

Conference Paper

Application of Genetic Algorithms Technique in the Generation of Academic Schedules

Aplicación de la Técnica de Algoritmos Genéticos en la Generación de Horarios Académicos

Jason Gómez, Gloria Arcos-Medina, and Danilo Pástor

Escuela Superior Politécnica de Chimborazo. Facultad de Informática y Electrónica, Riobamba, Ecuador

Abstract

This paper presents an application to solve the problem of low efficiency in the generation of academic schedules for an educational institution through metaheuristic techniques. The agile SCRUM methodology was used for the planning and development of the system, in addition a sample of four processes for the evaluation of system efficiency was determined. In the planning phase, 15 user stories, 13 technical stories and 19 system sprints were identified. In the development phase the design of the architecture, interfaces, database was carried out. The process for working the genetic algorithms in the generation of schedules through the processes of initialization, selection, crossing, mutation and recombination was carried out until compliance with an objective function thus obtaining an academic schedule with the conditions that the educational institution requires. To evaluate the efficiency of the system, the ISO / IEC 25010 standard was used through the sub-characteristics of time behavior and resource utilization. The results obtained in time behavior reveal that with the use of the system it is reduced by 99.12%, compared to the manual processes thus showing a total efficiency of 93.75%.

Resumen. Este trabajo presenta una aplicación para resolver el problema de la poca eficiencia en la generación de horarios académicos para una institución educativa a través de técnicas metaheurísticas. Se utilizó la metodología ágil SCRUM para la planificación y desarrollo del sistema, además se determinó una muestra de cuatro procesos para la evaluación de eficiencia del sistema. En la fase de planificación se identificaron 15 historias de usuario, 13 historias técnicas y 19 sprints del sistema. En la fase de desarrollo se realizó el diseño de la arquitectura, interfaces, base de datos, además se señaló el proceso de como los algoritmos genéticos trabajan en la generación de horarios mediante los procesos de inicialización, selección, cruce, mutación y recombinación hasta cumplir con una función objetivo obteniendo así un horario académico con las condiciones que la institución educativa requiere. Para evaluar la eficiencia del sistema se utilizó la norma ISO/IEC 25010 a través de las subcaracterísticas de comportamiento de tiempos y utilización de recursos. Los resultados obtenidos en cuanto a tiempos revelan que con el uso del sistema se

Corresponding Author:

Jason Gómez

jasongomez71@yahoo.es

Received: 24 December 2019

Accepted: 2 January 2020

Published: 8 January 2020

Publishing services provided by
Knowledge E

© Jason Gómez et al. This article is distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Selection and Peer-review under the responsibility of the SIIPRIN-CITEGC Conference Committee.

OPEN ACCESS

reduce en un 99.12% en comparación a los procesos manuales, mostrando así una eficiencia total del 93.75%.

Keywords: Time Generation, Genetic Algorithms, Metaheuristic Techniques, Software Development, Efficiency.

Palabras clave: Generación de Horarios, Algoritmos Genéticos, Técnicas Metaheurísticas, Desarrollo de Software, Eficiencia.

1. Introducción

La generación de horarios académicos en una institución es un proceso administrativo que consiste en asignar a un docente una programación semanal, a cada uno de los cursos, esto se lo hace cada inicio del periodo académico ya que muchas veces la información de los periodos anteriores no se puede utilizar debido a ciertos cambios que se puedan presentar [1]. En cada periodo académico se realiza un horario académico para los docentes con lineamientos y restricciones establecidas en un determinado tiempo (una semana) [2]. Al no considerarse un proceso simple este lo transforman en un problema de optimización ya que consiste en asignar recursos y estos se los debe organizar de la mejor manera [1].

Existen varias técnicas metaheurísticas utilizadas para resolver problemas de optimización y búsqueda. En la revisión de literatura se han encontrado estudios que utilizan el algoritmo de búsqueda tabú, para la resolución del problema de programación de horarios universitarios y proponen un modelo matemático que provoca que la implementación resulte compleja [5]. Otra propuesta, [6] realiza el diseño de un modelo matemático, y plantea una técnica de solución mediante la selección de una búsqueda aleatoria, con la ayuda de operadores de la búsqueda tabú. En el estudio realizado por [7], se propone una variante de los criterios de aspiración utilizado por el algoritmo de búsqueda Tabú la cual implementa una función de probabilidad al momento de evaluar un elemento clasificado como tabú, esta propuesta demostró un alto nivel de confiabilidad de este algoritmo, en relación a los criterios existentes.

En cuanto a la literatura relacionada con colonias de hormigas, [8] demuestra en su estudio que se puede utilizar este tipo de técnica para resolver el problema de asignación de salones y horarios para estudiantes matriculados a través de un modelo

matemático, pero también se menciona que es difícil obtener una solución mejorada a la que se obtuvo, además de tener una demora considerable para generar la solución.

Para el desarrollo de este sistema web de generación de horarios se ha tomado en cuenta ciertas características importantes encontradas en trabajos analizados que han sido realizados mediante algoritmos genéticos en donde se implementan elementos de selección, recombinación y mutación de los algoritmos genéticos pero con distintos lineamientos como tomar en cuenta horas libres al momento de generar los horarios [3]. Otro autor utiliza los algoritmos genéticos combinándolos con ritmos cognitivos permitiendo que ciertas asignaturas se las considere dictar en ciertos intervalos de tiempo específicos [4]. Finalmente, en el trabajo [2] se propone realizar un sistema mucho más complejo y adopta en su artículo la distribución de salones, distribución de cursos, disponibilidad horaria y académica de los docentes pretendiendo que el sistema pueda ser utilizado para cualquier tipo de institución.

El objetivo que tiene el presente artículo es de desarrollar un sistema web para asignación de horarios académicos a través de una técnica metaheurística, de tal forma que se pueda optimizar el proceso en función del rendimiento, permitiendo de esta manera que las instituciones mejoren sus procesos que actualmente los llevan de forma manual.

El resto del documento está organizado de la siguiente manera: la sección 2 describe el marco teórico que ayuda a fundamentar el presente trabajo; la sección 3 especifica la propuesta metodológica de diseño del algoritmo genético y la forma de validar su eficiencia; la sección 4 realiza un análisis de los resultados obtenidos de la evaluación de la eficiencia del algoritmo, y finalmente la sección 5 muestra las conclusiones del presente trabajo y los posibles trabajos futuros.

2. Marco Teórico

2.1. Técnicas metaheurísticas

Hablar de metaheurística es referirse a técnicas inteligentes que tratan de aportar soluciones de calidad a un problema empleando recursos con un buen rendimiento [9].

Las técnicas metaheurísticas son diseñadas para resolver problemas de optimización y búsqueda, en los que los heurísticos clásicos no son efectivos [10]. Parten de una solución inicial y a medida que el algoritmo se vaya ejecutando se va optimizando la solución [1].

Entre las ventajas más destacables que plantea [4] que tiene gran éxito en la práctica, son algoritmos de propósito general y pueden ser fácilmente implementados. También presenta ciertas desventajas ya que son algoritmos aproximados mas no exactos, son probabilísticos y las bases teóricas no son estrictamente establecidas.

2.1.1. Clasificación de las técnicas metaheurísticas

A. Metaheurísticas basadas en trayectorias

Las metaheurísticas basadas en trayectorias son las que utilizan una sola solución en el proceso de búsqueda es decir empiezan con una solución de partida y terminan la trayectoria hasta encontrar una solución final [5]. En este tipo de metaheurísticas podemos encontrar:

Algoritmo de búsqueda tabú

Es un proceso metaheurístico que se distingue de los demás por utilizar estrategias especiales y memoria distintiva para la solución de problemas, además se basa en un proceso secuencial hasta cumplir con la condición de parada dispuesta [6].

GRASP (Greedy Randomize Adaptative Search Procedure)

Es una técnica creada para la resolución de problemas difíciles de optimización combinatoria, se basa en 2 fases principales que son: la fase de construcción y la fase de mejoramiento [2].

B. Metaheurísticas basadas en poblaciones

Son técnicas que utilizan un conjunto de soluciones que se optimizan simultáneamente en el transcurso de la búsqueda. Se pueden reconocer dos aspectos comunes en los algoritmos heurísticos basados en la población: exploración y explotación. La exploración es la capacidad de expandir el espacio de búsqueda, donde la explotación es la capacidad de encontrar los óptimos en torno a una buena solución. Desde un punto de vista diferente, los miembros de un algoritmo de búsqueda basado en la población pasan tres pasos en cada iteración para realizar los conceptos de exploración y explotación: autoadaptación, cooperación y competencia [18].

Algoritmo de colonia de hormigas

Están basados en una colonia de hormigas artificial que trabajan de manera conjunta y por cada iteración fabrican una solución al problema. Contiene aristas que representan las posibles opciones que tienen las hormigas y cuentan con 2 tipos de información: heurística que es la que no puede ser modificada mientras el algoritmo se está ejecutando y la información de los rastros de feromonas artificiales que puede ser modificada dependiendo de las soluciones que vayan encontrando las hormigas [7].

Algoritmos genéticos

Son aquellos sistemas que operan sobre una población o conjunto de soluciones representadas como cromosomas. Durante la ejecución, el algoritmo cruza los individuos de mayor aptitud para renovar la población y elimina los de menor aptitud. Al final, la cadena de mayor aptitud es la solución al problema [8].

2.1.2. Estudio comparativo entre algoritmos metaheurísticos

Dentro de las técnicas metaheurísticas basadas en trayectorias y basadas en poblaciones se pudo observar 4 técnicas que son: GRASP, búsqueda tabú, colonia de hormigas y algoritmos genéticos que se muestra en la Tabla 1, en donde se realiza una comparación entre las características de cada una de ellas.

Los algoritmos metaheurísticos presentan características importantes en su funcionamiento de los cuales se han tomado 4 para compararlos como se puede observar en la tabla 1, en donde se selecciona a los algoritmos genéticos frente a las otras 3 técnicas expuestas ya que, en su funcionamiento adoptan criterios favorables al rendimiento al momento de implementarlos en el sistema, por ejemplo en la adaptación donde los algoritmos genéticos no necesitan de conocimientos previos para la encontrar la solución al problema, en la adecuación de parámetros resulta más simple para codificarlos e implementarlos y a diferencia del resto de algoritmos, además respecto a las características de la cantidad de soluciones y su tiempo de solución, posee varias soluciones en menores tiempos.

Según [2], se presenta un cuadro de calificaciones de las técnicas metaheurísticas de acuerdo a distintas características, puntuando entre 0 y 1 cada una de ellas. 0 equivaldrá al puntaje más bajo y 1 el más alto como se muestra en la Tabla 2.

TABLE 1: Comparación de los algoritmos metaheurísticos.

Características	Algoritmos			
	GRASP	Búsqueda Tabú	Colonia de hormigas	Algoritmos genéticos
Año de creación	1989	1986	1996	1975
Adaptación	Se adapta acorde a las condiciones del problema.	Requiere una solución inicial	Requiere de una solución inicial	No necesitan de conocimientos previos del problema a resolver.
Tiempo de solución	Alto	Bajo	Alto	Bajo
Determinístico/ Aleatorio	Aleatorio	Determinístico	Aleatorio	Aleatorio
Adecuación de parámetros	Compleja	Compleja	Medio	Simple
Uso de memoria	No	Si	Si	No
Basada en procesos	Ninguno	Ninguno	Biológicos	Biológicos
Soluciones que maneja	1	1	N soluciones	N soluciones

TABLE 2: Cuadro comparativo valorado de las características de las técnicas metaheurísticas.

Característica	GRASP	Tabú	Colonia de hormigas	Algoritmos genéticos
Simplicidad	0.75	0.75	0.5	1
Independencia	1	1	1	1
Coherencia	1	0.75	0.75	1
Efectividad	0.75	1	1	1
Eficiencia	0.5	0.75	1	1
Adaptabilidad	0.75	1	1	1
Robustez	0.75	0.75	0.75	0.75
Interactividad	0.75	0.75	0.75	0.75
Autonomía	1	1	1	1
Diversidad	0.5	0.5	1	1
Total	7.75/10	8.25/10	8.75/10	9.5/10

De acuerdo con el estudio comparativo realizado en la tabla 2 se puede interpretar que la técnica metaheurística óptima a utilizar en la implementación del sistema son los algoritmos genéticos ya que obtienen la puntuación más alta en comparación a las otras técnicas; entre sus características contempladas como las más significativas se tiene: la simplicidad al momento de ajustar los parámetros al sistema, efectividad al generar las soluciones del problema, adaptabilidad a distintas plataformas de desarrollo e independencia ya que no necesita de otros algoritmos para ser implementados.

3. Metodología

3.1. Metodología de desarrollo de software

La metodología que se empleó para el desarrollo del sistema es la metodología de gestión de proyectos del desarrollo de software denominado SCRUM, es una metodología ágil, adaptable y flexible que permite realizar cambios en cada ciclo de trabajo y las tareas se realizan en base a lo planificado para cada uno de los miembros del grupo de trabajo de manera independiente, el cual trata sobre una planificación del desarrollo del proyecto.

SCRUM presenta 3 fases para el desarrollo del software que son: Fase de Planificación en donde se establecieron 19 sprints, 15 historias de usuario y 13 historias técnicas. Fase de Desarrollo se aplica el procedimiento para el desarrollo de horarios académicos a través de algoritmos genéticos que se derivan de las técnicas metaheurísticas y la Fase de Cierre, donde se evalúa el trabajo realizado.

3.2. Algoritmo genético

Por otra parte, para llegar a obtener una solución óptima aplicando técnicas metaheurísticas, se siguieron ciertos pasos que definen a los algoritmos genéticos, a continuación, se explica cómo se trabaja mediante un proceso algorítmico de los estos algoritmos (Ver Figura 1):

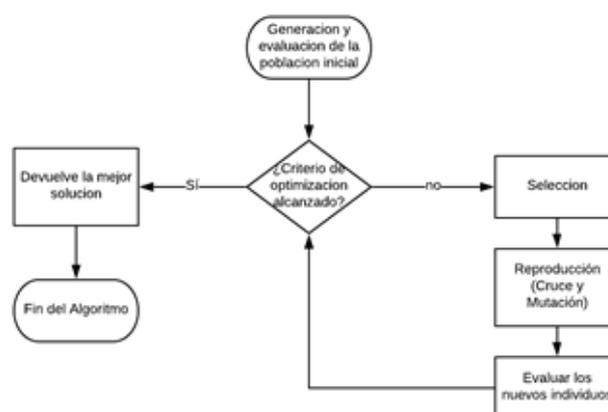


Figure 1: Diagrama de Flujo de Algoritmos Genéticos.

1. Se genera una población inicial aleatoria de n-cromosomas
2. Se evalúa la función de aptitud de cada cromosoma x de la población

3. Si la condición de término está satisfecha, se detiene el algoritmo, se devuelve la mejor solución de la población actual y se va al paso 7.
4. Se genera una nueva población repitiendo los siguientes pasos, hasta que se cumple la condición de parada.
5. Se selección los 2 cromosomas padres, de una población, según su aptitud
6. Con una probabilidad de emparejamiento, los padres se emparejan para formar a un nuevo descendiente. Si no se realiza emparejamiento alguno, el descendiente es la copia exacta de los padres.
7. Se realiza una mutación por cambios, el nuevo descendiente muta en alguna posición de su cromosoma.
8. La nueva población generada es aplicada para la nueva iteración del algoritmo
9. Vuelve al paso 2
10. Fin del algoritmo

3.2.1. Inicialización

El Algoritmo Genético comienza con la generación de una población inicial de horarios. En la representación utilizada cada elemento de la población consiste en dos genes, uno representa el horario con las materias de la clase y el otro el horario con los docentes de la clase.

Cada gen es una matriz de dos dimensiones donde las filas corresponden a las horas determinadas de las clases y las columnas a los días de la semana. La intersección de una fila y una columna contiene la materia y el profesor que son requeridos en una hora de clase en particular. Esta representación se muestra en la Tabla 3.

TABLE 3: Representación del Horario.

HORA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
1	(M1, D1)	(M1, D1)	(M7, D7)	(M2, D2)	(M4, D4)
2	(M2, D2)	(M3, D3)	(M4, D4)	(M4, D4)	(M4, D4)
3	(M2, D2)	(M3, D3)	(M3, D3)	(M7, D7)	(M6, D6)
4	(M4, D4)	(M6, D6)	(M5, D5)	(M5, D5)	(M5, D5)
5	(M3, D3)	(M7, D7)	(M8, D8)	(M5, D5)	(M8, D8)
6	(M8, D8)	(M5, D5)	(M8, D8)	(M8, D8)	(M1, D1)
7	(M5, D5)	(M2, D2)	(M1, D1)	(M8, D8)	(M2, D2)

Así, por ejemplo: La Materia 1 (M1) y el docente 1 (D1) son requeridos en la primera hora el día lunes. Esta representación asegura que no habrá conflictos en cada uno de los cursos ya que cada uno de los requerimientos de las materias y los docentes se los tratará por separado. Se realiza primero una lista de requerimientos para proceder a la construcción de los horarios. Cada posición es una tupla Materia-Docente. Si el Docente debe encontrarse con un curso n veces a la semana, la tupla curso-materia-docentes aparecerá n veces en la lista de requerimientos.

La población inicial es generada de forma aleatoria, produciendo así n horarios diferentes. El algoritmo va ubicando cada tupla materia-docente de la lista de requerimientos de acuerdo al grado de saturación, por ejemplo: el número de veces que la tupla puede ser ubicada en una posición del horario sin que haya conflictos. El algoritmo va comparando que las materias y los docentes cumpla con la función de aptitud y se detiene ya sea por haber encontrado la solución factible o porque se haya alcanzado el número máximo de generaciones.

3.2.2. Selección

Si el problema tiene restricciones débiles al igual que restricciones duras solo se utiliza el criterio de terminación. La adaptación de un individuo es el número de restricciones violadas en el horario programado. En las restricciones no debe haber conflictos de materias y docentes en los cursos. Además, la adaptación del horario es la suma de los conflictos de materias y docentes que tienen lugar en el horario generado. Para el proceso de selección se utilizará el método de selección por torneo. Este método empieza eligiendo al azar t elementos de la población, pero después se van seleccionando y comparando los individuos por pares para encontrar al individuo que mejor se adapta.

3.2.3. Cruce

Una vez ya generados aleatoriamente los horarios de los cursos, el algoritmo permite que este operador de cruce vaya mezclando el material genético de materias y docentes y a su vez, ir generando nuevas soluciones hasta que se encuentre una solución factible de horario académico.

3.2.4. Mutación

El operador de mutación escoge al azar cuando se realiza un cambio en el horario de una clase. El cambio se realiza ubicándose en una posición en el horario con el que algún docente tenga conflicto debido a la falta de la programación de una clase, luego se ubica en otra fila de la clase en donde al docente se le haga conveniente cambiar su hora de clase. Así, el intercambio se lo puede realizar ya sea por filas o columnas (Ver Figura 2). Este tipo de cambio que se realiza se lo llama mutación por cambios.



Figure 2: Mutación por cambios.

Una vez realizados los procesos de los operadores genéticos evalúa los nuevos individuos y procede con el reemplazamiento de la nueva solución generada.

3.3. Determinación de la eficiencia

Para medir la eficiencia del sistema se hizo mediante el uso de la norma ISO/IEC 25010, tomando en cuenta las sub-características de comportamiento de tiempos y utilización de recursos.

Para ello se tomaron en cuenta 4 procesos como muestras que son los procesos con los que más interactúan el usuario. Estos son:

1. Agregar las materias que va a dictar un docente.
2. Agregar materias a cada curso.
3. Generar que los horarios sin cruces de materias y docentes.
4. Ajustar el horario en alguna hora que desee cambiar el docente.

En base a estos 4 procesos se procedió a medir la eficiencia del sistema.

Para la evaluación de las métricas de comportamiento de tiempos y de la utilización de recursos se hizo uso de fichas técnicas, de acuerdo a [16] la finalidad de las fichas para es que el usuario se sienta conforme con los resultados obtenidos de cada uno de los procesos que se ejecutan en el sistema para la generación de horarios académicos. A continuación, se muestra la ficha técnica en la Tabla 4.

TABLE 4: Ficha técnica para medir la eficiencia.

Nombre	Determina el nombre de la métrica escogida.
Propósito	Especifica el fundamento por el cual se escogió la métrica.
Método de la aplicación	Proporciona un perfil de aplicación.
Medición, Fórmula	Provee una fórmula de medición e interpreta los datos que se van a utilizar.
Interpretación	Brinda el valor y el rango preferido y recomendado.
Tipo de escala	Determina el tipo de escala usado para la métrica.
Tipo de medida	Determina el tipo de medida que se va a elegir.
Fuente de medición	Determina la fuente de datos usados en la medición.
ISO/IEC 12207 SLCP	Determina el proceso o procesos del ciclo de vida del software donde la métrica es aplicable.
Audiencia	Determina el tipo de usuario para analizar la métrica.

4. Resultados

En este apartado se presenta los resultados obtenidos mediante los parámetros de *comportamiento de tiempos y de utilización de recursos*.

4.1. Comportamiento de tiempo

Este parámetro se lo mide con el objetivo de ir obteniendo los tiempos de respuesta de los procesos cuando se los hace manualmente y con el sistema web, obteniendo así resultados que nos permitan verificar si existe una disminución de tiempo en los procesos que se realizan para la generación de horarios académicos.

Para llevar a cabo el análisis del tiempo de respuesta se utilizó el cronometro y se tomaron en cuenta los procesos que más inconvenientes causan en la institución al momento de realizar los horarios académicos.

Los resultados se pueden observar en la tabla 5, en la que se muestra que con la utilización del sistema se logró una disminución considerable del tiempo. Cabe indicar, que la medida de tiempo fue tomada en minutos y se la efectuó de forma manual. Todos los procesos que están en la tabla 5 se demoran en total 540 minutos, los cuales serán

considerados como el 100% para realizar la debida comparación y sacar el porcentaje con el que ayudaría a mejorar el proceso de asignación de horarios con el sistema web. Los tiempos que se obtuvo con el sistema son de 4.74 minutos lo que representa un 99.12% de reducción de los tiempos a comparación de los procesos manuales.

TABLE 5: Tiempos de respuesta de procesos manuales de generación de horarios.

Tarea	Sin sistema (minutos)	Con sistema (minutos)	Reducción de tiempos con el sistema
Agregar materias que va a dictar un docente	90	1.59	88.41
Agregar materias a un curso	30	1.76	28.24
Generar los horarios sin cruces de materias ni docentes	240	1.24	238.76
Ajuste de horas de clase del docente en el horario académico	180	0.15	179.85
TOTAL	540	4.74	535.26

4.2. Utilización de recursos

4.2.1. Uso de la memoria RAM

Para la medición del uso de la memoria RAM se lo hizo con una unidad de medida en MB a través del administrador de tareas con el que cuenta el computador.

Para la calificación de esta subcaracterística se tomará en cuenta que mientras menos se consuma memoria más alta será la calificación establecida. A continuación, en las Tabla 6 y 7 se presenta los indicadores de evaluación y sus resultados.

TABLE 6: Indicador de evaluación para el uso de memoria RAM.

Calificación	Tiempo	Valor cuantitativo
100%	[0-150] MB	Excelente
90%	[151-250] MB	Muy bueno
75%	[251-350] MB	Bueno
50%	[351-450] MB	Aceptable
20%	[451-550] MB	Regular
0%	[551-∞] MB	Malo

TABLE 7: Resultados del uso de memoria RAM.

Tarea	Uso de la memoria RAM (MB)
Agregar materias que va a dictar un docente	130.5
Agregar materias a un curso	129.8
Generar los horarios sin cruces de materias ni docentes	132.4
Ajuste de horas de clase del docente en el horario académico	148.1
TOTAL	540.8
PROMEDIO	135.2

Analizando la Tabla el promedio obtenido en esta subcaracterística de uso de memoria RAM que es de 135.2MB podemos constatar que se encuentra en el rango de 0-150 adoptando así una calificación de 100% y un valor cuantitativo de Excelente.

4.2.2. Uso del procesador

Para la calificación de esta subcaracterística se tomará en cuenta que mientras menos se utilice el procesador más alta será la calificación establecida. A continuación, en la Tabla 8 y 9 se presenta los indicadores de evaluación y sus resultados.

TABLE 8: Indicador de evaluación para el uso del procesador.

Calificación	Tiempo	Valor cuantitativo
100%	[0-0.5] %	Excelente
90%	[1-1.5] %	Muy bueno
75%	[1.6-2.5] %	Bueno
50%	[2.6-3.5] %	Aceptable
20%	[3.6-4.5] %	Regular
0%	[4.6-∞] %	Malo

TABLE 9: Resultados del uso del procesador.

Tarea	Uso del procesador (%)
Agregar materias que va a dictar un docente	0.7
Agregar materias a un curso	2.6
Generar los horarios sin cruces de materias ni docentes	1.4
Ajuste de horas de clase del docente en el horario académico	1.7
TOTAL	6.4
PROMEDIO	1.6

Analizando el promedio obtenido en esta subcaracterísticas de uso del procesador que es de 1.6% podemos constatar que se encuentra en el rango de 1.6-2.5 adoptando así una calificación de 75% y un valor cuantitativo de Bueno.

A continuación, en la Tabla 10 se señalan todos los valores reales recopilados mediante la métrica de la eficiencia junto con sus subcaracterísticas de comportamientos de tiempo y utilización de recursos.

TABLE 10: Resumen de resultados recopilados.

Característica	Subcaracterística	Métrica	Ponderación	% medido	% con Ponderación
Eficiencia	Tiempo de respuesta	Tiempo de respuesta	50%	100%	50%
	Utilización de recursos	Uso de memoria RAM	25%	100%	25%
		Uso del procesador	25%	75%	18.75
Promedio			100%	93.75%	93.75%

Como se puede observar los porcentajes que se mostraron en la tabla XI que arrojaron como resultado el 93.75% de efectividad del sistema web, estos resultados se dieron debido a que en la métrica del uso del procesador no pudo conseguir una buena calificación ya que se obtuvo un valor cuantitativo de Bueno.

Es por esto que el sistema no obtuvo un nivel de eficiencia deseado del 100% pero por los resultados mostrados el sistema presenta un nivel Excelente de eficiencia como se puede observar en el indicador de evaluación de la Tabla 12 ya que se encuentra en el rango del 91-100%.

TABLE 11: Indicadores de evaluación para la eficiencia total del sistema.

Calificación	Valor cuantitativo
91-100%	Excelente
74-90%	Muy bueno
75%	Bueno
50%	Aceptable
20%	Regular
0%	Malo

5. Conclusiones

Se analizaron e identificaron los procesos que actualmente se llevan a cabo en la institución educativa para la generación de horarios académicos demostrando que el

sistema permite realizar estos procesos de manera automatizada beneficiando así a la institución con un sistema eficiente evitando inconvenientes de tiempos y recursos.

Se analizaron 4 técnicas metaheurísticas que son apropiadas para aplicar en el desarrollo de sistemas de generación de horarios académicos las cuales permiten hallar soluciones aproximadamente exactas y en un tiempo óptimo.

Se escogieron los algoritmos genéticos frente a otras 3 técnicas metaheurísticas como los más adecuados ya que de acuerdo a la mayor parte de características como adaptabilidad, efectividad, simplicidad, etc., demuestran con una diferencia considerable que son los óptimos para utilizar en el desarrollo del sistema.

Para el desarrollo del sistema usando la metodología SCRUM se obtuvo un total de 19 sprints, 15 historias de usuario y 13 historias técnicas.

En la evaluación de comportamiento de tiempos con el sistema se redujo un 99.12% en comparación a los procesos manuales que se realizan en la institución. En la evaluación para la utilización de recursos en cuanto al uso de memoria se obtuvo un promedio de 135.2 MB recayendo en el rango de [0-150]MB obteniendo un valor cuantitativo de Excelente y el uso del procesador se obtuvo un promedio de 1.6% recayendo en el rango de [1.6-2.5]% obteniendo un valor cuantitativo de Bueno.

Para evaluar la eficiencia total se la realizó a través de la norma ISO/IEC 25010 la cual se obtuvo un nivel de eficiencia excelente con el 93.75% del sistema web de generación de horarios académicos de la institución educativa.

References

- [1] J. A. Ahumada, "Generación de horarios académicos en INACAP utilizando algoritmos genéticos," 2014.
- [2] M. Andres, G. Cubillos, E. Hamid, P. Quiroga, R. Emilio, and S. Ruiz, "Problema del school timetabling y algoritmos genéticos" pp. 259--276, 2013.
- [3] L. Nuñez, "Aplicación de algoritmos genéticos en la generación de horarios de clase en la Escuela de Enfermería de la Universidad Nacional de Chimborazo," 2016.
- [4] O. D. Castrillón, "Combinación entre algoritmos genéticos y aleatorios para la programación de horarios de clases basado en ritmos cognitivos," vol. 25, no. 4, pp. 51--62, 2014.
- [5] C. D. Oliva and G. Ramírez, "Algoritmo de tipo búsqueda tabú para un problema de programación de horarios universitarios vespertinos," *Inge Cuc*, vol. 9, no. 2, pp. xx--xx, 2013.

- [6] M. D. Lozano, "Diseño de un algoritmo para realizar la programación de horarios de la carrera de Ingeniería Industrial de la Pontificia Universidad Javeriana," 2016.
- [7] O. Chavez, P. Pozos, and J. L. Gomez, "Aspiración probabilístico aplicada a la generación de horarios escolares," vol. 22, no. 1, pp. 153--177, 2015.
- [8] J. Marin Lozada, D. L. Hoyos B, C. A. Peñuela, and J. J. Santa Chávez, "Herramientas heurísticas para la asignación óptima de horarios de clase," *Av. Investig. en Ing.*, vol. 10, no. 1, pp. 68--74, 2013.
- [9] J. A. M. Pérez, "Metaheurísticas?: Conceptos y propiedades," 2004.
- [10] F. Herrera, "Introducción a los algoritmos metaheurísticos introducción a los algoritmos," 2010.
- [11] N. Alancay, S. M. Villagra, and N. A. Villagra, "Metaheurísticas de trayectoria y poblacional aplicadas a problemas de optimización combinatoria," *Inf. Científicos - Técnicos UNPA*, vol. 8, no. 1, p. 202, 2018.
- [12] J. Gallart Suárez, F. A. Manchego, A. A. Nole, G. B. Nicho, and M. T. Anticono, "Generación inteligente de horarios empleando heurísticas GRASP con Búsqueda Tabú para la Pontificia Universidad Católica del Perú," no. November 2015, 2009.
- [13] C. A. Algarín Robles, "Optimización por colonia de hormigas: aplicaciones y tendencias," *Rev. Ing. Solidar.*, vol. 6, pp. 83--89, 2010.
- [14] M. César and J. A. Montoya, *Revista EIA.*, no. 8. Escuela de ingeniería de Antioquia, 2007.
- [15] J. V. Paredes and P. M. R. Basalo, "Estudio y aplicación de metaheurísticas y comparación con métodos exhaustivos," 2016.
- [16] A. A. Vivanco Villamar, "Evaluación de calidad del sistema integrado proyecto," p. 224, 2011.
- [17] E. C. VALENCIA ESPINOZA, "Desarrollo de una aplicación web para la gestión de la información de un centro odontológico usando el framework Zend.," p. 200, 2018.
- [18] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Inf. Sci.* 179, 2232--2248, (2009).