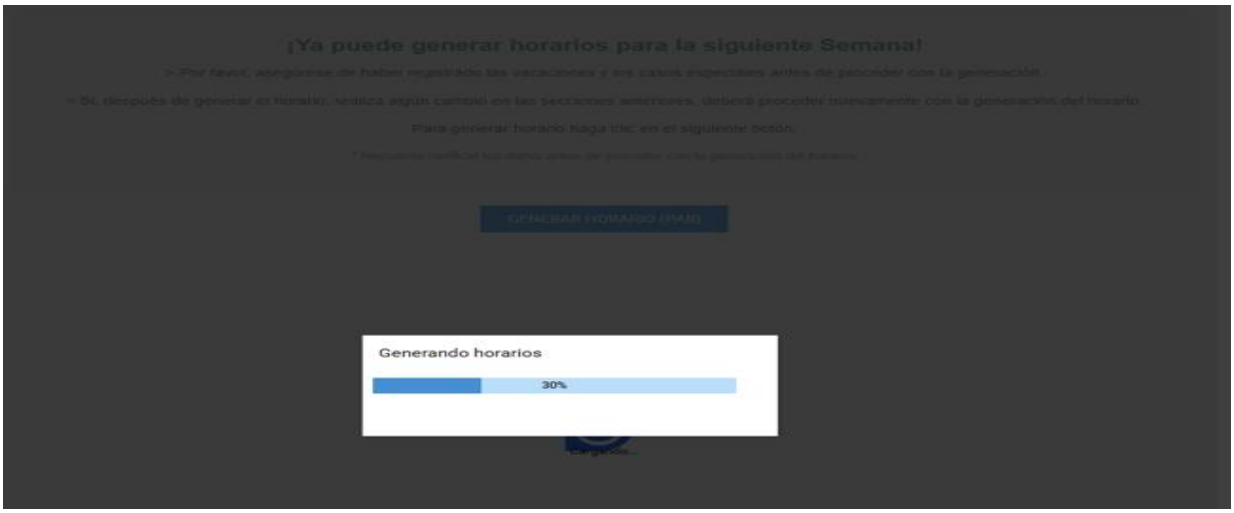

Figura 8. *Generación de horarios laborales.*

La Figura 8 muestra el proceso de generación de horarios, donde el sistema ejecuta el Algoritmo Genético y presenta el avance de la operación. De acuerdo con los resultados experimentales obtenidos, el tiempo promedio se ubicó aproximadamente entre los 200 y 300 segundos con la configuración óptima encontrada (Tabla 7). No obstante, puede variar en función de los recursos computacionales disponibles.

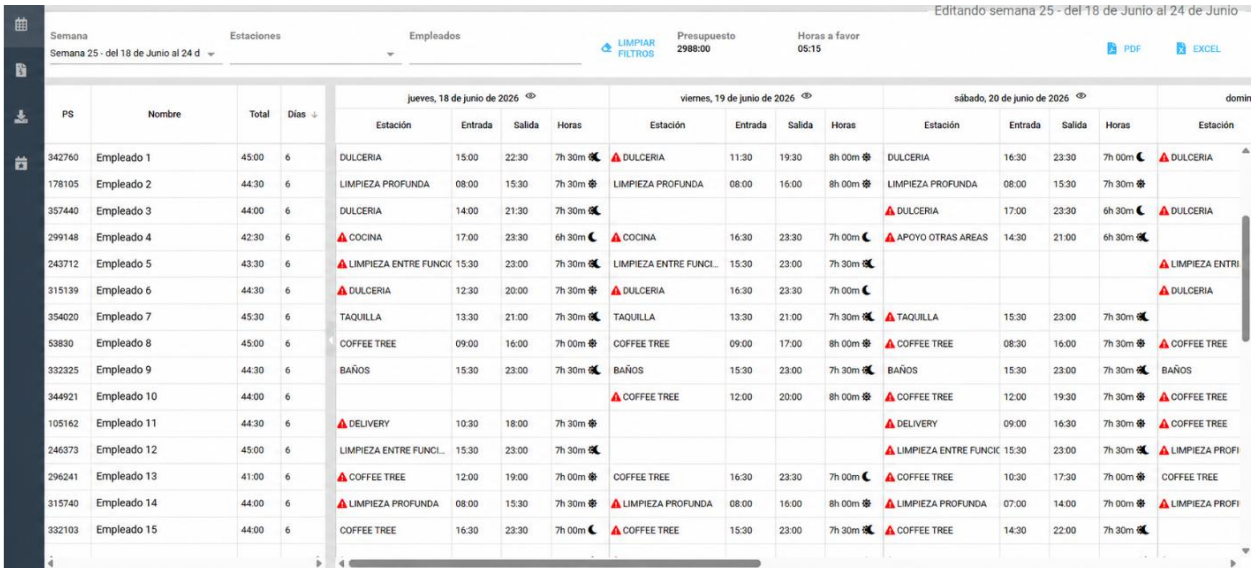

Figura 9. *Resultado final de horarios laborales.*

La Figura 9 muestra el resultado final del proceso de generación, donde se presentan los horarios asignados a cada empleado. En esta interfaz se visualizan las estaciones asignadas, los horarios de entrada y salida, la duración de los turnos y la distribución semanal de las asignaciones.

A diferencia del proceso manual, en el cual la validación de horarios depende directamente de la revisión del gerente o responsable de programación, el sistema permite generar una propuesta estructurada a partir de las restricciones previamente definidas. Esto mejora técnicamente el proceso al integrar de manera automática la disponibilidad del personal, la capacitación por puesto, los horarios operativos del cine y las reglas laborales consideradas en el modelo.

Asimismo, la visualización centralizada facilita la identificación de asignaciones por empleado, día y estación, reduciendo la necesidad de revisar manualmente múltiples fuentes de información. De esta manera, el sistema contribuye a disminuir errores asociados a solapamientos, asignaciones fuera de disponibilidad o incumplimientos de restricciones operativas. Además, la posibilidad de exportar los horarios en formatos XLSX y PDF facilita su consulta, distribución y validación posterior.

La interfaz final no solo muestra el resultado del Algoritmo Genético, sino que representa una mejora técnica frente a la programación manual, al automatizar la generación, validación y presentación de los horarios laborales. En conjunto, las interfaces desarrolladas integran las etapas de configuración, ejecución y visualización del sistema, por lo que la solución queda preparada para su implementación en pruebas piloto en entornos reales.

7. Conclusiones y recomendaciones

7.1. Conclusiones

En la presente investigación se cumplió de manera satisfactoria el objetivo general planteado, consistente en diseñar e implementar un sistema automatizado para la generación de horarios laborales en una cadena de cines mediante el uso de Algoritmos Genéticos. Para ello, se desarrolló un modelo de optimización con representación estructurada de soluciones, una función objetivo basada en restricciones fuertes y débiles, operadores evolutivos adaptados al problema y una interfaz gráfica que permite la configuración, generación y visualización de los horarios propuestos.

En relación con los objetivos específicos, se desarrolló un modelo matemático formal que representa adecuadamente la complejidad del problema de asignación de turnos, integrando restricciones legales, laborales y organizacionales. Asimismo, se diseñó una función objetivo robusta que prioriza el cumplimiento estricto de las restricciones fuertes, permitiendo una optimización progresiva de las restricciones débiles. De manera complementaria, se implementó un sistema funcional, acompañado de una interfaz gráfica de usuario que facilita la interacción con el modelo, permitiendo la generación, visualización y validación de los horarios propuestos. Finalmente, se evaluó experimentalmente el desempeño del algoritmo mediante distintas configuraciones de población, generaciones y probabilidad de cruzamiento, lo que permitió determinar una configuración de parámetros adecuada para el problema estudiado. La configuración óptima identificada fue de 500 generaciones y 300 individuos de población, con una penalización promedio aproximada de 197,748, penalización por restricciones fuertes cercana a cero y un tiempo promedio de generación entre 200 y 300 segundos. Esta integración posiciona la solución como un prototipo funcional aplicable en contextos operativos reales y como una herramienta de apoyo a la toma de decisiones.

Los resultados experimentales obtenidos evidenciaron una mejora significativa en la calidad de las soluciones generadas, particularmente en la reducción de penalizaciones

asociadas a las restricciones del problema. Se observó un comportamiento característico de los Algoritmos Genéticos, donde el proceso evolutivo inicia con una fase de exploración del espacio de soluciones y progresa hacia una fase de explotación enfocada en la mejora de las soluciones más prometedoras. Este comportamiento permitió alcanzar configuraciones donde las restricciones fuertes son mínimas o prácticamente nulas, manteniendo la mayor parte de la penalización residual asociada a restricciones débiles. Asimismo, la experimentación facilitó la identificación de configuraciones adecuadas de parámetros, evidenciando el equilibrio entre calidad de solución y costo computacional.

No obstante, se identificaron algunas limitaciones relevantes. El incremento en el tiempo de ejecución representa un factor a considerar en escenarios que demanden respuestas en tiempo cercano a real. Adicionalmente, aunque el algoritmo logró reducir de manera significativa las restricciones fuertes, no siempre alcanza soluciones completamente óptimas respecto a las restricciones débiles, lo cual abre la posibilidad de mejoras mediante ajustes en la función objetivo, estrategias de penalización o el diseño de operadores genéticos más especializados. Asimismo, los experimentos fueron realizados sobre un conjunto específico de datos, por lo que el comportamiento del modelo podría variar ante escenarios de mayor escala o con condiciones operativas distintas.

En el ámbito científico, esta investigación proporciona una formulación estructurada del problema de asignación de turnos con una clara diferenciación entre tipos de restricciones, así como evidencia empírica sobre el desempeño de los Algoritmos Genéticos en problemas de alta complejidad combinatoria. En el ámbito práctico, la solución desarrollada constituye una alternativa viable para automatizar la asignación de turnos en la industria cinematográfica. Al integrarse con una interfaz gráfica de usuario, favorece la eficiencia operativa, reduce tiempos de planificación y apoya el cumplimiento de normativas laborales.

En conclusión, el uso de Algoritmos Genéticos dentro de un sistema computacional integral demostró ser una estrategia efectiva, robusta y escalable para la resolución del problema de asignación de turnos laborales en entornos dinámicos y altamente restringidos. Este trabajo no solo valida la aplicabilidad de técnicas metaheurísticas en contextos reales, sino que también sienta las bases para futuras investigaciones orientadas al desarrollo de modelos híbridos, particularmente mediante la integración con técnicas de aprendizaje automático, con el propósito de mejorar la adaptabilidad, precisión y eficiencia del sistema propuesto.

7.2. Recomendaciones

Con base en los resultados obtenidos a partir de la implementación y evaluación del Algoritmo Genético propuesto para la generación automática de horarios, se plantean las siguientes recomendaciones, orientadas tanto a su aplicación práctica como a la continuidad de la investigación.

7.2.1. Recomendaciones para futuros trabajos de investigación

1. Explorar estrategias de ajuste automático de parámetros del Algoritmo Genético, con el objetivo de reducir la necesidad de calibración manual y mejorar la robustez del modelo.
2. Evaluar la incorporación de enfoques híbridos, combinando Algoritmos Genéticos con técnicas de búsqueda local u otras metaheurísticas para mejorar el desempeño global.
3. Extender el modelo hacia un enfoque multiobjetivo, que permita analizar de manera independiente distintos criterios relevantes, como la satisfacción de preferencias, la equidad en la carga laboral y el cumplimiento de restricciones.

7.2.2. Recomendaciones para la institución educativa

1. Se recomienda que la institución cuente con infraestructura computacional adecuada (equipos con capacidad de procesamiento suficiente) para la ejecución de algoritmos,

especialmente cuando se trabajen escenarios de gran tamaño o múltiples ejecuciones experimentales.

2. Disponer de software especializado y licencias que permitan el desarrollo, prueba y validación de modelos de optimización, como entornos de programación, y herramientas de análisis de datos.

7.3. Trabajo futuro

Como continuación del presente trabajo, se identifican diversas líneas de investigación orientadas a fortalecer y ampliar la propuesta desarrollada. En primer lugar, se propone la incorporación de enfoques híbridos que integren el Algoritmo Genético con otras técnicas metaheurísticas, con el objetivo de mejorar la calidad de las soluciones obtenidas y acelerar el proceso.

En segundo lugar, se sugiere la implementación de mecanismos de ajuste dinámico de parámetros, mediante los cuales el tamaño de la población o número de generaciones se modifiquen de manera automática durante la ejecución del algoritmo, en función de su desempeño y del nivel de diversidad de la población.

Asimismo, se plantea extender el modelo hacia un enfoque multiobjetivo, que permita optimizar de forma simultánea diversos criterios relevantes, tales como el cumplimiento de restricciones laborales, la equidad en la distribución de turnos y la consideración de las preferencias individuales de los empleados, evitando la agregación de estos aspectos en una única función de penalización.

Otra línea de investigación consiste en la integración de modelos de predicción de demanda, basados en datos históricos, con el fin de anticipar variaciones en la carga operativa

y ajustar de manera más precisa la asignación de personal en los distintos periodos y puestos de trabajo.

Finalmente, se considera la implementación y validación del modelo en un entorno real.

8. Referencias

Acuña-Galván, A., Flores-Ruiz, J., & Reyes-Santiago, E. (2022). Generación de horarios académicos con algoritmos genéticos en Java. *Revista de Tecnología Educativa, 10*(2), 55–70.

aTurnos. (2021). *Gestión de turnos y planificación de personal*. https://www.aturnos.com

Bizneo HR. (2022). *Bizneo HR software*. https://www.bizneo.com

Bożejko, W., Pempera, J., & Smutnicki, C. (2021). Hybrid neuro-tabu search for resource scheduling in the automotive industry. *Applied Soft Computing, 101*, 107041. https://doi.org/10.1016/j.asoc.2020.107041

Burduk, A., Świątek, J., & Wójcik, P. (2022). Neuro-tabu search for production scheduling optimization. *Journal of Manufacturing Systems, 63*, 165–178. https://doi.org/10.1016/j.jmsy.2022.01.008

ClickUp. (2022). *ClickUp project management*. https://clickup.com

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer.

Crepinsek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM Computing Surveys, 45(3), 1–33. https://doi.org/10.1145/2480741.2480752

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). SAGE Publications.

Cuartas, J., & Muñoz, A. (2024). Optimización de horarios académicos mediante algoritmos genéticos en la Universidad EIA. *Revista Colombiana de Computación*.

Darwin, C. (1859). On the Origin of Species by Means of Natural Selection. John Murray.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

De, R., Zambrano, C., & Paredes, M. (2024). Automatización de horarios académicos en Ecuador con algoritmos genéticos. *Revista Científica de Ingeniería*.

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. Journal of Global Optimization, 6(2), 109–133. https://doi.org/10.1007/BF01096763

Fortin, F. A., De Rainville, F. M., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*(70), 2171–2175.

Gad, A. F. (2021). PyGAD: An intuitive genetic algorithm Python library. arXiv. https://doi.org/10.48550/arXiv.2106.06158

Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing, 1*(3), 190–206. https://doi.org/10.1287/ijoc.1.3.190

Glover, F., & Laguna, M. (1997). *Tabu search*. Kluwer Academic Publishers.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Gómez, J., Vargas, M., & Pérez, D. (2020). Comparación de metaheurísticas en la generación de horarios académicos. *Revista Ingeniería e Investigación, 40*(1), 45–56.

Guan, X., Li, Y., & Chen, Z. (2021). A hybrid PSO-ACO algorithm for the nurse rostering problem with multiple objectives. *Expert Systems with Applications, 171*, 114629. https://doi.org/10.1016/j.eswa.2020.114629

Gurobi Optimization, LLC. (2022). Gurobi optimizer reference manual. https://www.gurobi.com

Henry Nelson, A., Artigas Fuentes, F. J., & Hernández Candelaria, M. R. (2021). La planificación docente utilizando algoritmos genéticos. *Didasc@lia: Didáctica y Educación, 12*(4), 1–11.

Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, M. del P. (2018). *Metodología de la investigación* (6.ª ed.). McGraw-Hill Education.

Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kletzander, M., & Musliu, N. (2018). Solving the general employee scheduling problem. *European Journal of Operational Research, 264*(2), 639–654. https://doi.org/10.1016/j.ejor.2017.06.028

Li, Q., Zhao, H., & Wang, J. (2024). Differential evolution for workforce scheduling in manufacturing. *Computers & Industrial Engineering, 180*, 109176. https://doi.org/10.1016/j.cie.2023.109176

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT Press.

Mendel, G. (1866). Versuche über Pflanzen-Hybriden. Verhandlungen des naturforschenden Vereines in Brünn, 4, 3–47.

Ochoa Buitrago, S. D. (2012). *Análisis de variabilidad en asignación de personal para la unidad de urgencias de un hospital* [Tesis de maestría, Universidad de los Andes]. Repositorio Uniandes. https://repositorio.uniandes.edu.co

Odoo S.A. (2023). *Odoo documentation*. https://www.odoo.com

OptaPlanner. (2023). Open source constraint satisfaction solver. https://www.optaplanner.org

Perron, L., & Furnon, V. (2025). *OR-Tools* (Version 9.12) [Software]. Google.

PGPlanning. (2021). *PGPlanning software*. https://www.pgplanning.com

Plano WFM. (2021). *Plano workforce management*. https://www.plano-wfm.com

Rodríguez, N., Martínez, J., Flores, J. J., & Graff, M. (2014). Solving a scholar timetabling problem using a genetic algorithm – Study case: Instituto Tecnológico de Zitácuaro. 2014 13th Mexican International Conference on Artificial Intelligence. IEEE. https://ieeexplore.ieee.org/document/7222864

Sahraeian, A. (2024). Shift scheduling: A comprehensive literature review. *Computers & Industrial Engineering, 185*, 109768. https://doi.org/10.1016/j.cie.2023.109768

SDP Era. (2021). *SDP Era Plan software*. https://www.sdpera.com

Sesame HR. (2022). *Gestión de recursos humanos y control horario*. https://www.sesamehr.com

Shiftboard. (2021). *Workforce scheduling software*. https://www.shiftboard.com

Soriano, J., Pérez, M., & García, P. (2020). Integrated employee scheduling with known employee demand, including breaks, overtime, and preferences. *Journal of Scheduling, 23*(5), 623–640. https://doi.org/10.1007/s10951-020-00652-3

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Wiley.

Terán-Pozo, J., López, G., & Arévalo, M. (2022). Generación automática de horarios académicos utilizando algoritmos genéticos en Python. *Revista Politécnica*.

UKG. (2022). *Kronos workforce management solutions*. https://www.ukg.com

Uribe Pérez, O. (2016). *Modelo de asignación de turnos para la programación de conductores en la operación de sistemas de transporte masivo* [Tesis de pregrado, Universidad Tecnológica de Pereira]. Repositorio UTP. https://repositorio.utp.edu.co

Velásquez-Restrepo, P. A., Rodríguez-Quintero, A. K., & colaboradores. (2016). Metodologías cuantitativas para la optimización del servicio de urgencias: Una revisión de la literatura. *Gerencia y Políticas de Salud, 15*(30), 100–118.

Viñas, S., & Rodríguez, N. (2018). Software para la generación automática de horarios académicos. *Avances en Ciencias e Ingeniería, 9*(17), 97–110.

Woffu. (2021). *Gestión de ausencias y vacaciones*. https://www.woffu.com

Workday. (2022). *Workday scheduling & workforce planning*. https://www.workday.com

Woxi Digital. (2021). Woxi digital HR software. https://www.woxi.es

Xu, Y., & Wang, X. (2022). A hybrid integer programming and artificial bee colony algorithm for staff scheduling in call centers. Computers & Industrial Engineering, 166, 108312. https://doi.org/10.1016/j.cie.2022.108312