# GENERACIÓN AUTOMÁTICA DE RESÚMENES DE MÚLTIPLES DOCUMENTOS CON UN ENFOQUE HIPERHEURÍSTICO BASADO EN ALGORITMOS MEMÉTICOS

**ANEXOS**

**DIANA PILAR ASTUDILLO MEDINA**

**Director: Dr. (c) MARTHA ELIANA MENDOZA BECERRA**

**UNIVERSIDAD DEL CAUCA**
**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**
**DEPARTAMENTO DE SISTEMAS**
**GRUPO DE I+D EN TECNOLOGÍAS DE LA INFORMACIÓN**
**RECUPERACIÓN DE LA INFORMACIÓN**
**POPAYÁN, Abril 2013**

# Tabla de Contenido

ANEXO A - METODOLOGÍA RECOPILACIÓN DE REQUERIMIENTOS Y RESTRICCIONES DEL ALGORITMO ..... 1
1 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA ..... 2
1.1 DESCRIPCIÓN DEL PROBLEMA ..... 2
1.1.1 Estudio de las heurísticas de selección, reemplazo y búsqueda local ..... 2
1.1.2 Especificaciones a implementar del algoritmo obtenido con el enfoque hiperheurístico ..... 7
1.2 DEFINICIÓN DE LA ARQUITECTURA ..... 8
1.2.1 Hiperheurística ..... 8
1.2.2 Algoritmo Memético ..... 9
1.3 DEFINICIÓN DEL ENTORNO DE EVALUACIÓN ..... 10
1.3.1 Conjunto de documentos a utilizar ..... 10
1.3.2 Características del entorno ..... 11
ANEXO B - METODOLOGÍA DESARROLLO DE LA HIPERHEURÍSTICA ..... 12
2 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA RUP ..... 13
2.1 INICIACIÓN ..... 13
2.1.1 Diagrama general de casos de uso del sistema ..... 13
2.2 ELABORACIÓN ..... 13
2.2.1 Caso de uso de alto nivel ..... 14
2.2.2 Diagrama de clases. ..... 14
2.2.3 Arquitectura base: ..... 17
2.3 CONSTRUCCIÓN ..... 18
2.3.1 Ciclos de Desarrollo ..... 19
2.3.2 Casos de uso reales ..... 20
2.3.3 Pruebas de caja negra ..... 21
2.3.4 Modulo evaluación de resultados ..... 22
2.3.5 Módulo tabulación de resultados ..... 23
2.4 TRANSICIÓN ..... 24
ANEXO C –ALGORITMO MEMÉTICO PARA GENERACIÓN DE RESÚMENES ..... 25

3 EJEMPLO DEL ALGORITMO MEMÉTICO ....26
ANEXO D – SELECCIÓN DE FUNCIÓN OBJETIVO....29
4 SELECCIÓN DE FUNCIÓN OBJETIVO ....30
4.1 Comparación funciones objetivo ....30
ANEXO E – EXPERIMENTOS DE LA HIPERHEURÍSTICA ....32
5 EXPERIMENTOS ....33
5.1 Resultados de las experimentaciones....34
5.2 ESTADÍSTICAS DE LAS MEJORES CONFIGURACIONES ....35
ANEXO F – EXPERIMENTOS DE AFINACIÓN....40
6 AFINACIÓN PRELIMINAR DEL ALGORITMO MEMÉTICO ....41
6.1 AFINACIÓN DE PARÁMETROS PARA EL PRIMER CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL....41
6.2 AFINACIÓN DE PARÁMETROS PARA EL SEGUNDO CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL....42
6.3 RESULTADOS DE LA CONFIGURACIÓN INVERTIDA PARA EL PRIMER CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL ....42
6.4 RESULTADOS DEL SEGUNDO CONJUNTO DE HEURISTICAS DE BAJO NIVEL PARA DUC2005....43
6.5 MEJOR CONFIGURACIÓN PARA LOS CONJUNTOS DE DOCUMENTOS DE DUC2005 Y DUC2007 ....44
7 Bibliografía ....46

# LISTA DE TABLAS

Tabla 1. Ventajas y desventajas de los esquemas de selección, cruce y reemplazo .....4
Tabla 2. Ventajas, Desventajas y Aplicaciones de las búsquedas locales .....6
Tabla 3. Características documentos de DUC 2005 .....10
Tabla 4. Características documentos de DUC 2007 .....11
Tabla 5. Descripción de cada una de las clases .....17
Tabla 6. Caso de uso real: realizar evaluación de colección de documentos .....21
Tabla 7. Clases equivalentes módulo 1 .....21
Tabla 8. Batería de pruebas módulo 1 .....22
Tabla 9. Clases equivalentes módulo 2 .....22
Tabla 10. Batería de pruebas módulo 2 .....23
Tabla 11. Clases equivalentes módulo 3 .....23
Tabla 12. Batería de pruebas módulo 3 .....23
Tabla 13. Población .....26
Tabla 14. Padres seleccionados .....26
Tabla 15. Agentes hijo .....26
Tabla 16. Vecinos con distancia de hamming uno .....27
Tabla 17. Vecinos con distancia de hamming dos .....27
Tabla 18. Nuevo Agente .....27
Tabla 19. Grupo de agentes aleatorios .....27
Tabla 20.Nueva Población .....28
Tabla 21. Resultados comparación función objetivo .....31
Tabla 22. Conjuntos de heurísticas de bajo nivel .....33
Tabla 23. Combinaciones para realizar las experimentaciones .....34
Tabla 24. Resultados del primer conjunto de heurísticas de bajo nivel para DUC2005 .....34
Tabla 25. Resultados del primer conjunto de heurísticas de bajo nivel para DUC2007 .....34
Tabla 26. Resultados del segundo conjunto de heurísticas de bajo nivel para DUC2005 .....35
Tabla 27. Resultados del segundo conjunto de heurísticas de bajo nivel para DUC2007 .....35
Tabla 28. Mejores resultados de los conjuntos de heurísticas de bajo nivel .....35
Tabla 29. Estadísticas de la mejor configuración del primer conjunto de heurísticas de bajo nivel para DUC2005 .....36

Tabla 30. Estadísticas de la mejor configuración del primer conjunto de heurísticas de bajo nivel para DUC2007 ..... 37
Tabla 31. Estadísticas de la mejor configuración del segundo conjunto de heurísticas de bajo nivel para DUC2005 ..... 37
Tabla 32. Estadísticas de la mejor configuración del segundo conjunto de heurísticas de bajo nivel para DUC2007 ..... 38
Tabla 33. Mejores configuraciones con el primer conjunto de heurísticas de bajo nivel ..... 39
Tabla 34. Mejores configuraciones con el segundo conjunto de heurísticas de bajo nivel ..... 39
Tabla 35. Afinación de parámetros del primer conjunto de heurísticas de bajo nivel ..... 41
Tabla 36. Afinación de parámetros del segundo conjunto de heurísticas de bajo nivel ..... 42
Tabla 37. Afinación de parámetros con la configuración de DUC2007 ..... 42
Tabla 38. Nuevas configuraciones con el primer conjunto de heurísticas de bajo nivel ..... 43
Tabla 39.Resultados de las medidas de Rouge con la nueva configuración ..... 43
Tabla 40. Resultados de la configuración del segundo conjunto de heurísticas de bajo nivel sin afinación de parámetros ..... 43
Tabla 41. Nueva configuración con el segundo conjunto de heurísticas de bajo nivel ..... 43
Tabla 42.Resultados con la configuración del segundo conjunto de heurísticas de bajo nivel ..... 44
Tabla 43.Configuración de esquemas de bajo nivel para los conjuntos de documentos de DUC2005 y DUC2007 ..... 44
Tabla 44. Resultados con el intercambio de parámetros afinados ..... 45
Tabla 45. Comparación de resultados con las valores de las afinaciones de los parámetros originales e intercambio de parámetros afinados ..... 45

# LISTA DE FIGURAS

Figura 1. Estructura de los documentos de evaluación.....8
Figura 2. Arquitectura de la Hiperheurística para generar la configuración del Algoritmo Memético .....9
Figura 3. Arquitectura del Algoritmo Memético .....9
Figura 4. Diagrama de casos de uso del entorno hiperheurístico .....13
Figura 5. Casos de uso de alto nivel Hiperheurística .....14
Figura 6. Diagrama clases pre-procesamiento.....15
Figura 7. Diagrama clases Hiperheurística .....16
Figura 8. Arquitectura del Sistema. .....19

# ANEXO A - METODOLOGÍA RECOPILACIÓN DE REQUERIMIENTOS Y RESTRICCIONES DEL ALGORITMO

# 1 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA

La metodología para la parte teórica e investigativa correspondiente a la recopilación de requerimientos y restricciones del algoritmo involucra 3 etapas: Descripción del problema, Definición de la Arquitectura y Definición del entorno de evaluación.

## 1.1 DESCRIPCIÓN DEL PROBLEMA

Se hace el estudio de las heurísticas de alto nivel (ruleta y torneo probabilístico); las heurísticas de bajo nivel a utilizar en los esquemas de selección, cruce y reemplazo (que tengan en cuenta las particularidades del problema específico de generación automática de resúmenes de textos), y los algoritmos de búsqueda local iterativa y de vecindad para optimizar cada individuo de la población. Además, se realiza una investigación para determinar si las heurísticas de bajo y alto nivel ya se encuentran implementadas. Como resultado de esta fase se obtendrá un documento con las especificaciones de los algoritmos a implementar.

### 1.1.1 Estudio de las heurísticas de selección, reemplazo y búsqueda local

Para la selección de las heurísticas de: selección, cruce, y búsqueda local se realizó una tabla con las ventajas y desventajas de las mismas.

A continuación se presentan dos tablas correspondientes a: las ventajas y desventajas de los esquemas de selección, cruce y reemplazo (ver Tabla 1); y las ventajas, desventajas y algunas aplicaciones para las búsquedas locales (ver Tabla 2).

| Nombre | Ventajas | Desventajas |
|---|---|---|
| Selección por ruleta | No es posible escoger dos veces consecutivas el mismo elemento.<br>Puede ser forzado a que sea alta la probabilidad de que no sean elementos próximos en la población.<br>Permite que los mejores individuos sean elegidos con una mayor probabilidad [1].<br>Es un método sencillo. | A medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$) |
| Selección por torneo determinístico | Selecciona los mejores individuos [1]. | No tiene en cuenta a los individuos menos aptos.<br>No es muy útil cuando se utiliza una población grande, porque va a necesitar más tiempo para buscar el ganador del torneo en el grupo seleccionado al azar. |
| Selección por torneo Probabilístico | Esta técnica evita que los mejores individuos ganen dominio en comparación de los menos aptos, manteniendo la diversidad en la población [2].<br>Explorar nuevas soluciones del espacio de búsqueda. | . |
| Selección Uniforme | Tiende a seleccionar individuos aptos y menos aptos.<br>Mantiene la diversidad de los individuos que selecciona para la descendencia [3]. | Posible pérdida de rendimiento porque tiene que buscar el individuo más cercano con respecto al valor de aptitud seleccionado [4]. |
| Selección basada en el rango | Esta técnica evita que los individuos más aptos siempre sean elegidos a expensas de los menos aptos [1]. | Costo computacional extra ya que se debe reordenar la población. |
| Selección con emparejamiento restringido | Evita la convergencia prematura [1].<br>Preserva la diversidad de la población, escogiendo los mejores y peores agentes de la población.<br>Fomenta la creación de nuevas generaciones [5].<br>Explora nuevas soluciones del espacio de búsqueda. | |
| Cruce un punto | Es una técnica sencilla y clásica [6-8]. | Puede producir descendencia de baja aptitud en la primera generación. |
| Cruce dos puntos | Se conserva la cabeza y la cola del individuo padre. | Reduce el rendimiento del algoritmo. |

| | | |
|---|---|---|
| | Es mejor para poblaciones grandes, mayores o iguales a 50 [9]. | Está limitado al tamaño de la población. |
| Cruce uniforme | Cada gen de la descendencia tiene la misma probabilidad de pertenecer a uno u otro padre.<br>Permite variedad en los memes de un agente, porque selecciona aleatoriamente los memes de cada padre, para crear los descendientes [10, 11].<br>Explora nuevos memes en el espacio que está siendo buscado [12]. | |
| Reemplazo Peores Individuos | Disminuye las soluciones de baja calidad para mantener la calidad de la población [1].<br>La población mantiene las mejores soluciones generadas. | |
| Reemplazo por Competencia Restringida | Los nuevos individuos tienen mayor probabilidad de reemplazar individuos de la población similares a ellos según su genotipo [13]. | |
| Reemplazo Aleatorio | Diversidad de la nueva población [1]. | |
| Reemplazo de Padres | | Disminuye la calidad de la población si los descendientes tienen una aptitud baja [1]. |
| Reemplazo de Individuos Similares | Se mantendrá a la nueva población en un estado equilibrado [1]. | |

Tabla 1. Ventajas y desventajas de los esquemas de selección, cruce y reemplazo

| Nombre | Ventajas | Desventajas | Aplicaciones |
|---|---|---|---|
| Búsqueda Por vecindad [14] | Evitan quedarse en mínimos locales.<br>Escapa de un mínimo local cambiando de forma sistemática la estructura de entornos. | | Se ha aplicado a diversos problemas clásicos de la Inteligencia Artificial, entre ellos destacan los problemas de satisfacción, el aprendizaje en redes bayesiana, clasificación y planificación.<br>También su |

| | | | |
|---|---|---|---|
| | | | aplicación en problemas de optimización combinatoria, entre los que se encuentran los problemas de empaquetado, localización, p-mediana y de rutas. |
| Búsqueda Iterada [15] | Escapa de mínimos locales aplicando perturbaciones al mínimo local actual. | Si la perturbación es demasiado grande el efecto será similar a arrancar desde una nueva solución. | Asignación cuadrática (QAP), particionamiento de grafos, máxima satisfacción (MAX-SAT), y en el problema del árbol de Steiner restringido a grafos. |
| Búsqueda Local Guiada [16] | Estimula la diversificación de la búsqueda. | El ajuste de parámetros como la penalización y la regularización. | Ha sido aplicado exitosamente al problema del viajante del comercio (TSP), problemas de asignación de frecuencias de radio enlace (RLPA), enrutamiento de vehículos, entre otros. |
| Búsqueda Tabú [17] | Al utilizar una lista de los vecinos visitados, evita que se vuelvan a generar.<br>Las restricciones tabú y el criterio de aspiración juegan un papel dual en la restricción y guía del proceso de búsqueda. | El ajuste de parámetros, como el tamaño de la lista tabú, la elección de los movimientos que se deben registrar y la definición del criterio de aspiración.<br>La eficiencia depende cómo este modelado el problema. | Entre sus aplicaciones tradicionales se encuentran.<br>Problemas de teoría de grafos, localización y asignación, planificación y enrutamiento, entre otras.<br>También se usa para solucionar problemas de transporte, diseño, optimización de estructuras, etc. |

| | | | |
|---|---|---|---|
| Recocido Simulado [18] | Su naturaleza aleatoria permite convergencia asintótica a la solución óptima bajo condiciones moderadas. | La convergencia requiere de tiempo exponencial, convirtiendo el Recocido Simulado en no práctico como instrumento para procurarse soluciones óptimas. | Se ha aplicado a varios problemas de optimización combinatoria en los que se destacan los problemas de localización, empaquetado, asignación, localización, entre otros. |
| Búsqueda Voraz Adaptable Aleatoria [19] | Construye soluciones de alta calidad. | Falta de estructuras de memoria, para no volver a evaluar la solución visitada. No permite diversidad de la población. | Ha sido aplicada en problemas de enrutamiento, localización, árbol mínimo de Steiner, optimización en grafos, en áreas de manufactura, sistemas de potencia, estadística, programación matemática, entre otros. |

Tabla 2. Ventajas, Desventajas y Aplicaciones de las búsquedas locales

### 1.1.2 Especificaciones a implementar del algoritmo obtenido con el enfoque hiperheurístico

Se debe desarrollar una aplicación de escritorio la cual permita seleccionar los documentos a resumir, finalmente la aplicación debe entregar al usuario un resumen y una hoja de Excel donde estén los resultados de la evaluación del resumen.

#### *1.1.2.1 Algoritmo Memético (AM)*

**Entradas:**

- **Documentos:** Los documentos para la ejecución de las experimentaciones no tienen extensión y el usuario puede ingresar un conjunto de documentos o un grupo de conjuntos de documentos que serán procesados para la generación del resumen. El contenido de un documento está sujeto a la estructura que se muestra en la Figura 1, la cual es la dispuesta por DUC.

- **Parámetros**: El usuario debe ajustar en la parte del código los parámetros del algoritmo que haya seleccionado, si así lo desea, de lo contrario se ejecutará con los parámetros que se afinaron en las experimentaciones de laboratorio. Los parámetros requeridos para el algoritmo son: la probabilidad de optimización (PO), el tamaño de la población (TP), Lambda de la función objetivo (LF) y Máxima longitud del resumen (MLR).

- **Longitud del resumen deseado:** La longitud del resumen debe ser aproximadamente de 250 palabras para la realización de las experimentaciones, debido a que los resúmenes modelo proporcionados por DUC tienen estas longitudes, sin embargo el usuario tiene la posibilidad de realizar un resumen con una cantidad mayor o menor de palabras cambiando este valor en la parte del código.

**Salidas:** La salida del sistema es el resumen generado por el Algoritmo Memético. Y la evaluación, en la cual se compara el resumen generado con los resúmenes ideales proporcionados por DUC. Para esto, el sistema después de haber realizado el resumen, debe leer los resúmenes modelo y realizar la evaluación por medio de ROUGE 1.5.5, hay que tener en cuenta que ROUGE nos puede arrojar el promedio de todos los resúmenes modelo o el mejor resultado. Finalmente se obtiene una hoja de Excel donde están todos los valores de ROUGE para cada uno de los conjuntos de documentos.

**Condiciones:** Ejecución del AM cuya función objetivo consta de los factores Cobertura (FC) y Eliminación de Redundancia (FR) como se puede observar en la Ecuación 1:

$$f(x) = \lambda * FC - (1 - \lambda) * FR \qquad \text{Ecuación 1}$$

Donde λ, es un valor entre 0 y 1 .

Posteriormente los resúmenes obtenidos se evalúan por medio de ROUGE.

Para que el algoritmo se ejecute correctamente, se deben haber introducido todos los parámetros requeridos: PO, TP, LF y MLR.

```
<DOC>
        <HEADLINE>
        Titulo documento
        </HEADLINE>
        <TEXT>
                <P>
                Texto del documento
                </P>
        </TEXT>
</DOC>
```

Figura 1. Estructura de los documentos de evaluación

## 1.2 DEFINICIÓN DE LA ARQUITECTURA

Esto se refiere a patrones que brindan un esquema de referencia útil para guiarse en el desarrollo de software dentro de un sistema informático. Los objetivos que persigue esta etapa son: que el software pueda ser sostenible, esto es, fácilmente analizable, modificable, corregible; también se tienen en cuenta otro factor tal como la escalabilidad [20].

La parte fundamental del sistema propuesto es realizar la ejecución de la hiperheurística y el algoritmo obtenido de esta a partir de 2 factores fundamentales: cobertura, y redundancia.

### 1.2.1 Hiperheurística

Lo primero que se requiere en el sistema es la etapa de pre-procesamiento en la cual se realiza la segmentación de oraciones, filtro de palabras vacías y lematización. Con ésta etapa se logra una considerable reducción de palabras poco significativas que podrían generar ruido para el proceso de selección de oraciones relevantes. Además, se realiza la representación de documentos de origen cuyo objetivo es facilitar la interpretación de los documentos para que puedan ser procesados por la configuración del Algoritmo Memético.

Posteriormente se procede a realizar la hiperheurística con una de las dos selecciones de alto nivel. Luego con la selección de alto nivel se eligen las heurísticas de bajo nivel. Las heurísticas elegidas son utilizadas para ejecutar el algoritmo memético, que es el encargado de seleccionar las oraciones relevantes con ayuda de la función objetivo basada en cobertura y relevancia. Además dentro de la ejecución del Algoritmo Memético se lleva el contador de las heurísticas que obtienen buenos resultados en los agentes de la población y los que no.

Por último se obtiene un archivo de Excel con los contadores y probabilidades, que sirven para obtener la configuración del algoritmo memético.

Se definió una arquitectura de la Hiperheurística para generar la configuración del Algoritmo Memético que se muestra en la Figura 2.

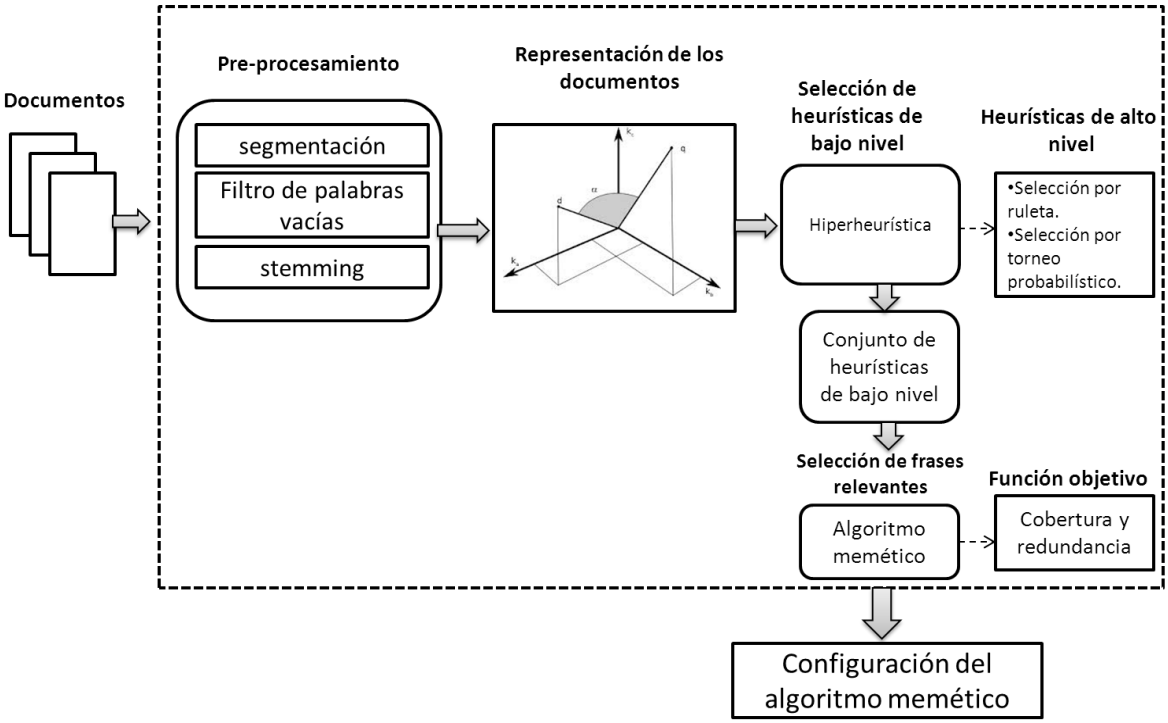


Figura 2. Arquitectura de la Hiperheurística para generar la configuración del Algoritmo Memético

### 1.2.2 Algoritmo Memético

Para el algoritmo obtenido con el enfoque hiperheurístico también se maneja la etapa de pre-procesamiento.

Posteriormente se realiza el proceso de selección de oraciones relevantes con el Algoritmo Memético teniendo en cuenta la función objetivo basada en cobertura y redundancia, cuyo resultado es el conjunto de oraciones que conforman el resumen. En la Figura 3 se muestra la arquitectura del Algoritmo Memético.

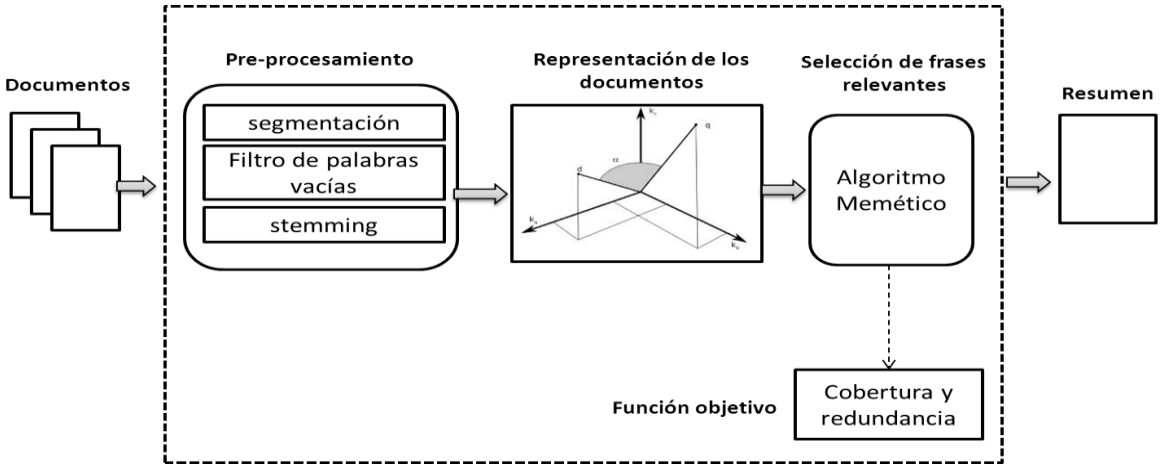


Figura 3. Arquitectura del Algoritmo Memético

## 1.3 DEFINICIÓN DEL ENTORNO DE EVALUACIÓN

En este punto se realiza la selección de los documentos proporcionados por DUC[1], con los cuales se hace la evaluación del algoritmo y también se define el entorno con el cual se realiza la evaluación, como resultado de esta fase se obtiene el conjunto de documentos a utilizar y las características del entorno.

### 1.3.1 Conjunto de documentos a utilizar

Para la selección de los documentos proporcionados por DUC se tuvo en cuenta los documentos que utiliza los métodos de referencia para esta investigación, ya que de esta forma se va a poder comparar los resultados obtenidos con nuestro algoritmo propuesto con los de referencia. Uno de los métodos de referencia con quien se compara es MCMR-PSO [21], este método hace uso de un algoritmo evolutivo para la generación de resúmenes automáticos para múltiples documentos.

Los documentos proporcionados por DUC incluyen:
- Documentos
- Resúmenes, resultados, tablas con resultados de evaluaciones, soporte adicional y software.

A continuación se muestra los documentos utilizados para la evaluación del Algoritmo Memético:

La Tabla 3 y Tabla 4 muestra un breve resumen de las características de DUC2005 y DUC2007 respectivamente.

| DUC | |
|---|---|
| Año | 2005 |
| Datos | Los evaluadores de NIST[2] seleccionan temas de interés, cada tema tiene al menos 35 documentos relevantes asociados, los evaluadores leen los documentos para cada tema y seleccionan un subconjunto de 25 a 50 documentos relevantes. |
| Tareas | • Resumen de múltiples documentos enfocado en consultas complejas.<br>• Generación de un resumen breve, bien organizado, fluido y con un nivel de granularidad. El resumen no debe tener más de 250 palabras. |
| Evaluación | NIST calcula dos medidas de ROUGE oficiales: recuerdo con ROUGE-2 y ROUGE-SU4. |
| Observaciones | |

Tabla 3. Características documentos de DUC 2005

[1] Conferencias de Coprensión de Documentos (Document Understanding Conference, DUC)

[2] NIST, National Institute of Standards and Technology

| DUC | |
|---|---|
| Año | 2007 |
| Datos | Los datos provienen del corpus de AQUAIN que comprenden artículos de noticias, los evaluadores crean un tema y seleccionan un subconjunto de 25 documentos relevantes. |
| Tareas | |
| Evaluación | |
| Observaciones | |

Tabla 4. Características documentos de DUC 2007

### 1.3.2 Características del entorno

Para la realización de las experimentaciones de laboratorio y para el desarrollo del software se utilizó un equipo con las siguientes características:
**Sistema operativo:** Windows XP SP3, 32 bits.
**Procesador:** Intel(R) Pentium(R) 4 CPU 3.00 GHz, 2992 Mhz.
**Ram:** 1GB.
**Hard disk:** 74.50 GB
**Fabricante del sistema:** Dell Computer Corporation
**Modelo del sistema:** Dimension 8300

El entorno de desarrollo fue Microsoft Visual Studio 2010 con el lenguaje c# .net.

Para la evaluación de la calidad de los resúmenes generados por los algoritmos se hizo uso de la herramienta ROUGE la cual estaba realizada en el lenguaje Perl y fue ejecutada en el entorno de Windows.

# ANEXO B - METODOLOGÍA DESARROLLO DE LA HIPERHEURÍSTICA

# 2 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA RUP

La metodología para la elaboración de la hiperheurística y la evaluación es una instanciación del Proceso Unificado, la cual tuvo en cuenta las fases de Iniciación, Elaboración, Construcción y Transición, se incluyó en la fase de transición la realización de pruebas experimentales y análisis de resultados, a continuación se describen cada una de las fases.

## 2.1 INICIACIÓN

En esta fase se realizará un diseño preliminar del entorno Hiperheurístico y de la arquitectura general del sistema teniendo en cuenta los requisitos planteados en la primera etapa. Como resultado de esta fase se obtendrá: Un diagrama general de casos de uso del sistema.

### 2.1.1 Diagrama general de casos de uso del sistema

El diagrama de casos de uso muestra el posible comportamiento del entorno hiperheurístico. Por tal motivo, es útil para determinar los requisitos funcionales del entorno, es decir, representan las funciones que el entorno puede ejecutar o realizar. En la Figura 4, se muestra el diagrama de casos de uso para el entorno hiperheurístico.

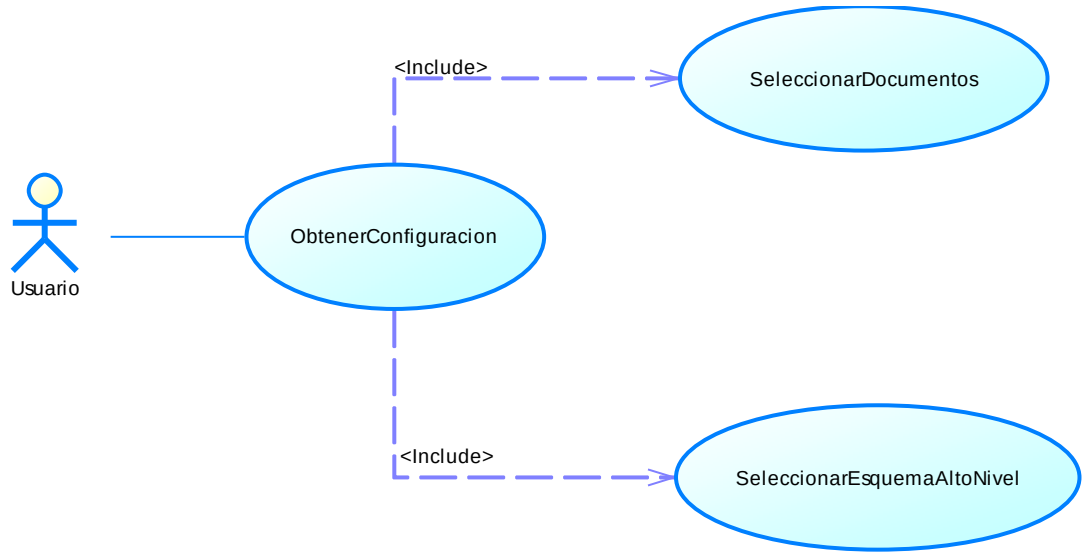


Figura 4. Diagrama de casos de uso del entorno hiperheurístico

## 2.2 ELABORACIÓN

En esta fase se comprendió con mayor detalle los requerimientos para el modelado y diseño del entorno Hiperheurístico, además el afinamiento de la arquitectura general del sistema. Como resultado de esta fase se obtuvo: Casos de Uso de alto nivel, Diagrama de Clases y Arquitectura base.

### 2.2.1 Caso de uso de alto nivel

En la Figura 5 se muestra la operación que el usuario del sistema puede realizar que es obtener la configuración del Algoritmo Memético.

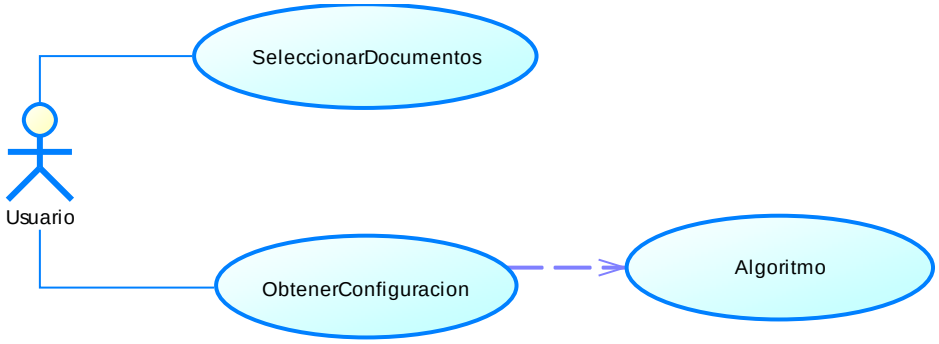


Figura 5. Casos de uso de alto nivel Hiperheurística

### 2.2.2 Diagrama de clases.

El sistema para obtener la configuración del Algoritmo Memético, se encuentra dividido en dos módulos

- Pre-procesamiento
- Hiperheurística

En la Figura 6 y Figura 7 se muestran los diagramas de clase para cada uno de los módulos y en la Tabla 5 se describe la funcionalidad de cada una de las clases.

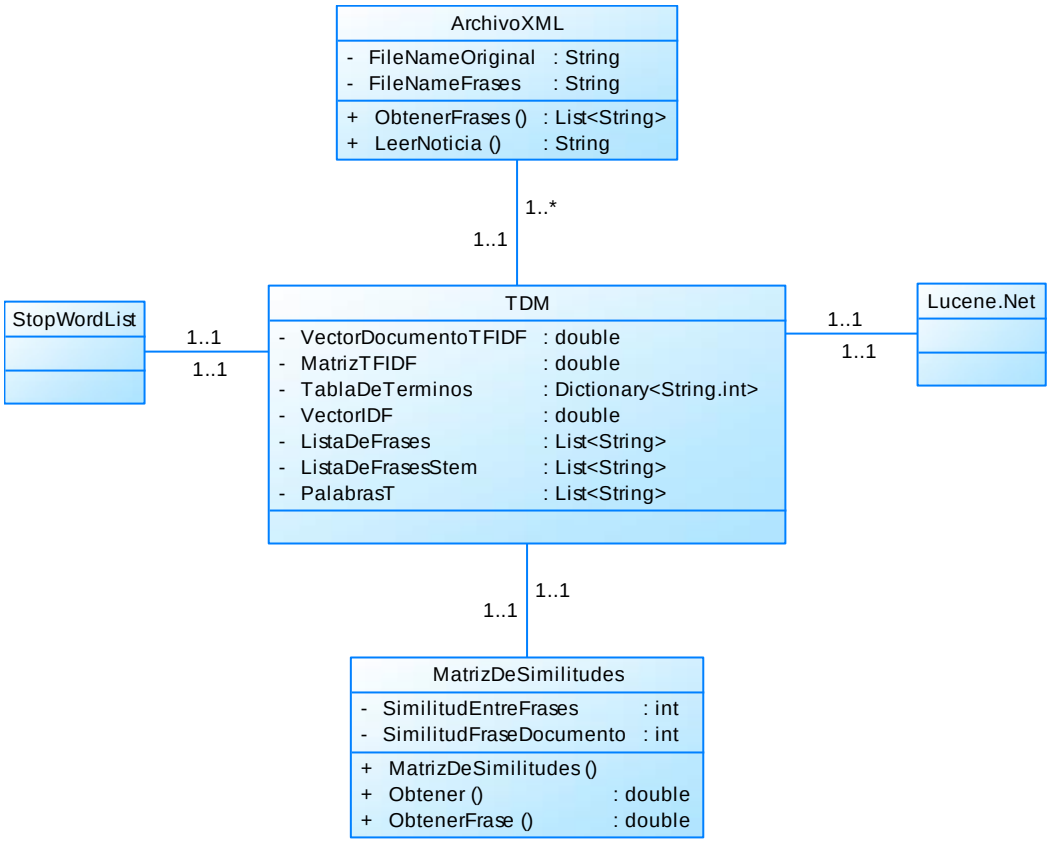


Figura 6. Diagrama clases pre-procesamiento

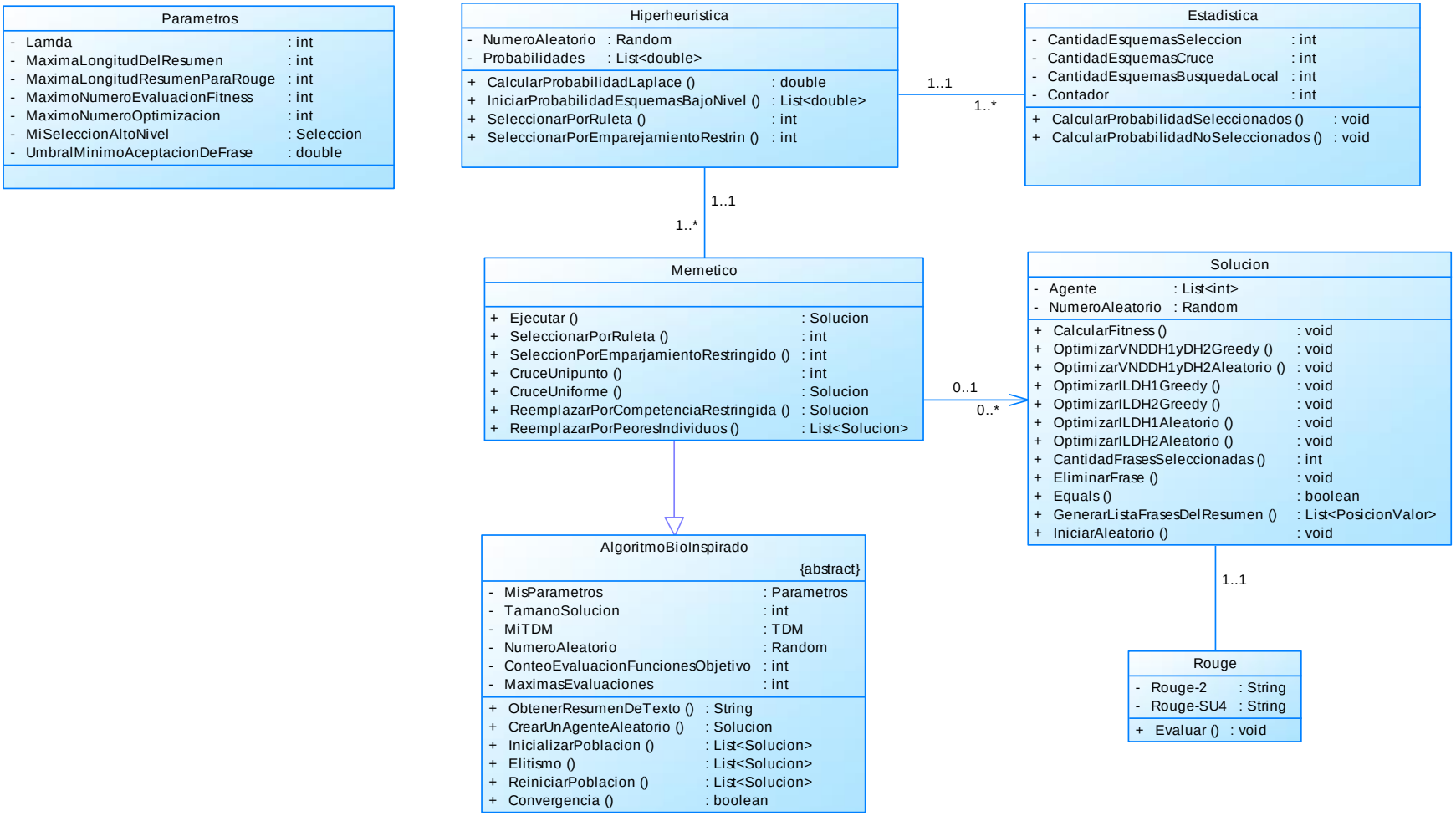


Figura 7. Diagrama clases Hiperheurística

| CLASE | FUNCIÓN |
|---|---|
| ArchivoXML | En esta clase se implementan las funciones que permiten leer los documentos de DUC y realizar la segmentación de las oraciones |
| StopWordList | Es una clase estática que contiene todas las palabras vacías (stopword) que se eliminarán de los documentos a resumir. |
| Lucene | Esta clase permite hacer eliminación de palabras vacías (stopwords), aplicar lematización (stemming), segmentación de oraciones en palabras. |
| TDM | Clase donde se implementan las funciones que permiten realizar la matriz TF-IDF y la matriz de pesos. |
| MatrizDeSimilitudes | Implementa funciones que permiten el cálculo de la matriz triangular superior de la similitud entre los términos y el documento. |
| AlgoritmoBioInspirado | Clase abstracta que implementa las funciones comunes para el algoritmo memético. |
| Memético | Representa la funcionalidad (pasos) algoritmo para la solución del problema. |
| Hiperheurística | Contiene la funcionalidad para elegir los esquemas de selección, cruce y búsqueda local para la ejecución del algoritmo memético. |
| Estadísticas | Clase utilizada para hallar las probabilidades de los esquemas de bajo nivel: selección, cruce y búsqueda local cuando se ejecuta el Algoritmo Memético. |
| Solución | Clase que implementa la funcionalidad necesaria para la ejecución del Algoritmo Memético. Entre sus funciones cabe resaltar el cálculo de la función objetivo que permite conocer qué tan buena es la combinación de heurísticas de bajo nivel realizada en el algoritmo memético. |
| Rouge | Realiza el llamado a ROUGE-1.5.5.pl para realizar la evaluación del resumen generado. |
| Parámetros | Clase utilizada para almacenar todas las variables necesarias para la correcta ejecución de la hiperheurística. |

Tabla 5. Descripción de cada una de las clases

### 2.2.3 Arquitectura base:

Para el sistema se definió una arquitectura multinivel que consta de 3 niveles, lógica de presentación, lógica de negocio y persistencia. Entre las ventajas que se obtienen mediante el uso de este tipo de arquitectura están la flexibilidad, la escalabilidad y el mantenimiento eficiente del sistema. En la Figura 8 se muestra la arquitectura del sistema y sus componentes.

A continuación se hace una breve descripción de las funciones que se realizan en cada uno de los niveles de la arquitectura.

- *Presentación:* En este nivel se incluyen el componente de la interfaz del usuario que permiten seleccionar la opción necesaria para la ejecución de la hiperheurística. Este nivel se comunica únicamente con la capa de negocio por medio de las interfaces.

- *Lógica de Negocio*: Este nivel se divide en tres módulos:

    - Módulo de Pre-procesamiento: Contiene la lógica necesaria para seleccionar los documentos y hacer los cálculos requeridos para el proceso de generación de resúmenes de múltiples documentos teniendo en cuenta segmentación de oraciones, cálculo de matriz TF-IDF.

    - Módulo Hiperheurística: Contiene la hiperheurística que realizará la selección de los esquemas de bajo nivel: selección, cruce y búsqueda local que serán utilizados en el algoritmo memético; y las estadísticas que son necesarias para hallar las probabilidades de los esquemas de bajo nivel.

    - Módulo Memético: Contiene la lógica necesaria para realizar la generación de resúmenes de múltiples documentos.

- *Persistencia:* En este nivel es donde residen los datos de las probabilidades de los esquemas de bajo nivel y medidas de Rouge. Tiene como objetivo almacenar los resultados de la ejecución de las experimentaciones en una hoja de Excel este nivel contiene el siguiente subnivel:

    - Lógica de servicios: Implementa la persistencia de la información obtenida por el programa ocultando los detalles de los repositorios de datos a los niveles superiores.

## 2.3 CONSTRUCCIÓN

Una vez realizadas las fases de Iniciación y Elaboración, se obtuvo un prototipo funcional de la hiperheurística a través de las siguientes actividades:

- **Análisis**: Se hace una profundización sobre los artefactos generados durante la fase de elaboración para la construcción del sistema.

- **Diseño**: Se realizaron los casos de uso reales que sirvieron como guía para la construcción de las diferentes funcionalidades.

    - **Implementación**: Se implementó el sistema (en el ciclo uno la hiperheurística junto con las heurísticas de alto y bajo nivel) con los artefactos obtenidos en las anteriores actividades (Análisis y Diseño).

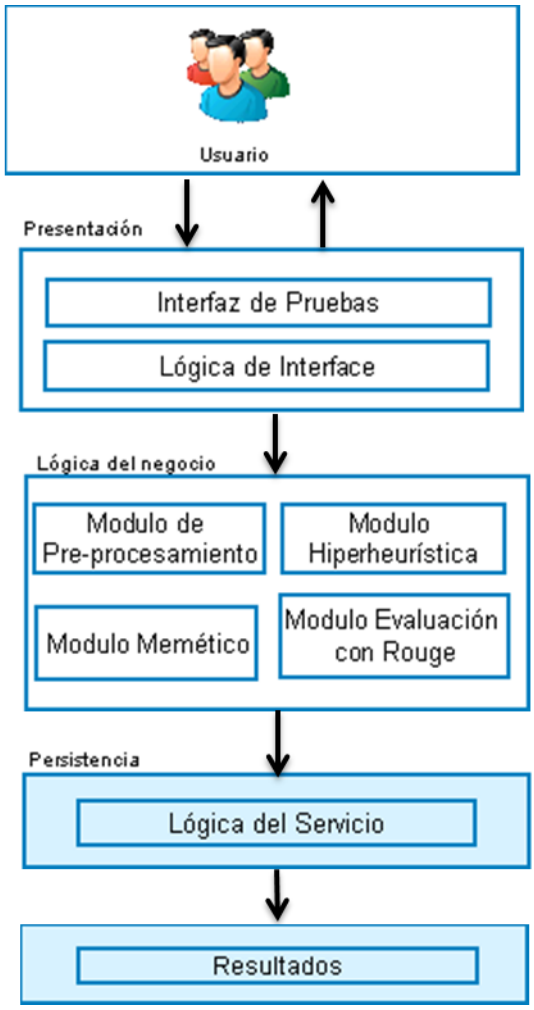


Figura 8. Arquitectura del Sistema.

- **Pruebas:** Al finalizar la implementación del entorno Hiperheurístico se define un conjunto de pruebas de caja negra, que serán aplicadas a las siguientes funcionalidades: 1. Algoritmo de generación automática de resúmenes de múltiples documentos 2. Algoritmo de evaluación de los resultados 3. Tabulación y reporte de resultados de las comparaciones realizadas.

### 2.3.1 Ciclos de Desarrollo

Los ciclos de desarrollo permitieron dividir la funcionalidad del sistema en funciones más pequeñas que facilitaron la labor de construcción del sistema cumpliendo con cada una de las fases mencionadas anteriormente. Los ciclos desarrollados fueron:

- Ciclo 1. En este ciclo se realizó:
    - Hiperheurística: tomando como base el esquema [22] se adaptó la hiperheurística para selección de las heurísticas de bajo nivel.

- La adaptación de las heurísticas de alto para la hiperheurística y heurísticas de bajo nivel para el algoritmo memético.
- Reporte de resultados: Se implementó una función que permitió guardar los resultados en una hoja de Excel para después realizar el análisis de las experimentaciones.

- Ciclo 2. En este ciclo se realizó:

  - Afinamiento del algoritmo memético obtenido con el enfoque hiperheurístico.

### 2.3.2 Casos de uso reales

A continuación se muestra los casos de uso reales del sistema

| **CASO DE USO REAL: REALIZAR RESUMEN DE COLECCIÓN DE DOCUMENTOS** | |
|---|---|
| **Actores**: Usuario. | |
| **Propósito**: Realizar evaluación de los documentos seleccionados | |
| **Resumen**: El usuario selecciona la colección de documentos que desea evaluar, ejecuta todo el proceso y finalmente se entrega una hoja de Excel con los resultados de la evaluación que contienen las probabilidades de uso de los esquemas de bajo nivel. | |
| **Tipo**: Primario. | |
| **CURSO NORMAL DE LOS EVENTOS** | |
| **Acción del actor** | **Respuesta del sistema** |
| 1. El usuario ejecuta la opción que da inicio a este caso de uso. | |
| | 2. El sistema presenta al usuario una interfaz con las opción: Directorio documentos. |
| 3. El usuario da clic en el botón Examinar y selecciona los documentos | 4. El sistema toma los documentos y los guarda en memoria temporal. |
| 5. El usuario da click en obtener la configuración | 6. El sistema procesa la opción:<br>a. El sistema realiza el pre-procesamiento de los documentos, lo que incluye, segmentación de oraciones, construcción de matriz de similitudes de términos por documento.<br>b. Realización de la hiperheurística para escoger los esquemas de bajo nivel.<br>c. Realización del resumen aplicando las heurísticas de bajo nivel seleccionadas.<br>d. Evaluación del resumen por medio de la herramienta ROUGE.<br>e. Generar hoja de Excel con los resultados de la evaluación, contadores y probabilidades de los |

| | |
|---|---|
| | esquemas de bajo nivel. |
| | 7. Una vez terminada la ejecución, el sistema finaliza y en el directorio evaluación se crea una hoja de Excel con los resultados de la evaluación, contadores y probabilidades. |
| **CURSO ALTERNO** | |
| **Acción del actor** | **Respuesta del sistema** |
| 8. El usuario no selecciona los documentos a evaluar. | 9. El sistema le informa que debe seleccionar los documentos. |

Tabla 6. Caso de uso real: realizar evaluación de colección de documentos

### 2.3.3 Pruebas de caja negra

Las pruebas de caja negra fueron aplicadas al algoritmo memético y están divididas en cuatro módulos los cuales se describen a continuación:

#### *2.3.3.1 Módulo generación automática de resúmenes*

En este módulo se seleccionan un conjunto de documentos a resumir los cuales deben cumplir con el formato que se mencionó en la Figura 1. Estructura de los documentos de evaluación, como resultado final este módulo entrega una lista de oraciones que contienen una cantidad aproximada de palabras igual a la longitud del resumen que es 250.

En este módulo se debe validar que los documentos que se leen del archivo cumplan con el formato especificado, en caso contrario se muestra un error.

- Tabla de clases equivalentes
  Dr = Directorio conjunto de documentos

| Asume | Id | Condición | Clases Correctas | Clases Erróneas |
|---|---|---|---|---|
| | A | N° de parámetros | { n = 1 } **1** | { n < 1 } **2.1** { no selecciono ningún documento } **2.2** |
| A | B | Datos correctos | { Dr debe ser un directorio que contenga documentos válidos} **3.1** | { Dr no contiene directorios validos} **4.1** |

Tabla 7. Clases equivalentes módulo 1

- Batería de pruebas

| | **Entradas** | **Salidas** | **Clases Cubiertas** | **Valores Limite** | **Salidas** |
|---|---|---|---|---|---|
| Clases correctas | Conjuntos de documentos válidos. | Resumen realizado con éxito | 1, 3.1 | | Resumen realizado con éxito |
| Clases erróneas | Sin parámetros | Debe seleccionar un conjunto de documentos para resumir | 2.1,2.2 | | |
| | Dr con documentos no validos | No se pueden leer los documentos a resumir | 4.1 | | |

Tabla 8. Batería de pruebas módulo 1

### 2.3.4 Modulo evaluación de resultados

Este módulo recibe como entrada un archivo con el resumen realizado por el algoritmo, los resúmenes modelo que se utilizan para la evaluación son seleccionados de manera automática por la herramienta ROUGE 1.5.5. La salida de este modelo es un archivo con los resultados de la evaluación.

El archivo resumen no tiene ningún tipo de etiqueta, sólo contiene las oraciones seleccionadas y no tiene extensión.

- Tabla de clases equivalentes
  Ar = Archivo resumen

| **Asume** | **Id** | **Condición** | **Clases Correctas** | **Clases Erróneas** |
|---|---|---|---|---|
| | A | N° de parámetros | { n = 1 } **1** | { n < 1 } **2** |
| | B | Datos correctos | Ar es archivo resumen **3** | Ar no contiene ningún archivo resumen **4** |

Tabla 9. Clases equivalentes módulo 2

- Batería de pruebas

| | **Entradas** | **Salidas** | **Clases Cubiertas** | **Valores Limite** | **Salidas** |
|---|---|---|---|---|---|
| Clases correctas | Ar es un archivo resumen valido | Archivo con resultados | 1,3 | | |
| Clases erróneas | Ar no contiene ningún archivo resumen valido | Error! no se puede leer el resumen | 4 | | |
| | Ar es nulo | Debes seleccionar el archivo resumen | 2 | | |

Tabla 10. Batería de pruebas módulo 2

### 2.3.5 Módulo tabulación de resultados

Este módulo toma el archivo de evaluación de resultados, lee los valores y genera una hoja de Excel donde se puede analizar los resultados.

El archivo de evaluación de resultado es un archivo .txt sin ningún tipo de etiqueta.

- Tabla de clases equivalentes

AER = Archivo de evaluación de resultados

| **Asume** | **Id** | **Condición** | **Clases Correctas** | **Clases Erróneas** |
|---|---|---|---|---|
| | A | N° de parámetros | { n = 1 } **1** | { n < 1 } **2** |
| | B | Datos correctos | AER es archivo valido **3** | AER no contiene ningún archivo valido **4** |

Tabla 11. Clases equivalentes módulo 3

- Batería de pruebas

| | **Entradas** | **Salidas** | **Clases Cubiertas** | **Valores Limite** | **Salidas** |
|---|---|---|---|---|---|
| Clases correctas | AER es un archivo valido | Hoja de Excel creada | 1,3 | | |
| Clases erróneas | AER no contiene ningún archivo valido | Error! no se puede crear la hoja de Excel | 4 | | |
| | AER es nulo | Debes seleccionar el archivo resumen | 2 | | |

Tabla 12. Batería de pruebas módulo 3

## 2.4 TRANSICIÓN

En esta fase se verifica la funcionalidad del sistema, se realiza el afinamiento de parámetros al algoritmo memético para obtener mejores resultados, y la evaluación por medio de la utilización de documentos de DUC. Finalmente se realiza el análisis de los resultados obtenidos en la evaluación.

# ANEXO C –ALGORITMO MEMÉTICO PARA GENERACIÓN DE RESÚMENES

## 3 EJEMPLO DEL ALGORITMO MEMÉTICO

El siguiente ejemplo muestra en detalle la operación del algoritmo memético cuando es aplicado a una población que contiene 10 agentes. Los agentes de la población se pueden observar en cada una de las filas de la Tabla 13, la cual se encuentra ordenada según la función objetivo de mayor a menor.

| S1 | S2 | S3 | S4 | S5 | S6 | S7 | F(x) |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 5,3242 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 5,1321 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 4,3152 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 4,3012 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3,5432 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3,1456 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2,5469 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2,1365 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1,0259 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1,0012 |

Tabla 13. Población

A. Se eligen dos agentes con la selección de emparejamiento restringido, estos serán los padres utilizados en el cruce. El primer agente Padre1 se selecciona aleatoriamente y el Padre2 teniendo en cuenta que sea similar al Padre1. En la Tabla 14 se puede ver los padres que han sido seleccionados.

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | F(x) |
|---|---|---|---|---|---|---|---|---|
| Padre1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 4,3152 |
| Padre2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 4,3012 |

Tabla 14. Padres seleccionados

B. Se cruzan los agentes padre por medio del cruce unipunto para crear la descendencia. Los agentes Padre1 y Padre2 son cortados en la posición 4. Luego la cabeza del Padre1 y la cola del Padre2 harán parte del Hijo1. Y la cabeza del Padre2 y la cola del Padre1 harán parte del Hijo2. En la Tabla 15 se encuentran los hijos obtenidos con el cruce de padres.

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | F(x) |
|---|---|---|---|---|---|---|---|---|
| Hijo1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4,4012 |
| Hijo2 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 4,2014 |

Tabla 15. Agentes hijo

C. Se optimizan los agentes hijo por medio de la búsqueda local greedy con distancia de hamming uno y dos. En este ejemplo se va optimizar el Hijo1 una vez, para el que se crean dos agentes vecinos con distancia de hamming uno y dos agentes vecinos con distancia de hamming dos.

Vecinos con distancia de hamming uno (ver Tabla 16): al Vecino1 se le adiciona un meme al Hijo1 con cobertura más alta. Los memes se encuentran en una lista ordenados descendentemente por cobertura. Y para el Vecino2 se adiciona el meme siguiente de la lista, siempre y cuando no se encuentre ese meme en el Hijo1.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **F(x)** |
|---|---|---|---|---|---|---|---|---|
| Hijo1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4,4012 |
| Vecino1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 4,3001 |
| Vecino2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 4,3120 |

Tabla 16. Vecinos con distancia de hamming uno

Vecinos con distancia de hamming dos (ver Tabla 17): al Vecino1 se le elimina del Hijo1 un meme con peor cobertura y se adiciona un meme con cobertura más alta. Los memes se manejan con la lista ordenada por cobertura. Se sigue el mismo proceso para el Vecino2 teniendo en cuenta que el meme que se adiciona no se encuentre en el Hijo1.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **F(x)** |
|---|---|---|---|---|---|---|---|---|
| Hijo1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4,4012 |
| Vecino1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 4,4321 |
| Vecino2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 4,3130 |

Tabla 17. Vecinos con distancia de hamming dos

La optimización del Hijo1 consiste en buscar en la vecindad con distancia de hamming uno el mejor vecino. Como no hay un mejor vecino aquí, entonces se procede a buscarlo en la vecindad con distancia de hamming dos. Como se puede observar en la Tabla 17, el mejor vecino es el Vecino1. Por lo tanto el Vecino1 será el nuevo agente como se observa en la Tabla 18 .

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **F(x)** |
|---|---|---|---|---|---|---|---|---|
| NuevoAgente | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 4,4321 |

Tabla 18. Nuevo Agente

D. Se actualiza la población: este paso determina si el nuevo agente de la Tabla 18 hará parte de la nueva población. Teniendo en cuenta el procedimiento del reemplazo por de los peores individuos se realiza lo siguiente:
   - Se forma un grupo de agentes de los peores individuos de la población de tamaño 3 como se puede ver en la Tabla 19.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **F(x)** |
|---|---|---|---|---|---|---|---|---|
| a | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2,1365 |
| b | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1,0259 |
| c | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1,0012 |

Tabla 19. Grupo de agentes aleatorios

   - Se reemplaza el NuevoAgente en el peor agente del grupo de la Tabla 19, el peor agente está ubicado en la fila c.

E. La nueva población se puede observar en la Tabla 20.

| S1 | S2 | S3 | S4 | S5 | S6 | S7 | F(x) |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 5,3242 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 5,1321 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 4,3152 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 4,3012 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3,5432 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3,1456 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2,5469 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2,1365 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1,0259 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 4,4321 |

Tabla 20.Nueva Población

La fila sombreada de la Tabla 20 es el NuevoAgente que reemplazo al peor agente de un grupo de los peores agentes.

# ANEXO D – SELECCIÓN DE FUNCIÓN OBJETIVO

# 4 SELECCIÓN DE FUNCIÓN OBJETIVO

En este proyecto se planteó el uso de una función objetivo basada en máxima cobertura y mínima redundancia, teniendo en cuenta que investigaciones que contemplan estos factores en la función objetivo han mostrado buenos resultados con respecto al estado del arte [21, 23-25]. Una de estas investigaciones es el algoritmo basado en PSO cuya función objetivo está compuesta por estos dos factores (MCMR PSO) [21], y que fue tomada como base para la función objetivo del algoritmo propuesto en este proyecto.

El factor de redundancia fue tomado de la misma forma como se planteó en MCMR; pero al factor de cobertura se le realizó una variación calculando como la similitud del texto del resumen candidato con el centroide de la colección de documentos y no el cálculo de la similitud de cada oración del resumen con respecto al centroide de la colección de documentos, además se incluyó un coeficiente para ponderar la importancia de los dos factores. Esta modificación permite que se pueda afinar la importancia de cada factor, lo que no ocurre en la función objetivo MCMR, en la cual se da mayor importancia al factor de cobertura dado que la formula suma varias veces la similitud de cada una de las oraciones del resumen con respecto a la colección de documentos.

De esta forma la función objetivo es la combinación de los factores de cobertura (FC) y redundancia (FR) como se observa en la Ecuación 2, los cuales están controlados por el coeficiente lambda (λ) el cual le da flexibilidad a la función objetivo permitiendo que se dé mayor o menor peso a cada uno de los factores. El coeficiente λ varía entre 0 y 1.

$$f(x) = \lambda * FC - (1 - \lambda) * FR \qquad \text{Ecuación 2}$$

Donde

$$FC = Sim\ (R, D) \qquad \text{Ecuación 3}$$

$$FR = \frac{2}{n \times (n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Sim(S_i, S_j) \qquad \text{Ecuación 4}$$

**Factor de cobertura (FC)**. Como se observa en la Ecuación 3, *R* representa el texto con todas las oraciones del resumen candidato; *D*, representa todas las oraciones de la colección de documentos (en este caso es el centroide de la colección de documentos); y *Sim*(*R*,*D*), es la similitud de cosenos entre el vector de términos de *R* y el vector de términos de *D*. por lo tanto este factor toma valores entre cero y uno.

**Factor de redundancia (FR).**como se observa en la Ecuación 4, $S_i$ y $S_j$ son oraciones del resumen $Sim(S_i, S_j)$ es la similitud entre las dos oraciones y *n* es la cantidad de oraciones que hay el resumen.

## 4.1 Comparación funciones objetivo

A continuación se muestra los resultados obtenidos con la función objetivo de MCMR-PSO y nuestra función objetivo modificado.

| | Rouge-2 | Rouge-su4 |
|---|---|---|
| **F. AM** | 0.0812 | 0.1394 |
| **F. MCMR** | 0.0755 | 0.1334 |

Tabla 21. Resultados comparación función objetivo

Como podemos observar en la Tabla 21, los mejores resultados se presentan con la función objetivo modificada. Por tanto la función que se utilizó para la presente investigación se le realizó la modificación de la función de cobertura como se presenta en la Ecuación 3.

# ANEXO E – EXPERIMENTOS DE LA HIPERHEURÍSTICA

# 5 EXPERIMENTOS

En la Tabla 22 se muestran los conjuntos de heurísticas de bajo nivel utilizados en las experimentaciones de la hiperheurística.

| **CONJUNTO UNO** | **CONJUNTO DOS** |
|---|---|
| Selección por Ruleta<br>Selección por emparejamiento restringido | Selección por Ruleta<br>Selección por emparejamiento restringido<br>Selección basada en rango |
| Cruce unipunto<br>Cruce uniforme | Cruce unipunto<br>Cruce uniforme |
| Búsqueda por vecindad:<br>• Búsqueda por vecindad greedy con distancia de hamming 1 y 2.<br>• Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2.<br><br>Búsqueda local iterada:<br>• Búsqueda por vecindad greedy con distancia de hamming uno.<br>• Búsqueda por vecindad greedy con distancia de hamming dos.<br>• Búsqueda por vecindad aleatorio con distancia de hamming uno.<br>• Búsqueda por vecindad aleatoria con distancia de hamming dos. | Búsqueda por vecindad:<br>• Búsqueda por vecindad greedy con distancia de hamming 1 y 2.<br>• Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2.<br><br>Búsqueda local iterada:<br>• Búsqueda por vecindad greedy con distancia de hamming uno.<br>• Búsqueda por vecindad greedy con distancia de hamming dos.<br><br>Búsqueda local guiada. |
| Reemplazo por competencia restringida<br>Reemplazo de peores individuos | Reemplazo por competencia restringida<br>Reemplazo de peores individuos |

Tabla 22. Conjuntos de heurísticas de bajo nivel

Se realizó una segunda experimentación de la hiperheurística con unas variaciones en el conjunto de heurísticas de bajo nivel agregando la selección basada en rango y la búsqueda local guiada, debido a los buenos resultados obtenidos en la tesis de pregrado del programa de Ingeniería de Sistemas de la Universidad del Cauca, titulada “Generación Automática de Resúmenes de Un Solo Documento Basada en Algoritmos Meméticos” bajo la misma dirección de la profesora Martha Mendoza. Además se quitaron dos de las búsquedas locales iteradas que se manejaban de forma aleatoria porque éstas dieron bajos resultados en la experimentación con el primer conjunto de datos.

Para encontrar la mejor configuración con el primer y segundo conjunto de heurísticas se tuvo en cuenta cuatro experimentaciones en cada experimento de la hiperheurística como se observa en la Tabla 23.

| Esquemas de selección de Alto Nivel | Reemplazo |
|---|---|
| Selección por Ruleta. | Por competencia restringida |
| | De peores individuos |
| Selección por emparejamiento restringido. | Por competencia restringida |
| | De peores individuos |

Tabla 23. Combinaciones para realizar las experimentaciones

## 5.1 Resultados de las experimentaciones

En las Tablas Tabla 24Tabla 25Tabla 26Tabla 27 se muestran los resultados de las medidas de Rouge para DUC2005 y DUC2007 de las combinaciones que se realizaron para las experimentaciones de la hiperheurística. En la Tabla 28 se muestran las filas sombreadas de las Tablas Tabla 24Tabla 25Tabla 26Tabla 27 que representan las mejores combinaciones de esquema de selección de alto nivel y reemplazo para cada conjunto de heurísticas de bajo nivel y el conjunto de documentos DUC.

| Esquemas de selección de Alto Nivel | Reemplazo | Rouge-2 | Rouge-SU4 |
|---|---|---|---|
| Selección por Ruleta. | Por competencia restringida | 0.0808 | 0.1391 |
| | De peores individuos | 0.0795 | 0.1375 |
| Selección por emparejamiento restringido. | Por competencia restringida | 0.0801 | 0.1379 |
| | De peores individuos | 0.0801 | 0.1377 |

Tabla 24. Resultados del primer conjunto de heurísticas de bajo nivel para DUC2005

| Esquemas de selección de Alto Nivel | Reemplazo | Rouge-2 | Rouge-SU4 |
|---|---|---|---|
| Selección por Ruleta. | Por competencia restringida | 0.1149 | 0.1655 |
| | De peores individuos | 0.1151 | 0.1656 |
| Selección por emparejamiento restringido. | Por competencia restringida | 0.1152 | 0.1657 |
| | De peores individuos | 0.1151 | 0.1656 |

Tabla 25. Resultados del primer conjunto de heurísticas de bajo nivel para DUC2007

| Esquemas de selección de Alto Nivel | Reemplazo | Rouge-2 | Rouge-SU4 |
|---|---|---|---|
| Selección por Ruleta. | Por competencia restringida | 0.0812 | 0.1391 |
| | De peores individuos | 0.0795 | 0.1374 |
| Selección por emparejamiento restringido. | Por competencia restringida | 0.0800 | 0.1378 |
| | De peores individuos | 0.0802 | 0.1379 |

Tabla 26. Resultados del segundo conjunto de heurísticas de bajo nivel para DUC2005

| Esquemas de selección de Alto Nivel | Reemplazo | Rouge-2 | Rouge-SU4 |
|---|---|---|---|
| Selección por Ruleta. | Por competencia restringida | 0.11492 | 0.1656 |
| | De peores individuos | 0.11494 | 0.16564 |
| Selección por emparejamiento restringido. | Por competencia restringida | 0.1152 | 0.1657 |
| | De peores individuos | 0.1153 | 0.1658 |

Tabla 27. Resultados del segundo conjunto de heurísticas de bajo nivel para DUC2007

| Conjuntos de heurísticas de bajo nivel | DUC | Esquemas de selección de alto nivel | Reemplazo | Rouge-2 | Rouge-SU4 |
|---|---|---|---|---|---|
| Primer conjunto | 2005 | Selección por ruleta | Reemplazo por competencia restringida | 0.0808 | 0.1391 |
| | 2007 | Selección por emparejamiento restringido | Reemplazo por competencia restringida | 0.1152 | 0.1657 |
| Segundo conjunto | 2005 | Selección por ruleta | Reemplazo por competencia restringida | 0.0812 | 0.1391 |
| | 2007 | Selección por emparejamiento restringido | Reemplazo de los peores individuos | 0.1153 | 0.1658 |

Tabla 28. Mejores resultados de los conjuntos de heurísticas de bajo nivel

## 5.2 ESTADÍSTICAS DE LAS MEJORES CONFIGURACIONES

De acuerdo a los resultados de los conjuntos de heurísticas de bajo nivel (Ver Tabla 28). Se procede a elegir los esquemas de bajo nivel teniendo en cuenta sus probabilidades, la

probabilidad indica que tan frecuente fue utilizado un esquema de bajo nivel en el algoritmo memético.

| **Esquema** | **Esquemas de bajo nivel** | **%** |
|---|---|---|
| Selección | Selección por ruleta | 11,49 |
| | Selección por emparejamiento restringido | 88,51 |
| Cruce | Unipunto | 11,18 |
| | Uniforme | 88,82 |
| Búsqueda local | Búsqueda por vecindad greedy con distancia de hamming 1 y 2. | 98,08 |
| | Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2. | 1,92 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 y 2 | 0,00018 |
| | Búsqueda local iterada por vecindad aleatoria distancia de hamming 1 y 2 | 0,00004 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 | 0,00004 |
| | Búsqueda local iterada por vecindad aleatoria distancia de hamming 2 | 0,00009 |

Tabla 29. Estadísticas de la mejor configuración del primer conjunto de heurísticas de bajo nivel para DUC2005

En la Tabla 29, se observa que los esquemas de bajo nivel más utilizados fueron: selección por emparejamiento restringido con un 88.51%, cruce uniforme con un 88.82% y búsqueda por vecindad greedy con distancia de hamming 1 y 2 con 98,08%.

| **Esquema** | **Esquemas de bajo nivel** | **%** |
|---|---|---|
| Selección | Selección por ruleta | 22,92 |
| | Selección por emparejamiento restringido | 77,08 |
| Cruce | Unipunto | 57,63 |
| | Uniforme | 42,37 |
| Búsqueda local | Búsqueda por vecindad greedy con distancia de hamming 1 y 2. | 56,75 |
| | Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2. | 43,25 |

| | | |
|---|---|---|
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 y 2 | 0,00009 |
| | Búsqueda local iterada por vecindad aleatoria distancia de hamming 1 y 2 | 0,0004 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 | 0,0004 |
| | Búsqueda local iterada por vecindad aleatoria distancia de hamming 2 | 0,0002 |

Tabla 30. Estadísticas de la mejor configuración del primer conjunto de heurísticas de bajo nivel para DUC2007

En la Tabla 30 los esquemas de bajo nivel más utilizados fueron: selección por emparejamiento restringido, cruce unipunto y búsqueda por vecindad greedy con distancia de hamming 1 y 2 con porcentajes de 77.08%, 57.63% y 56.75% respectivamente.

| **Esquema** | **Esquemas de bajo nivel** | **%** |
|---|---|---|
| Selección | Selección por ruleta | 3,08 |
| | Selección por emparejamiento restringido | 1,31 |
| | Selección basada en rango | 95,61 |
| Cruce | Unipunto | 18,20 |
| | Uniforme | 81,80 |
| Búsqueda local | Búsqueda por vecindad greedy con distancia de hamming 1 y 2. | 96,49 |
| | Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2. | 3,28 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 y 2 | 0,0001 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 | 0,00005 |
| | Búsqueda local guiada | 0,23 |

Tabla 31. Estadísticas de la mejor configuración del segundo conjunto de heurísticas de bajo nivel para DUC2005

En la Tabla 31, se puede observar que los esquemas de bajo nivel más utilizados fueron: selección basada en rango con un 95.61%, cruce uniforme con un 81.80% y búsqueda por vecindad greedy con distancia de hamming 1 y 2 con 96,49%.

| **Esquema** | **Esquemas de bajo nivel** | **%** |
|---|---|---|
| Selección | Selección por ruleta | 33,64 |
| | Selección por emparejamiento restringido | 33,72 |
| | Selección basada en rango | 32,64 |
| Cruce | Unipunto | 51,02 |
| | Uniforme | 48,98 |
| Búsqueda local | Búsqueda por vecindad greedy con distancia de hamming 1 y 2. | 21,48 |
| | Búsqueda por vecindad aleatoria con distancia de hamming 1 y 2. | 21,26 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 y 2 | 19,34 |
| | Búsqueda local iterada por vecindad greedy distancia de hamming 1 | 18,25 |
| | Búsqueda local guiada | 19,67 |

Tabla 32. Estadísticas de la mejor configuración del segundo conjunto de heurísticas de bajo nivel para DUC2007

En la Tabla 32 los esquemas de bajo nivel más utilizados fueron: selección por emparejamiento restringido, cruce unipunto y búsqueda por vecindad greedy con distancia de hamming 1 y 2 con porcentajes de 33.72%, 51.02% y 21.48% respectivamente.

Las Tabla 33 y Tabla 34, muestran las mejores configuraciones que se obtuvieron con la hiperheurística en cada uno de los conjuntos de heurísticas de bajo nivel, respectivamente.

| | **DUC2005** | **DUC2007** |
|---|---|---|
| Selección de Alto Nivel | Ruleta | Torneo Probabilístico |
| Selección | Emparejamiento Restringido | Emparejamiento Restringido |
| Cruce | Uniforme | Unipunto |
| Búsqueda Local | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy |
| Reemplazo | Competencia Restringida | Competencia Restringida |
| Rouge-2 | 0.0808 | 0.1152 |
| Rouge-SU4 | 0.1391 | 0.1657 |

Tabla 33. Mejores configuraciones con el primer conjunto de heurísticas de bajo nivel

| | **DUC2005** | **DUC2007** |
|---|---|---|
| Selección de Alto Nivel | Ruleta | Torneo Probabilístico |
| Selección | Basado en rango | Emparejamiento Restringido |
| Cruce | Uniforme | Unipunto |
| Búsqueda Local | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy |
| Reemplazo | Competencia Restringida | Peores Individuos |
| Rouge-2 | 0.0812 | 0.1153 |
| Rouge-SU4 | 0.1391 | 0.1658 |

Tabla 34. Mejores configuraciones con el segundo conjunto de heurísticas de bajo nivel

# ANEXO F – EXPERIMENTOS DE AFINACIÓN

# 6 AFINACIÓN PRELIMINAR DEL ALGORITMO MEMÉTICO

Los parámetros del algoritmo memético obtenido desde el enfoque hiperheurístico afinados fueron:

- La probabilidad de optimización (PO), este parámetro tomó los valores de {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}.
- El tamaño de la población (TP), para este valor se partió de un tamaño de 30 con incrementos de 10 hasta llegar a 100.
- Lambda de la función objetivo (LF), se partió de 0.7 con incrementos de 0.2 hasta llegar a 0.98.
- Máxima longitud del resumen (MLR), para este valor se partió de 270 con incrementos de 20 hasta llegar a 390.

La afinación consistió en tomar cada parámetro del algoritmo memético, buscar el mejor valor de afinación con el primer parámetro, luego con el siguiente, así hasta terminar con los parámetros a afinar.

Para las experimentaciones de las secciones 6.3 y 6.4 se utilizaron los valores de los parámetros por defecto (Probabilidad de optimización: 0.4; tamaño de población: 50; lambda: 0.86, máxima longitud del resumen: 290), porque se quería conocer el comportamiento de la configuración invertida para el primer conjunto de heurísticas de bajo nivel y la configuración obtenida en el segundo conjunto de heurísticas de bajo nivel para DUC2007 con DUC2005, para encontrar la mejor configuración en ambos conjuntos de documentos.

## 6.1 AFINACIÓN DE PARÁMETROS PARA EL PRIMER CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL

La afinación se realizó para las mejores configuraciones de DUC2005 y DUC2007 (ver Tabla 34), obteniendo las afinaciones en los parámetros y los valores de las medidas de Rouge como se puede observar en la Tabla 35.

| **Parámetros** | **DUC2005** | **DUC2007** |
|---|---|---|
| Probabilidad de Optimización | 0.2 | 0.1 |
| Tamaño de la Población | 50 | 80 |
| Lambda | 0.86 | 0.86 |
| Máxima Longitud del Resumen | 290 | 270 |
| | | |
| **Medidas** | | |
| Rouge-2 | 0.0809 | 0.1140 |
| Rouge-SU4 | 0.1397 | 0.1659 |

Tabla 35. Afinación de parámetros del primer conjunto de heurísticas de bajo nivel

## 6.2 AFINACIÓN DE PARÁMETROS PARA EL SEGUNDO CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL

La afinación se realizó para la configuración de DUC2007, porque se quería obtener mejores resultados en las medidas de Rouge-2 y Rouge-Su4 para este conjunto de documentos, debido a que sus valores en estas medidas en la afinación de parámetros del primer conjunto de heurísticas de bajo nivel no superaron a la referencia MCMR-PSO.

| **Parámetros** | **DUC2007** |
|---|---|
| Probabilidad de Optimización | 0.4 |
| Tamaño de la Población | 50 |
| Lambda | 0.86 |
| Máxima Longitud del Resumen | 270 |
| | |
| **Medidas** | |
| Rouge-2 | 0.1146 |
| Rouge-SU4 | 0.1660 |

Tabla 36. Afinación de parámetros del segundo conjunto de heurísticas de bajo nivel

Como se puede observar en la Tabla 36 los resultados de Rouge tuvieron una mejora con respecto a los resultados obtenidos en la afinación de parámetros del primer conjunto de heurísticas de bajo nivel de la Tabla 35 para DUC2007.

Adicional, se realizó la afinación de los parámetros para DUC2005 (Ver Tabla 37), pero con la configuración de heurísticas de bajo nivel que se obtuvo en DUC2007. Esto se hizo con el fin de saber cómo se comportaba la configuración obtenida para DUC2007 con el conjunto DUC2005.

| **Parámetros** | **DUC2005** |
|---|---|
| Probabilidad de Optimización | 0.5 |
| Tamaño de la Población | 70 |
| Lambda | 0.86 |
| Máxima Longitud del Resumen | 290 |
| | |
| **Medidas** | |
| Rouge-2 | 0.0812 |
| Rouge-SU4 | 0.1394 |

Tabla 37. Afinación de parámetros con la configuración de DUC2007

## 6.3 RESULTADOS DE LA CONFIGURACIÓN INVERTIDA PARA EL PRIMER CONJUNTO DE HEURÍSTICAS DE BAJO NIVEL

Con el objetivo de encontrar mejores resultados en ambos conjuntos de documentos con las medidas de Rouge se ejecutó la configuración de DUC2005 para DUC2007 y viceversa con los valores de los parámetros por defecto. En la Tabla 38 se pueden observar las nuevas configuraciones.

| | **DUC2005** | **DUC2007** |
|---|---|---|
| Selección | Emparejamiento Restringido | Emparejamiento Restringido |
| Cruce | Unipunto | Uniforme |
| Búsqueda Local | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy |
| Reemplazo | Competencia Restringida | Competencia Restringida |

Tabla 38. Nuevas configuraciones con el primer conjunto de heurísticas de bajo nivel

Los resultados que se obtuvieron para las medidas de Rouge-2 y Rouge-SU4 de las nuevas configuraciones se muestran en la Tabla 39.

| | **DUC2005** | **DUC2007** |
|---|---|---|
| Rouge-2 | 0.0807 | 0.1132 |
| Rouge-SU4 | 0.1389 | 0.1651 |

Tabla 39.Resultados de las medidas de Rouge con la nueva configuración

## 6.4 RESULTADOS DEL SEGUNDO CONJUNTO DE HEURISTICAS DE BAJO NIVEL PARA DUC2005

Antes de utilizar la configuración de DUC2007 de la Tabla 36 para el conjunto de documentos de DUC2005. Se debe tener en cuenta los resultados de las medidas de Rouge para DUC2007 con los parámetros por defecto como se puede observar en la Tabla 40.

| | **DUC2007** |
|---|---|
| Rouge-2 | 0.1138 |
| Rouge-SU4 | 0.1652 |

Tabla 40. Resultados de la configuración del segundo conjunto de heurísticas de bajo nivel sin afinación de parámetros

| | **DUC2005** |
|---|---|
| Selección | Emparejamiento Restringido |
| Cruce | Unipunto |
| Búsqueda Local | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy |
| Reemplazo | Peores Individuos |

Tabla 41. Nueva configuración con el segundo conjunto de heurísticas de bajo nivel

En la Tabla 42 se puede observar los resultados obtenidos para las medidas Rouge con la nueva configuración de heurísticas de bajo nivel para DUC2005 (Ver Tabla 41) con los parámetros por defecto.

| | **DUC2005** |
|---|---|
| Rouge-2 | 0.0808 |
| Rouge-SU4 | 0.1390 |

Tabla 42.Resultados con la configuración del segundo conjunto de heurísticas de bajo nivel

## 6.5 MEJOR CONFIGURACIÓN PARA LOS CONJUNTOS DE DOCUMENTOS DE DUC2005 Y DUC2007

A partir de los resultados de las Tabla 39, Tabla 40 y Tabla 42 se puede observar que la mejor configuración de las heurísticas de bajo nivel es la que se obtuvo con la configuración del segundo conjunto de estas heurísticas, porque superaron en ambos conjuntos de documentos las medidas de Rouge-2 y Rouge-SU4 a la configuración del primer conjunto.

Por lo tanto la configuración final para los conjuntos de documentos de DUC2005 y DUC2007 es la que se muestra en la Tabla 43.

| | **Esquemas de bajo nivel** |
|---|---|
| Selección | Emparejamiento Restringido |
| Cruce | Unipunto |
| Búsqueda Local | Búsqueda de vecindad con Distancia de hamming uno y dos Greedy |
| Reemplazo | Peores Individuos |

Tabla 43.Configuración de esquemas de bajo nivel para los conjuntos de documentos de DUC2005 y DUC2007

Además para tener el mismo valor de parámetros en ambos conjuntos de datos de DUC, se realizaron dos experimentaciones con la configuración de la Tabla 43. Estos experimentos consistieron en usar los parámetros afinados de DUC2005 (Ver Tabla 37) para DUC2007 y los parámetros de DUC2007 (Ver Tabla 36) para DUC2005.

| **Parámetros** | **DUC2005** | **DUC2007** |
|---|---|---|
| Probabilidad de Optimización | 0.4 | 0.5 |
| Tamaño de la Población | 50 | 70 |
| Lambda | 0.86 | 0.86 |
| Máxima Longitud del Resumen | 270 | 290 |
| **Medidas** | | |
| Rouge-2 | 0.8138 | 0.1140 |
| Rouge-SU4 | 0.1394 | 0.1655 |

Tabla 44. Resultados con el intercambio de parámetros afinados

| | **Valores originales** | | **Nuevos valores** | |
|---|---|---|---|---|
| **Parámetros** | DUC2005 | DUC2007 | DUC2005 | DUC2007 |
| Probabilidad de Optimización | 0.5 | 0.4 | 0.4 | 0.5 |
| Tamaño de la Población | 70 | 50 | 50 | 70 |
| Lambda | 0.86 | 0.86 | 0.86 | 0.86 |
| Máxima Longitud del Resumen | 290 | 270 | 270 | 290 |
| | | | | |
| **Medidas** | | | | |
| Rouge-2 | 0.0812 | 0.1146 | 0.8138 | 0.1140 |
| Rouge-SU4 | 0.1394 | 0.1660 | 0.1394 | 0.1655 |

Tabla 45. Comparación de resultados con las valores de las afinaciones de los parámetros originales e intercambio de parámetros afinados

Se puede observar en la Tabla 45 que se obtienen mejores resultados en las medidas de Rouge-2 y Rouge-SU4 con los parámetros afinados de DUC2007 (Ver Tabla 36). Por lo tanto se eligen estos parámetros.

# 7 Bibliografía

[1] S.N.Sivanandam and S.N.Deepa, "Introduction to Genetic Algorithms," S. B. Heidelberg, Ed. New York, 2008.

[2] C. A. C. Coello and E. M. Montes, "Constraint-Handling in Genetic Algorithms Through the Use of Dominance-based T ournament Selection," *Advanced Engineering Informatics*, 2002.

[3] M. Hutter, "Fitness uniform selection to preserve genetic diversity," *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, vol. 1, pp. 783-788 2002.

[4] S. Legg, M. Hutter, and A. Kumar, "Tournament versus Fitness Uniform Selection," *In Proceeding of the 2004 Congress on Evolutionary Computation*, 2004.

[5] D. Beasley, D. R. Bull, and R. R. Martin, "An Overview of Genetic Algorithms: Part 2, Research Topics," *University Computing*, vol. 15, pp. 170-181, 1993.

[6] R. Poli and W. B. Landong, "Schema Theory for Genetic Programming with One-point Crossover and Point Mutation " University of Birmingham, UK 1998.

[7] A. H. Wright, "Genetic algorithms for real parameter optimization," *Foundations of Genetic Algorithms*, 1991.

[8] D. Beasley, D. R. Bull, and R. R. Martin, "An Overview of Genetic Algorithms : Part 1, Fundamentals," *University Computing*, vol. 15, pp. 58-69, 1993.

[9] A. D. J. Kenneth and M. S. William, "An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms," in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*: Springer-Verlag, 1991.

[10] W. M. Spears and K. A. D. Jong, "On the Virtues of Parameterized Uniform Crossover," *In Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230-236, 1991.

[11] H. Xiao-Bing and P. Ezequiel Di, "An efficient genetic algorithm with uniform crossover for air traffic control," *Comput. Oper. Res.*, vol. 36, pp. 245-259, 2009.

[12] William M. Spears and K. A. D. Jong, "On the Virtues of Parameterized Uniform Crossover " *In Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991.

[13] O. M. Shir, "Niching in Derandomized Evolution Strategies and its Applications in Quantum Control," Leiden, 2008, p. 256.

[14] P. Hansen and N. MladenoviÄ‡, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, pp. 449-467, 2001.

[15] O. C. M. H. R. Lourenco, and T. Stützle, "Iterated Local Search," *Handbook of Metaheuristics*, vol. 7, pp. 321-353, 2003.

[16] C. Voudouris and E. Tsang, "Guided Local Search," University of Essex, Colchester, Technical Report CSM-247 1995.

[17] F. Glover, "Tabu Search Fundamentals and uses," University of Colorado, Boulder, Colorado 1995.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, 1983.

[19] T. A. Feo and M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures," *Journal of Global Optimization*, vol. 6, pp. 109-133, 1995.

[20] P. Avgeriou and U. Zdun, "Architectural patterns revisited – a pattern language," *In 10th European Conference on Pattern Languages of Programs (EuroPlop 2005), Irsee,* pp. 1-39, 2005.

[21] R. M. A. Rasim M. Alguliev, Makrufa S. Hajirahimova, Chingiz A. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," 2011.

[22] P. Rattadilok, "An Investigation and Extension of a Hyper-heuristic Framework," *Slovenian Society Informatika,* pp. 523–534, 2009.

[23] M. Ryan, "A study of global inference algorithms in multi-document summarization," in *Proceedings of the 29th European conference on IR research* Rome, Italy: Springer-Verlag, 2007.

[24] G. R. Saggion H, "Multi-document summarization by cluster/profile relevance and redundancy removal," in *Proceedings of the Document Understanding Conference 2004* Boston, USA: NIST, 2004.

[25] B. Hachey, G. Murray, and D. Reitter, "The embra system at duc 2005: Query-oriented multi-document summarization with a very large latent semantic space (2005)," *in Proceedings of the Document Understanding Conference (DUC) 2005*, 2005.