



**UJAT**

UNIVERSIDAD JUÁREZ  
AUTÓNOMA DE TABASCO

“ESTUDIO EN LA DUDA. ACCIÓN EN LA FE”



DIVISIÓN  
ACADÉMICA DE  
INFORMÁTICA Y  
SISTEMAS

# Universidad Juárez Autónoma de Tabasco

Tesis de Maestría

## Generación de menús nutritivos basados en *frameworks* de metaheurísticas

Que presenta

**Isela Jiménez Castellano**

Para obtener el grado de

**Maestro en Ciencias de la Computación**

Directora

**Dra. Betania Hernández Ocaña**

**Dr. Oscar Alberto Chávez Bosquez**

Jurado:	<b>Dra. Juana Canul Reich</b>	Presidente
	<b>Dr. José Hernández Torruco</b>	Secretario
	<b>Dr. José Adán Hernández Nolasco</b>	Vocal

Cuerpo Académico  
**Inteligencia Artificial**

Línea de Generación y Aplicación del Conocimiento  
**Representación y manejo del conocimiento**

Cunduacán, Tabasco, México

Noviembre 2019

## Carta de autorización

El que suscribe, autoriza por medio del presente escrito a la Universidad Juárez Autónoma de Tabasco para que autorice tanto física como digitalmente la Tesis de Maestría "**Generación de menús nutritivos basados en frameworks de metaheurísticas**", de la cual soy autora y titular de los Derechos de Autor.

La finalidad de uso por parte de la Universidad Juárez Autónoma de Tabasco de la tesis antes mencionada, será única y exclusivamente para difusión, educación sin fines de lucro; autorización que se hace de manera enunciativa mas no limitativa para subirla a la Red Abierta de Bibliotecas Digitales (RABID) y a cualquier otra Red Académica con las que la Universidad tenga relación institucional.

Por lo antes mencionado, libero a la Universidad Juárez Autónoma de Tabasco de cualquier reclamación legal que pudiera ejercer respecto al uso y manipulación de la Tesis mencionada y para los fines estipulados en este documento.

Se firma la presente autorización en la ciudad de Villahermosa Tabasco a los 07 días del mes de noviembre del año 2019.

ATENTAMENTE,



ISELA JIMENEZ CASTELLANO



# UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

"ESTUDIO EN LA DUDA. ACCIÓN EN LA FE"

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS

Cunduacán, Tabasco a 31 de octubre de 2019.

En la Universidad Juárez Autónoma de Tabasco, de acuerdo al Reglamento de Estudios de Posgrado vigente, se revisó el trabajo de investigación titulado "GENERACIÓN DE MENÚS NUTRITIVOS BASADOS EN FRAMEWORKS DE METAHEURÍSTICAS", realizado por el C. **Isela Jiménez Castellano**, para obtener el Grado de Maestro en Ciencias de la Computación bajo la modalidad de Tesis.

Los integrantes del jurado, después de revisar el trabajo, lo declararon aceptado. Firmando la presente el 31 del mes de octubre de 2019.

  
Dra. Juana Canul Reich

  
Dr. José Hernández Torruco

  
Dr. José Adán Hernández Nolasco




Oficio No.2568/2019/DAIS-D  
6 de noviembre de 2019.

**C. Isela Jiménez Castellano**  
**Matricula 172H13001**

En virtud de que cumple satisfactoriamente los requisitos establecidos en el Reglamento General de Estudio de Posgrado vigente en la Universidad, informo a Usted que se autoriza la impresión del trabajo Recepcional **“GENERACIÓN DE MENÚS NUTRITIVOS BASADOS EN FRAMEWORKS DE METAHEURÍSTICAS”**, para presentar examen y obtener el Grado de Maestro en Ciencias de la Computación bajo la modalidad de Tesis

Sin otro particular, aprovecho la ocasión para saludarle.

Atentamente

  
M.T.E. Oscar Alberto Gonzalez Gonzalez  
Director



DIVISION ACADEMICA DE INFORMATICA Y SISTEMAS

C.c.p.- MASI. Arturo Corona Ferreira. - Encargado del Despacho de la Coordinación de Posgrado.  
Archivo  
Consecutivo.

# Índice general

Tabla de contenido	I
Índice de Figuras	III
Índice de Tablas	IV
Resumen	1
<b>1. Generalidades</b>	<b>2</b>
1.1. Introducción . . . . .	2
1.2. Planteamiento del problema . . . . .	4
1.2.1. Definición del problema . . . . .	4
1.3. Delimitación de la investigación . . . . .	5
1.3.1. Alcances . . . . .	5
1.3.2. Limitaciones . . . . .	5
1.4. Preguntas de investigación e hipótesis . . . . .	6
1.4.1. Preguntas de investigación . . . . .	6
1.4.2. Hipótesis . . . . .	7
1.5. Objetivos . . . . .	7
1.5.1. Objetivo general . . . . .	7
1.5.2. Objetivos específicos . . . . .	7
1.6. Justificación . . . . .	8
1.7. Metodología utilizada . . . . .	8
<b>2. Marco teórico</b>	<b>10</b>
2.1. Conceptos y teorías fundamentales de la investigación . . . . .	10
2.1.1. Generación de menús . . . . .	10
2.1.2. Información dietética y nutrición . . . . .	11

2.1.3.	Requerimiento energético . . . . .	14
2.1.4.	Nutrientes . . . . .	15
2.1.5.	Leyes de la correcta alimentación . . . . .	17
2.2.	Literatura relacionada . . . . .	17
2.2.1.	Generación de menús empleando técnicas de IA . . . . .	18
2.2.2.	Trabajos relacionados en la UJAT . . . . .	18
2.2.3.	Aplicación de <i>Frameworks</i> de metaheurísticas en problemas reales . . . . .	19
2.3.	Marco tecnológico . . . . .	20
2.3.1.	Java . . . . .	20
2.3.2.	Programación orientada a objetos . . . . .	20
2.3.3.	UML . . . . .	21
2.3.4.	Metaheurísticas . . . . .	21
2.3.5.	<i>Frameworks</i> de metaheurísticas . . . . .	22
<b>3.</b>	<b>Evaluación de <i>frameworks</i> de metaheurísticas</b>	<b>25</b>
3.1.	Análisis de <i>frameworks</i> . . . . .	25
3.2.	Prototipo Lanzador de Framework de Metaheurísticas (LFM)	26
<b>4.</b>	<b>Modelo del problema de la generación de menús nutritivos</b>	<b>34</b>
4.1.	Especificaciones del menú . . . . .	35
4.2.	Base de alimentos . . . . .	37
4.3.	Modelo matemático de la función objetivo y restricciones . . . . .	37
<b>5.</b>	<b>Experimentos y resultados</b>	<b>42</b>
5.1.	Diagrama de dominio del generador de menús nutritivos . . . . .	42
5.2.	Descripción del Prototipo Generador de Menús nutritivos (GM)	43
5.3.	Pruebas del prototipo GM . . . . .	48
5.4.	Discusión . . . . .	50
<b>6.</b>	<b>Conclusión, contribución y trabajos futuros</b>	<b>54</b>
6.1.	Conclusión . . . . .	54
6.2.	Contribución . . . . .	55
6.3.	Trabajos futuros . . . . .	55
6.4.	Producción académica derivada . . . . .	55
	<b>Bibliografía</b>	<b>57</b>

## Índice de figuras

2.1. Campaña “Chécate, mídete, muévete”. Fuente:IMSS-2013. . . . .	12
2.2. Plato del bien comer. Fuente: NOM-043-SSA2-2012. . . . .	14
3.1. Prototipo LFM. . . . .	27
3.2. Mejores resultados por instancia. . . . .	33
3.3. Media de los resultados por instancia. . . . .	33
5.1. Diagrama de dominio. . . . .	44
5.2. Generación del menú empleando el Prototipo GM. . . . .	46
5.3. Prototipo GM. . . . .	47

# Índice de tablas

2.1.	Menú con los 5 grupos del plato del bien comer. . . . .	13
2.2.	Distribución de macronutrientes y micronutrientes. . . . .	16
2.3.	Tesis relacionadas con la generación de menús nutritivos en la División Académica De Informática y Sistemas (DAIS). . . . .	19
3.1.	Resultados de los algoritmos de JACOF (SNF= Solución no factible, EMI= Error de memoria insuficiente). . . . .	29
3.2.	Resultados de los algoritmos de JAMES (SNF= Solución no factible). . . . .	30
3.3.	Resultados de los algoritmos de MOEA (SNF = Solución no factible). . . . .	31
3.4.	Resultados generales (SNF= Solución no factible, EMI= Error de memoria insuficiente). . . . .	32
4.1.	Porcentaje agregado al REI de acuerdo a la actividad física. . . . .	37
4.2.	Información general y nutrimental registrada en la base de datos de alimentos de la UJAT en el grupo de verduras. . . . .	37
4.3.	Número de alimentos por cada grupo en la base de alimentos. . . . .	38
5.1.	Características de 20 usuarios entre 18 y 60 años. . . . .	49
5.2.	Resultados obtenidos por el generador de menús para 20 usuarios (se indican los resultados del MOEA, el REI de los usuarios y las calorías requeridas en cada tiempo de comida). . . . .	50
5.3.	Diferencia porcentual de las calorías recomendadas por el generador de menús. . . . .	51
5.4.	Diferencia de error entre calorías. . . . .	52



# Resumen

En el presente trabajo se propone el desarrollo de un prototipo de software generador de menús nutritivos usando *frameworks* de metaheurísticas. Estos *frameworks* buscan optimizar un modelo matemático propuesto basado en las Leyes de la correcta alimentación; cuyo propósito es generar un menú nutritivo personalizado para los 5 tiempos de comida (desayuno, colación matutina, almuerzo, colación vespertina y cena).

Los *frameworks* fueron probados en el problema de la mochila específicamente en el prototipo LFM, el cual fue desarrollado para evaluar y analizar los *frameworks* de acuerdo a su funcionalidad, usabilidad y eficiencia. El prototipo hace uso de métodos metaheurísticos de búsqueda global, computación evolutiva e inteligencia colectiva.

Como propuesta para resolver el problema de la generación de menús se utilizó el *framework* MOEA, del cual se seleccionó el algoritmo NSGA-II. Se desarrolló el prototipo GM para generar un menú nutricional personalizado que sea variado y equilibrado desde el punto de vista nutricional, además de considerar dietas específicas con platillos del sureste mexicano.

Al generar el menú se consideraron las características de los usuarios y alimentos específicos de la región, para cumplir con el modelo propuesto y con las especificaciones del menú. Los menús generados por el algoritmo contenían los alimentos adecuados para el usuario y cumplen con las calorías recomendadas durante el día.

El diseño presenta una interfaz sencilla e intuitiva que no requiere de conocimientos previos en computación u optimización, además de contribuir al conocimiento de las necesidades nutricionales en la sociedad.

# Capítulo 1

## Generalidades

### 1.1. Introducción

En la actualidad uno de los principales problemas de la sociedad es la forma inadecuada de alimentarse. El ritmo de vida acelerado de las personas que permanecen en constante movimiento, en ocasiones no permite dedicar suficiente tiempo a la adquisición, preparación y consumo de los alimentos.

Sin darse cuenta, las personas han adoptado rutinas en las que ignoran sus necesidades y mantienen malos hábitos alimentarios. Tal es el caso de comer en poco tiempo y consumir alimentos calificados como “comida rápida”.

Cuando se habla de una buena o mala alimentación, se trata de un acto voluntario en la elección de los alimentos, mientras que la nutrición son todos los procesos que se dan en el organismo tras la ingesta de los alimentos. Los alimentos aportan nutrientes energéticos que permiten el correcto funcionamiento de nuestro organismo y la continuidad de la vida [41].

Existe evidencia [10] de que los estilos de vida condicionan el nivel de salud y dentro de los hábitos saludables una alimentación equilibrada es fundamental. Para controlar el consumo de alimentos y bebidas se crearon las dietas [61], éstas contienen diferentes raciones y porciones de los alimentos de acuerdo a la región y adecuadas a una persona para perder, aumentar o mantener el peso.

Por ello, una elección acertada de los alimentos en el contexto de una dieta variada, equilibrada y en cantidad suficiente, permitirá el desarrollo correcto del individuo [19].

Para generar un menú adecuado al usuario, los especialistas en nutrición

han tomado en cuenta las Leyes de la correcta alimentación: Ley de la cantidad, Ley de la calidad, Ley de la armonía y Ley de la adecuación [3]. El cumplimiento de las 4 Leyes en conjunto permiten que la dieta sea balanceada y beneficiosa para mantener una buena salud [65].

En el contexto de las Ciencias de la computación, el problema de la generación de menús es relacionado frecuentemente con el Problema de la Mochila multidimensional (KP, por sus siglas en inglés *Knapsack Problem*) [57], el cual es un problema clásico de Optimización Combinatoria (COP por las siglas en inglés de *Combinatorial Optimization Problem*) que pertenece a la familia de problemas NP-completos.

Se puede decir que esta clase de problemas (también conocidos informalmente como intratables), son los más difíciles de resolver ya que un algoritmo de fuerza bruta para problemas NP-Completos utiliza tiempo exponencial con respecto al tamaño de la entrada. Se desconoce si hay mejores algoritmos, por lo cual es necesario utilizar diferentes enfoques que permitan obtener una solución adecuada en un tiempo razonable [20].

En el área de Inteligencia artificial (IA) existen diversos métodos que permiten resolver este tipo de problemas, una manera de solucionarlos es aplicando algoritmos heurísticos. Las heurísticas son métodos o principios que permiten obtener soluciones de alta calidad a un costo razonable de recursos [31, 54, 56]. Se utilizan en diversos campos como la robótica, los problemas de satisfacción de restricciones, entre otros [35].

Las metaheurísticas designan un método computacional para la resolución de problemas, en ocasiones implementando alguna forma de optimización estocástica. Aunque no garantizan que se encuentre una solución óptima [1], pueden hacer pocas suposiciones sobre el Problema de Optimización (PO) que se está resolviendo y son útiles para gran variedad de problemas [11].

Las metaheurísticas han alcanzado un alto prestigio, como demuestran la amplia gama de problemas a los que se han aplicado con éxito en la literatura y al gran número de revistas, libros y conferencias dedicados a este tema. Se han adaptado en contextos como: minería de datos, física, biología, logística, control y procesamiento de imágenes, entre otros [25].

Debido al auge que tiene el uso de metaheurísticas para resolver PO, existen *frameworks* que implementan gran variedad de algoritmos, con los cuales se reutiliza código existente, se agiliza el proceso de desarrollo y se promueven buenas prácticas de programación.

Un *framework* también conocido como entorno o marco de trabajo, es un conjunto de prácticas empleadas en el desarrollo de sistemas de software para

enfrentar y resolver problemas [30, 58].

De acuerdo a lo anterior, se propone implementar un prototipo de software que permita generar menús nutritivos aplicando *frameworks* de metaheurísticas. El documento se encuentra organizado en 6 capítulos, cuyo contenido se describe a continuación:

En el Capítulo 1 se establecen los siguientes aspectos: descripción del problema a resolver, objetivos, preguntas, hipótesis, justificación y la metodología utilizada durante la investigación.

El Capítulo 2 describe el estado del arte en el contexto de las Ciencias de la Computación, específicamente en el área de la Inteligencia Artificial, y presenta una revisión bibliográfica de los conceptos principales como la generación de menús nutritivos, las Leyes de la correcta alimentación y aspectos tecnológicos del área.

El Capítulo 3 muestra la comparación de *frameworks* de metaheurísticas, las pruebas realizadas y los resultados obtenidos para la selección del *framework* aplicado en la generación de menús.

En el Capítulo 4 se presenta la formulación del problema, especificaciones generales, descripciones acerca del diseño y desarrollo del prototipo generador de menús, la base de datos y otros elementos que fundamentan la propuesta.

En el Capítulo 5 inicia con el desarrollo y diseño del prototipo generador de menús nutritivos personalizados como propuesta de solución.

En el Capítulo 6 se explican las conclusiones, contribuciones y trabajos futuros.

## 1.2. Planteamiento del problema

### 1.2.1. Definición del problema

En la generación de menús nutritivos los expertos en nutrición requieren de citas previas para conocer las características del usuario (edad, peso, género, talla, actividad física, preferencias e intolerancias). Al obtener estos datos, el especialista recomienda los alimentos que debe consumir. Este proceso generalmente se elabora de forma manual y ocasiona demoras en el tiempo de elaboración y entrega del menú.

Generar menús nutritivos es un proceso complejo, ya que al incluir variedad de alimentos en cantidades adecuadas, preferencias y otros aspectos

en nutrición, el número de restricciones que debe analizar el especialista aumenta.

Realizar un menú de manera diaria, semanal o mensual requiere de tiempo, dedicación y esfuerzo. La computación ha venido a automatizar y optimizar diversas tareas en muchas áreas. Una propuesta de optimización es la generación de menús con ayuda de la computación, ya que existen metaheurísticas que permiten optimizar este tipo de problemas.

La generación de menús puede modelarse de manera numérica o combinatoria, no obstante es más efectivo hacerlo de manera numérica. Al optimizar el consumo energético diario de una persona se puede aplicar como técnica de IA las metaheurísticas: Algoritmos evolutivos (AE), Búsqueda global (BG) o Inteligencia colectiva (AIC).

Actualmente existen aplicaciones de apoyo al nutriólogo que permiten conocer la cantidad de nutrientes que contienen los alimentos y llevar un control de cada paciente. Sin embargo, no existe una herramienta o software que sea de acceso libre y que valide las Leyes de la correcta alimentación. En consecuencia, se propone una solución al problema utilizando *frameworks* de metaheurísticas y las Leyes de la correcta alimentación.

## 1.3. Delimitación de la investigación

### 1.3.1. Alcances

- El prototipo desarrollado es una aplicación de escritorio multiplataforma con una base de datos en formato CSV (del inglés *comma separated values*).
- Se analizaron tres *frameworks* de metaheurísticas citados en la literatura.
- Se usó una base de datos de alimentos diseñada en la División Académica de Ciencias de la Salud de la UJAT.
- Se seleccionó un conjunto específico de algoritmos por cada *framework*.

### 1.3.2. Limitaciones

- Los menús generados son para pacientes sin patologías entre 18 y 60 años de edad.

- Debido a que se trata de un proyecto de investigación no comercial, solo se utilizan *frameworks* de acceso libre.
- Solo se analizan *frameworks* implementados en lenguaje multiplataforma, específicamente sobre la plataforma Java Standard Edition.
- El prototipo está limitado a la funcionalidad del *framework*.

## 1.4. Preguntas de investigación e hipótesis

### 1.4.1. Preguntas de investigación

- ¿Cómo modelar el problema de generación de menús basado en las Leyes de la correcta alimentación, para que funcione con un *framework* de metaheurísticas?

Dado que cada *framework* utiliza su propio formato para representar el problema a resolver, será necesario evaluar los *frameworks* y seleccionar el que obtenga las mejores soluciones, así como diseñar un modelo genérico que contemple los elementos principales del proceso de generación de menús junto con las Leyes de la correcta alimentación en forma de restricciones, y que pueda servir como entrada para el *framework* a utilizar.

- ¿Bajo qué *framework* de metaheurísticas es más sencillo implementar el modelo para la generación de menús?

Desde el punto de vista del desarrollo de software, interesa conocer qué *framework* reduce el tiempo de desarrollo a partir de un problema genérico, es decir encontrar el *framework* con el mejor diseño. A la vez que el *framework* seleccionado permita integrar el modelo basado en las Leyes de la correcta alimentación de forma más adecuada. Al utilizar un *framework* se tiene la desventaja de que el desarrollador debe adaptarse a su especificación, sin embargo esto se convierte en una ventaja mayúscula si el *framework* está bien diseñado.

- ¿El *framework* de metaheurísticas seleccionado genera menús nutritivos válidos?

Es necesario identificar si el menú generado por el *framework* cumple con las Leyes de la correcta alimentación y satisface al máximo las restricciones del usuario.

### 1.4.2. Hipótesis

*A partir de la evaluación de diferentes frameworks de metaheurísticas es posible generar menús nutritivos personalizados que cumplan con las Leyes de la correcta alimentación.*

La hipótesis de investigación contempla las restricciones de los usuarios y compromete a generar menús personalizados en un tiempo razonable, considerando las Leyes de la correcta alimentación.

## 1.5. Objetivos

### 1.5.1. Objetivo general

Generar menús nutritivos basados en las Leyes de la correcta alimentación empleando *frameworks* de metaheurísticas.

### 1.5.2. Objetivos específicos

- Evaluar los *frameworks* de metaheurísticas en el problema de la mochila.
- Integrar las Leyes de la correcta alimentación en un modelo de optimización basado en restricciones que funcione con un *framework* de metaheurísticas.
- Construir un prototipo de software que genere menús nutritivos personalizados.

## 1.6. Justificación

La investigación acerca de la generación de menús nutritivos permite conocer las aplicaciones de software en nutrición existentes, comparar las características de los usuarios consideradas en cada desarrollo y las técnicas de IA aplicadas en el problema.

La generación del menú se complica al realizar combinaciones de alimentos (platos/bebidas), por lo que la integración de las Leyes de la correcta alimentación son importantes para calcular y clasificar los menús correctamente. Además, las metaheurísticas constituyen una alternativa importante para resolver esta clase de problemas.

Este problema ha sido ampliamente estudiado a lo largo de los años, en la literatura se ha encontrado evidencia de trabajos relacionados para usuarios con patologías o aplicaciones de apoyo al nutriólogo, aunque estas tecnologías carecen de información o solo permiten llevar un control de los usuarios.

Los nutriólogos tienen diferentes maneras de generar menús nutritivos, siendo su principal apoyo el manejo de tablas que representan las raciones que debe consumir una persona durante el día, el reparto de los grupos de alimentos, entre otras estrategias. El nutriólogo puede afrontar la complejidad de la planificación de menús e intentar satisfacer los requerimientos del usuario, tiempo, costos y calidad.

En este documento se plantea la creación de un prototipo de software que genere menús personalizados que incluyan las 5 comidas del día (desayuno, colación matutina, almuerzo, colación vespertina y cena), incluyendo un conjunto de recetas de la región mexicana y *frameworks* de metaheurísticas como solución.

## 1.7. Metodología utilizada

Esta investigación es de tipo mixta, aborda un estudio cuantitativo acerca de los datos proporcionados en las encuestas de alimentación y nutrición en México, así como cualitativa al incluir descripciones y características acerca de la generación de menús. Los pasos a seguir son los siguientes:

- Revisión y análisis de información acerca de las Leyes de la correcta alimentación y la generación de menús.
- Análisis y selección de *frameworks* de metaheurísticas disponibles.



- Diseño basado en restricciones del modelo que cumpla con las Leyes de la correcta alimentación.
- Implementación del modelo sobre el *framework* de metaheurísticas seleccionado.
- Desarrollo de un prototipo de software multiplataforma para la generación de menús nutritivos.
- Integración de la base de datos con platillos del sureste mexicano.

# Capítulo 2

## Marco teórico

### 2.1. Conceptos y teorías fundamentales de la investigación

En esta sección se presentan los conceptos de la investigación como la generación de menús, las Leyes de la correcta alimentación, teoría general (nutrición, alimentación, requerimiento energético), hasta la revisión de la literatura y aspectos tecnológicos del área.

#### 2.1.1. Generación de menús

En la actualidad las personas dedican poco tiempo a planificar el menú de la semana, esta situación conduce a que se consuman con frecuencia platillos precocinados o alimentos con raciones inadecuadas. Para planificar un menú equilibrado, nutritivo y saludable se debe tener en cuenta el número de comidas o ingestas que se requiere distribuir en un menú diario.

La cultura de cada país establece que las comidas se componen de algún líquido, de preparaciones frías o calientes, saladas o dulces. Por lo tanto, de acuerdo a las preferencias del usuario se piensa que para que una comida sea completa debe contener los micronutrientes y macronutrientes necesarios para satisfacer las necesidades del cuerpo humano [37].

Un menú es un plan de alimentación que incluye alimentos y preparaciones para satisfacer los requerimientos nutricionales de las personas, este se divide en raciones para la obtención de los nutrientes necesarios [24].

### 2.1.2. Información dietética y nutrición

Los hábitos alimentarios comienzan en los primeros años de vida. Estos son comportamientos conscientes, colectivos y repetitivos que conducen a las personas a seleccionar, consumir y utilizar determinados alimentos o dietas, a partir de la influencia social y cultural.

En el mundo las dietas insalubres y la falta de actividad física están entre los principales factores de riesgo para la salud, una dieta saludable ayuda a protegernos de las enfermedades y de la malnutrición en todas sus formas.

El estado de salud de una persona depende de la calidad de la nutrición, en este proceso se deben consumir alimentos de todos los grupos alimenticios para aportar los nutrientes necesarios en cantidades adecuadas. Por lo tanto, si se quiere mejorar el estado nutricional se deben mantener buenos hábitos alimenticios [49].

Para conseguir y mantener el funcionamiento óptimo del organismo es necesario comer sano. Esta es una necesidad básica condicionada por la época, la región en la que viven, el gusto personal y la disponibilidad de alimentos, lo que a su vez está limitado por el ambiente, la tecnología y el nivel socioeconómico [26].

La OMS define la nutrición como la acción de ingerir alimentos en relación con las necesidades dietéticas del organismo. “Una buena nutrición (una dieta suficiente y equilibrada combinada con el ejercicio físico regular) es un elemento fundamental de la buena salud”.

La alimentación consiste en suministrar alimentos al organismo e incluye la selección, preparación e ingestión de alimentos que proporcionan nutrientes y vitaminas [45].

ENSANUT es un proyecto del Instituto Nacional de Salud Pública y la Secretaría de Salud Federal, que permite conocer cuál es el estado de salud y las condiciones nutricionales de los diversos grupos que forman la población mexicana.

La ENSANUT MC [23] consiste en una serie de preguntas sobre el estado de salud y nutrición. Abarca el consumo de alimentos, servicios de salud y programas sociales de ayuda alimentaria a los que se tiene acceso. Entre las actividades más recientes se evaluó la exposición de la población a la campaña “CHÉCATE, MÍDETE, MUÉVETE”<sup>1</sup> presentada en la Figura 2.1 e implementada en el país con un diseño estratégico basado en la promoción de la salud.

<sup>1</sup><https://checatemitetemuevete.gob.mx>



Figura 2.1. Campaña “Chécate, mídete, muévete”. Fuente:IMSS-2013.

- “CHÉCATE” promueve el asistir a su clínica de salud para conocer su peso y medir la circunferencia de su cintura.
- “MÍDETE” hace referencia a disminuir el consumo de grasas, azúcares y sal en sus comidas.
- “MUÉVETE” se refiere a realizar ejercicio cotidianamente.

Esta iniciativa intenta promover, fortalecer hábitos y estilos de vida saludables en las personas, con una orientación novedosa y diferente para el beneficio de la salud individual y colectiva.

En la Tabla 2.1 se presenta un menú saludable que incluye las recomendaciones del plato del bien comer en recetas elaboradas por la nutrióloga Xaviera Cabada, coordinadora de Salud Alimentaria en “El poder del Consumidor<sup>2</sup>”.

El plato del bien comer [13] es una guía de alimentación que forma parte de la Norma Oficial Mexicana para la promoción y educación para la salud en materia alimentaria, la cual establece criterios para la orientación nutritiva en México. Consiste de los elementos mostrados gráficamente en la Figura 2.2, dirigido a brindar a la población opciones prácticas con respaldo científico, para tener una alimentación correcta que pueda adecuarse a sus necesidades y posibilidades.

Además, facilita la identificación de los tres grupos de alimentos (verduras y frutas, cereales y tubérculos, leguminosas y proteínas de origen animal), la combinación y variación de la alimentación, la selección de menús diarios con los tres grupos de alimentos y el aporte de energía y nutrientes a través de la alimentación correcta.

<sup>2</sup><https://elpoderdelconsumidor.org/2016/01/un-menu-practico-saludable-y-barato-67>

**Tabla 2.1.** Menú con los 5 grupos del plato del bien comer.

---

<b>Desayuno</b>
Omelet con espinacas frescas con garbanzo de guarnición. Tortilla de maíz nixtamalizado. Té o café al gusto.
Grupos: Verduras, leguminosas, alimentos de origen animal y cereales. Precio aproximado: \$ 17.50 pesos.
<b>Almuerzo</b>
Sopa de lentejas con pan integral tostado en trozos. Guisado de germinado de soya con vegetales, sancochados en aceite de oliva. Agua fresca de fruta o agua simple.
Grupos: Leguminosas, cereales y verduras. Precio aproximado: \$21 pesos (por una porción grande).
<b>Cena</b>
Pan integral de amaranto con avena y un plátano en rebanadas. Licuado de nuez (puede contener leche de vaca, almendra, soya, arroz o coco).
Grupos: Frutas, verduras, leguminosas, alimentos de origen animal y cereales. Precio aproximado: \$18 pesos.
<b>Colación</b>
Piñones. Precio aproximado: \$8 pesos.

---

## El Plato del Bien Comer

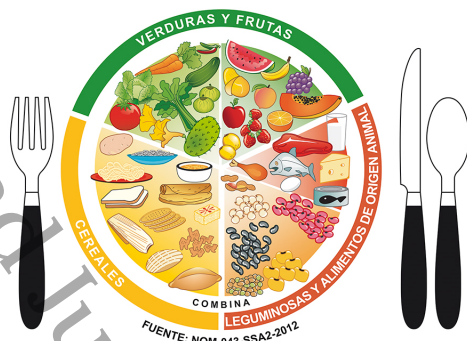


Figura 2.2. Plato del bien comer. Fuente: NOM-043-SSA2-2012.

### 2.1.3. Requerimiento energético

El requerimiento energético se refiere a la cantidad de energía adquirida a través de los alimentos, capaz de cubrir las exigencias calóricas del organismo y conservar un buen estado de salud [63].

Si el consumo de energía a corto plazo es superior o inferior al gasto, es posible que se origine una transformación en las reservas de energía del cuerpo. Si este desequilibrio se conserva en un plazo prolongado, se pueden producir cambios en la composición corporal y efectos negativos en la salud [64].

La cantidad de energía de los alimentos se expresa en calorías (cal) y la unidad internacional para medir dicha energía es el joule (J), siendo:

$$1\text{cal} = 4.184\text{J} \quad (2.1)$$

El gasto energético total diario de una persona representa la energía que el organismo consume. Está constituido por la suma del gasto energético basal, efecto termógeno de los alimentos y la actividad física, que se describen a continuación [9]:

- Gasto energético basal:** Es el gasto energético por unidad de tiempo de una persona en condiciones basales, estar saludable, en total reposo físico, mental, en un ambiente tranquilo y con temperatura agradable. Así medido, representa el gasto de energía mínimo para mantener las funciones vitales del organismo.

- **Efecto termógeno de los alimentos:** Designa el incremento de gasto calórico después de los procesos que se presentan en el consumo de los alimentos (digestión, absorción, transporte, almacenamiento de alimentos y nutrientes).
- **Gasto por actividad física:** Describe la cantidad de energía que consume el cuerpo y representa el elemento más variable del gasto energético total.

#### 2.1.4. Nutrientes

Los nutrientes o nutrimentos son sustancias contenidas en los alimentos que consumimos para cubrir las necesidades del cuerpo [45].

##### Nutrientes esenciales y no esenciales

- **Agua:** El organismo es completamente dependiente del agua, está es una sustancia inorgánica compuesta por dos átomos de hidrógeno y uno de oxígeno (H<sub>2</sub>O). Se estima que la ingesta de agua debe estar relacionada con la ingesta de energía. Las necesidades aumentan cuanto mayor es la actividad, la sudoración o la carga de solutos de la dieta [44].
- **Azúcar:** Es un carbohidrato que el cuerpo convierte en glucosa. La glucosa es el hidrato de carbono más importante, es imprescindible para el funcionamiento del cerebro y para obtener energía [53].
- **Alcohol:** Es una fuente de caloría vacía y no es un nutriente. Cuando se consume en exceso este puede interferir gravemente con el estado nutricional, ya sea alterando la ingestión del alimento, su absorción o la utilización de los nutrientes por el organismo [50].

##### Macronutrientes

El organismo necesita una adecuada cantidad de macronutrientes (gramos) para funcionar correctamente, también pueden ser llamados nutrientes proveedores de energía <sup>3</sup>. La energía se mide en calorías y es esencial pa-

<sup>3</sup>Algunos autores consideran el agua como un macronutriente

**Tabla 2.2.** Distribución de macronutrientes y micronutrientes.

Tipo de nutriente	Grupo	Clasificación química	Calorías (kcal/gr)
Carbohidratos	Macronutriente	Orgánico	4
Lípidos	Macronutriente	Orgánico	4
Proteínas	Macronutriente	Orgánico	9
Vitaminas	Micronutriente	Orgánico	-
Minerales	Micronutriente	Inorgánico	-

ra el crecimiento, reparación y desarrollo de nuevos tejidos, conducción de impulsos nerviosos y regulación de procesos corporales.

En esta categoría se incluyen [27, 45, 59]:

- **Carbohidratos (Hidratos de carbono)** : Forman la mayor parte de los alimentos almacenados en el cuerpo y constituyen una parte fundamental de la alimentación humana, para el posterior uso de la energía. Su función principal es ser un excelente combustible metabólico, son importantes para la oxidación de las grasas y también se pueden convertir en proteínas.
- **Proteínas:** Son compuestos orgánicos que constituyen la mayor parte de la estructura celular. Funcionan como enzimas, queratina, anticuerpos, ayudan al crecimiento y aumentan el sistema inmune. Sus unidades constituyentes son los aminoácidos, fundamentales para reparar y mantener el cuerpo humano.
- **Lípidos (Grasas)** : Sirven como solventes para las hormonas y las vitaminas liposolubles. Su función principal es la reserva energética, ya que en condiciones de inanición proporcionan energía. Son altamente necesarios para el funcionamiento apropiado de las membranas celulares y mantener estable la temperatura corporal.

Los nutrientes esenciales en la Tabla 2.2 muestran las calorías y la clasificación química de los macronutrientes y micronutrientes.



## Micronutrientes

Los micronutrientes son requeridos en pequeñas cantidades y no proporcionan energía al cuerpo. Estos son extremadamente importantes para la actividad normal y para facilitar muchas reacciones químicas que ocurren dentro del organismo humano e incluyen [45]:

- **Vitaminas:** Son nutrientes orgánicos que se requieren en cantidades pequeñas para el funcionamiento normal del metabolismo (crecimiento y desarrollo) y para la regulación de la función celular. Las vitaminas, las enzimas y otras sustancias deben tomarse en la dieta, no pueden ser sintetizados por el cuerpo y son esenciales para mantener la salud.
- **Minerales:** Son sustancias inorgánicas necesarias para la activación de las enzimas, se requieren en pequeñas cantidades y forman parte de la estructura de los tejidos en el cuerpo humano.

### 2.1.5. Leyes de la correcta alimentación

El Dr. Pedro Escudero considerado el padre de la nutrición en Latinoamérica, propuso en 1935 las Leyes de la correcta alimentación [3, 7, 52]. Describió de manera formal en el Capítulo 4 y presentadas a continuación:

- **Ley de la cantidad:** Establece que la cantidad de alimentos ingerida debe proporcionar la energía que el cuerpo necesita.
- **Ley de la calidad:** Indica que se deben consumir alimentos de todos los grupos.
- **Ley de la armonía:** Señala que los nutrientes ingeridos deben ser distribuidos adecuadamente en el organismo.
- **Ley de la adecuación:** Consiste en adecuar la alimentación a las necesidades de las personas, ya sean nutritivas, sociales o psicológicas.

## 2.2. Literatura relacionada

En esta sección se presenta una revisión de la literatura que incluye trabajos recientes que describen el problema de la generación de menús y plantean una posible solución utilizando metaheurísticas.

### 2.2.1. Generación de menús empleando técnicas de IA

Los trabajos más citados en el tema de la generación de menús empleando metaheurísticas se describen a continuación:

En el trabajo de investigación “Planificación de menús dietéticos utilizando un método evolutivo”[4] se propone una solución al problema de menús mediante el algoritmo NSGA-II, para generar menús semanales compuestos por 5 comidas diarias en personas adultas sin requisitos dietéticos en un hospital local. Se utilizó una base de datos de composición de alimentos de Eslovenia para obtener un menú en función del coste, la calidad y funcionalidad de los alimentos. La autora desarrollo un trabajo posterior [5] como contribución del mismo.

Otro trabajo que utiliza el algoritmo NSGA-II es el [42] mediante una formulación multiobjetivo que incluye el coste del alimento y las preferencias del usuario. Los autores consideran las especificaciones del usuario para generar un menú adecuado de acuerdo a sus características.

En el artículo [15] se utiliza un algoritmo de programación lineal y un operador de fusión de creencias (*belief merging*) para reforzar la base de conocimiento. En este trabajo se consideran los aspectos del usuario como: preferencias alimenticias, género, intolerancias, técnicas de preparación del alimento y otras características.

Posteriormente en el artículo [16], presentan un modelo de dominio completo para la generación de menús nutritivos que cumplen con las 4 Leyes de la correcta alimentación. La verificación y validación formal del modelo se realizó en USE (*UML-based Specification Environment*) y se desarrolló un prototipo de software para generar el menú.

Los trabajos de investigación presentados han sido probados con distintas técnicas de IA y abordados en la generación de menús con diferentes platos y con restricciones en los gustos personales de los usuarios, así como la creación de menús equilibrados y con dietas personalizadas.

### 2.2.2. Trabajos relacionados en la UJAT

Las metaheurísticas se han usado ampliamente en la generación de menús nutritivos en la Universidad. Recientemente existe inquietud por este tema, la UJAT tiene interés particular en resolver este problema y se ha demostrado con esfuerzos a través de las tesis de licenciatura y posgrado agrupadas en la Tabla 2.3.

**Tabla 2.3.** Tesis relacionadas con la generación de menús nutritivos en la División Académica De Informática y Sistemas (DAIS).

Nombre de la tesis	Estrategia de solución	Requerimiento energético	Macro-nutrientes	Micro-nutrientes	Preferencias
Generación automatizada de menús nutricionales utilizando Búsqueda Tabú.	Búsqueda Tabú	✓	✓	-	-
Generación automatizada de menús nutricionales utilizando Algoritmos Genéticos.	Algoritmos Genéticos	✓	✓	-	-
Optimización Global con Res-tricciones usando el Algoritmo Basado en el Forrajeo de Bacterias.	Forrajeo de Bacterias	✓	✓	-	-
Modelando la generación de menús nutritivos empleando técnicas de Inteligencia Artificial.	Búsqueda Tabú, Fusión de Creencias	✓	✓	✓	✓

Como se puede observar en la tabla, los primeros trabajos no consideran las Leyes de la correcta alimentación solo el requerimiento energético y los macronutrientes. En un solo trabajo se incluyeron las leyes, los macronutrientes y micronutrientes con dos técnicas de IA en la solución del problema.

Los trabajos que se han realizado en la UJAT utilizan distintos algoritmos para resolver el problema, aunque actualmente ninguno ha sido probado con pacientes reales. En los proyectos de tesis presentados se realizaron prototipos generadores de menús que fueron probados con diferente número de usuarios (adultos entre 18 y 60 años) sanos (sin patologías).

Los programas desarrollados fueron validados utilizando lenguajes de modelado como UML, además de realizar pruebas de usabilidad que en la mayoría de los casos no contemplan todas las variables de la generación de menús.

### 2.2.3. Aplicación de *Frameworks* de metaheurísticas en problemas reales

En la literatura existen una gran cantidad de trabajos relacionados con el estudio y análisis de *frameworks* para resolver problemas de optimización, los cuales se presentan a continuación:

El trabajo [21] presenta una descripción generalizada del *framework* JAMES, incluye la implementación de un problema de selección con componen-

tes predefinidos, el algoritmo *hill climbing* y la ejecución de un algoritmo de descenso aleatorio.

En [66] se enfocan en estudiar los desarrollos recientes en los MOEA y en MOP. Este estudio despliega un análisis de *frameworks* evolutivos, diferentes MOEAs y MOP dinámicos, costosos y ruidosos. Además de problemas de referencia, funcionalidades y medidas de rendimiento de los algoritmos.

En [2], proponen un enfoque basado en el Algoritmo de colonia de hormigas (ACO) para resolver problemas multiobjetivo. En el cual se plantea un algoritmo ACO genérico con cuatro instancias, se describe el KP y se emplea el algoritmo para resolverlo. Los resultados obtenidos con ACO fueron comparados con AE para elegir cuál de las variantes de ACO se adapta mejor al problema.

## 2.3. Marco tecnológico

### 2.3.1. Java

Java es un lenguaje de programación orientado a objetos y una plataforma informática, comercializada por primera vez en 1995 por *Sun Microsystems*. Hay muchas aplicaciones y sitios web que requieren Java, este lenguaje es uno de los más populares en uso ya que es rápido, seguro y fiable. Fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (WORA, por la siglas en inglés *write once, run anywhere*), lo que quiere decir el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra [36].

### 2.3.2. Programación orientada a objetos

La programación orientada a objetos (POO) es un paradigma de programación basado en varias técnicas: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Los objetos son entidades que tienen un determinado estado (atributo), comportamiento (método) e identidad, y manipulan los datos de entrada para la obtención de datos de salida específicos [18].

- Los atributos y los métodos están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta.
- La identidad es una propiedad de un objeto (identificador de una variable o una constante).
- Los métodos son mecanismos de interacción entre los objetos, esta comunicación favorece el cambio de estado en los propios objetos y lleva a tratarlos como unidades indivisibles en las que no se separa el estado y el comportamiento.

### 2.3.3. UML

El lenguaje unificado de modelado o UML (*Unified Modeling Language*), es una representación gráfica de métodos que describen la estructura y el comportamiento de los objetos relativos al desarrollo de sistemas de información, permite modelar y describir formalmente tales sistemas para que puedan ser entendidos por diferentes usuarios [28, 60].

Algunas de las propiedades de UML como lenguaje de modelado estándar son:

- Concurrencia: es un lenguaje adecuado y distribuido a las necesidades de conectividad actual y futura.
- Modela estructuras complejas y es ampliamente utilizado en la industria de software.
- Las estructuras más importantes que soportan tienen su fundamento en el lenguaje orientado a objetos, tales como: clases, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones que sirven para evaluar el estado de las máquinas.

### 2.3.4. Metaheurísticas

En el trabajo “Metaheurísticas: una revisión actualizada” se definen los siguientes conceptos [12]:

- El término heurística se emplea para referirse a una técnica, método o procedimiento que trata de aportar soluciones a un problema con un buen rendimiento, proviene de una palabra griega relacionada con el concepto de “encontrar” y se vincula a la supuesta exclamación “eureka” de Arquímedes.
- El término metaheurísticas se obtiene de anteponer a heurística el sufijo “meta” que significa “más allá” o “a un nivel superior”, apareció por primera vez en el artículo seminal sobre búsqueda tabú de Fred Glover en 1986. Desde entonces han surgido multitud de propuestas para diseñar buenos procedimientos para resolver ciertos problemas que al ampliar su campo de aplicación, han adoptado la denominación de metaheurísticas.
- Las metaheurísticas son estrategias generales de diseño de procedimientos heurísticos para la resolución de problemas con un alto rendimiento. Existen metaheurísticas centradas en el uso de algún tipo de recurso computacional como las redes neuronales, los sistemas de hormiga o la programación por restricciones.

En la literatura se encuentran dos grandes grupos de metaheurísticas, las basadas en la evolución natural de las especies conocidos como AE [48] y los AIC, los cuales simulan el comportamiento de ciertas especies simples e inteligentes en la búsqueda de alimento o refugio [22].

Los más populares son los AE debido a su éxito en la solución de PO, entre los cuales se encuentran los Algoritmos Genéticos (AG) [34], además de algunas propuestas de AIC que han tomado relevancia en estos últimos años como los Algoritmos de Colonia de Hormigas (ACO).

Algunas metaheurísticas conocidas son BG, que permiten una búsqueda exhaustiva del óptimo en un espacio de búsqueda, el cual es definido por las variables de decisión y las restricciones del problema a resolver [40].

### 2.3.5. *Frameworks* de metaheurísticas

Un *framework* es un conjunto de prácticas empleadas en el desarrollo de software para enfrentar y resolver problemas de forma sencilla, segura y mantienen un comportamiento útil, definido e identificable [58]. Existen *frameworks* para multitud de rutinas y algoritmos en prácticamente cualquier

lenguaje de programación, el caso de las metaheurísticas no es la excepción [21, 38, 51].

Los *Frameworks* son utilizados ampliamente en las áreas de computación, ya que reducen el tiempo de elaboración e implementación de cualquier tipo de desarrollo [46]. Los *frameworks* de metaheurísticas existentes implementan gran variedad de algoritmos heurísticos de diferente tipo: AE, BG, AIC, entre otros. Los *Frameworks* a utilizar son los siguientes:

- **Java Metaheuristics Search framework (JAMES)** [32]: es un *framework* para la optimización discreta usando metaheurísticas de búsqueda global. Este *framework* proporciona guías de implementación para una serie de problemas, donde el rendimiento de los algoritmos y la influencia de los valores de los parámetros se pueden estudiar fácilmente, esto ocurre al definir el problema, seleccionar una estrategia de optimización adecuada y aplicar un algoritmo de búsqueda para encontrar la mejor solución. JAMES está compuesto de tres módulos:
  1. Principal: Proporciona una amplia variedad de algoritmos de BG como: Descenso Aleatorio (*Random Descent - RD*), Búsqueda de Vecindad Variable (*Variable Neighbourhood Search - VNS*), Templado Paralelo (*Parallel Tempering - PT*), entre otros.
  2. Extensiones: Proporciona herramientas adicionales para la especificación de problemas.
  3. Ejemplos: Proporciona ejemplos de problemas y soluciones aplicando diferentes algoritmos de optimización.

La documentación y el sitio web de JAMES es completamente de código abierto, el código fuente está bajo Licencia Permisiva de Apache 2.0, la documentación y el sitio web se encuentran publicados bajo la Licencia Internacional Creative Commons Attribution 4.0.

- **MultiObjective Evolutionary Algorithms (MOEA)** [38]: es un *framework* para desarrollo y experimentación con algoritmos evolutivos multiobjetivo (MOEAs). Su objetivo es proporcionar una colección completa de algoritmos y herramientas para la optimización de objetivos únicos y multiobjetivos.

Dentro de la clasificación de AG, MOEA incluye los siguientes: Algoritmo Genético de Ordenación No Dominado II (*Non-dominated Sorting*

*Genetic Algorithm II - NSGA-II*); Algoritmo Genético de Ordenación No Dominado Basado en Puntos de Referencia (*Reference-Point Based Non-dominated Sorting Genetic Algorithm - NSGA-III*); Algoritmo Genético con Elitismo (objetivo único) (*Genetic Algorithm with Elitism (Single Objective) - GA*), Algoritmo Genético para el Vector Evaluado (*Vector Evaluated Genetic Algorithm - VEGA*), por mencionar algunos algoritmos del *framework*. Además de incluir algoritmos de la biblioteca JMetal y PISA, los cuales están disponibles como un plugin JAR.

MOEA es de código abierto, proporciona herramientas para el diseño rápido, desarrollo, ejecución y estadísticas de prueba en algoritmos de optimización. Adicionalmente, cuenta con un módulo de descargas que proporciona los recursos necesarios para el desarrollo de aplicaciones y una guía de inicio rápido con los pasos para configurar y ejecutar ejemplos de problemas. La documentación y el sitio web son publicados bajo la licencia *GNU Lesser General*.

- **Java Ant Colony Framework (JACOF)** [51]: es un *framework* para la optimización de ACO. JACOF implementa las variantes más importantes de ACO como lo son el sistema de hormigas (*Ant System - AS*), sistema elitista de hormigas (*Elitist Ant System - EAS*), sistema de colonia de hormigas (*Ant Colony System - ACS*), sistema de hormigas basado en rangos (*Rank-based Ant System - ASRank*), y sistema de hormigas Max-Min (*Max-Min Ant System - MMAS*).

Este *framework* proporciona problemas clásicos en la literatura y es compatible con bibliotecas que incluyen instancias para estos problemas. Para utilizar JACOF es necesario abordar un problema, elegir una variante de ACO y ejecutar el algoritmo. La documentación y los archivos de descarga son presentados bajo la plataforma GitHub con licencia de uso libre.



## Capítulo 3

# Evaluación de *frameworks* de metaheurísticas

### 3.1. Análisis de *frameworks*

Se analizaron los *frameworks* JAMES, MOEA y JACOF de las familias de metaheurísticas Algoritmos evolutivos (AE), Búsqueda global (BG) o Inteligencia colectiva (AIC) con la finalidad de identificar la facilidad de implementación de cada *framework* y resolver el KP, una de las variantes del problema estándar surgida a partir de su aplicación para resolver problemas.

El KP consiste en un conjunto de elementos con un peso y valor específicos que son seleccionados para maximizar el valor obtenido y el peso total de los elementos elegidos sin exceder la capacidad de la mochila [29]. Para la selección de los *frameworks* de metaheurísticas se consideraron los siguientes criterios:

- Búsqueda de *frameworks* con las siguientes características:
  1. Familia: Que el *framework* incluya metaheurísticas de una misma familia, es decir, que este especializado en AE, AIC o BG.
  2. Multiplataforma: Que se ejecute en diferentes sistemas operativos de escritorio.
  3. Actualizado recientemente: Aunque el *framework* se haya desarrollado años atrás, se desea que la última actualización sea de al menos el año 2017.

4. Licencia libre: Sin restricciones de uso y código fuente disponible.
  - Revisión de la literatura en aplicaciones de los *frameworks* de metaheurísticas: Los *frameworks* deben resolver de forma clara y precisa al menos un problema de optimización.
  - Uso e implementación de cada *framework*: Los *frameworks* deben contener problemas de prueba y documentación acerca del código a utilizar.
  - Identificación de los parámetros de los *frameworks*: Los parámetros del código deben ser definidos y representados en un ejemplo de implementación del algoritmo seleccionado.

Para MOEA se seleccionaron 4 algoritmos genéticos nativos del *framework*: NSGA-II, NSGA-III, GA y VEGA. JAMES utiliza los algoritmos Parallel Tempering, Random Descent y Variable Neighbourhood Search, por su amplia aplicación en la literatura. Para JACOF se implementan las 5 variantes de ACO: AS, ACS, EAS, ASRank y MMAS.

En la literatura se encontró que MOEA posee una colección completa de algoritmos y herramientas para la optimización de objetivos únicos y multi-objetivos, lo que permite que sea uno de los *frameworks* más utilizados en la familia de metaheurísticas evolutivas. JAMES proporciona 13 algoritmos de BG que pueden ser aplicados a cualquier PO. JACOF utiliza 5 algoritmos como variantes de solución con una configuración de parámetros diferente para cada algoritmo.

## 3.2. Prototipo Lanzador de Framework de Metaheurísticas (LFM)

El prototipo LFM, fue desarrollado bajo la plataforma Java (Open JDK 7) usando exclusivamente bibliotecas de licencia libre. El prototipo resuelve el KP utilizando los algoritmos mencionados de cada uno los tres *frameworks* y se encuentra disponible en la plataforma GitHub (<http://github.com/chavezbosquez/prototipo-lfm>). Las pruebas se realizaron en una computadora Alienware M17x Intel Core i7-2670QM CPU @ 2.20GHz bajo el sistema operativo Ubuntu 16.04.3 LTS 64-bit y la interfaz se puede observar en la Figura 3.1.

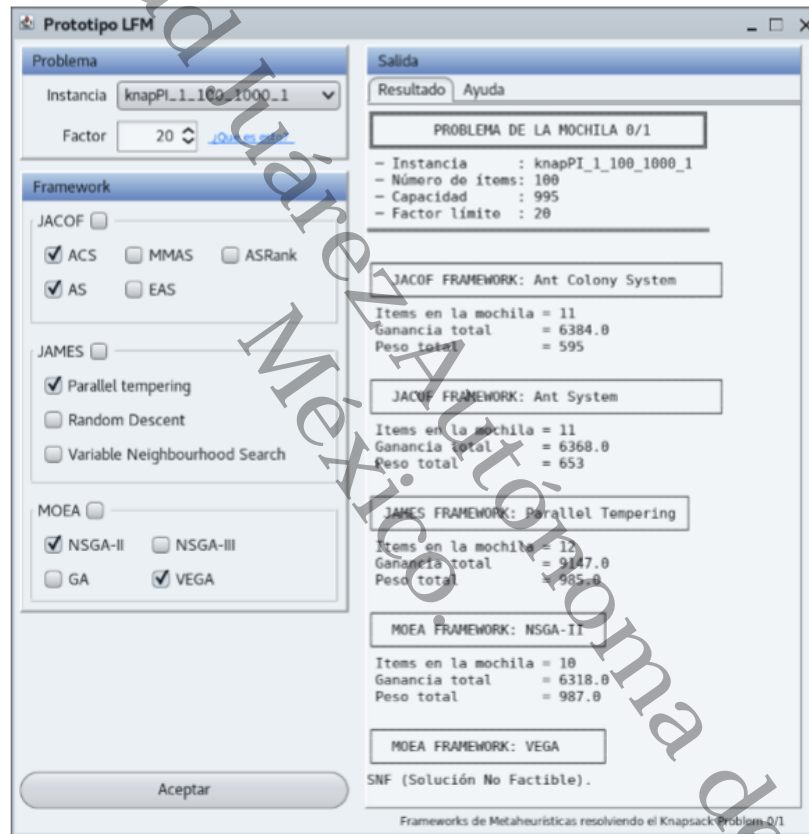


Figura 3.1. Prototipo LFM.

Se realizaron 30 ejecuciones de cada uno de los 12 algoritmos seleccionados para resolver 7 instancias del KP 0-1 y así identificar qué *framework* contiene los algoritmos más eficientes. La sintonización de parámetros del Prototipo LFM para cada *framework* es la siguiente:

**MOEA** : La configuración de parámetros de los algoritmos genéticos NSGA-II, NSGA-III, GA Y VEGA de este *framework* son:

- Evaluaciones:  $\text{factor} \times 100000$ , evalúa la aptitud de cada individuo en la población, los individuos más aptos son seleccionados y el genoma de cada individuo es modificado para formar una nueva generación.

**JAMES** : El algoritmo *Parallel Tempering* tiene los siguientes parámetros:

- $temp_{min}$ : Temperatura mínima que muestrea una región más pequeña y puede quedar en mínimos locales.
- $temp_{max}$ : Temperatura máxima donde la temperatura más alta intercambia soluciones con la temperatura más baja y puede muestrear una mayor cantidad de espacio.
- Número de réplicas: Tiene un valor de 10 y se ordena de acuerdo a la temperatura.

**JACOF** : Dependiendo la variante de OCH, los valores de los parámetros son:

- $numHormigas$ : 10.
- $\alpha$ : 1.0 Factor de influencia de feromonas.
- $\beta$ : 2.0 Información heurística.
- $\rho$ : 0.1 Coeficiente de evaporación de feromonas.
- $\omega$ : 0.1 Parámetro local de decaimiento de feromonas.
- $Q_0$ : 0.9. Parámetro adicional que corresponde al nivel de exploración de las hormigas.

Las Tablas 3.1, 3.2 y 3.3 muestran los resultados obtenidos por los algoritmos de cada *framework* en las 7 instancias del KP. La Tabla 3.4, presenta los resultados del mejor algoritmo de cada *framework* en cada una de las 7 instancias.

**Tabla 3.1.** Resultados de los algoritmos de JACOF (SNF= Solución no factible, EMI= Error de memoria insuficiente).

Instancia	Métrica	ACS			AS			ASRank			MMAS			EAS		
		Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso
KP1	Mejor	11	6384	595	11	6384	595	11	6384	595	11	6384	595	11	6384	595
	Ítems: 100	Mediana	11	6384	595	11	6368	629	11	6251.5	608	11	6361	595	11	6279
Capacidad: 995	Peor	11	6198	639	10	6059	617	11	5996	662	10	5943	550	11	6142	630
	Media	-	6369.36	-	-	6305.16	-	-	6243.66	-	-	6311.16	-	-	6287.63	-
	STD	-	40.75	-	-	86.13	-	-	110.87	-	-	97.91	-	-	74.73	-
KP2-KP5			SNF			SNF			SNF			SNF			SNF	
KP6-KP7			EMI			EMI			EMI			EMI			EMI	

El formato de las cuatro tablas es el mismo: en la primera columna se muestran las características de cada instancia (número de artículos y capacidad), la segunda columna contiene las métricas utilizadas (mejor solución, peor, media, mediana y desviación estándar), y finalmente el número de artículos en la mochila, la ganancia y el peso, por cada algoritmo. Los algoritmos están ordenados de izquierda a derecha de acuerdo a la mejor solución obtenida.

En la Tabla 3.1 se puede observar que las 5 variantes de OCH únicamente resuelven la instancia más sencilla del problema, el KP1. En cuanto a las instancias KP6 y KP7, consideradas las más difíciles, los 5 algoritmos lanzan una `OutOfMemoryException`, que ocurre cuando la máquina virtual de Java carece de la memoria suficiente para instanciar nuevos objetos. Cabe mencionar que no se realizó ninguna configuración adicional para ejecutar el Prototipo, se utilizó la memoria que utiliza de manera predeterminada la máquina virtual de Java.

La Tabla 3.2 contiene los resultados obtenidos por los 3 algoritmos de JAMES, el algoritmo de *Parallel Tempering* resuelve las 7 instancias del problema, resaltando el hecho que la desviación estándar es igual a 0 para las primeras 5 instancias, pero muy alta para las otras 2 instancias, consideradas más difíciles. Por otra parte, el algoritmo *Random Descent* resuelve únicamente las 3 primeras instancias del problema, eso sí, con desviación estándar igual a 0. Para el caso del algoritmo *Variable Neighbourhood Search*, éste resuelve únicamente las 2 primeras instancias y con alta variabilidad, como lo demuestra su desviación estándar.

Los resultados obtenidos por los AE de MOEA se muestran en la Tabla 3.3. Los 4 algoritmos seleccionados fueron ejecutados con el mismo número de evaluaciones, de los cuales solamente el NSGA-II y NSGA-III encuentran soluciones para las primeras 4 instancias del problema. GA sola-

Capítulo 3. Evaluación de *frameworks* de metaheurísticas

Tabla 3.2. Resultados de los algoritmos de JAMES (SNF= Solución no factible).

Instancia	Métrica	Templado paralelo			Descenso aleatorio			Variable Neighbourhood Search		
		Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso
<b>KP1</b> Ítems: 100 Capacidad: 995	Mejor	12	9147	985	8	6919	981	8	6871	959
	Mediana	12	9147	985	8	6895	975.5	7	6150	965.5
	Peor	12	9147	985	7	6295	970	6	5429	972
	Media	-	4268.6	-	-	900.13	-	-	410	-
	STD	-	0	-	-	304.84	-	-	1019.64	-
<b>KP2</b> Ítems: 200 Capacidad: 1008	Mejor	16	11238	987	11	9245	998	12	9738	1004
	Mediana	16	11238	987	11	9245	998	11	9134	1000.5
	Peor	16	11238	987	11	9245	998	10	8530	997
	Media	-	1123.8	-	-	308.16	-	-	608.93	-
	STD	-	0	-	-	0	-	-	854.18	-
<b>KP3</b> Ítems: 500 Capacidad: 2543	Mejor	42	28857	2543	40	35844	5001		SNF	
	Mediana	42	28857	2543	40	35844	5001		SNF	
	Peor	42	28857	2543	40	35844	5001		SNF	
	Media	-	3847.6	-	-	1194.8	-		SNF	
	STD	-	0	-	-	0	-		SNF	
<b>KP4</b> Ítems: 1000 Capacidad: 5002	Mejor	81	54403	4998		SNF			SNF	
	Mediana	77	53894	4998		SNF			SNF	
	Peor	71	52653	4998		SNF			SNF	
	Media	-	8937.56	-		SNF			SNF	
	STD	-	842.66	-		SNF			SNF	
<b>KP5</b> Ítems: 2000 Capacidad: 10011	Mejor	153	109759	10011		SNF			SNF	
	Mediana	153	109759	10011		SNF			SNF	
	Peor	153	109759	10011		SNF			SNF	
	Media	-	3658.63	-		SNF			SNF	
	STD	-	0	-		SNF			SNF	
<b>KP6</b> Ítems: 5000 Capacidad: 25016	Mejor	287	239149	25009		SNF			SNF	
	Mediana	271	228332	24955		SNF			SNF	
	Peor	26	14070	14926		SNF			SNF	
	Media	-	16051.7	-		SNF			SNF	
	STD	-	126942.09	-		SNF			SNF	
<b>KP7</b> Ítems: 10000 Capacidad: 49877	Mejor	503	432592	49868		SNF			SNF	
	Mediana	296	273071	49314		SNF			SNF	
	Peor	45	23026	22170		SNF			SNF	
	Media	-	24289.63	-		SNF			SNF	
	STD	-	206443.6	-		SNF			SNF	

**Tabla 3.3.** Resultados de los algoritmos de MOEA (SNF = Solución no factible).

Instancia	Métrica	NSGA-II			NSGA-III			GA			VEGA		
		Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso
<b>KP1</b>	Mejor	11	6384	993	11	6384	993	11	6384	993			
	Ítems: 100	Mediana	11	6368	989	10.5	6251.5	977	10	6318	981		
	Capacidad:	Peor	11	6116	992	9	5753	962	10	6132	973		
	<b>995</b>	Media	-	6334.76	-	-	6210.03	-	-	6310	-		
	STD	-	75.60	-	-	160.04	-	-	62.81	-			
<b>KP2</b>	Mejor	18	10463	993	18	10463	993	17	10001	996			
	Ítems: 200	Mediana	18	10390	997.5	18	10263.5	1000	16	9196	985		
	Capacidad:	Peor	17	10021	1005	16	9926	1001	15	8080	847		
	<b>1008</b>	Media	-	10337.46	-	-	10250.46	-	-	9148.63	-		
	STD	-	139.33	-	-	182.60	-	-	457.36	-			
<b>KP3</b>	Mejor	13	29505	2537	42	29217	2540						
	Ítems: 500	Mediana	42	29315	2537	42	28481.5	2534.5					
	Capacidad:	Peor	41	28868	2510	40	27774	2541					
	<b>2543</b>	Media	-	29271.63	-	-	28445.96	-					
	STD	-	162.500	-	-	340.73	-						
<b>KP4</b>	Mejor	71	44484	5001	64	36642	4963						
	Ítems: 1000	Mediana	68	40919.5	4936	61	32339	4939					
	Capacidad:	Peor	64	36663	4827	50	23164	4889					
	<b>5002</b>	Media	-	40915.4	-	-	30475.76	-					
	STD	-	2026.91	-	-	2867.59	-						
<b>KP5 - KP7</b>			SNF		SNF		SNF		SNF		SNF		

mente resuelve 2 instancias, mientras que VEGA no encontró solución para ninguna de las instancias. En cuanto a las instancias KP5 a KP7, ninguno de los algoritmos encontró una solución válida.

En la Figura 3.2 se señala el gráfico con las mejores soluciones para el KP 0-1 obtenidas por el mejor algoritmo de cada *framework*, derivado de las 30 ejecuciones por instancia.

En la Figura 3.3 se presenta la media de los resultados del mejor algoritmo de cada *framework*. Se puede observar que, a pesar de que el *framework* JAMES obtuvo los mejores resultados, MOEA sobresale en la distribución de la media aritmética.

Capítulo 3. Evaluación de *frameworks* de metaheurísticas

**Tabla 3.4.** Resultados generales (SNF= Solución no factible, EMI= Error de memoria insuficiente).

Instancia	Métrica	Templado paralelo (JAMES)			NSGA-II (MOEA)			ACS (JACOF)		
		Ítems	Ganancia	Peso	Ítems	Ganancia	Peso	Ítems	Ganancia	Peso
<b>KP1</b> Ítems: 100 Capacidad: 995	Mejor	12	9147	985	11	6384	993	11	6384	595
	Mediana	12	9147	985	11	6368	989	-	6384	-
	Peor	12	9147	985	11	6116	992	11	6198	639
	Media	-	4268.6	-	-	6334.76	-	-	6369.36	-
	STD	-	0	-	-	75.60	-	-	40.75	-
<b>KP2</b> Ítems: 200 Capacidad: 1008	Mejor	16	11238	987	18	10463	993	-	SNF	-
	Mediana	16	11238	987	18	10390	997.5	-	SNF	-
	Peor	16	11238	987	17	10021	1005	-	SNF	-
	Media	-	1123.8	-	-	10337.46	-	-	SNF	-
	STD	-	0	-	-	139.33	-	-	SNF	-
<b>KP3</b> Ítems: 500 Capacidad: 2543	Mejor	42	28857	2543	43	29505	2537	-	SNF	-
	Mediana	42	28857	2543	42	29315	2537	-	SNF	-
	Peor	42	28857	2543	41	28868	2510	-	SNF	-
	Media	-	3847.6	-	-	29271.63	-	-	SNF	-
	STD	-	0	-	-	162.50	-	-	SNF	-
<b>KP4</b> Ítems: 1000 Capacidad: 5002	Mejor	81	54403	4998	71	44484	5001	-	SNF	-
	Mediana	77	53894	4998	68	40919.5	4936	-	SNF	-
	Peor	71	52653	4998	64	36663	4827	-	SNF	-
	Media	-	8937.56	-	-	40915.4	-	-	SNF	-
	STD	-	842.66	-	-	2026.91	-	-	SNF	-
<b>KP5</b> Ítems: 2000 Capacidad: 10011	Mejor	153	109759	10011	-	SNF	-	-	SNF	-
	Mediana	153	109759	10011	-	SNF	-	-	SNF	-
	Peor	153	109759	10011	-	SNF	-	-	SNF	-
	Media	-	3658.63	-	-	SNF	-	-	SNF	-
	STD	-	0	-	-	SNF	-	-	SNF	-
<b>KP6</b> Ítems: 5000 Capacidad: 25016	Mejor	287	239149	25009	-	SNF	-	-	EMI	-
	Mediana	271	228332	24955	-	SNF	-	-	EMI	-
	Peor	26	14070	14926	-	SNF	-	-	EMI	-
	Media	-	16051.7	-	-	SNF	-	-	EMI	-
	STD	-	126942.09	-	-	SNF	-	-	EMI	-
<b>KP7</b> Ítems: 10000 Capacidad: 49877	Mejor	503	432592	49868	-	SNF	-	-	EMI	-
	Mediana	296	273071	49314	-	SNF	-	-	EMI	-
	Peor	45	23026	22170	-	SNF	-	-	EMI	-
	Media	-	24289.63	-	-	SNF	-	-	EMI	-
	STD	-	206443.6	-	-	SNF	-	-	EMI	-



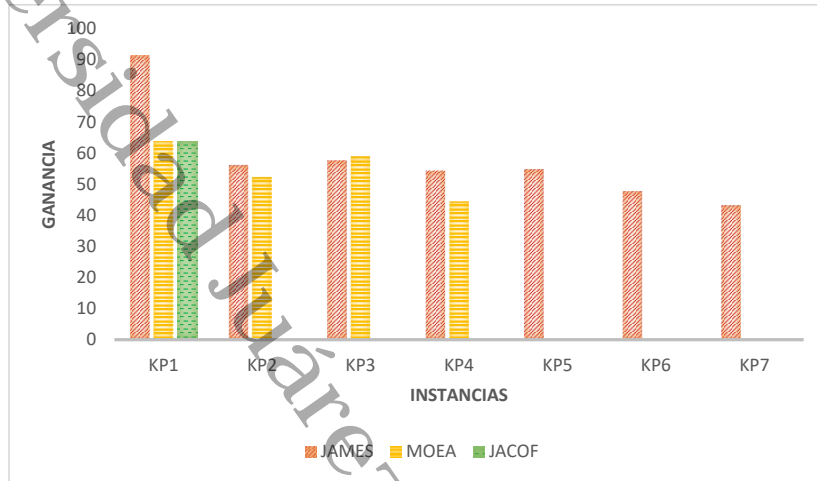


Figura 3.2. Mejores resultados por instancia.

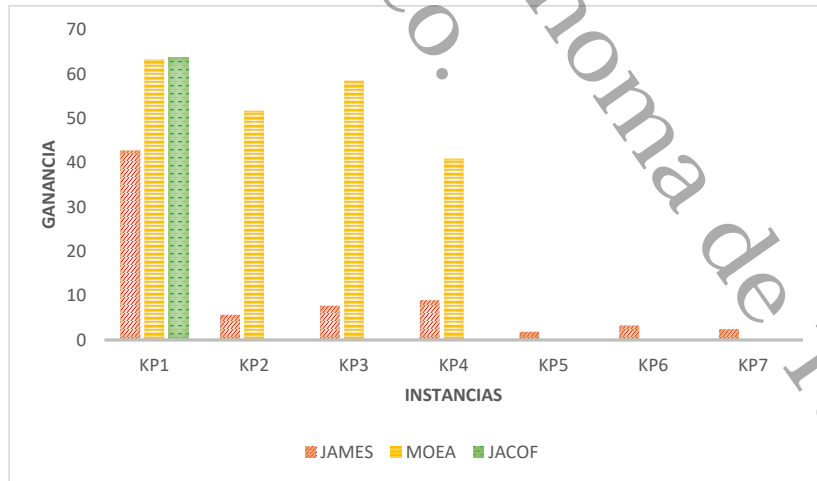


Figura 3.3. Media de los resultados por instancia.

## Capítulo 4

# Modelo del problema de la generación de menús nutritivos

El problema de generación de menús es comparado frecuentemente con el problema de la mochila. Este problema constituye uno de los más simples de la programación lineal [62]. Aparece como subproblema en otros problemas más complejos y tiene diversas aplicaciones prácticas en la toma de decisiones del mundo real, tales como la búsqueda de patrones de corte para materias primas que generen el menor desperdicio posible [33], la optimización de recursos computacionales (computación en malla) [55], el abastecimiento de vehículos de transporte y entrega de productos de diferentes tamaños, colocados en compartimentos de igual o distinta capacidad [14], entre otros.

Matemáticamente se expresa como [6]:

$$\text{maximizar } \sum_{j=1}^n w_j x_j, \quad (4.1)$$

$$\text{sujeto a } \sum_{j=1}^n w_j x_j \leq c, \quad (4.2)$$

$$\text{tal que } x_j \in \{0, 1\}, j = 1, \dots, n \quad (4.3)$$

donde:

$x_j$  : Variables de decisión,

$w_j$  : Peso  $w$  del artículo  $j$ ,

$c$  : Capacidad total de la mochila,

$n$  : Número de artículos.

El modelo matemático define que cada elemento corresponde a la variable  $x_j$  cuyo valor puede ser 1 si el elemento  $j$  se introduce a la mochila y 0 si se descarta. Además, se debe considerar que un elemento solo puede ser elegido si no se ha excedido el peso  $w_j$ .

## 4.1. Especificaciones del menú

El menú nutritivo descrito [39] considera las siguientes especificaciones:

1. Será para usuarios sin sobrepeso o desnutrición, y sin patologías.
2. Está compuesto por alimentos agrupados en 5 tiempos de comida:
  - Desayuno.
  - Colación matutina.
  - Almuerzo.
  - Colación vespertina.
  - Cena.
3. Los grupos de alimentos contemplados en este trabajo forman parte de una base de datos de alimentos creada por nutriólogas de la División Académica de Ciencias de la salud de la UJAT [17]:
  - Lácteos.
  - Alimentos de origen animal.
  - Frutas.
  - Verduras.
  - Cereales sin grasa.
  - Cereales con grasa.
  - Leguminosas.
  - Bebidas generales.
  - Bebidas desayuno.
  - Acompañamiento.

4. La colación es una comida ligera entre el desayuno y el almuerzo, y entre el almuerzo y la cena. Puede estar compuesta de un cereal con grasa y/o fruta y/o verdura.
5. El desayuno puede estar compuesto por una bebida desayuno (la cual puede contener leche) y/o cereales sin grasa y/o frutas y/o verduras y/o tortillas y/o alimentos de origen animal y/o lácteos.
6. El almuerzo puede estar compuesto por una bebida general (la cual no contiene leche) y/o cereales sin grasa y/o frutas y/o verduras y/o tortillas y/o leguminosas y/o alimentos de origen animal, y/o lácteos.
7. La cena puede estar compuesta por una bebida general y/o cereales sin grasa y/o frutas y/o verduras.
8. Cada tiempo de comida está compuesto por alimentos variados que deben cumplir con el aporte de las necesidades de energía diarias, en el modelo matemático se menciona el número de alimentos que se deben incluir en cada tiempo de comida pertenecientes a diferentes grupos de alimentos.
9. El Requerimiento Energético total diario de un Individuo se calcula usando la ecuación de Harris-Benedict revisada por Mifflin & St Jeor [8]. Para el género femenino:

$$GEB_{mujer} = (10 \times peso) + (6.25 \times estatura) - (5 \times edad) - 161 \quad (4.4)$$

Para el género masculino:

$$GEB_{hombre} = (10 \times peso) + (6.25 \times estatura) - (5 \times edad) + 5 \quad (4.5)$$

10. El efecto termógeno de los alimentos consiste en el gasto energético que se necesita para procesar los alimentos el cual corresponde al 10 % del Requerimiento Energético Individual (REI) y debe ser agregado a este mismo para calcular el gasto total.
11. El gasto por actividad física se agrega al REI y los porcentajes varían dependiendo del tipo de actividad que realiza [47], los cuales se muestran en la Tabla 4.1:

**Tabla 4.1.** Porcentaje agregado al REI de acuerdo a la actividad física.

Nivel de actividad	Ejemplo	% REI
Reposo	Encamado	20 %
Ligero	Profesional	37.5 %
Moderado	Ama de casa	55 %
Intenso	Obrero	72.5 %

**Tabla 4.2.** Información general y nutrimental registrada en la base de datos de alimentos de la UJAT en el grupo de verduras.

Index	Código	Nombre	Cant.	Medida	Gr/MI	Cal.	HC	Prot	F	Fibra
4	PILAR-V4	Ensalada de brócoli con pimienta	1	Taza	100	182	11.25	5.75	12.65	3.33

## 4.2. Base de alimentos

La base de alimentos es un registro de información general y nutrimental por alimento almacenada en archivos CSV, un ejemplo se describe en la Tabla 4.2.

En la Tabla 4.3 se presenta el número de alimentos contenidos en los 10 grupos. Esta base de alimentos es utilizada en las pruebas del algoritmo NSGA-II y para generar los menús nutricionales.

## 4.3. Modelo matemático de la función objetivo y restricciones

El modelo matemático propuesto se basa en la formulación [39]. La función objetivo se presenta en la Ecuación 4.6 en la cual se busca minimizar el error entre las calorías que requiere el usuario y las calorías del menú generado por el algoritmo NSGA-II con un margen de error del 10 %:

$$E = \sum_{m=1}^c KcalComida_m(\vec{k}) - REI \quad (4.6)$$

donde  $c$  es el número de comidas del menú (en este trabajo son 5),  $\vec{k}$  es un vector de números enteros donde cada valor hace referencia a un índice

**Tabla 4.3.** Número de alimentos por cada grupo en la base de alimentos.

Nivel de actividad	Ejemplo
Verduras	27
Frutas	50
Cereales sin grasa	29
Cereales con grasa	18
Leguminosas	20
Alimentos de origen animal	26
Bebidas generales	12
Bebidas desayuno	16
Acompañamientos	4
Lácteos	19

en la base de datos en un grupo de alimento específico, el  $\vec{k}$  tiene rangos distintos ya que depende del tipo de comida y del grupo de alimento.

El algoritmo debe encontrar cinco vectores  $\vec{k}$  de números enteros y cada valor del vector debe estar entre el rango  $[1, indexTgrupo_t]$ , donde  $indexTgrupo_t$  es el número total de alimentos (índices) en uno de los diez grupos de alimentos en la base de datos,  $t = 1 : 10$ . El tamaño del vector  $\vec{k}$  varía dependiendo del número de elementos que se consideran en cada comida.

$KcalComida_m(\vec{k})$  son las calorías de cada alimento de las comidas del menú y REI es el requerimiento energético diario de un individuo el cual es calculado usando la Ecuación 4.4 o 4.5 dependiendo del género y es agregado el efecto termógeno de los alimentos  $(REI + 0.10REI)$  y el gasto por actividad física  $(REI + [0.20|0.375|0.55|0.725] \times REI)$ .

La restricción  $g1$  permite cumplir con la Ley de la cantidad formulada en la Ecuación 4.7.

$$g1 = \left| \sum_{m=1}^c KcalComida_m(\vec{k}) - REI \right| - (0.10REI) \leq 0 \quad (4.7)$$

Con las restricciones  $g2$ ,  $g3$  y  $g4$  se permite cumplir con la Ley de la armonía la cual establece que del aporte calórico diario, los carbohidratos deben cubrir entre el 55 y 60 %, las proteínas deben cubrir entre el 10 y 15 % y los lípidos entre el 25 y 30 %. La restricción  $g2$  es calculada en la Ecuación

4.8.

$$g2 = \sum_{m=1}^c CarbComida_m(\vec{k}) \leq 1.10CarbReq \ \& \ \sum_{m=1}^c CarbComida_m(\vec{k}) \geq 0.90CarbReq \quad (4.8)$$

donde  $CarbComida_m(\vec{k})$  son los carbohidratos totales de cada una de las comidas del menú y  $CarbReq$  son los carbohidratos requeridos por el individuo calculados con la Ecuación 4.9.

$$CarReq = \frac{carb}{100} \times \frac{REI}{4} \quad (4.9)$$

donde  $carb$  es una variable aleatoria entre el rango [55, 60].

La restricción  $g3$  formaliza parte de la Ley de la armonía, la cual establece que del aporte calórico diario las proteínas deben cubrir entre el 10 y 15 %.

$$g3 = \sum_{m=1}^c ProtComida_m(\vec{k}) \leq 1.10ProtReq \ \& \ \sum_{m=1}^c ProtComida_m(\vec{k}) \geq 0.90ProtReq \quad (4.10)$$

donde  $ProtComida_m(\vec{k})$  son las proteínas totales de cada una de las comidas del menú y  $ProtReq$  son las proteínas requeridas por el individuo calculadas con la Ecuación 4.11.

$$ProtReq = \frac{prot}{100} \times \frac{REI}{4} \quad (4.11)$$

donde  $prot$  es una variable aleatoria entre el rango [10, 15]. La  $g4$  es la última fórmula que formaliza la Ley de la armonía, la cual valida los lípidos del menú con la Ecuación 4.12.

$$g4 = \sum_{m=1}^c LipComida_m(\vec{k}) \leq 1.10LipReq \ \& \ \sum_{m=1}^c LipComida_m(\vec{k}) \geq 0.90LipReq \quad (4.12)$$

donde  $LipComida_m(\vec{k})$  son los lípidos totales de cada una de las comidas del menú y  $LipReq$  son los lípidos requeridos por el individuo calculados con la Ecuación 4.13.

$$LipReq = \frac{lip}{100} \times \frac{REI}{9} \quad (4.13)$$

donde  $lip$  es una variable aleatoria entre el rango  $[25, 30]$ . La restricción  $g5$  corresponde al requerimiento energético diario que debe consumir un individuo durante el desayuno. En este trabajo el porcentaje establecido para el desayuno es del 25 % y la restricción se calcula con la Ecuación 4.14.

$$g5 = \sum_{k=1}^n KcalComida_1(\vec{k}) \leq 1.10Pd \ \& \ \sum_{k=1}^n KcalComida_1(\vec{k}) \geq 0.90Pd \quad (4.14)$$

donde  $n$  es el número de alimentos del desayuno, la especificación 5 para el modelo matemático menciona que son 6 alimentos contemplados en el desayuno de diferentes grupos de alimento.  $KcalComida_1(\vec{k})$  corresponde a las calorías de los alimentos en el desayuno (comida 1).  $Pd$  es el porcentaje de calorías que debe tener la comida desayuno, en este caso corresponde al 25 % del REI.

La restricción  $g6$  corresponde al requerimiento energético diario que debe consumir un individuo durante el almuerzo. En este trabajo el porcentaje establecido para el almuerzo es del 35 % y la restricción se calcula con la Ecuación 4.15.

$$g6 = \sum_{k=1}^n KcalComida_3(\vec{k}) \leq 1.10Pa \ \& \ \sum_{k=1}^n KcalComida_3(\vec{k}) \geq 0.90Pa \quad (4.15)$$

donde  $n$  es el número de alimentos del almuerzo que en este trabajo son 7, los cuales fueron mencionados en la especificación 6.  $KcalComida_3(\vec{k})$  corresponde a las calorías de los alimentos en el almuerzo (comida 3).  $Pa$  es el porcentaje de calorías que debe tener la comida almuerzo.

La restricción  $g7$  corresponde al requerimiento energético diario que debe consumir un individuo durante la cena. En este trabajo el porcentaje establecido para la cena es del 20 % y la restricción se calcula con la Ecuación 4.16.

$$g7 = \sum_{k=1}^n KcalComida_5(\vec{k}) \leq 1.10Pc \ \& \ \sum_{k=1}^n KcalComida_5(\vec{k}) \geq 0.90Pc \quad (4.16)$$

donde  $n$  son los alimentos de la cena que en este trabajo son 4 y fueron mencionados en la especificación 7.  $KcalComida_5(\vec{k})$  corresponde a las calorías de los alimentos en la cena (comida 5).  $Pc$  es el porcentaje de calorías que debe tener la comida cena. Por último, las restricciones  $g8$  y  $g9$  corresponden



al requerimiento energético diario que debe consumir un individuo durante las colaciones. En este trabajo el porcentaje establecido para cada colación es el 10 %. Ambas restricciones se aplican de manera similar a cada una de las colaciones. En la Ecuación 4.17 se válida la colación consumida entre el desayuno y almuerzo y en la Ecuación 4.18 se válida la colación consumida entre el almuerzo y cena.

$$g8 = \sum_{k=1}^n KcalComida_2(\vec{k}) \leq 1.10Pi \ \& \ \sum_{k=1}^n KcalComida_2(\vec{k}) \geq 0.90Pi \quad (4.17)$$

$$g9 = \sum_{k=1}^n KcalComida_4(\vec{k}) \leq 1.10Pi \ \& \ \sum_{k=1}^n KcalComida_4(\vec{k}) \geq 0.90Pi \quad (4.18)$$

donde  $n$  son los alimentos de cada uno de las colaciones, en este trabajo son 3 alimentos y fueron mencionados en la especificación 4.  $KcalComida_2(\vec{k})$  y  $KcalComida_4(\vec{k})$  corresponden a las calorías de los alimentos en cada colación (comida 2 y 4).  $Pi$  es el porcentaje de calorías que debe tener la comida colación.

En este modelo no se contemplan restricciones para la Ley de la adecuación y la Ley de la calidad, sin embargo para esta última ley se cuidó que en el menú existieran alimentos de los 5 grupos del plato del bien comer: frutas, verduras, cereales, leguminosas y alimentos de origen animal.

## Capítulo 5

# Experimentos y resultados

En este capítulo se presenta el generador de menús nutritivos. Los menús son personalizados para un usuario e incluyen las 5 comidas del día (desayuno, colación matutina, almuerzo, colación vespertina y cena). El público al que va dirigido es a usuarios sin patologías, con los que la cantidad diaria recomendada y adecuada a nutrientes varía con respecto a la actividad física que realiza, la edad, el peso, entre otros aspectos.

### 5.1. Diagrama de dominio del generador de menús nutritivos

En el menú nutritivo se deben validar las Leyes de la correcta alimentación a partir de los macronutrientes esenciales de los alimentos. Cada elemento del menú está representado en gramos y por medición (taza, porción, pieza, entre otros) esto permite obtener las calorías y nutrientes generales de los alimentos, los cuales deben satisfacer el REI.

En la Figura 5.1 se muestra el diagrama de dominio del prototipo GM (Generador de menús), el cual describe el procedimiento que se lleva a cabo para generar el menú. Los elementos considerados en el diagrama son los siguientes:

- **Enumeraciones.** Consisten en un conjunto de valores con nombre que se comportan como constantes en el lenguaje de programación utilizado. A una variable que ha sido declarada de tipo enumerado se le puede asignar como valor cualquiera de los miembros de esa enumeración [17].

1. **Genero:** característica de dos valores, correspondientes a un hombre (masculino) o una mujer (femenino).
  2. **ActividadFisica:** conjunto de 4 tipos de actividad física: reposo, ligera, moderada e intensa.
  3. **TiempoDeComida:** ingesta de los alimentos durante el día (desayuno, colación matutina, almuerzo, colación vespertina y cena).
  4. **GrupoAlimenticio:** conjunto de alimentos que proporcionan nutrientes esenciales y la energía que el cuerpo necesita para mantener el crecimiento normal y la salud del individuo. La clasificación de grupos es tomada de las tablas de composición de alimentos [43].
- **Clases:** se utilizan para representar variables (estado) y métodos (comportamiento). Las clases modeladas en el diagrama de dominio son las siguientes:
    1. **Usuario:** características particulares de un individuo (edad, peso, altura, etc.)
    2. **Recetas:** composición de alimentos que incluye las características de cada alimento (nombre, cantidad, medida y macronutrientes).
    3. **Comidas:** conjunto de alimentos y bebidas para cada tiempo de comida.
    4. **Menú:** conjunto de alimentos para los 5 tiempos de comida, que cumplan con los requerimientos del usuario.

## 5.2. Descripción del Prototipo Generador de Menús nutritivos (GM)

De las pruebas realizadas en el prototipo LFM se obtuvo que el algoritmo NSGA-II es el más apropiado para implementar en el problema de generación de menús. Por lo que el *framework* utilizado es MOEA, específicamente se ha implementado el algoritmo genético NSGA-II. El MOEA selecciona aleatoriamente los alimentos que integran el menú, genera un código binario, selecciona la posición que corresponde a la receta, obtiene el nombre y las calorías de cada alimento o bebida.

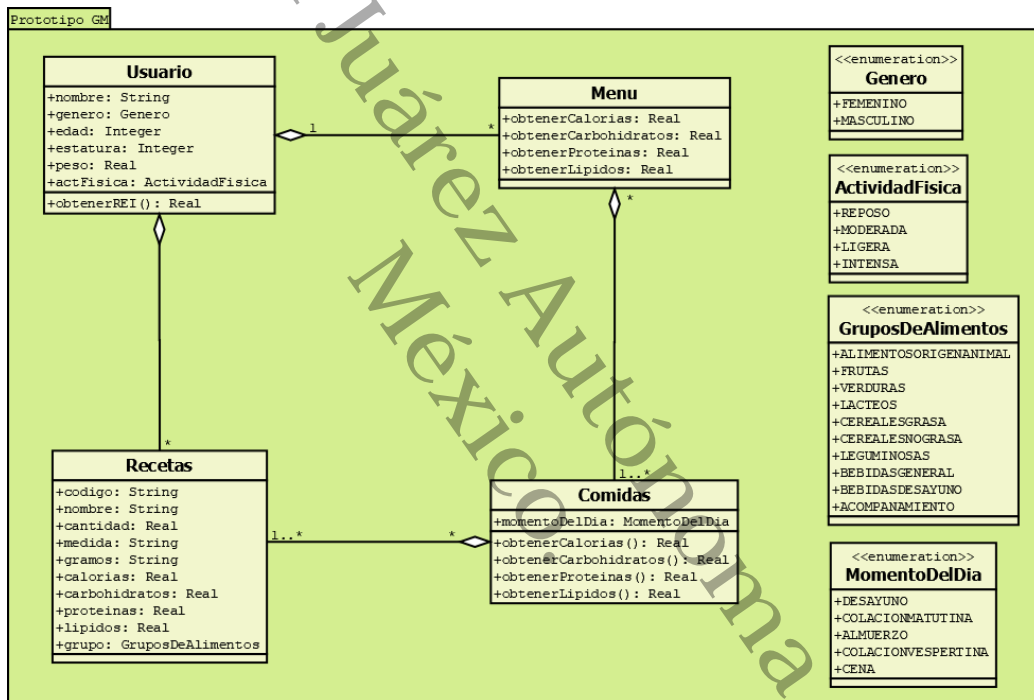


Figura 5.1. Diagrama de dominio.

El prototipo GM fue desarrollado en una computadora Dell G7 con procesador Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz 2.21 GHz, memoria RAM de 16 GB y Sistema Operativo Windows 10 Home de 64 bits.

Las bibliotecas empleadas son:

1. MOEA framework versión 2.12 <sup>1</sup>
2. OpenCSV versión 4.5 <sup>2</sup>

Inicialmente se realizaron pruebas usando una mochila y sin restricciones de alimentos en cada tiempo de comida. Los menús obtenidos en estas primeras pruebas alcanzaban las calorías requeridas por el usuario y no cumplían con las Especificaciones del menú (sección 4.1).

Las siguientes versiones del prototipo consideraron 2 aspectos importantes:

1. Se diseñaron 5 mochilas, cada una representa un tiempo de comida y está limitada a la capacidad de la mochila (corresponde al porcentaje de requerimiento energético del usuario), para cada tiempo de comida se debe incluir una cantidad de alimentos que cubran el REI. Los porcentajes de cada tiempo de comida se encuentran en la Sección 4.3 y son los siguientes:
  - Desayuno: 25 %.
  - Colación matutina: 10 %.
  - Almuerzo: 35 %.
  - Colación vespertina: 10 %.
  - Cena: 20 %.
2. El menú cubre un total de 23 alimentos en un día, en los cuales se incluye un número de alimentos específico: 6 en el desayuno, 3 en las colaciones, 7 en el almuerzo y 4 en la cena. Estos alimentos en conjunto generan el menú de acuerdo al REI del usuario.

En la Figura 5.2 se muestra un esquema con los pasos que necesarios para generar un menú nutritivo empleando el Prototipo GM:

---

<sup>1</sup><http://moeaframework.org>

<sup>2</sup><http://opencsv.sourceforge.net>

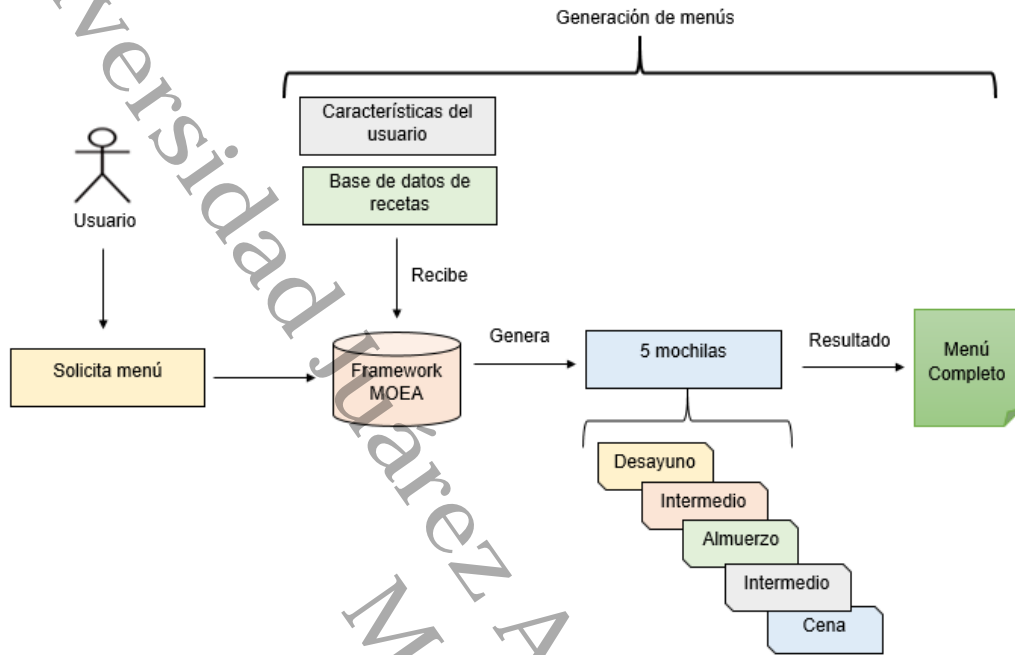


Figura 5.2. Generación del menú empleando el Prototipo GM.

- El usuario solicita el menú e ingresa sus características.
- El *framework* MOEA recibe los datos del usuario y la base de datos de las recetas.
- Se ejecuta el KP y el algoritmo NSGA-II del *framework* para generar un código binario.
- Se producen 5 códigos binarios aleatorios equivalentes a 5 mochilas independientes.
- Se genera un menú completo con alimentos/bebidas en cada tiempo de comida.

En la Figura 5.3 se observa la interfaz gráfica para generar un menú, los elementos que la integran son los siguientes:

- Usuario: Se asignan los datos de entrada correspondientes a las características del usuario (nombre, altura, peso, edad, actividad física y género) para calcular las calorías requeridas.

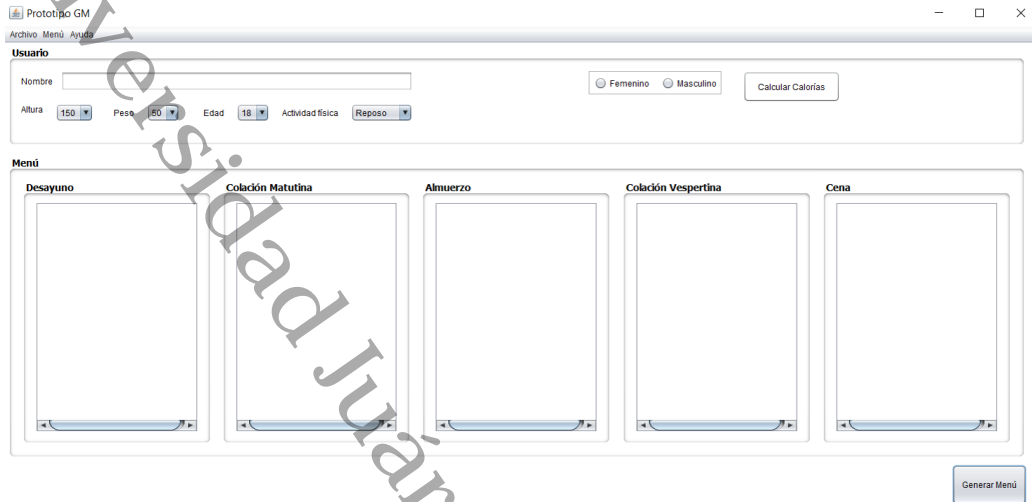


Figura 5.3. Prototipo GM.

- Menú: Está compuesto por 5 tiempos de comida independientes. El menú se genera aleatoriamente al inicio y a partir de los requerimientos del usuario selecciona los alimentos que se incluyen en el menú sin exceder el porcentaje requerido en cada tiempo de comida, por último es mostrado en el panel de salida al dar un clic sobre el botón generar menú.
- Se incluye la pestaña de archivo y el módulo de ayuda.

En la Figura 5.3 se presentan las características de una mujer que requiere 1856 calorías. Se probó el menú con los 5 tiempos de comida y se observó que en su mayoría los alimentos seleccionados por el *framework* son frutas y verduras.

Los parámetros del algoritmo NSGA-II utilizados para generar el menú son:

- **Ejecutor:** Proporciona todas las funciones necesarias para ejecutar el algoritmo sobre el problema especificado.
- **Problema:** Nombre de la clase a resolver (en este caso se indican 5 clases independientes correspondientes a las mochilas de los tiempos de comida).

- **Algoritmo:** Selecciona el algoritmo NSGA-II para resolver el problema.
- **Evaluaciones:** Evalúa la aptitud de cada individuo en la población, los individuos más aptos son seleccionados y el genoma de cada individuo es modificado para formar una nueva generación.
- **Número de evaluaciones realizadas:** 5000.
- El ejecutor permite evaluaciones distribuidas en múltiples núcleos o computadoras, puntos de control, instrumentación, entre otros.

### 5.3. Pruebas del prototipo GM

Las pruebas del Prototipo Generador de Menús se realizaron con 20 usuarios (12 hombres y 8 mujeres) presentados en la Tabla 5.1. Para cada usuario se generó un menú diferente, cuyos alimentos obtenidos son variados dependiendo del tiempo de comida y de acuerdo a las especificaciones del menú.

En la Tabla 5.2 se presentan los valores correspondientes al REI del usuario y la cantidad de calorías que éste debe consumir en cada tiempo de comida. En esta tabla se reflejan también las calorías de los menús generados por el Prototipo GM. Se puede observar que para cada usuario las calorías obtenidas por el generador de menús son diferentes, esto depende del porcentaje de cada tiempo de comida como en la Sección 5.2.

En la Tabla 5.3 se observa que para el usuario 1, el valor de REI es de 3320 cal, mientras que el total de calorías obtenidas por el menú es de 3316.5 cal, con un porcentaje de calorías del menú del 99.89%. Para un individuo de la tabla se alcanza el 100% de calorías, lo que indica que el REI y las calorías encontradas por el *framework* son cantidades iguales.

El tiempo de ejecución para los 5 tiempos de comida fue de 5 a 6 segundos por menú, es decir, aproximadamente 1 segundo por mochila. El REI obtenido a partir de los datos ingresados por el usuario no mostró una diferencia significativa en relación con las calorías obtenidas por el generador de menús (ver Tabla 5.4).

El algoritmo se ejecutó 2 veces por usuario con un total de 5000 evaluaciones. Cabe destacar que para el *framework* MOEA solo se necesita definir dos métodos: `newSolution` y `evaluation()`. El método `newSolution` define la representación del problema (el número y los tipos de sus variables de de-



**Tabla 5.1.** Características de 20 usuarios entre 18 y 60 años.

Usuario	Género	Peso	Altura	Edad	Actividad física
1	Masculino	72	180	20	Intenso
2	Femenino	62	157	22	Moderado
3	Femenino	50	165	23	Moderado
4	Femenino	60	172	25	Ligero
5	Masculino	70	170	25	Ligero
6	Masculino	68	170	26	Ligero
7	Masculino	65	172	28	Moderado
8	Femenino	68	170	35	Reposo
9	Femenino	68	157	45	Moderado
10	Masculino	80	165	55	Intenso
11	Masculino	73	173	30	Moderado
12	Femenino	80	153	33	Moderado
13	Masculino	78	167	36	Moderado
14	Masculino	113	175	25	Moderado
15	Masculino	81	184	29	Moderado
16	Masculino	84	183	31	Moderado
17	Femenino	56	153	31	Ligero
18	Masculino	105	172	33	Moderado
19	Masculino	88	186	25	Moderado
20	Femenino	61	160	60	Moderado

**Tabla 5.2.** Resultados obtenidos por el generador de menús para 20 usuarios (se indican los resultados del MOEA, el REI de los usuarios y las calorías requeridas en cada tiempo de comida).

Usuario	REI (cal)	MOEA (cal)	Desayuno	Colación matutina	Almuerzo	Colación vespertina	Cena
1	3320	3316.5	830	332	1162	332	664
2	2269	2263.0	567	227	794	227	454
3	2140	2141.0	535	214	749	214	428
4	2100	2101.1	525	210	735	210	420
5	2483	2483.8	621	248	869	248	497
6	2447	2443.4	612	245	856	245	489
7	2710	2707.9	678	271	948	271	542
8	1856	1856.0	465	185	650	185	371
9	2174	2170.7	544	217	761	217	435
10	2962	2958.3	741	296	1037	296	592
11	2840	2837.5	710	284	994	284	568
12	2439	2430.8	610	244	853	244	488
13	2811	2806.2	703	281	984	281	562
14	3586	3580.4	897	359	1254	359	717
15	3103	3101.5	776	310	1086	310	621
16	3126	3119.6	782	313	1093	313	625
17	1816	1814.6	454	182	635	182	363
18	3350	3345.3	838	335	1172	335	670
19	3277	3271.5	819	328	1147	328	655
20	1960	1951.9	490	196	686	196	392

cisión). El método de `evaluation()` toma una solución y calcula sus valores de función objetivo y posteriormente se utiliza el ejecutor.

Cuando el algoritmo realiza la selección de alimentos de manera aleatoria en la base de datos de recetas, un bajo número de alimentos se repiten y se ofrece una solución factible al problema.

A partir de estos resultados se puede mencionar que es importante tener variedad de alimentos en cuanto a grupos alimenticios, y considerar que se tiene un margen de error del 10% en las calorías del menú con respecto al REI del usuario. De acuerdo a los resultados obtenidos esta diferencia no es significativa, cumple con las restricciones y requerimientos del usuario.

## 5.4. Discusión

En la Tabla 5.4 se presenta la diferencia de error entre los valores del REI y el obtenido por MOEA para cada individuo, paciente o usuario. Se puede observar que MOEA es capaz de generar resultados muy cercanos al REI de cada paciente o usuario, en promedio la diferencia entre los valores obtenidos de MOEA y el REI es de 1.39 calorías.

Por otra parte, para corroborar si existe una diferencia significativa entre

**Tabla 5.3.** Diferencia porcentual de las calorías recomendadas por el generador de menús.

Usuario	REI (cal)	MOEA (cal)	% cal del menú
1	3320	3316.5	99.89 %
2	2269	2263.0	99.74 %
3	2140	2141.0	100.05 %
4	2100	2101.1	100.05 %
5	2483	2483.8	100.03 %
6	2447	2443.4	99.85 %
7	2710	2707.9	99.92 %
8	1856	1856.0	100 %
9	2174	2170.7	99.85 %
10	2962	2958.3	99.88 %
11	2840	2837.5	99.91
12	2439	2430.8	99.66
13	2811	2806.2	99.83
14	3586	3580.4	99.84
15	3103	3101.5	99.95
16	3126	3119.6	99.8
17	1816	1814.6	99.92
18	3350	3345.3	99.86
19	3277	3271.5	99.83
20	1960	1951.9	99.59

Tabla 5.4. Diferencia de error entre calorías.

Usuario	REI (cal)	MOEA (cal)	Diferencia de error
1	3320	3316.5	3.5
2	2269	2263.0	6
3	2140	2141.0	-1
4	2100	2101.1	-1.1
5	2483	2483.8	-0.8
6	2447	2443.4	3.6
7	2710	2707.9	2.1
8	1856	1856.0	0
9	2174	2170.7	3.3
10	2962	2958.3	3.7
11	2840	2837.5	2.4
12	2439	2430.8	8.2
13	2811	2806.2	4.8
14	3586	3580.4	5.6
15	3103	3101.5	1.5
16	3126	3119.6	6.4
17	1816	1814.6	1.4
18	3350	3345.3	4.7
19	3277	3271.5	5.5
20	1960	1951.9	8.1
		Promedio total	3,39

estos valores, se hace uso de la prueba no paramétrica *Wilcoxon Signed-Rank Test* en una calculadora en línea<sup>3</sup> tomando los valores de las columnas 2 y 3 de la Tabla 5.4. Para esta prueba se tomó una muestra con 20 individuos, donde se descartó al usuario 8 por obtener una diferencia de cero en ambas columnas y se redujo el tamaño de la muestra a 19 individuos.

La diferencia de error entre las calorías de los usuarios de la prueba es de 0 a 8.2, en comparación con el REI y los resultados del generador de menús. El resultado de la prueba de *Wilcoxon Signed-Rank Test* mostró que es significativo en  $p < .05$ . Lo que indica que hay diferencia entre ambos conjuntos de valores (REI y MOEA).

---

<sup>3</sup><https://www.socscistatistics.com/tests/signedranks/default2.aspx>

## Capítulo 6

# Conclusión, contribución y trabajos futuros

### 6.1. Conclusión

La generación de menús es un problema que se ha estudiado desde hace muchos años. Aun así, no se ha encontrado evidencia de trabajos que empleen *frameworks* de metaheurísticas y las Leyes de la correcta alimentación para resolver este problema.

En este trabajo se identificaron las técnicas que han empleado otros autores para resolver este problema, así como la representación de los atributos y operaciones de los prototipos realizados mediante el lenguaje UML con el fin de visualizar las relaciones entre las clases que involucran el sistema.

Se realizó un estudio y análisis de los *frameworks* a utilizar que incluían características específicas que les hacía interesantes de comparación. Posteriormente se desarrolló un prototipo para evaluar los *frameworks* de metaheurísticas, específicamente en el problema de la mochila. Para este problema se probaron 12 algoritmos incluidos en los 3 *frameworks* seleccionados (MOEA, JAMES Y JACOF).

De las pruebas realizadas, se seleccionó el algoritmo NSGA-II del *framework* MOEA por obtener el mejor rendimiento. Este algoritmo fue implementado en el desarrollo de un prototipo generador de menús nutritivos. El prototipo es una aplicación de escritorio multiplataforma programada en el lenguaje Java.

Este prototipo permite al usuario obtener un menú para los 5 tiempos de

comida, a partir de su REI y satisface las Leyes de la correcta alimentación. El menú generado, minimiza la diferencia entre las calorías del usuario y las calorías totales del menú.

## 6.2. Contribución

- Prototipo generador de menús nutritivos que cumple con las Leyes de la correcta alimentación utilizando frameworks de metaheurísticas.
- Prototipo que compara y evalúa los algoritmos de los frameworks de metaheurísticas en un problema de optimización.
- Adaptación y propuesta al framework MOEA en el problema de generación de menús.
- Propuesta de solución a un problema de trascendencia actual y de alto impacto social como lo es la generación de menús, aplicando técnicas de IA novedosas.

## 6.3. Trabajos futuros

- Se plantea el uso de otras metaheurísticas para resolver el problema.
- Modelar el problema de generación de menús incluyendo las Leyes de la correcta alimentación como función objetivo y las calorías del usuario como restricción del problema.

## 6.4. Producción académica derivada

Como resultado de este proyecto, se publicaron los siguientes artículos de investigación:

- Jiménez-Castellano, I., Hernández-Ocaña, B., & Chávez-Bosquez, O. (2018). *Frameworks* de metaheurísticas para resolver problemas de optimización. Semana de Difusión y Divulgación Científica. UJAT, 11p.
- Jiménez-Castellano, I., Hernández-Ocaña, B., Hernández-Torruco, J., & Chávez-Bosquez, O. (2019). *Frameworks* basados en metaheurísticas

para resolver el problema de la mochila. Revista INGENIERÍA UC, 26(1), 31-43. <http://servicio.bc.uc.edu.ve/ingenieria/revista/index.htm>

Universidad Juárez Autónoma de Tabasco.  
México.



## Bibliografía

- [1] AdaptiveBox Studio. Meta-Heuristic Repository. <http://www.adaptivebox.net/CILib>, 2017.
- [2] I. Alaya, C. Solnon, and K. Ghedira. Ant colony optimization for multi-objective optimization problems. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 1, pages 450–457. IEEE, 2007.
- [3] R. M. Antico. El profesor Dr. Pedro Escudero y su obra. *Revista Digital de Ciencias - Facultad de Ciencias Exactas, Naturales y de Salud*, 7(6), 2007.
- [4] Koroušić-Seljak B. Dietary Menu Planning Using an Evolutionary Method. In *International Conference on Intelligent Engineering Systems*, pages 108–113, 2006.
- [5] Koroušić-Seljak B. Computer-based dietary menu planning. *Journal of food composition and analysis*, 22(5):414–420, 2009.
- [6] E. Balas and E. Zemel. An algorithm for large zero-one knapsack problems. *Operations Research*, 28(5):1130–1154, 1980.
- [7] R. Barritta de Defranchi and J.K. Nelson. Evolution and trends of the dietetics profession in the United States of America and in Argentina. North and South united by similar challenges. *Archivos Latinoamericanos de Nutrición*, 59(2):113–119, 2009.
- [8] R. Berman. *Boosting your metabolism for dummies*. John Wiley & sons, Inc., 1 edición, 2013.

- [9] R. Blasco Redondo. Gasto energético en reposo. Métodos de evaluación y aplicaciones. *Revista Española de Nutrición Comunitaria*, 21:243–251, 2015.
- [10] G. Blázquez Abellán, J. D. López-Torres Hidalgo, J. Rabanales-Sotos, J. López-Torres López, and C. Ll. Jiménez. Alimentación saludable y autopercepción de salud. *Atención Primaria*, 48(8):535–542, 2016.
- [11] C. Blum and A. Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [12] J. Brito, C. Campos, F. García, M. García, B. Melián, and J. Moreno. Metaheurísticas: Una revisión actualizada. *Documentos de Trabajo del DEIOC. Universidad de La Laguna*, 2, 2004.
- [13] Centro de Investigación en Alimentación y Desarrollo (CIAD). Plato del bien comer, 2016.
- [14] E. Chajakis and M. Guignard. A model for delivery of groceries in vehicle with multiple compartments and lagrangean approximation schemes. In *Proceedings of Congreso Latino Ibero-Americano de Investigación de Operaciones e Ingeniería de Sistemas*, 1992.
- [15] O. Chávez-Bosquez, J. Marchi, and P. Parra-Pozos. Nutritional Menu Planning: A Hybrid Approach and Preliminary Tests. *Research in Computing Science*, 82:93–104, 2014.
- [16] O. Chávez-Bosquez and P. Pozos-Parra. The Latin American laws of correct nutrition: Review, unified interpretation, model and tools. *Computers in biology and medicine*, 70:67–79, 2016.
- [17] O. A. Chávez Bosquez. *Modelando la generación de menús nutritivos empleando técnicas de inteligencia artificial*. PhD thesis, Universidad Juárez Autónoma de Tabasco, México, 2016.
- [18] P. Coad and E. Yourdon. *Object Oriented Design*. Prentice Hall, 1 edición, 1991.
- [19] R. Corio Andújar and L. Arbonés Fincias. Nutrición y salud. *SEMER-GEN - Medicina de familia*, 35:443–449, 2009.

- [20] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. The MIT Press & McGraw Hill, 2 edición, 2001.
- [21] H. De Beukelaer, G. F. Davenport, G. De Meyer, and V. Fack. JAMES: A modern object-oriented Java framework for discrete optimization using local search metaheuristics. In *4th International symposium and 26th National conference on Operational Research*, pages 134–138. Hellenic Operational Research Society, 2015.
- [22] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.
- [23] ENSANUT. Encuesta Nacional de Salud y Nutrición. <https://ensanut.insp.mx>, 2017.
- [24] R.I. Esquivel Hernández, S.M. Martínez Correa, and J.L. Martínez Correa. *Nutrición y Salud*. Manual Moderno, México, 2 edición, 2005.
- [25] J. Fajardo Calderín. *Soft Computing en problemas de optimización dinámicos*. PhD thesis, Universidad de Granada, España, 2016.
- [26] FAO and Ministerio de Educación de la Nación República Argentina. *Educación alimentaria y nutricional*. Ministerio de Educación de la Nación, República Argentina, Argentina, 2009.
- [27] Food Pyramid. The 6 Essential Nutrients. <http://www.foodpyramid.com/6-essential-nutrients>, 2009-2015.
- [28] M. Fowler. *UML gota a gota*. Addison Wesley Longman Inc., México, 1 edición, 1999.
- [29] A. Fuentes-Penna, D. Vélez-Díaz, S. Moreno-Gutiérrez, M. Martínez-Cervantes, and O. Sánchez-Muñoz. Problema de la mochila (Knapsack problem). *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, 3(6), 2015.
- [30] J. M. Galindo Haro and J. M. Camps Riba. *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE*. PhD thesis, Escuela Técnica Superior de Ingeniería Informática, Valencia, España, 2008.

- [31] M. García Torres. *Aplicación de técnicas metaheurísticas en minería de datos*. PhD thesis, Universidad de La Laguna, España, 2006.
- [32] Ghent University. JAMES: Java Metaheuristics Search Framework. <https://www.jamesframework.org>, 2017.
- [33] P. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem. *Operations research*, 9(6):849–859, 1961.
- [34] D. E. Goldberg. Genetic algorithms in search, optimization and machine learning. *Addison-Wesley Publishing Co*, 1989.
- [35] M. Goldenberg. The Heuristic Search Research Framework. *Knowledge-Based Systems*, 129(Supplement C), 2017.
- [36] J. Gosling and Sun Microsystems. Software Java. <https://www.oracle.com/java>, 2019.
- [37] G. Gottau. Cómo planificar un menú saludable. <https://www.directoalpaladar.com/salud/como-planificar-un-menu-saludable>, 2013.
- [38] Hadka, D. MOEA: MultiObjective Evolutionary Algorithms. <http://moeaframework.org>, 2017.
- [39] B. Hernández Ocaña. *Optimización global con restricciones usando el algoritmo basado en el forrajeo de bacterias*. PhD thesis, Universidad Juárez Autónoma de Tabasco, México, 2016.
- [40] H. Hoos and T. Stützle. *Stochastic local search: Foundations and applications*. Elsevier, 2004.
- [41] J. A. Hurtado Sánchez. *Capítulo 74 - Nutrición y alimentación: evidencia en las recomendaciones y nuevas aportaciones metodológicas*. Elsevier España, Madrid, 2014.
- [42] E. Kaldirim and Z. Kose. Application of a multi-objective genetic algorithm to the modified diet problem. In *Genetic and Evolutionary Computation Conference (GECCO)*, volume 6, 2006.

- [43] J. A. Ledesma-Solano, A. Chávez-Villasana, Pérez-Gil F., E. Mendoza-Martínez, and C. Calvo-Carrillo. *Composición de alimentos Miriam Muñoz de Chávez. Valor nutritivo de los alimentos de mayor consumo*. McGraw-Hill, 2 edición, 2009.
- [44] A. Lorenzo Corchón. Energía, nutrientes y otros componentes de la dieta. <https://www.asturnatura.com/articulos/nutricion/energia-nutrientes-componentes-dieta/index.php>, 2017.
- [45] J. A. Lozano Teruel. *La nutrición es con-ciencia*. Salud. Editum: Ediciones de la Universidad de Murcia, Murcia, 2 edición, 2011.
- [46] G. Martínez Villalobos, G. Camacho Sánchez, and D. Biancha Gutierrez. Diseño de framework web para el desarrollo dinámico de aplicaciones. *Scientia Et Technica*, 16:178–183, 2010.
- [47] J. Mataix. Tratado de Nutrición y Alimentación: Situaciones fisiológicas y patológicas. *España: Océano-Ergon*, 2, 2005.
- [48] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Germany, 2 edición, 2004.
- [49] F. Moragas. *Nutrición e higiene alimentaria*. lulu.com, Dixie, Washington USA, 2006.
- [50] R. Moreno Otero and J. R. Cortés. Nutrición y alcoholismo crónico. *Nutrición Hospitalaria*, 23:3–7, 2008.
- [51] T. Nascimento. Jacof: Java ant colony optimization framework. <https://github.com/thiagodnf/jacof>, 2018.
- [52] Q. Olascoaga. *Dietética, Nutrición normal*. Méndez Editores, México, 8 edición, 1998.
- [53] Organización de las Naciones Unidas para la Alimentación y la Agricultura. Necesidades nutricionales. <http://www.fao.org/docrep/014/am401s/am401s03.pdf>, 2017.
- [54] M. Pardalos-Panos and M Resende. *Handbook of applied optimization*. Oxford University Press, 2002.

- [55] R. Parra-Hernandez, D. Vanderster, and N. Dimopoulos. Resource management and knapsack formulations on the grid. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 94–101. IEEE Computer Society, 2004.
- [56] J. Pearl. *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley Pub. Co. Inc., USA, 1 edición, 1984.
- [57] J. M. Ramos Pérez. *Técnicas de optimización multi-objetivo aplicadas al problema de planificación de menús*. PhD thesis, Universidad de La Laguna, España, 2016.
- [58] D. Riehle. *Framework Design: A Role Modeling Approach*. PhD thesis, Universidad de Hamburgo, Zurich, Suiza, 2000.
- [59] R.A. Roth. *Nutrición y dietoterapia*. McGraw Hill Interamericana, México, 9 edición, 2007.
- [60] J. Rumbaugh, I. Jacobson, and G. Booch. *El lenguaje unificado de modelado. Manual de referencia*. Addison Wesley, Madrid, España, 1 edición, 2000.
- [61] J. Salas-Salvadó, A. Bonada i Sanjaume, R. Trallero Casañas, M. E. Saló i Solá, and R. Burgos Peláez. *Nutrición y dietética clínica*. Elsevier Masson, Barcelona, España, 3 edición, 2014.
- [62] B. Silva and L. Torres. Acerca de una versión dinámica del problema de la mochila. *Revista Politécnica*, 34(1):142, 2014.
- [63] A. Suverza Fernández and K. Haua Navarro. *El ABCD de la evaluación del estado de nutrición*. McGraw-Hill, México, 1 edición, 2010.
- [64] M. E. Téllez Villagómez. *Nutrición Clínica*. Manual Moderno, México, 1 edición, 2010.
- [65] M. L. Yankilevich and M. P. Álvarez. Conocer lo que consumimos para no dejarnos engañar. *Elemental Watson*, 2(4), 2011.
- [66] A. Zhou, B-Y Qu, H. Li, S-Z Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.