

Aplicación web para la asignación automática de horarios docentes mediante algoritmos genéticos

Web application for automatic teacher timetabling using genetic algorithms

César González-Camargo¹, Justmary Anderson-Fermín², José Chiru-Figueroa³

¹Universidad Tecnológica de Panamá, Facultad de Ingeniería de Sistemas Computacionales, Panamá, cesar.gonzalez9@utp.ac.pa, <https://orcid.org/0009-0006-8862-8733>

²Universidad Tecnológica de Panamá, Facultad de Ingeniería de Sistemas Computacionales, Panamá, justmary.anderson@utp.ac.pa, <https://orcid.org/0009-0001-2625-0556>

³Universidad Tecnológica de Panamá, Facultad de Ingeniería de Sistemas Computacionales, Panamá, jose.chiru1@utp.ac.pa, <https://orcid.org/0009-0002-1930-3573>

DOI: <https://doi.org/10.48204/j.colonciencias.v13n2.a9267>

Resumen

En las instituciones de educación superior, la planificación de horarios docentes es un proceso clave para garantizar la continuidad académica. En este contexto, el presente estudio abordó la problemática de la asignación manual de horarios docentes en estos entornos, caracterizada por alta complejidad combinable, predisposición a errores y consumo elevado de tiempo administrativo. El objetivo fue desarrollar y validar una aplicación web para la asignación automática de horarios docentes mediante algoritmos genéticos. La metodología aplicada fue de tipo experimental y cuantitativa, estructurada en fases de análisis, diseño, implementación y evaluación. El sistema fue desarrollado utilizando una arquitectura dockerizada basada en Python y SQL Server, integrando una función de aptitud multiobjetivo para gestionar restricciones duras (disponibilidad) y blandas (equidad). Los resultados obtenidos tras las pruebas con datos reales evidenciaron que el sistema redujo el tiempo de planificación en más de un 95%, eliminó los conflictos de horario en el escenario evaluado y mejoró el equilibrio de la carga académica entre los docentes. En consecuencia, la solución propuesta es viable, eficiente y escalable para optimizar la gestión académica.

Palabras clave: asignación automática, horarios docentes, inteligencia artificial, algoritmos genéticos, optimización académica.

Abstract

In higher education institutions, teacher scheduling is a key process for ensuring academic continuity. In this context, this study addressed the problem of manual teacher scheduling in these environments, characterized by high combinatorial complexity, susceptibility to errors, and high administrative time consumption. The objective was to develop and validate a web application for the automatic assignment of teaching schedules using genetic algorithms. The methodology applied was experimental and quantitative, structured in phases of analysis, design, implementation, and evaluation. The system was developed using a dockerized architecture based on Python and SQL Server, integrating a multi-objective fitness function to manage hard (availability) and soft (equity) constraints. The results obtained after testing with real data showed that the system reduced planning time by more than 95%, eliminated schedule conflicts in the evaluated scenario, and improved the balance of academic workload among teachers. Consequently, the proposed solution is viable, efficient, and scalable for optimizing academic management.

Keywords: automatic scheduling, teaching load, artificial intelligence, genetic algorithms, academic optimization.

Introducción

En las instituciones de educación superior, la planificación académica es un proceso estratégico para sincronizar la oferta educativa con los recursos disponibles y asegurar la continuidad y calidad del proceso enseñanza-aprendizaje (Babaei *et al.*, 2015). Aunque se cuenta con instrumentos de gestión académica institucional para registrar y certificar la organización docente, el proceso de asignación de profesores a las franjas horarias se sigue realizando de manera

manual. Esta tarea es asignada a los jefes de departamento, los cuales tienen que resolver un problema combinatorio complejo con métodos clásicos (Valverde Calderón, 2019) que escalan mal a medida que la institución crece (Zhu *et al.*, 2021). Esta dificultad se debe a que la planificación de horarios es un problema de optimización NP-hard, donde el número de posibles combinaciones crece exponencialmente, haciendo imposible una solución manual exhaustiva (Lewis & Thompson, 2015).

La dependencia de procesos manuales o semiautomatizados genera grandes problemas de eficiencia administrativa y clima laboral. Entre los principales problemas se encuentran los conflictos de disponibilidad y superposición en los horarios, que obligan a realizar correcciones de última hora y reelaboración costosas. Además, la ausencia de mecanismos algorítmicos crea desequilibrios en la carga docente, dando lugar a distribuciones horarias no siempre justas entre el profesorado. A ello se unen los tiempos extendidos de planificación, que pueden llegar a semanas, y la susceptibilidad a errores humanos al manipular grandes volúmenes de datos sin validaciones automatizadas en tiempo real (Wang *et al.*, 2024). Pero, además, la trazabilidad de las decisiones es baja, lo que hace difícilmente justificable de manera técnica por qué se asignó un determinado recurso en caso de auditorías o reclamos.

La dependencia de procesos manuales o semiautomatizados genera grandes problemas de eficiencia administrativa y clima laboral. Entre los principales problemas se encuentran los conflictos de disponibilidad y superposición en los horarios, que obligan a realizar correcciones de última hora y reelaboración costosas. Además, la ausencia de mecanismos algorítmicos crea desequilibrios en la carga docente, dando lugar a distribuciones horarias no siempre justas entre el profesorado. A ello se unen los tiempos extendidos de planificación, que pueden llegar a semanas, y la susceptibilidad a errores humanos al manipular grandes volúmenes de datos sin validaciones automatizadas en tiempo real (Wang *et al.*, 2024). Pero, además, la trazabilidad de las decisiones es baja, lo que hace difícilmente justificable de manera técnica por qué se asignó un determinado recurso en caso de auditorías o reclamos.

Ante estos retos, la literatura científica muestra que la automatización con inteligencia artificial (IA) es una solución posible y necesaria. Varias instituciones a nivel mundial (Ceschia *et al.*, 2023) como la Telkom University en Indonesia (Ramadhani *et al.*, 2025) o la Universidad de Sharjah en Emiratos Árabes Unidos (Lulu *et al.*, 2024), han utilizado con éxito algoritmos genéticos y otras metaheurísticas para resolver problemas de planificación de horarios (Ahumada Ahumada, 2014). Estas técnicas pueden tomar en cuenta miles de variables al mismo tiempo (restricciones contractuales, preferencias de los profesores, disponibilidad de aulas, etc.) para crear horarios optimizados que minimicen los conflictos y maximicen la satisfacción de todos.

En este contexto, los algoritmos genéticos, una técnica metaheurística de búsqueda y optimización perteneciente a IA, inspirado en la evolución biológica (Goldberg, 1989). Estos algoritmos crean una población inicial de soluciones que se va evolucionando a través de generaciones sucesivas mediante operadores de selección, cruce y mutación. Su habilidad para recorrer grandes espacios de búsqueda, encontrar buenas soluciones en tiempo razonable los hace apropiados para problemas multidimensionales y difíciles, como la asignación de horarios, en los que no hay una mejor solución lineal, sino un arreglo óptimo entre muchas restricciones.

Metodología

La investigación es de tipo aplicada y buscó desarrollar y validar una aplicación web para la asignación automática de horarios docentes usando algoritmos genéticos. Siguió un enfoque cuantitativo, midiendo de forma objetiva variables numéricas como el tiempo de procesamiento, el número de conflictos de horario y la carga docente. Para la validación de la solución tecnológica, se realizaron comparaciones utilizando datos auténticos de un período académico reciente. El algoritmo se ejecutó con los mismos datos que se utilizan en la planificación manual oficial (disponibilidad docente, oferta académica, etc.), permitiendo la comparación de ambos métodos en condiciones equivalentes.

La validación se estructuró en torno a tres indicadores clave de desempeño (KPIs) diseñados para medir objetivamente la mejora respecto al método tradicional:

1. Eficiencia Temporal: Se cronometró el tiempo requerido por el sistema para generar una solución completa y se comparó con el tiempo que tardan los encargados en el proceso manual.
2. Calidad y Factibilidad: Se midió la cantidad de "restricciones duras" violadas, probando la habilidad del sistema para encontrar soluciones sin errores técnicos.
3. Equidad y Distribución: Se analizó el equilibrio de la carga académica calculando la desviación estándar de las horas asignadas y así establecer qué tan equitativa fue la distribución del trabajo entre los profesores en ambos escenarios.

Herramientas y entorno de desarrollo

Para la creación del sistema se utilizó una arquitectura modular full-stack de código abierto que asegura escalabilidad y mantenimiento. El backend se codificó en Python 3.9 usando Flask 2.3, que maneja la lógica de negocios y ejecuta el algoritmo genético con bibliotecas científicas como Polars. Para el frontend se empleó HTML5, CSS3, y JavaScript.

La persistencia de datos se implementó con Microsoft SQL Server 2019, el entorno de desarrollo y despliegue se estandarizó con Docker, orquestando los servicios en contenedores separados para garantizar la portabilidad entre entornos de prueba y producción (Liao *et al.*, 2026). Las pruebas y validaciones del algoritmo se realizaron en una computadora con procesador Intel i5 de 8va generación, 16 GB de RAM y 500 GB de almacenamiento SSD.

Etapas del desarrollo de la aplicación web

Para el desarrollo de la aplicación web, se siguió una metodología sistemática de ingeniería de software estructurada en cuatro etapas principales:

Análisis: Recopilación de los requisitos funcionales y análisis de las restricciones (rígidas y flexibles) del proceso actual de la institución.

Diseño: Diseño de la arquitectura del software, el modelo de base de datos y la interfaz de usuario.

Implementación: Codificación de los módulos del sistema y del algoritmo genético.

Evaluación: Ejecución de pruebas con datos reales, destinada a validar el funcionamiento del sistema y cuantificar las mejoras logradas frente al método manual.

Diseño del sistema

La aplicación se estructuró bajo una arquitectura de tres capas:

Capa de Presentación (Frontend): Interfaz que permite a los usuarios autenticarse, y visualizar los resultados mediante tableros de control y reportes gráficos.

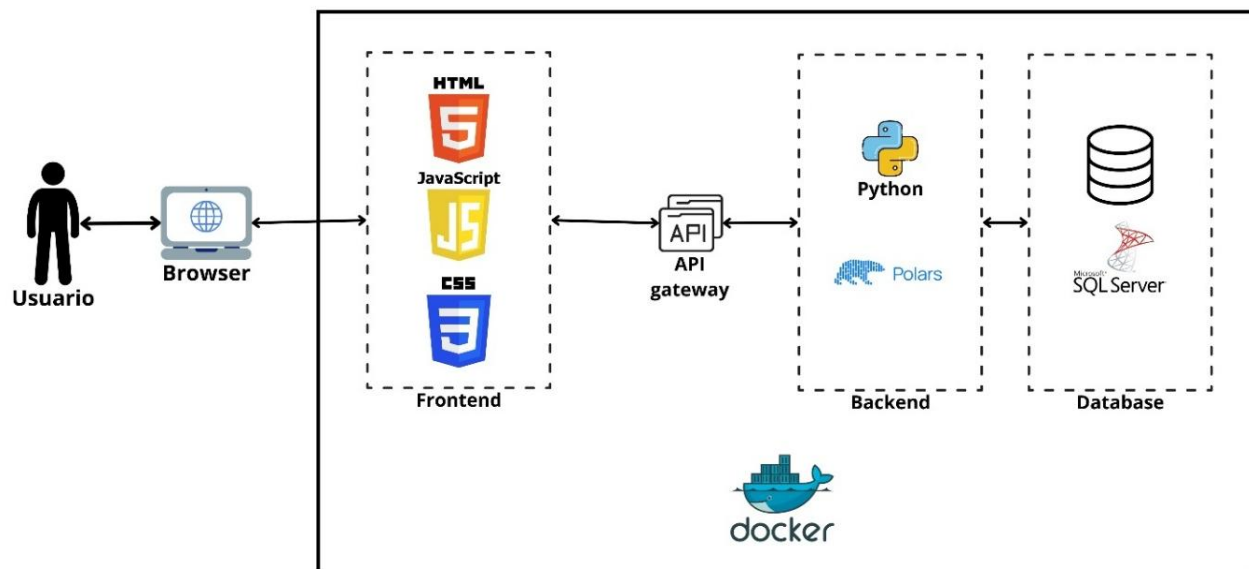
Capa de Lógica de Negocio (Backend): Núcleo del sistema. Este módulo procesa las consultas, ejecuta las rutinas de optimización, valida las restricciones académicas y gestiona la comunicación entre el usuario y los datos.

Capa de Datos (Persistencia): Alberga los registros de usuarios y los historiales de asignaciones creados.

En la Figura 1 se detalla la integración de los componentes tecnológicos y el flujo de información de la aplicación. El sistema emplea un modelo cliente-servidor donde el usuario interactúa mediante un navegador con una interfaz desarrollada en HTML5, JavaScript y CSS3. Esta comunicación se gestiona a través de un API Gateway, que sirve como puente hacia el backend programado en Python, utilizando la librería Polars para el procesamiento de alto rendimiento de datos. La información se centraliza en un servidor SQL Server, mientras que toda la infraestructura se encuentra dockerizada para asegurar la portabilidad y consistencia del entorno de ejecución.

Figura 1

Arquitectura del sistema



Fuente: Elaboración propia

Algoritmo genético implementado

Para abordar el problema de optimización combinable (Yakovlev *et al.*, 2018), se diseñó e implementó un algoritmo genético (AG) personalizado capaz de gestionar restricciones multidimensionales (Babaei *et al.*, 2015). La implementación se estructuró en torno a una arquitectura evolutiva que integra validaciones en tiempo real para garantizar la factibilidad de las soluciones desde las primeras generaciones (Alghamdi *et al.*, 2020).

Codificación y Representación

La unidad básica de procesamiento es el individuo, el cual representa una propuesta completa de asignación para todas las facultades. A nivel de genotipo, cada individuo se codifica como una lista de objetos “Asignación Docente”, donde cada gen vincula tres elementos clave: un docente, un grupo académico y un bloque horario predefinido.

Para reducir el espacio de búsqueda y asegurar la coherencia pedagógica, se implementó una fase previa de "agrupamiento", donde los horarios individuales se consolidan en bloques compactos (por ejemplo, segmentos de 3 horas continuas) antes de entrar al ciclo evolutivo.

Función de Aptitud (Fitness)

La calidad de cada solución se evalúa mediante una función multiobjetivo (Diallo & Tudose, 2024) que pondera el cumplimiento de restricciones estrictas y criterios de optimización deseables. El puntaje de aptitud se calcula penalizando las restricciones duras y mejorando las blandas:

Restricciones Duras: Inviabilidad contractual (exceder la carga horaria máxima del docente), choques de horario (asignar al mismo profesor en dos grupos simultáneos) y falta de perfil (asignar materias fuera de la especialidad del docente). El sistema aplica una validación inmediata que descarta o penaliza drásticamente a los individuos que violan estas reglas.

Restricciones Blandas:

Equidad de carga: Se minimiza la desviación estándar de las horas asignadas para evitar desproporciones entre profesores.

Compactación: Se premian los horarios que evitan "huecos" o ventanas libres excesivas en la jornada del docente.

Cobertura: Se maximiza el porcentaje total de grupos asignados.

Operadores Evolutivos y Parámetros

El motor evolutivo utiliza operadores especializados para explorar el espacio de soluciones sin perder la validez de las asignaciones:

Selección: Se emplea una estrategia elitista, garantizando que el mejor subconjunto de soluciones de una generación sobreviva inalterado a la siguiente, protegiendo así los hallazgos óptimos.

Cruzamiento: Se implementó el operador de "cruce de punto medio con validación". Esta toma dos individuos padres y genera un descendiente combinando la primera mitad de las asignaciones del padre A con la segunda mitad del padre B. Críticamente, cada gen heredado pasa por una verificación de conflictos; si una asignación genera un choque, es reparada o descartada instantáneamente.

Mutación Adaptativa: A diferencia de una tasa fija, se utilizó una tasa de mutación dinámica (iniciando típicamente en 0.1 o 10%). Si el algoritmo detecta estancamiento (el fitness no mejora tras N generaciones), la tasa de mutación aumenta automáticamente para introducir diversidad y escapar de óptimos locales. La mutación opera reasignando aleatoriamente un docente compatible a un grupo específico.

Estrategia de Ejecución

El algoritmo se diseñó para operar bajo un esquema de Convergencia Temprana Inteligente, lo que permite optimizar recursos computacionales sin sacrificar la calidad de los resultados.

La ejecución se rige por los siguientes parámetros y condiciones:

Población y Ciclo Evolutivo: Se genera una población inicial de 100 individuos, donde cada individuo es una propuesta completa de horario. El ciclo evolutivo está configurado para ejecutarse hasta un máximo de 200 generaciones, un límite determinado experimentalmente como suficiente para alcanzar la estabilidad en el fitness.

Elitismo Controlado: Para evitar la pérdida de soluciones prometedoras, se aplica una estrategia elitista que preserva el mejor subconjunto de individuos de cada generación. Estos "campeones" pasan intactos a la siguiente iteración, garantizando que la calidad de la

solución nunca retroceda, mientras el resto de la población se somete a los operadores de cruce y mutación.

Criterios de Parada

El proceso evolutivo se detiene automáticamente si se cumple alguna de las dos condiciones:

Si se alcanza el número máximo de generaciones (100).

Si se detecta estancamiento del fitness, definido como la ausencia de mejoras significativas en el puntaje de aptitud durante un número consecutivo de generaciones predefinido.

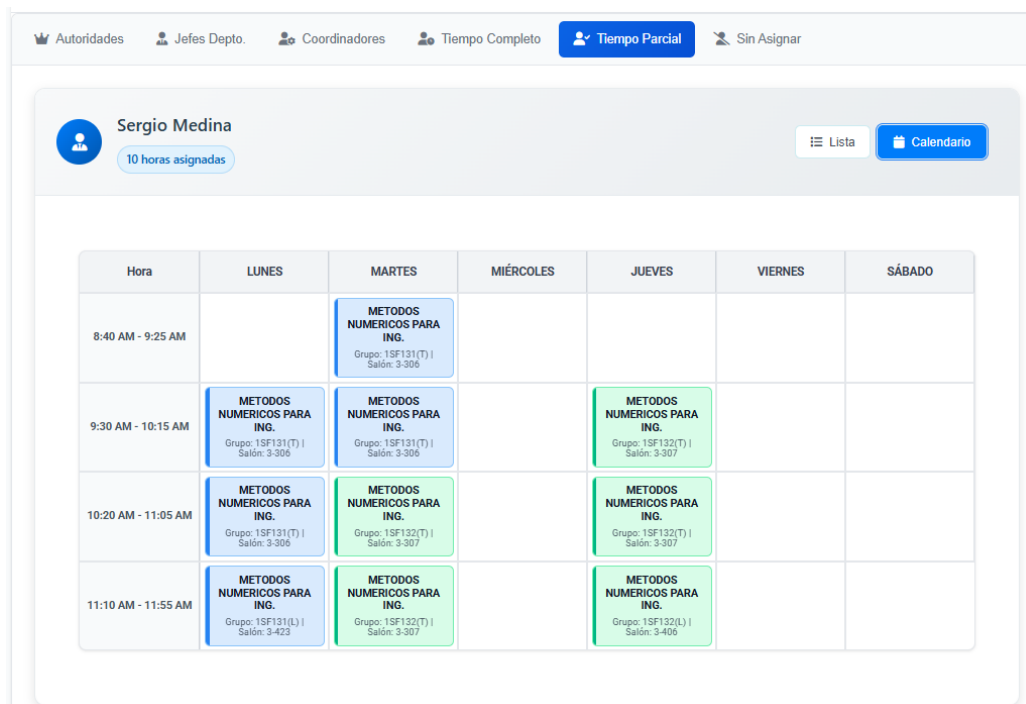
Resultados

Para evidenciar la operatividad del sistema y validar visualmente la calidad de las soluciones generadas por el algoritmo, se presentan a continuación las interfaces más relevantes de la aplicación. Correspondientes a las Figuras 2, 3, 4, 5 y 6.

La Figura 2 muestra la matriz de horarios resultante tras la ejecución del algoritmo genético. En esta vista de calendario semanal, las asignaciones se distribuyen visualmente por días (columnas) y franjas horarias (filas), permitiendo identificar de manera inmediata la ocupación de cada docente. Cada bloque de color representa una asignatura asignada, detallando el nombre del curso, el grupo y el aula correspondiente.

Figura 2

Matriz de horarios generada



La Figura 3 ofrece una perspectiva consolidada del cumplimiento de la asignación por asignatura, permitiendo monitorear el avance del proceso en tiempo real. En esta pantalla, el sistema lista las materias del departamento e indica, mediante barras de progreso y etiquetas de estado, el nivel de cobertura alcanzado para cada curso. Cada fila presenta el código y nombre de la asignatura, junto con una visualización clara de la proporción de grupos que ya cuentan con un docente titular frente al total programado. Los indicadores de estado en color verde confirman que todos los grupos de esas materias han sido cubiertos exitosamente, mientras que el botón de detalle permite al administrador acceder rápidamente a la lista de docentes encargados, facilitando la supervisión y asegurando la cobertura académica total

Figura 3

Materias con Profesor Asignado

■ Materias del Departamento: Departamento de Computación y Simulación de Sistemas

● Materias Asignadas ● Sin Asignar

Se encontraron 18 materias del departamento con profesores asignados

Código	Nombre de la Materia	Cobertura de Grupos	Profesores Asignados	Estado
0592	ESTRUCTURAS DISCRETAS PARA COMPUTACIÓN	8/8 100%	Ver profesores asignados	Completa
0684	ESTRUCTURA Y REPRESENTACIÓN DE DATOS	6/6 100%	Ver profesores asignados	Completa
0697	COMPUTACIÓN GRÁFICA Y VISUAL	3/3 100%	Ver profesores asignados	Completa
1195	ANÁLISIS DATOS Y TOMA DE DEC. PARA COMP.	4/4 100%	Ver profesores asignados	Completa
1196	HERRAMIENTAS DE COMPUTACIÓN GRÁFICA	4/4 100%	Ver profesores asignados	Completa
1197	INTELIGENCIA ARTIFICIAL	4/4 100%	Ver profesores asignados	Completa

Al interactuar con el listado de materias, el sistema despliega una ventana modal (Figura 4) que permite auditar la asignación específica de una asignatura. Este componente presenta un resumen ejecutivo en la parte superior con el total de profesores y grupos involucrados. En el cuerpo principal, se detalla la información de cada docente asignado, incluyendo su estatus contractual (Tiempo Completo, Parcial, etc.) y su categoría académica. Junto a cada perfil, se listan mediante etiquetas azules los códigos de los grupos que tiene bajo su responsabilidad. Esta vista es fundamental para validar que la distribución de la carga respete la equidad y las normativas departamentales, asegurando que cada sección cuente con el docente adecuado.

Figura 4

Profesores Asignados a una Materia



En contraparte, la Figura 5 permite gestionar las excepciones del proceso mediante la vista de "Materias no asignadas". Esta pantalla filtra y lista exclusivamente aquellas asignaturas que, tras la ejecución del algoritmo, aún tienen grupos pendientes de cobertura docente. Para cada registro, se informa el código, nombre de la materia y la cantidad exacta de grupos disponibles por turno (ej. Especial Diurno, Regular Nocturno). Al igual que en la Figura 3 la interfaz incorpora un botón interactivo "Ver detalle de grupos". Al accionarlo, el sistema despliega una ventana modal con la información granular de los grupos vacantes (horarios, días y aulas), facilitando al administrador el análisis de las causas como falta de disponibilidad horaria compatible y la toma de decisiones para la asignación manual o la búsqueda de nuevos docentes.

Figura 5

Materias no Asignadas

Materias del Departamento: Departamento de Computación y Simulación de Sistemas				
Materias Asignadas Sin Asignar				
Se encontraron 2 materias del departamento sin profesores asignados				
Código	Nombre de la Materia	Grupos Disponibles	Turnos	Ver Grupos
8823	DISEÑO VISUAL PARA LA EDUCACION	2 grupos	Especial Diurno Especial Nocturno	Ver detalle de grupos
8837	TALLER DE INFOR.APPLIC.A LA EDUC.	1 grupos	Regular Nocturno	Ver detalle de grupos

Por último, la Figura 6 ilustra el módulo de reportes, diseñado para garantizar la trazabilidad y facilitar la toma de decisiones. El sistema ofrece tres tipos de informes exportables en formato PDF:

Reporte Completo de Profesores: Genera un expediente detallado de todos los docentes asignados al departamento. Incluye información contractual (estatus de tiempo completo o parcial), carga horaria total, grupos a cargo y un resumen ejecutivo, además de anexar los horarios individuales de cada profesor.

Reporte de Conflictos: Documento crítico para la auditoría del algoritmo. Identifica cualquier colisión de horarios detectada y señala casos de sobrecarga académica (profesores con más horas de las permitidas), permitiendo una corrección rápida de anomalías.

Materias Sin Asignar: Consolida la información de aquellas asignaturas que quedaron sin cobertura docente. El reporte detalla el nombre de la materia, el grupo específico afectado, los días, turnos y aulas correspondientes, proporcionando al administrador la data necesaria para gestionar contrataciones o reasignaciones manuales.

Figura 6

Reportes disponibles

Reportes Disponibles


Reporte Completo de Profesores
 Listado detallado de todos los profesores y sus asignaciones
 


Reporte de Conflictos
 Análisis detallado de conflictos de horarios
 


Materias Sin Asignar
 Detalle de materias que no pudieron ser asignadas
 

A continuación, se presenta la Tabla 1, donde se compara el desempeño entre el método manual tradicional y los obtenidos mediante la aplicación web.

Tabla 1

Comparación de desempeño entre el método manual y el sistema propuesto basado en algoritmos genéticos

Métrica	Método Manual	Sistema Propuesto	Impacto
Tiempo de Ejecución	Semanas / Días	< 30 Minutos	Reducción > 95% (estimada)
Conflictos de Horario	8% - 15% (Frecuentes)	0% (en las pruebas)	Eliminación en el escenario evaluado.
Cobertura de Grupos	Variable (requiere ajuste)	90% - 100%	Optimización Alta
Equidad de Carga	Alta (no balanceada)	Baja (Balanceada)	Distribución más justa

Validación de	Visual/manual	Automática	Cumplimiento $\approx 90\%$
Restricciones	(propensa a error)	(durante ejecución)	

La tabla 1 presenta una estructura comparativa diseñada para detallar los indicadores técnicos y operativos evaluados durante la investigación. El contenido se organiza en cuatro columnas principales que especifican la métrica analizada, el estado del proceso manual tradicional, los valores obtenidos por el sistema propuesto y el impacto final derivado de la automatización.

Discusión

El desarrollo y validación de la aplicación web para la asignación automática de horarios docentes en la institución de estudio logra demostrar que los algoritmos genéticos optimizan en gran medida el proceso en comparación con la forma manual. Los resultados tabulados validan una mejora radical en la eficiencia operativa, donde el tiempo de planificación pasó de semanas a menos de 30 minutos, logrando una optimización de más del 95%. Esta disminución temporal no solo implica un aumento en la productividad administrativa, sino que confirma los resultados de (Ramadhani *et al.*, 2025) y (Lulu *et al.*, 2024), quienes encontraron mejoras operativas similares al automatizar procesos de planificación de horarios en universidades internacionales. Estos resultados demuestran que los algoritmos genéticos pueden superar las restricciones cognitivas y de tiempo de la planificación manual y pueden manejar la complejidad que está más allá de la capacidad humana.

Comparando la exactitud técnica, el sistema eliminó por completo los conflictos de disponibilidad horaria que históricamente oscilaban entre un 8% y un 15% en el proceso manual. La habilidad del sistema para hacer verificaciones en tiempo real aseguró que ninguna solución violara restricciones contractuales clave, como el límite de horas por profesor, alcanzando una confiabilidad que el proceso manual no puede garantizar sin revisiones posteriores exhaustivas (Salmasnia *et al.*, 2011). Este paso al 0% de conflictos y la validación automática continua dio una

fiabilidad del 90% en el cumplimiento de las normas institucionales, superando la susceptibilidad a errores visuales de la asignación manual.

Esta mejora en la exactitud técnica y la ausencia de conflictos coincide con los hallazgos de (Bensky & Saunders, 2025) que demuestran cómo los algoritmos genéticos en departamentos de ciencias logran una asignación de cargas docentes eficiente y, sobre todo, justa. Como en esta investigación, estos autores señalan que la habilidad del algoritmo para manejar múltiples restricciones al mismo tiempo permite a los gestores académicos mover su papel de 'hacedor manual' a 'supervisor estratégico' y asegurar que la asignación final sea equitativa y transparente para todo el profesorado.

En términos de cobertura, el sistema se optimizó, alcanzando entre el 90% y el 100% de cobertura de grupos académicos. En la métrica de “equidad de carga”, el algoritmo reemplazó el criterio discrecional anterior por una distribución equitativa de baja desviación, en línea con las tesis de (Abdipoor *et al.*, 2023) sobre la mejora de la calidad de vida laboral a través de la nivelación de cargas. El principal hallazgo en esta área es que el algoritmo, con su función de aptitud multiobjetivo, es capaz de distribuir la carga de manera más equitativa que los métodos convencionales, satisfaciendo los criterios de equidad institucional propuestos por (Agbolade *et al.*, 2024).

Sin embargo, el resultado del sistema está condicionado a la información de los docentes cuando esta información de partida es incompleta o inconsistente, el algoritmo mostró estas limitaciones en la cobertura final, lo que destaca la necesidad de procesos de limpieza de datos previos a la ejecución automatizada para evitar el principio "Garbage In, Garbage Out". Esta dependencia total a los archivos de entrada es la razón por la cual la fiabilidad técnica es de un 90% y no de un 100%, ya que el sistema depende de la exactitud de la información base proporcionada por la facultad.

Si bien la aplicación da solución a las restricciones técnicas y operacionales que se muestran en la Tabla 1, aún queda pendiente la implementación de mecanismos de retroalimentación cualitativa para evaluar aspectos subjetivos de la satisfacción del profesorado.

Finalmente, la arquitectura Cliente - Servidor con docker y la inmersión de reglas de negocio propias de la institución permite la escalabilidad para ser desplegada en otras unidades académicas sin realizar modificaciones.

Conclusión

La aplicación demuestra ser viable para la asignación automática de horarios docentes. Los resultados confirman que se puede disminuir el tiempo de planificación en más del 95%, y a la vez asegurar el cumplimiento riguroso de políticas institucionales y mejorar la repartición de carga laboral entre los docentes.

La implementación de algoritmos genéticos permite generar horarios sin conflictos de disponibilidad ni superposición, gestionando de forma eficiente tanto las restricciones obligatorias como los criterios de equidad. Esto se traduce en soluciones consistentes, válidas y alineadas con las necesidades reales del entorno académico.

Por otra parte, la arquitectura web empleada basada en tecnologías modernas y contenedores Docker, aseguró un sistema estable, escalable y fácil de mantener, con capacidad de adaptación a futuros cambios institucionales.

En conjunto, los resultados evidencian que la solución propuesta optimiza de manera significativa la gestión de horarios docentes, mejora la eficiencia administrativa, fortalece la transparencia y equidad del proceso de planificación académica.

Conflicto de interés

Los autores declaran que no existe conflicto de interés en la redacción de este artículo.

Participación de los autores

La participación de los autores se define de la siguiente manera: CG y JA; conceptualización y diseño del estudio: CG y JA; metodología: CG y JA; desarrollo del software: CG, JA. Validación y ejecución de las pruebas: CG y JCH; análisis formal de los datos: CG, JA; investigación: CG, JA, JCH; redacción, revisión y edición: CG, JA y JCH; supervisión y asesoría técnica: JCH. Todos los autores han revisado y aprobado la versión final del manuscrito para su publicación.

Nota aclaratoria

Este artículo es un derivado de la tesis titulada “Prototipo de una Aplicación Web para la Asignación Automática de Horarios Docentes mediante Inteligencia Artificial”.

Referencias bibliográficas

- Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023). Meta-heuristic approaches for the University Course Timetabling Problem. *Intelligent Systems with Applications*, 19, 200253. <https://doi.org/10.1016/j.iswa.2023.200253>
- Agbolade, S. J., Ayinla, B. I., Odeniyi, L. A., & Akinola, S. O. (2024). Optimisation of University Examination Timetable Using Hybridised Genetic and Greedy Algorithms: A Case Study of Computer Science Department, University of Ibadan. *International Journal of Computer*, 51(1), 1–16. <https://ijcjournal.org/InternationalJournalOfComputer/article/view/2230>
- Ahumada Ahumada, J. A. (2014). *Generación de horarios académicos en INACAP utilizando algoritmos genéticos* [Tesis de Magister, Universidad de Chile]. <https://repositorio.uchile.cl/handle/2250/131461>

- Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A Review of Optimization Algorithms for University Timetable Scheduling. *Engineering, Technology & Applied Science Research*, 10(6), 6410–6417. <https://doi.org/10.48084/etasr.3832>
- Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86, 43–59. <https://doi.org/10.1016/j.cie.2014.11.010>
- Bensky, T., & Saunders, K. (2025). Use of a genetic algorithm in university scheduling for equitable and efficient determination of teaching assignments. In *Department of Physics, California Polytechnic State University*. <https://doi.org/10.48550/arXiv.2509.06981>
- Ceschia, S., Di Gaspero, L., & Schaerf, A. (2023). Educational timetabling: Problems, benchmarks, and state-of-the-art results. *European Journal of Operational Research*, 308, 1–18. <https://doi.org/10.1016/j.ejor.2022.07.011>
- Diallo, F. P., & Tudose, C. (2024). Optimizing the Scheduling of Teaching Activities in a Faculty. *Applied Sciences*, 14(20), 9554. <https://doi.org/10.3390/app14209554>
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- Lewis, R., & Thompson, J. (2015). Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. *European Journal of Operational Research*, 240, 637–648. <https://doi.org/10.1016/j.ejor.2014.07.041>
- Liao, J., Zheng, J., & Chiru-Figueroa, J. (2026). Evaluación del rendimiento de un prototipo de aplicación web con mecánica gacha basado en microservicios para la promoción de la cultura panameña. *Revista Colón Ciencias, Tecnología y Negocios*, 13(1), 90–108. <https://doi.org/10.48204/j.colonciencias.v13n1.a9113>
- Lulu, I., Alowais, H., Turkey, A., Harous, S., & Hussain, A. (2024). Efficient Solution for Exam Timetabling Problem: A Case Study of the University of Sharjah. *2024 17th International*

Conference on Development in ESystem Engineering (DeSE), 101–106.

<https://doi.org/10.1109/DeSE63988.2024.10911922>

Ramadhani, P., Nasir, A., Abdullah, Z. M., Novianty, A., & Setianingsih, C. (2025). Telkom Bandung Vocational School Scheduling Application Using a Website-Based Genetic Algorithm. *CEPAT Journal of Computer Engineering: Progress, Application and Technology*, 2, 80–91. <https://doi.org/10.25124/cepat.v3i01.6709>

Salmasnia, A., Mokhtari, H., & Abadi, I. (2011). A robust scheduling of projects with time, cost, and quality considerations. *The International Journal of Advanced Manufacturing Technology*, 60(1–4), 631–642. <https://doi.org/10.1007/s00170-011-3627-5>

Valverde Calderón, L. A. (2019). *Sistema basado en algoritmo genético para mejorar la gestión académica de horarios universitarios en el año 2019* [Universidad César Vallejo]. <https://hdl.handle.net/20.500.12692/41736>

Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T., & Du, Z. (2024). Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 252(Part A), 124167. <https://doi.org/10.1016/j.eswa.2024.124167>

Yakovlev, S., Kartashov, O., & Yarovaya, O. (2018). On Class of Genetic Algorithms in Optimization Problems on Combinatorial Configurations. *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 374–377. <https://doi.org/10.1109/STC-CSIT.2018.8526746>

Zhu, K., Li, L. D., & Li, M. (2021). A Survey of Computational Intelligence in Educational Timetabling. *International Journal of Machine Learning and Computing*, 11, 40–47. <https://doi.org/10.18178/ijmlc.2021.11.1.1012>