



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Técnicas de optimización multi-objetivo
aplicadas al problema de
planificación de menús

*Multi-Objective Optimization Techniques
applied to the Menu Planning Problem*

Juan Manuel Ramos Pérez

La Laguna, 7 de junio de 2016

D^a. **Gara Miranda Valladares**, con N.I.F. 78.563.584-T profesora Ayudante Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Eduardo Manuel Segredo González**, con N.I.F. 78.564.242-Z investigador adscrito al Centre for Algorithms, Visualisation, and Evolving Systems de la Edinburgh Napier University, como co-tutor

C E R T I F I C A N

Que la presente memoria titulada:

“Técnicas de optimización multi-objetivo aplicadas al problema de planificación de menús”

ha sido realizada bajo su dirección por D. **Juan Manuel Ramos Pérez**, con N.I.F. 78.853.410-A.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de junio de 2016

Agradecimientos

Quisiera expresar mi reconocimiento y gratitud a Gara Miranda Valladares, por sus consejos, ayuda y dedicación. Su gran labor como tutora ha sido fundamental en el desarrollo de este trabajo.

También quisiera agradecer el apoyo y los ánimos recibidos por mi familia y amigos. Gracias a ellos el recorrido por este camino ha sido más agradable y llevadero.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El objetivo de este Trabajo de Fin de Grado ha sido el desarrollo de una aplicación mediante una formulación de optimización multi-objetivo cuyo propósito es la planificación de un menú escolar. La aplicación hace uso de métodos metaheurísticos en computación evolutiva, más concretamente algoritmos evolutivos de optimización multi-objetivo, teniendo como principal objetivo ofrecer un plan alimenticio variado, económico y equilibrado desde el punto de vista nutricional para un número de días determinado, asegurando así la adecuada alimentación de los niños en el comedor escolar. En el momento de generar un plan alimenticio también se tendrán en cuenta otros aspectos de suma importancia como las posibles alergias e intolerancias alimenticias, dietas específicas de una religión o estilo de vida, que pueda poseer una persona o conjunto de personas determinado.

La aplicación, además, presentaría una interfaz sencilla e intuitiva que no requiere de conocimientos especiales en el uso de tecnologías ni formación específica en cuestiones relacionadas con la optimización. El objetivo es mejorar la forma en la que se alimentan los niños, incluidos aquellos con necesidades nutricionales especiales, y contribuir con ello a reducir los altos niveles de malnutrición existentes en la sociedad actual.

Palabras clave: planificador de menús, menú escolar, alimentación, nutrición, optimización multi-objetivo, computación evolutiva, algoritmos evolutivos.

Abstract

The objective of this Final Degree Project has been to develop an application through a multi-objective optimization formulation whose purpose is the planification of a school meal plan. The application makes use of metaheuristic methods in evolutionary computation, and more specifically, evolutionary algorithms for multi-objective optimization. The main goal herein is to offer a varied, economic, and balanced plan from the nutritional point of view, considering a prefixed number of days, and ensuring a proper nutrition for children in school. At the time of generating a meal plan, we will also take into account other important aspects, such as possible allergies and food intolerances, and specific constraints regarding the religion or the lifestyle of a particular person or group of people.

The application also would present a simple and intuitive interface that requires no special knowledge in the use of technologies or specific formation on issues related to optimization. In this way, we aim to create a tool to improve the way children eat, including those with special nutritional needs and thus contributing to reduce the high levels of malnutrition in today's society..

Keywords: *menu planner, menu planning, school meal plan, alimentation, nutrition, multi-objective optimization, evolutionary computation, evolutionary algorithms.*

Índice general

1. Planificador de menú	1
1.1. Introducción	1
1.2. Definición del problema	1
1.3. Objetivos, restricciones y variables	3
1.4. Planificador de menú propuesto	4
1.5. Información dietética y nutricional	6
2. Estado del arte	8
2.1. Formulaciones del problema	8
2.2. Aplicaciones existentes	13
3. Menu Planning App	18
3.1. Propósito general	18
3.2. Tecnología utilizada	19
3.3. Descripción	21
3.3.1. Inicio	21
3.3.2. Ingredientes	22
3.3.3. Platos	24
3.3.4. Planificador	26
3.3.5. Plan alimenticio	28
4. Resolución del problema	31
4.1. Optimización Multi-objetivo	31
4.2. Algoritmos Evolutivos Multi-Objetivo	32
4.3. Algoritmo implementado	34
5. Resultados computacionales	42
5.1. Configuración de las pruebas	42
5.2. Resultados obtenidos	43
5.3. Conclusiones de las pruebas	47
6. Conclusiones y trabajo futuro	48
6.1. Conclusiones	48
6.2. Trabajo futuro	49

7. Conclusions and future work	51
7.1. Conclusions	51
7.2. Future work	52
8. Presupuesto	54
8.1. Presupuesto	54
Bibliografía	56

Índice de figuras

3.1. MySQL - Relaciones	20
3.2. App - Ventana de inicio	21
3.3. App - Ventana de ingredientes	23
3.4. App - Ventana de platos	25
3.5. App - Ventana de planificador	27
3.6. App - Ventana de plan alimenticio	29
5.1. Gráficas - Elección de platos	44
5.2. Gráficas - Tamaño de población	45
5.3. Gráficas - Probabilidad de cruce y mutación	46

Índice de tablas

2.1. Formulaciones del problema	12
2.2. Aplicaciones	17
5.1. Parámetros - Elección de platos	43
5.2. Parámetros - Tamaño de población	44
5.3. Parámetros - Probabilidad de cruce y mutación	45
8.1. Tiempo dedicado y presupuesto	55

Capítulo 1

Planificador de menús

1.1. Introducción

Uno de los principales problemas en la sociedad actual, tanto por la extensión del problema como por su importancia, es la forma inadecuada de alimentarse que sufre parte de la población. El ritmo de vida ajetreado de una sociedad moderna actual en la que las personas, sin distinción de género, trabajan y se desplazan de un lugar a otro de forma constante no deja en muchas ocasiones tiempo suficiente para dedicar a la adquisición y preparación de alimentos de calidad. Por este motivo, entre otros, en las últimas décadas se ha incrementado el consumo de las llamadas comidas rápidas y snacks, los cuales en su mayoría no aportan los nutrientes necesarios que una buena dieta sana y equilibrada debe poseer. El aumento de la presencia de este tipo de comidas en la sociedad ha impulsado a su consumo incluso cuando no se tiene una necesidad real por falta de tiempo. Esto unido a una poca frecuencia en la realización de actividad física induce a altos niveles de sobrepeso y obesidad en la población, los cuales no han dejado de ir en aumento en los últimos años.

Por otra parte, los recientes episodios de crisis económica en muchos países desarrollados han hecho aumentar los índices de pobreza de la población en los mismos, con los consecuentes problemas de desnutrición que se producen. Este problema de malnutrición, tanto por falta de nutrientes esenciales como por exceso, o por una ingesta en proporciones erróneas es especialmente grave cuando sucede en los niños. Por ello, es de vital importancia disponer de una dieta sana y equilibrada.

1.2. Definición del problema

El problema del planificador de menús consiste en la generación para el usuario de un plan dietético, generalmente semanal o mensual, compuesto por platos o menús saludables para las principales comidas del día. Para ello se debe contar

con una base de datos de platos y alimentos con sus respectivas especificaciones nutricionales y un método que sea capaz de organizar como mínimo un plan dietético basado en los estándares nutricionales RDA (*Recommended Dietary Allowances*). Es importante que el menú generado se adapte a las condiciones fisiológicas del usuario, principalmente su género y edad, ya que la cantidad de nutrientes recomendada varía según estos factores en particular, a los que indudablemente se puede añadir otros factores a tener en cuenta para una recomendación dietética más precisa.

Resulta interesante que el sistema no sólo se limite a ofrecer planes dietéticos nutricionalmente adecuados, sino que también sea capaz de adaptar dichos planes a necesidades particulares del usuario como por ejemplo sus preferencias personales sobre el tipo de comida, el coste económico de los menús, la variedad en los platos, características estéticas del alimento, métodos de preparación, incompatibilidades con determinados alimentos por cuestión de alergias, intolerancias o estilos de vida específicos, calidad del alimento basado en su en su temporada/periodo de producción, etc.

En este sentido, el problema contaría con objetivos más allá de ofrecer un menú saludable, por lo que estaríamos hablando de un problema multi-objetivo, y puesto que se trata de optimizar los valores antes mencionados, en el problema tratados como valores de una función objetivo, se trataría de un problema de optimización multi-objetivo. No obstante no todos los posibles aspectos a tener en cuenta antes mencionados serían tratados como objetivos, ya que la realización de una función objetivo para cada uno de ellos incrementaría notablemente la dificultad en la resolución del problema, por ello es que la mayoría de estos aspectos serían representados como restricciones del problema. Por lo que finalmente tendríamos un problema con una serie de funciones objetivo a optimizar sujetas a unas determinadas restricciones.

El problema del planificador de menús, o *menu planning*, debido a su condición multi-objetivo y multi-restricción, es comparado frecuentemente con el problema de la mochila multi-dimensional. En el problema de la mochila clásico se dispone de una mochila con una determinada capacidad en la que se tiene que introducir un número finito de objetos con un valor establecido, de tal forma que la suma del peso de los objetos no supere el peso máximo permitido por la mochila y al mismo tiempo se maximice el valor de los objetos introducidos. La diferencia con respecto al problema de la mochila multidimensional es que para este contamos con un número finito de mochilas, o de volúmenes dentro de una mochila, para los cuales tenemos que satisfacer el objetivo antes mencionado.

Formalmente se define de la siguiente forma:

$$\begin{aligned} & \text{maximizar} \quad z = \sum_{j=1}^n p_j x_j \\ & \text{sujeto a} \quad \sum_{j=1}^n w_{ij} x_j \leq c_i, \quad i = 1, \dots, m, \\ & \quad \quad \quad x_j \in \{0, 1\}, \quad j = 1, \dots, n. \end{aligned}$$

En donde n es el número de objetos con valor $p_j > 0$ y m el número de mochilas o volúmenes con capacidad $c_i > 0$. El peso del objeto j en el volumen i es representado por w_{ij} , y x_j indica si el objeto j está seleccionado (1) o no (0).

La similitud del problema de la mochila multi-dimensional con respecto al problema aquí tratado se debe a que en el *menu planning* contamos con una serie finita de días, los días que componen el tiempo total al que está dirigido el plan, equivalentes al número finito de mochilas. Adicionalmente disponemos de un conjunto de platos con unos valores nutricionales concretos que deben ser evaluados con el objetivo de que el menú, en su totalidad, cumpla con unos requisitos nutricionales establecidos para cada día, y por extensión, al número total de días establecido. Este aspecto en el problema de la mochila sería identificado por el conjunto de objetos a introducir en cada mochila. Cada objeto tiene un valor establecido y el objetivo es optimizar el valor total de los objetos almacenados en una mochila, y por extensión, al total de mochilas disponibles.

1.3. Objetivos, restricciones y variables

El asunto sobre qué aspectos considerar como objetivos del problema y cuáles considerar como restricciones puede diferir entre distintos autores. No obstante se tiende a asignar el coste del menú como un objetivo, representado por una función a minimizar. Los autores [1, 5, 8, 16, 20, 21, 23, 28, 29, 30, 39] que no tienen en consideración el coste del menú suelen incluir las preferencias del usuario sobre ciertos alimentos, nivel de adecuación o nivel de aceptación del mismo como un objetivo. En este caso, maximizar el nivel de adecuación del menú al usuario. Otra autora, B. Seijak [7] incluye características del alimento, como por ejemplo la calidad del mismo, dentro de los objetivos del problema.

El aspecto más importante del problema, conseguir un menú saludable y equilibrado desde el punto de vista nutricional, se agrupa generalmente en múltiples restricciones, ya que éstas obedecen a cumplir con unas cantidades recomendadas mínimas y máximas de diferentes nutrientes que obviamente no son las mismas para cada uno de ellos. Otras restricciones comunes son la variedad de

los platos, tiempo de preparación del alimento, alimentos que no pueden ser consumidos, etc.

Finalmente, según el nivel de completitud que se le quiera dar al planificador, puede haber numerosos aspectos a tener en cuenta. En primer lugar y más importante, los datos del usuario. Si el planificador de menús está indicado a personas a partir de la edad adolescente, las dietas recomendadas difieren de un individuo a otro principalmente según su género y edad. Adicionalmente se pueden añadir otros aspectos como el peso, la altura, el nivel de actividad física o los objetivos en términos de salud del usuario para así realizar un plan más preciso. Otro aspecto importante y que no tienen en consideración muchos de los proyectos revisados en la literatura es si el usuario en cuestión padece algún tipo de alergia, intolerancia, padecimiento, o si simplemente su estilo de vida le impide consumir determinados alimentos. La condición del usuario a pertenecer a uno de estos grupos se consideraría como una restricción específica para eliminar de su plan dietético el alimento que proceda.

Otros trabajos, como el de N. Funabiki *et al.* [28], tienen en cuenta el tiempo disponible del usuario para la preparación del alimento, por lo que esto también sería aportado al planificador.

1.4. Planificador de menús propuesto

En este proyecto se plantea la creación de un planificador de menús dirigido a un comedor escolar, lo cual conlleva una serie de particularidades que lo hacen diferir en cierto modo del problema general.

En primer lugar los comedores escolares trabajan principalmente en horario de mediodía, por lo que nuestro planificador está enfocado a la recomendación de menús para el almuerzo, excluyendo así el resto de comidas del día.

En segundo lugar, el público al que va dirigido son niños y adolescentes, con lo que la cantidad diaria recomendada y adecuada de nutrientes varía con respecto a la de un adulto. No obstante, la cantidad de nutrientes recomendados para niños entre 4 y 13 años no difiere en exceso por género, según datos de AEP (Asociación Española de Pediatría) recogidos en el *Libro blanco de la nutrición infantil* [2], por lo que la aplicación no hará esta distinción. Por otra parte, a diferencia de las aplicaciones planificadoras de menús convencionales, los menús en este proyecto estarán dirigidos a un gran número de personas y no a personas de forma individual, por lo que la recogida de datos físicos, objetivos y preferencias del usuario para realizar un plan personalizado más acorde a sus necesidades no cobra sentido en este caso.

En tercer lugar, otros aspectos que se han tenido en cuenta en otras aplicaciones del problema referentes al alimento, como las características del mismo

(calidad, temperatura, etc) e incluso tiempo de preparación no serán contemplados. Sí que serán considerados otros aspectos directamente relacionados con los alimentos como son su condición de ser o no alimentos de temporada y el coste económico de cada plato, el cual influye en el coste económico total del menú a realizar. Por otra parte, la variedad de los platos en el menú de un plan que se realizará para un número de días determinado es indispensable. Estos dos últimos aspectos, **coste** y **variedad** en los platos, serán los objetivos del problema.

Por último se tendrá en cuenta otra característica muy importante en la planificación de cualquier menú, y son las posibles intolerancias, alergias o incompatibilidades alimenticias presentes en una persona para consumir determinados alimentos. Estas particularidades en la realidad son tratadas de forma diferente, pero desde el punto de vista de la aplicación, se trata de un alimento que no puede ser consumido por el usuario, ya sea por intolerancia, alergia, o porque su condición de vida no lo permite (religión, dieta, etc). Por tanto se tratará como una restricción más del problema.

En resumen, la planificación de objetivos, restricciones y variables para este problema será la siguiente.

Objetivos

- Precio total del plan alimenticio
- Variedad de los platos

Restricciones

- Valores nutricionales recomendados
- Intolerancias alimenticias, alergias e incompatibilidades

Variables

- Cada uno de los menús diarios de almuerzo, divididos en primer plato, segundo plato y postre. El número de menús de almuerzo dependerá del número de días para el cual el usuario quiera crear el plan alimenticio.

1.5. Información dietética y nutricional

Finalmente, para la realización de un proyecto dedicado al problema del *Menu Planning*, es indispensable la utilización de bases de datos y documentación que recojan abundante información sobre dietética, nutrición, alimentos y cantidades recomendadas, alimentos de temporada, consejos nutricionales, etc.

En primer lugar tenemos [BEDCA](#) [36] (Base de Datos Española de Composición de Alimentos), desarrollada en conjunto con AESAN (Agencia Española de Seguridad Alimentaria y Nutrición) y avalada por el Ministerio de Sanidad, Servicios Sociales e Igualdad y el Ministerio de Ciencia e Innovación del Gobierno de España. Esta base de datos contiene, de forma muy completa, información nutricional de cientos de alimentos por cada 100 gramos de producto comestible. La búsqueda de alimentos se puede realizar en orden alfabético por nombre, por grupos de alimentos y por composición. Principalmente la base de datos contiene alimentos separados de forma individual, aunque también aparecen algunos conjuntos de alimentos, comidas o platos preparadas, como por ejemplo bocadillos, pizzas, paellas, etc.

También se ha consultado las Guías de Comedores Escolares [32] [4] proporcionadas por AESAN, las cuales ofrecen abundante información y consejos sobre ingestas y menús recomendados para niños y adolescentes, alimentos prohibidos según determinadas alergias, intolerancias o características dietéticas especiales, etc. Otras guías nutricionales de interés son las proporcionadas por la [UNED](#) (Universidad Nacional de Educación a Distancia) [40] y por el [Portal de salud](#) [12] de la Comunidad de Madrid.

Con respecto a los alimentos de temporada, existen numerosas webs donde poder consultar listas con esta información, algunas de ellas son las proporcionadas por la [OCU](#) (Organización de Consumidores y Usuarios) [31], y la web [Naturesan](#) [34].

Otro aspecto importante considerado en este trabajo son los posibles alérgenos e incompatibilidades alimenticias de los alimentos. Para los alérgenos en los alimentos la web [AEPNAA](#) (Asociación Española de Personas con Alergia a Alimentos y Látex) [3] ofrece abundante información al respecto. Así mismo, para obtener información acerca de la dieta de personas celíacas se he consultado [FACE](#) (Federación de Asociaciones de Celíacos de España) [17] y para la dieta de personas diabéticas se ha consultado la web [FD](#) (Fundación para la Diabetes) [18].

Para la posterior generación de platos en la aplicación se ha consultado la web de [Dietas.net](#) [14], en donde se muestra gran cantidad de platos ordenados por grupos alimenticios y con sus respectivos ingredientes y cantidades detallados.

Finalmente para la obtención de los precios de los alimentos se han consultado las siguientes webs correspondientes a mayoristas en productos alimenticios: [Mercatenerife](#) [27], [Mercalaspalmas](#) [26], [Makro](#) [25], así como el centro comercial [Carrefour](#) [9].

Capítulo 2

Estado del arte

El objetivo de diseñar una dieta adecuada a base de menús diarios nutricionalmente equilibrados, según las especificaciones RDA (*Recommended Dietary Allowances*), es un asunto de vital importancia en la actualidad que ha sido tratado en numerosas ocasiones. Por ello se ha realizado un estudio de los trabajos existentes al respecto así como de las diversas aplicaciones web y/o móviles que tratan el problema del planificador de menús.

2.1. Formulaciones del problema

En primer lugar, la búsqueda de información para este trabajo se ha centrado en la investigación de otras formulaciones del problema, prestando especial atención a las formulaciones multi-objetivo realizadas mediante algoritmos evolutivos. En este sentido, un trabajo conocido y muy completo es el de B. Seijk [7], en el que se resuelve el *Menu Planning* mediante el algoritmo evolutivo *Non-Dominated Sorting Genetic Algorithm* (NSGA-II). En este trabajo se realiza un menú semanal compuesto por cinco comidas diarias valiéndose de los objetivos de mínimo coste, calidad y funcionalidad de los alimentos, alimentos de temporada y otras características, además de las restricciones de preferencias personales del usuario y requisitos nutricionales. Además, en el artículo se hace una comparación del *Menu Planning* con el problema de la mochila multi-dimensional, con el que, como se ha comentado en el capítulo anterior, guarda una estrecha relación en su definición y forma de resolver.

Otro trabajo que utiliza el NSGA-II mediante una formulación multi-objetivo es el de E. Kaldirim *et al.* [15], aunque su trabajo está compuesto por un número menor de objetivos en comparación con el anterior, ya que reduce el problema a dos objetivos compuestos por el coste del alimento y la preferencia del mismo por parte del usuario mediante un ranking de 0 a 10 con el que clasifica los alimentos. Un aspecto positivo de este trabajo es que considera el género y la edad del usuario para generar un menú adecuado a dichos parámetros, mientras que B. Seijk [7] utiliza datos estándares para una persona adulta.

Es importante destacar que estos dos trabajos mencionados son los únicos encontrados en los cuales se realiza una formulación multi-objetivo del problema, puesto que en la mayoría de trabajos, aunque se realice una descripción multi-objetivo del problema, se acaban reduciendo a una formulación mono-objetivo. Un ejemplo de esto es el trabajo de P. Gallardo *et al.* [33]. En él se realiza un menú para cuatro comidas diarias cuyos objetivos son el coste del menú y el grado de adecuación. No obstante estos dos objetivos son reducidos a una única función objetivo mediante la aplicación del método *epsilon* restricción, o *epsilon-constraint method*.

De la misma forma, J. Hsiao *et al.* [24] utilizan en su trabajo, una aplicación móvil llamada Smart Diet, el método de la suma ponderada para reducir sus objetivos de precio del menú, preferencias del usuario y plan adecuado y equilibrado nutricionalmente a un solo objetivo. Con respecto a las restricciones del problema, al igual que el anterior autor, se tiene en consideración la diversidad de los menús así como otras restricciones nutricionales.

Un trabajo interesante con un enfoque no visto en otros anteriores es el propuesto por N. Funabiki *et al.* [28]. En este trabajo se plantea el problema del *Menu Planning* desde el enfoque *Two-phases cooking*, una forma de cocinar en la que en la primera fase se realizan los preparativos iniciales para cocinar un alimento (preparar, cortar, condimentar, etc) para ser almacenados y en una segunda fase terminar de ser cocinado y consumido. En pocas palabras, preparar un alimento para ser consumido en días posteriores. Debido a que este enfoque está pensado principalmente para personas con poca disponibilidad de tiempo, en su trabajo introducen gran cantidad de variables y restricciones referentes al tiempo disponible, tiempo de preparación, etc. Otros aspectos interesantes son el número de personas y de días para los que se quiere adaptar el plan. Dispone de una única función objetivo que trata de maximizar la elección de platos preferidos por el usuario y el problema finalmente se resuelve mediante un algoritmo heurístico voraz enfocado al problema de la mochila.

El trabajo de H. Alfaro [21] hace también una formulación mono-objetivo del problema cuyo objetivo y restricciones se centran principalmente en satisfacer la adecuación del menú según los requisitos nutricionales recomendados. Su trabajo utiliza aplicaciones de Microsoft (interfaz en Visual Basic, Access y Excel Solver) para tratar el problema.

Las siguientes propuestas mencionadas [6, 11, 39, 37, 23, 16] utilizan algoritmos evolutivos, más concretamente algoritmos genéticos, en sus formulaciones. Aunque, a diferencia de los dos primeros trabajos descritos en esta sección, estos lo hacen desde un enfoque mono-objetivo.

B. Gaál *et al.* [6] hacen especial hincapié en el aspecto multi-nivel de su trabajo. Esto es debido a que su planificador es capaz de proporcionar menús

individuales, diarios y semanales. Mediante una perspectiva divide y vencerás se puede desglosar un plan semanal en planes diarios, que a su vez acaban siendo conformados por conjuntos de platos únicos. Su planificador, llamado MenuGene, utiliza un algoritmo genético para resolver el problema. Datos personales del usuario como el género, edad y peso también son considerados por la aplicación.

El trabajo de D. Osthus [11] se enfoca principalmente en combatir la malnutrición, y para ello centra las restricciones del problema en conseguir unos niveles adecuados de vitamina A y retinol. La función objetivo del problema pretende minimizar el coste del menú. También hace uso de un algoritmo genético para la resolución del problema.

Otro trabajo que tiene en cuenta el problema de la mochila multi-dimensional para resolver el planificador de menús es el de T. Kashima *et al.* [39]. Mediante un algoritmo genético, los autores han decidido abordar este problema convirtiéndolo en un mono-objetivo teniendo como objetivo el nivel de satisfacción del usuario sobre determinado alimento, siendo las restricciones los requisitos nutricionales.

El siguiente proyecto, realizado por S. Gumustekin *et al.* [37] resuelve el problema mediante tres métodos diferentes con el objetivo de evaluar sus resultados y rendimiento. El primero es un algoritmo de optimización bayesiano, un tipo específico de algoritmo de estimación de distribución. En segundo lugar utiliza un algoritmo genético, y en tercer lugar un método de programación lineal. Su planificador únicamente ofrece menús para el desayuno, y su objetivo es minimizar el coste. El método que mejor resultados ofrece en términos de de minimización de coste y tiempo de ejecución es el de optimización bayesiano.

J. Bulka *et al.* [23] realizan un planificador de menús especialmente indicado para personas diabéticas, ya que no solo tienen en cuenta el equilibrio nutricional de los alimentos y las preferencias del usuario sino también el impacto de los alimentos en los niveles de glucosa en sangre. Para esta labor se basan nuevamente en la utilización de un algoritmo genético.

Finalmente F. Wang *et al.* [16] utilizan otro algoritmo genético auto adaptativo en combinación con un sistema multi-agente en su proyecto. Cada agente realiza una función específica, basadas principalmente en la evaluación de los resultados, la optimización de la función de razonamiento y la mejora de la base de conocimiento.

El grupo de propuestas citadas a continuación [1, 29, 20, 5, 8, 22, 30] utiliza otros enfoques y técnicas de inteligencia artificial diferentes a los algoritmos evolutivos.

En primer lugar dentro de esta última sección está el trabajo de A. Kale *et al.* [1]. Su planificador está dirigido a la creación de menús para niños en la India. La elección del menú se basa en las características físicas de cada niño, su nivel de actividad física, la disponibilidad de los alimentos y las preferencias de los mismos. Para ello utilizan el algoritmo ID3 con el software Weka, diseñado para el aprendizaje automático y la minería de datos.

N. Jothi *et al.* [29] utilizan un sistema basado en conocimiento en su trabajo. Para recomendar un plan dietético al usuario, el sistema comprueba si existen casos previamente resueltos en base a los datos personales del usuario, como la edad, el género, el peso, preferencias alimenticias, etc. En caso afirmativo el sistema recomienda el menú del caso similar, en caso contrario se utiliza un sub-sistema basado en razonamiento basado en reglas (RBR) para ofrecen una solución diferente.

La siguiente propuesta, realizada por G. Petot *et al.* [20], ha sido referenciada con frecuencia en numerosos artículos y revisiones del problema. Los autores fusionan dos sistemas expertos en uno capaz de ofrecer mejores resultados. En primer lugar está el sistema CAMP, un sistema de razonamiento basado en casos y utilizado para el planificador de menús. En segundo lugar está PRISM, un sistema RBR utilizado para la selección inteligente de menús. La fusión da lugar al sistema CAMPER, el cual combina, según sus autores, la habilidad para satisfacer restricciones nutricionales de CAMP con la creatividad para combinar platos de PRISM para dar un mejor resultado. Este sistema, aunque tiene en consideración las preferencias alimenticias del usuario, permite cambiar un plato sugerido por otro.

El trabajo de A. Khan *et al.* [5] también utiliza un sistema de razonamiento basado en casos. Este sistema utiliza la idea de las *ripple-down rules* para permitir a un experto añadir reglas directamente en la base de conocimiento. El sistema cuenta con dos modos, un modo de consulta que permite generar un menú adecuado a las necesidades del usuario y un modo de adquisición de conocimiento, el cual permite añadir reglas a la base de conocimiento en caso de que el usuario no esté de acuerdo con la recomendación ofrecida. Este sistema es la evolución de un sistema anterior utilizado en un hospital, por lo que se tiene en consideración particularidades del usuario como alergias, intolerancias u otro tipo de restricciones alimenticias.

C. Lee *et al.* [8] utilizan lógica difusa en su trabajo. Concretamente se combina un sistema difuso de tipo 2 con el modelo ontológico para crear un set de mecanismos de lógica difusa, cada uno de los cuales es asignado a una tarea específica en el planificador de menús. Tareas como la de recomendar una dieta, planificarla o la descripción semántica, propia de los sistemas difusos, permiten dar solución al problema.

Tabla 2.1: Formulaciones del problema

	Meta-heurísticas	Métodos exactos	Otros métodos
Mono-objetivo	B. Gaál [6] D. Osthus [11] T. Kashima [39] S. Gumustekin [37] J. Bulka [23] F. Wang [16]	N. Funabiki [28] J. Aberg [22]	Gallardo P.R. [33] J. Hsiao [24] H. Alfaro [21] A. Kale [1] N. Jothi [29] G. Petot [20] A. Khan [5] C. Lee [8] O. Chávez [30]
Multi-objetivo	B. Seijak [7] E. Kaldirim [15]		

En el siguiente trabajo, realizado por J. Aberg [22], se utiliza un método *branch and bound* con un *forward-checking* parcial, una versión mejorada de *backtracking*. Esta propuesta tiene la particularidad de presentarse como un problema de satisfacción de restricciones, en el que no se trata ningún objetivo ya que todo se considera como un conjunto de restricciones sobre variables. Otro aspecto a tener en cuenta de este trabajo, que también lo consideraba A. Khan *et al.* [5] pero que no está presente en la mayoría de ellos, es la restricción dietética a los usuarios por cuestión de alergias, intolerancias u otros motivos médicos.

Finalmente está el trabajo de O. Chávez *et al.* [30] en el cual se utiliza, junto con una base de conocimiento, una combinación de frameworks. Uno es un algoritmo de programación lineal y otro un operador de fusión de creencias (*belief merging*) cuyo objetivo es reforzar la base de conocimiento. Este proyecto destaca por su gran cantidad de aspectos del usuario a tener en cuenta, como sus preferencias alimenticias, género, edad, peso, nivel de actividad física, intolerancias e incluso técnica de preparación del alimento.

En resumen, las distintas formulaciones del problema según los diferentes autores pueden ser agrupadas en la Tabla 2.1.

Por último, y después de realizar esta revisión por diferentes formulaciones del problema, se puede concluir lo siguiente. Prácticamente la totalidad de los trabajos consideran las preferencias del usuario por determinados alimentos así como, de forma evidente, el equilibrio nutricional de los mismos a la hora de recomendar un menú. El coste económico del menú es un aspecto también considerado y tratado en numerosas ocasiones como un objetivo del problema.

Tener en cuenta variables relacionadas con características fisiológicas del usuario, como por ejemplo la edad, género, peso, nivel de actividad física, etc, es sin lugar a dudas un añadido importante, ya que dependiendo de estos valores el menú recomendado puede variar en gran medida.

Si hay algo que destacar de las revisiones realizadas es que la mayoría de los proyectos se centran en la creación de un menú de propósito general para cualquier usuario, generalmente un adulto. La aplicación que se plantea en este proyecto está especialmente enfocada a menús escolares, y por lo tanto, el tipo de usuarios al que se dirige es a niños y adolescentes, cuyos planes alimenticios recomendados difieren al de una persona adulta. También debido a esto, los menús recomendados en este proyecto se dirigen a un gran conjunto de personas, no a personas de forma individual, por lo que algunas restricciones relacionadas con las características físicas o niveles de actividad del usuario no serán consideradas.

Otro punto muy importante a destacar es que se han observado muy pocos proyectos en la literatura consultada que consideren como objetivos y/o restricciones aspectos referentes a los principales alérgenos o intolerancias alimenticias, así como tampoco posibles restricciones en los menús por cuestión de religión o estilo de vida, algo que se pretende implementar en este proyecto.

2.2. Aplicaciones existentes

Para complementar a la literatura relacionada con el problema se ha realizado una búsqueda sobre aplicaciones web y/o móviles existentes relacionadas. A continuación se hará un repaso sobre las características de algunas de ellas:

- **Tesco Real Food:** es una aplicación [web](#) que permite crear un plan de menús para toda la semana. Por cada día la aplicación sugiere un plato para desayuno, almuerzo y cena. Además, ofrece información nutricional sobre la cantidad de calorías, azúcar, grasas, grasas saturadas y sal que contiene cada menú diario. Entre otras cosas, la aplicación permite intercambiar platos sugeridos por otros, que podrán ser elegidos manualmente o generados por la aplicación, y acceder a la compra online de los diferentes ingredientes que componen los platos del menú. La aplicación también permite seleccionar la generación de un plan para vegetarianos, comidas sin huevo, sin gluten, baja en calorías, baja en grasas, sin lactosa, o sin cereales. Además, permite añadir ingredientes preferidos o rechazados para realizar una mejor adaptación al plan. Otro añadido es la indicación de alimentos de temporada y la posibilidad de tenerlo en cuenta para generar el menú.

Realizando algunas pruebas en esta aplicación se ha comprobado que en

términos generales funciona bastante bien. En las consultas realizadas la aplicación ha generado menús diversos según las condiciones impuestas, y la opción de cambiar un plato propuesto por otro resulta bastante útil. El inconveniente más notable es la opción de generación de menú habiendo añadido un ingrediente preferente y según qué ingrediente se haya indicado. Por ejemplo, si se añade como preferencia un alimento no muy común en los menús, como puede ser algún tipo de marisco o molusco, la aplicación recomendará muy pocos o ningún plato a lo largo de la semana que contenga ese ingrediente. Sin embargo si se añade como preferencia un alimento mucho más común en los menús, como por ejemplo el pollo, la aplicación recomendará platos que contienen pollo prácticamente durante todos los días para los que se realiza el plan. Sería conveniente un mayor equilibrio en este sentido, y quizás la opción de que el usuario pueda elegir un alimento preferente o rechazado de un total de alimentos proporcionados por la aplicación, para evitar así que el usuario pueda introducir un ingrediente mal escrito, poco común o inexistente en la base de datos de platos de la aplicación.

- **Eat This Much:** es una aplicación [web](#), disponible también en la *App Store* y próximamente en *Google Play*, que genera automáticamente un menú para un día, compuesto por de uno a nueve platos, basado principalmente en la cantidad de calorías que el usuario quiera consumir. En caso de que el usuario no tenga una idea exacta de la cantidad de calorías que debe o quiere consumir al día, la aplicación hace una recomendación en base a una serie de datos personales, como el género, peso, talla, cantidad de grasa corporal, nivel de actividad física y objetivos que pretenda conseguir. La aplicación también permite seleccionar un menú editando la cantidad de grasas, carbohidratos, proteínas, colesterol, fibra y sodio que se quiera. También se puede decidir en cuántos platos dividir la cantidad de calorías diarias que se quiere consumir, desde un plato hasta nueve al día. Otras características que ofrece es la información nutricional tanto del menú completo como para cada uno de los platos de forma individual, también la posibilidad de elegir menús de dietas específicas (mediterránea, apta para vegetarianos, apta para veganos, etc).

Tras la realización de algunas pruebas con la aplicación se ha comprobado que en general funciona correctamente para el propósito que tiene. Esta aplicación está más enfocada a generar una dieta en la que predomina el control sobre las cantidades de nutrientes deseados, principalmente calorías, carbohidratos, proteínas y grasas. En este sentido hace un buen trabajo ya que los menús recomendados cumplen generalmente con estas limitaciones, además de advertir de las posibles inconsistencias en los datos introducidos por el usuario. Por ejemplo, no es posible pedir un menú bajo en calorías y alto en grasas o carbohidratos al mismo tiem-

po. Otro punto positivo es la gran cantidad de información nutricional que contienen sus menús. Como inconvenientes, el número de opciones a editar para personalizar un plan es menor que en la aplicación anterior, y quizás la variedad de alimentos no es muy extensa, ya que los platos se repiten al cambiarlos por otros con cierta frecuencia.

- **Free Custom Fitness Meal Planner:** es una aplicación [web](#) que genera un plan de comida para el día, compuesto por de tres a seis platos. La aplicación se basa en los datos físicos del usuario (género, talla, peso, nivel de actividad física y objetivos físicos a alcanzar) para ofrecer una lista de múltiples opciones para cada uno de los platos, basada también en la preferencia en el tipo de dieta (mediterránea, vegetariana, vegana, paleo, etc). La aplicación no ofrece un plato concreto para cada comida, sino que en base a las preferencias y datos del usuario se ofrece una serie de platos a elegir, compatibles con dicha información administrada, para cada una de las comidas. Una vez elegido un plato se muestra información nutricional detallada sobre el mismo.

Esta aplicación es sin lugar a dudas la peor de las cuatro aquí analizadas. Tanto en menús propuestos como en el diseño, el cual es muy discreto en detalles y hasta confuso en el modo en el que muestra la información. La cantidad de platos propuestos, como se ha comentado anteriormente, es muy reducida y se repite en las diferentes comidas del día. Lo único bueno es la información nutricional aportada por cada comida, que por otra parte no es tan numerosa como en otras aplicaciones. Da la sensación de ser una aplicación realizada como parte de un proyecto o prueba para la realización de un planificador de menús, pero no para ser utilizada por el público en general.

- **Eat Right Ontario:** es una aplicación [web](#) que genera un menú diario para toda la semana compuesto diariamente por tres comidas, con sus respectivas bebidas, y dos snacks. Se empieza por indicar el género del usuario, la edad y el tipo de plan que prefiere (mantener el peso, dieta para deportista, embarazadas, vegetarianos, fácil de cocinar, etc). Seguidamente se genera el menú diario semanal, en el que se puede intercambiar platos sugeridos. También se indica para cada menú la cantidad de tipo de comida existente, separada por: fruta y vegetales, productos de grano, carne y pescado, y leche y derivados. Finalmente la aplicación permite hacer una lista de la compra de ingredientes basado en el menú.

Esta aplicación no es peor que la anterior pero sí peor que las dos primeras. La razón principal es su escasa cantidad de platos en su base de datos, que si bien tiene los suficientes para no repetirse a lo largo de una semana, sí que se acabarían repitiendo si se hace un plan para varias semanas. Incluso si se quiere cambiar un plato sugerido por otro, las opciones disponibles son pocas. Este problema es notable en las comidas, y en las

bebidas la variedad es aún menor. Otro punto negativo a destacar es que no ofrece información nutricional detallada de los alimentos, simplemente la clasificación por grupos de alimentos antes mencionado.

En la siguiente tabla se muestra, de forma resumida, una valoración las principales características de cada una de las aplicaciones.

Tabla 2.2: Aplicaciones

	Plataforma	Tipo de plan	Comidas al día	Variedad de platos	Cambio de platos	Datos del usuario	Necesidades especiales	Personalización	Información nutricional	Usabilidad y diseño
Tesco Real Food	web	semanal	3	alta	sí	ninguno	alto	alta	moderada	bueno
Eat This Much	web	diario	de 1 a 9	moderada	sí	alto	alto	alta	alta	bueno
Free Custom Fitness MP	web	diario	de 3 a 6	baja	sí	alto	moderado	moderada	alta	malo
Eat Right Ontario	web	semanal	5	baja	sí	bajo	alto	baja	baja	bueno

Capítulo 3

Menu Planning App

Para abordar el problema del *Menu Planning* de una forma más eficiente y cómoda, sobre todo en el aspecto visual, se ha diseñado y desarrollado una aplicación de escritorio con interfaz gráfica que permitirá visualizar y gestionar todo lo relativo a los ingredientes, platos y menús recomendados. La interfaz nos permitirá definir la instancia del problema (elegir los platos a considerar y el número de días del plan alimenticio), pero también la resolución del mismo, así como la visualización de las soluciones obtenidas.

3.1. Propósito general

La idea tras el desarrollo de esta aplicación es que cualquier usuario, independientemente del nivel de conocimientos y manejo de aplicaciones informáticas que posea, sea capaz de beneficiarse de las técnicas y métodos realizados en este trabajo para la administración y gestión de los menús en un centro escolar.

Para ello, la aplicación desarrollada cuenta con una interfaz sencilla y fácil de manejar, con un diseño intuitivo que permite, en un periodo muy corto de tiempo, comprender todo lo que la aplicación contiene y de lo que es capaz de hacer. Las tareas a realizar por el usuario solamente se limitarán a tres aspectos principalmente. En primer lugar la introducción de datos correspondientes a los platos disponibles en el centro escolar así como cada uno de sus ingredientes. En segundo lugar, la consulta de dichos datos con opción a la modificación de los mismos. Y por último, en tercer lugar, la creación de forma automática de un plan de menús para un número de días elegido por el usuario para el comedor escolar. Todas estas tareas se realizarán desde la aplicación de forma sencilla, sin que el usuario tenga que preocuparse de otros aspectos relativos al problema.

3.2. Tecnología utilizada

Para el diseño y desarrollo de la aplicación se ha utilizado el software Qt, un programa multiplataforma ampliamente utilizado en el desarrollo de aplicaciones con interfaz gráfica de usuario. Qt [35] ha sido desarrollada como un software libre de código abierto, por lo cual además de poseer una edición comercial también cuenta con una edición completa *Open Source*. El lenguaje de programación nativo de Qt es C++, aunque existe una serie de diferentes lenguajes que pueden ser utilizados a través de *bindings*. No obstante y debido entre otras cuestiones a la optimalidad y eficiencia de C++ [38] como lenguaje para el desarrollo de aplicaciones se ha decidido usar este lenguaje para la creación del programa.

Una de las características más llamativas de Qt es el posible desarrollo de aplicaciones multiplataforma, para que éstas sean ejecutadas en diferentes sistemas operativos y dispositivos. Para este trabajo en concreto, una aplicación dedicada a un centro escolar, y a pesar de las múltiples posibilidades que ofrece Qt, se ha acordado que lo más conveniente es realizar una aplicación de escritorio que trabaje sobre el sistema Windows, ya que es dicha plataforma y sistema lo que se considera que se encontraría de forma más frecuente en un centro educativo y en el que los posibles usuarios de esta aplicación se encuentren más adaptados.

Otro aspecto obligatorio debido a la naturaleza de esta aplicación es la utilización de una base de datos para el almacenamiento del gran volumen de datos correspondientes a los ingredientes y platos disponibles en el centro escolar, con sus respectivas características e información nutricional. Para ello, y debido a que la base de datos no es significativamente compleja se ha escogido MySQL [41] como base de datos debido a su sencillez y facilidad de uso, ya que la información a almacenar por parte de la aplicación no es excesivamente grande ni compleja. Esta base de datos está dividida en tres tablas (ver Figura 3.1): alimentos, platos e ingredientes.

- **Tabla de alimentos:** se encarga de almacenar cada uno de los alimentos de forma individual así como toda la información necesaria correspondiente a los mismos. Los datos a almacenar son los siguientes: id del alimento, nombre, cantidad en gramos, precio, meses de temporada, alérgenos, incompatibilidades alimenticias, ingrediente principal, grupo alimenticio. Los datos referentes a los nutrientes son los siguientes: ácido fólico (μg), calcio (mg), energía (kcal), fósforo (mg), grasa total (g), hierro (mg), magnesio (mg), potasio (mg), proteínas (g), selenio (μg), sodio (mg), vitamina A (μg), vitamina B1 (mg), vitamina B2 (mg), vitamina B6 (mg), vitamina B12 (μg), vitamina C (mg), vitamina D (μg), vitamina E (mg), yodo

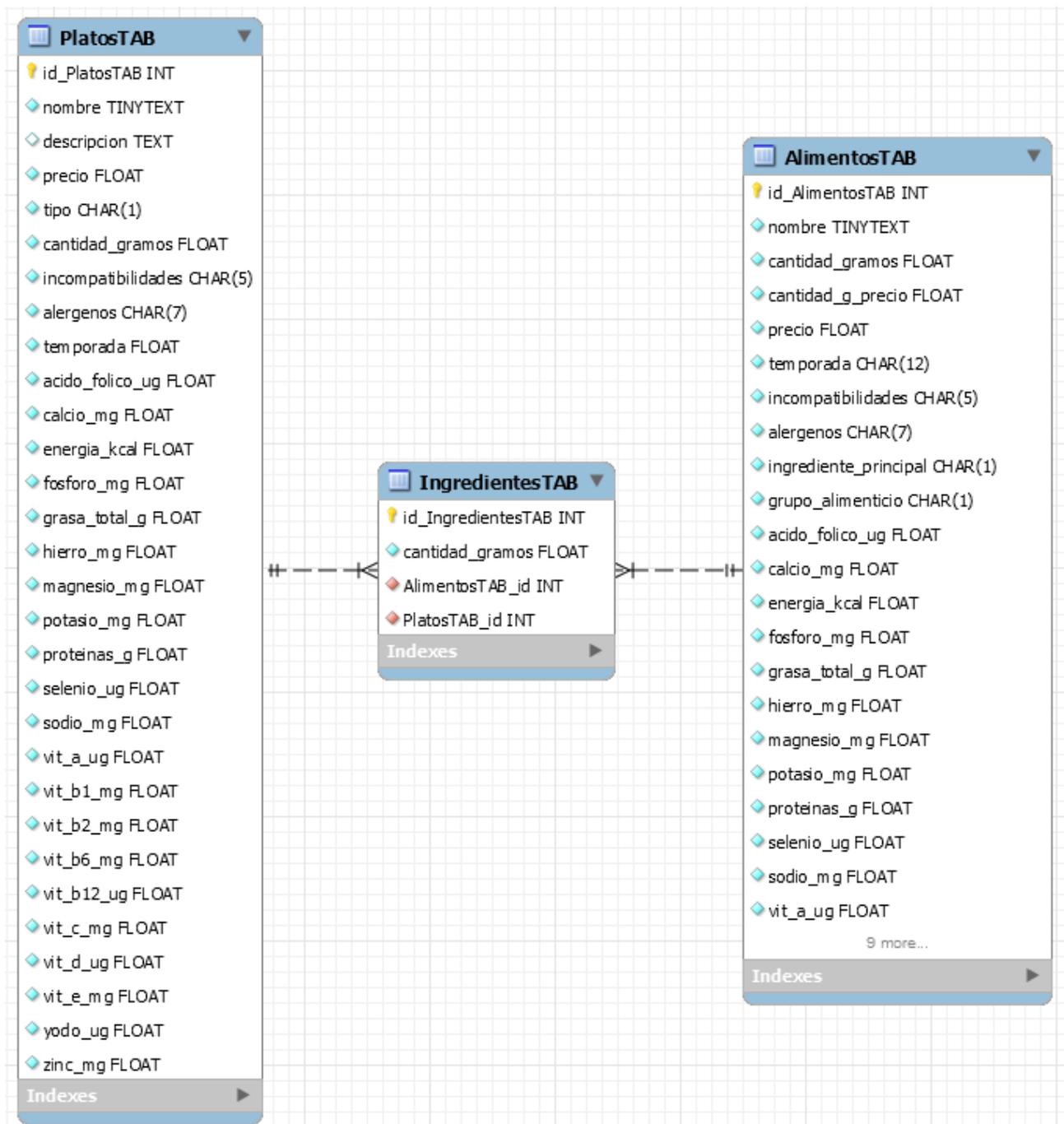


Figura 3.1: MySQL - Relaciones

(μg) y zinc (mg). Siendo gramos (g), miligramos (mg), microgramos (μg) y kilocalorías (kcal).

- Tabla de platos:** almacena cada uno de los platos creados por el usuario. Los datos que recoge son los siguientes: id del plato, nombre, descripción (opcional), precio, tipo (primer plato, segundo plato o postre), cantidad en gramos, alérgenos, incompatibilidades alimenticias, meses de temporada, así como todos los datos referentes a los nutrientes existentes en la tabla de alimentos.

- **Tabla de ingredientes:** es una tabla puente entre la tabla de alimentos y la tabla de platos que sirve para indicar cuáles de los alimentos están presentes en cada uno de los platos. Los datos recogidos por esta tabla son: id del ingrediente, cantidad en gramos, id del alimento e id del plato.

3.3. Descripción

En esta sección se visualizará cada una de las diferentes ventanas de la aplicación y se hará una breve descripción de las mismas, de su contenido y de las posibles acciones a realizar en cada una de ellas.

3.3.1. Inicio

Es la pantalla inicial que se encuentra nada más ejecutar la aplicación (ver Figura 3.2). Se utiliza como ventana de presentación y la única acción posible a realizar es la conexión y desconexión de la base de datos mediante el botón correspondiente a esta acción situado en la parte inferior derecha.

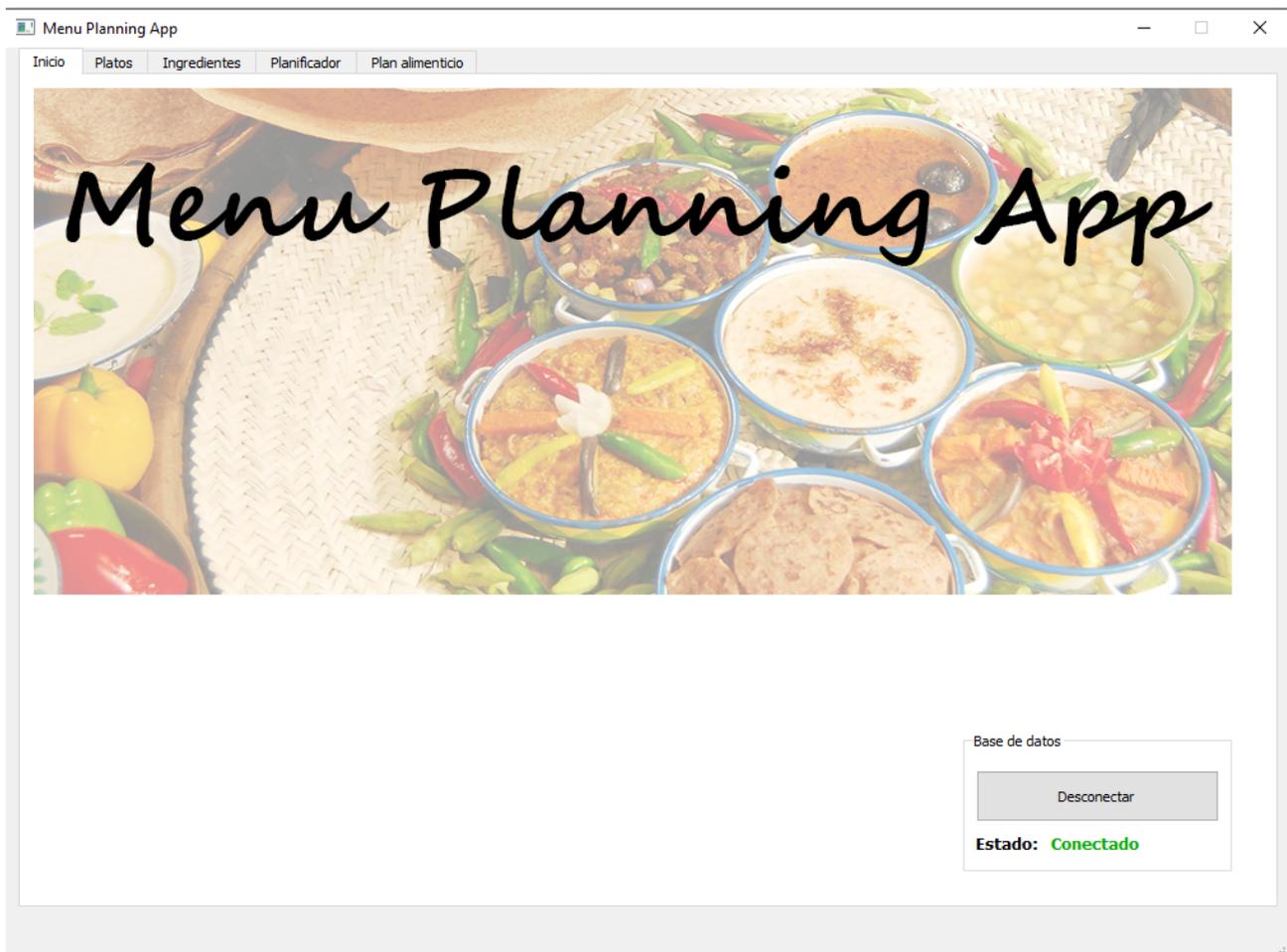


Figura 3.2: App - Ventana de inicio

3.3.2. Ingredientes

La pantalla de ingredientes es donde se almacenan todos los alimentos, por separado, existentes en la base de datos (ver Figura 3.3). Esta parte de la aplicación es de suma importancia ya que desde aquí se permite modificar la información nutricional, características, período de temporada, incompatibilidades y alérgenos de todos los ingredientes existentes, características que influirán directamente en los platos creados y por extensión, en los menús planificados. Desde aquí también se puede añadir nuevos ingredientes o eliminarlos.

Esta pantalla es similar a la anterior referente a los platos, salvo a que a diferencia de esta, todos los campos de texto y *check box* disponibles serán habilitados tanto al añadir un ingrediente nuevo como al modificar uno ya existente, de tal forma que no habrá ningún dato referente a los ingredientes que se rellene de forma automática por la aplicación. Si bien es verdad que si el usuario no introduce ningún dato en un campo determinado éste será rellenado por un valor por defecto, al ser los ingredientes, por separado, el elemento más simple que se puede desglosar de un plato, no hay ninguna característica de un ingrediente que pueda ser deducida de alguna información previa contenida en la aplicación. Por ello se recomienda al usuario rellenar manualmente todos los datos de cada ingrediente.

Al igual que en la pantalla dedicada a los platos, en la parte inferior derecha de la ventana se encuentran los botones **Añadir**, **Modificar** y **Eliminar**, y justo encima de ellos el listado con todos los ingredientes añadidos a la base de datos. Si se pulsa sobre el botón **Añadir** todos los campos de texto se habilitarán para su edición. La principal diferencia con respecto a los platos son los *check box* referentes a período de temporada, incompatibilidades y alérgenos del alimento, si el ingrediente en cuestión es un ingrediente principal o no, y el grupo alimenticio al que pertenece.

Los *check box* referentes a alérgenos e incompatibilidades servirán para limitar los platos recomendados en el menú en personas que posean algún tipo de incompatibilidad alimenticia y alergia de las especificadas en la aplicación.

Los alérgenos contemplados para servir de restricción a la creación de ciertos menús son los siguientes: alergia a los cereales, frutos secos, legumbres, marisco, pescado, proteína de huevo y proteína de la leche de vaca. De la misma forma, las incompatibilidades alimenticias contempladas son las siguientes: alimento no apto para celíacos, diabéticos, semivegetarianos (no comen carne de animales terrestres pero sí pescado), vegetarianos (no comen carne de ningún animal) y veganos.

El período de temporada se utilizará a título informativo para saber en que momento del año un ingrediente es de mayor calidad (si es de temporada) o no.

Figura 3.3: App - Ventana de ingredientes

La condición de que un ingrediente sea principal o no quiere decir que si su presencia en un plato suele ser abundante en cantidad (principal) o no. Esto servirá en la creación de menús para evitar que en un mismo menú del día se consuma el mismo ingrediente principal tanto en el primer plato como en el segundo.

Los grupos alimenticios disponibles son los siguientes: carne, cereal, fruta, lácteo, legumbre, marisco, pasta, pescado, verdura y otro. De la misma forma, el grupo alimenticio de un alimento servirá para evitar que en un mismo menú diario se repitan platos pertenecientes a un mismo grupo alimenticio, aportando de esta forma mayor variedad al plan alimenticio.

Cabe destacar que al igual que ocurre con la creación de platos, la aplicación no permite la creación de dos o más ingredientes con el mismo nombre.

Con respecto a la modificación y eliminación de un ingrediente existente el procedimiento es muy similar al anteriormente visto. Si se selecciona un ingrediente de la lista y se pulsa sobre el botón **Modificar** se habilitarán todos los campos de texto permitiendo la modificación de los mismos. Y si se selecciona un ingrediente de la lista y se pulsa sobre el botón **Eliminar**, la aplicación ad-

vertirá al usuario mediante una ventana emergente de que se procederá a la eliminación de un ingrediente. Si el usuario acepta dicha advertencia el ingrediente será eliminado de la aplicación y de la base de datos de forma permanente.

3.3.3. Platos

Desde esta pantalla se podrá gestionar todo lo relativo a los platos, de forma individual para cada uno de ellos (ver Figura 3.4). Las funcionalidades básicas son las de añadir nuevos platos a la base de datos, con sus respectivas características e ingredientes, modificar y/o eliminar platos ya existentes. Como se puede observar en la parte superior de la pantalla, las principales características a añadir en un plato son el nombre, la cantidad medida en gramos del alimento, su precio en euros y el tipo de plato (primer plato, segundo plato o postre). Adicionalmente se puede añadir una descripción al mismo, aunque esta no es obligatoria. Cabe destacar que todos los campos de texto, excepto el dedicado a la descripción y a los ingredientes del plato, son obligatorios.

Un plato puede ser creado desde el botón **Añadir** situado en la parte inferior derecha de la ventana. En este momento los campos de texto se habilitarán para poder escribir en ellos. Obligatoriamente debe escribirse un nombre para añadir el plato a la base de datos, en caso contrario la aplicación mostrará un mensaje informativo en una ventana emergente advirtiendo al usuario de que no es posible la creación de un plato sin nombre. La aplicación permite, sin embargo, crear un plato sin ningún ingrediente y almacenarlo de esta forma en la base de datos, el usuario en un futuro podría añadir ingredientes a dicho plato vacío sin problemas.

Como se ha comentado anteriormente, todos los campos de texto a excepción del campo **Descripción** son obligatorios. El tipo de plato por defecto es primer plato, y los campos de texto numéricos serán rellenados a cero también por defecto. Todos los campos de texto, excepto el campo **Nombre**, **Descripción** y **Cantidad en gramos** de un ingrediente, permanecerán deshabilitados y serán rellenados con valores numéricos dependiendo de los ingredientes añadidos al plato y su cantidad en gramos. De esta forma, la cantidad en gramos del plato será la suma de gramos de los ingredientes añadidos. El precio del plato corresponderá con la suma de los precios de cada uno de los ingredientes y los valores nutricionales del plato corresponderán a la suma de valores nutricionales de cada ingrediente en función de la cantidad especificada. Además, dependiendo de los ingredientes añadidos al plato, la aplicación calculará automáticamente los posibles alérgenos o incompatibilidades alimenticias que pueda contener, mostrándose de forma gráfica mediante círculos de colores representativos.

Después de haberle puesto nombre a un nuevo plato, pulsando el botón **Guardar**, a la izquierda de los botones **Añadir**, **Modificar** y **Eliminar**, se añadirá el

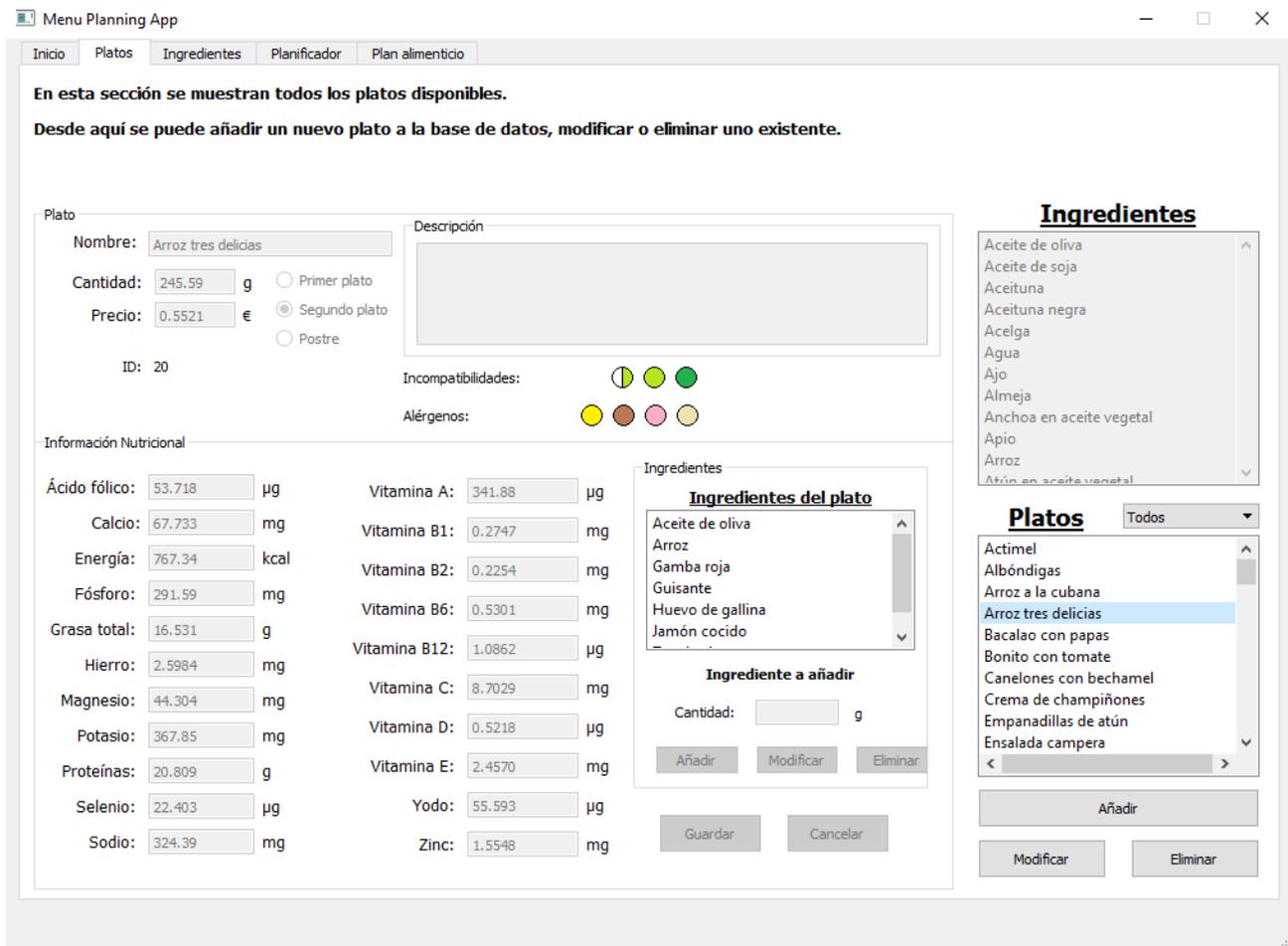


Figura 3.4: App - Ventana de platos

plato a la base de datos y aparecerá en el listado referente a platos en la parte derecha de la aplicación, desde donde se pueden visualizar todos los platos e incluso filtrar su visualización por tipo de plato. Cabe destacar que no se permitirá crear dos o más platos con el mismo nombre. Una vez guardado el plato la aplicación preguntará al usuario mediante una ventana emergente si desea añadir ingredientes al plato creado. En este momento, si el usuario acepta, el listado correspondiente a los ingredientes disponibles se habilitará y pulsando sobre uno de ellos, introduciendo su cantidad en gramos en el campo de texto correspondiente y pulsando el botón **Añadir** de la sección de ingredientes se guardará dicho alimento como ingrediente del plato. En caso de no especificarse una cantidad se rellenará automáticamente a cero.

Una vez creado un plato se puede modificar seleccionando dicho plato en el listado y pulsando el botón **Modificar** en la parte inferior derecha de la ventana. En este momento se habilitarán los campos de texto pertinentes permitiendo la modificación del mismo, de forma similar a la creación de un nuevo plato. Los posibles ingredientes añadidos al plato también puede ser modificados en tres aspectos: añadir un nuevo ingrediente al plato, modificar la cantidad en gramos de un ingrediente existente o eliminar un ingrediente del plato. Para

realizar estas acciones basta con seleccionar el ingrediente deseado de la lista de ingredientes del plato y utilizar los botones habilitados en dicha sección. Evidentemente, cualquier cambio realizado en los ingredientes se verá reflejado automáticamente en las características nutricionales, de cantidad, precio del plato, alérgenos e incompatibilidades alimenticias.

Para eliminar un plato de la base de datos basta con seleccionarlo en el listado correspondiente y pulsar el botón **Eliminar** situado en la parte inferior derecha de la ventana. En este momento la aplicación informará al usuario mediante una ventana emergente de que se procederá a la eliminación de un plato. Si el usuario acepta esta advertencia el plato será eliminado de la base de datos permanentemente.

En el desarrollo de esta aplicación, todos los datos referentes tanto a los ingredientes como a los platos han sido obtenidos de las referencias de información dietética y nutricional citadas en el primer capítulo de esta memoria. Al obtener la información de diferentes fuentes la introducción de los datos en la aplicación ha sido de forma manual.

3.3.4. Planificador

En esta ventana el usuario podrá realizar la acción principal de este trabajo, la creación de un plan alimenticio para el centro escolar. Desde aquí se podrá personalizar los principales datos a tener en cuenta para la creación del plan (ver Figura 3.5).

En primer lugar los platos que se desea que sean considerados para el menú. Es posible que, aunque se hayan agregado determinados platos en la aplicación éstos, por algún motivo (falta de medios en el comedor escolar para cocinarlo, falta de alimentos, etc), no se quieran tener en cuenta a la hora de crear un plan alimenticio. Para ello el usuario simplemente tiene que seleccionar los platos que desee considerar agregándolos a la tabla de platos seleccionados. Cabe destacar que para que sea posible la generación de un plan alimenticio es obligatorio seleccionar al menos un plato de cada tipo, es decir, al menos un primer plato, un segundo plato y un postre. De lo contrario la aplicación no permitirá la creación de un plan alimenticio.

En segundo lugar el usuario podrá seleccionar mediante el calendario situado en la parte superior derecha de la ventana la fecha de inicio y fin que tendrá el plan alimenticio. Una vez seleccionadas ambas fechas, la aplicación calculará el número total de días correspondientes. Al tratarse de una aplicación dirigida al comedor de un centro escolar, los fines de semana existentes entre la fecha de inicio y fin no serán considerados, por lo que no se generará un menú para

The screenshot shows the 'Planificador' window of the 'Menu Planning App'. It is divided into several sections:

- Platos:** A list of available dishes on the left and a list of 'Platos seleccionados' on the right. The selected dishes include Bonito con tomate, Crema de champiñones, Ensalada campera, Ensalada de garbanzos, Ensalada de pollo con mayo, Espaguetis con tomate, Filete de pollo a la plancha, Flan, Huevos rellenos, Judías blancas estofadas, Lentejas, Leche frita, Manzana, and Mixtura de frutas.
- Duración del plan:** A calendar for June 2016. The selected dates are the 2nd, 3rd, 4th, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, 14th, 15th, 16th, 17th, 18th, 19th, 20th, 21st, 22nd, 23rd, 24th, 25th, 26th, and 27th.
- Ingestas recomendadas de nutrientes:** A table of recommended nutrient intakes for the lunch period (40% of daily intake).

Nutriente	Cantidad	Unidad
Ácido fólico	135	µg
Calcio	585	mg
Energía	1012	kcal
Fósforo	562.5	mg
Grasa total	31.72	g
Hierro	8.55	mg
Magnesio	112.5	mg
Potasio	2025	mg
Proteínas	27.08	g
Selenio	25.75	µg
Sodio	870	mg
Vitamina A	450	µg
Vitamina B1	0.41	mg
Vitamina B2	0.63	mg
Vitamina B6	0.54	mg
Vitamina B12	2.28	µg
Vitamina C	27	mg
Vitamina D	4.65	µg
Vitamina E	6.3	mg
Yodo	67.5	µg
Zinc	6.75	mg
- Restricciones alimenticias:** A section for food allergies and incompatibilities. Under 'Alérgenos', 'Marisco' is checked. Under 'Incompatibilidades', 'Cereales', 'Frutos secos', 'Legumbres', 'Pescado', 'Proteína de huevo', and 'Proteína de leche de vaca' are all unchecked.

Figura 3.5: App - Ventana de planificador

esos días. Es importante la fecha del año considerada para realizar planes alimenticios ya que de esta forma se podrá detectar los alimentos de temporada correspondientes a dicha fecha, lo que implica un aumento de calidad en los menús.

En tercer lugar en la parte inferior de la ventana se puede observar las cantidades referentes a la ingesta de nutrientes recomendada en el almuerzo para niños en periodo escolar. La cantidad recomendada en el almuerzo corresponde a aproximadamente al cuarenta por ciento de la cantidad diaria recomendada. Estos datos pueden estar sujetos a modificación en la vida real y por ello también se permite la modificación de los mismos desde la aplicación. Es importante que el usuario esté de acuerdo con estas cantidades, ya que las restricciones del problema, referentes a la cantidad recomendada de nutrientes para la evaluación de los menús, y por ende, la decisión de si un menú es adecuado nutricionalmente o no, depende de estas cifras aquí establecidas.

En cuarto lugar aparece en la parte inferior derecha y agrupado en dos pequeñas ventanas, un listado con los posibles alérgenos e incompatibilidades alimenticias a evitar en la generación del menú. Los *checkbox* que sean seleccionados harán que el plan alimenticio carezca de platos en los que estén presentes

las alergias e incompatibilidades correspondientes. Es importante aclarar que siempre se generará un plan alimenticio en el que no se tenga en consideración la exclusión de ningún plato por alergia o incompatibilidad alimenticia. Si se selecciona alguna alergia o incompatibilidad a evitar, se generará en un post-procesado un único segundo plan alimenticio, idéntico al primero, salvo en aquellos platos en los que estén presentes los alérgenos o incompatibilidades a evitar. Se ha tomado esta decisión debido a que los casos en los que alguna persona padezca de alguna alergia o incompatibilidad alimenticia son generalmente casos aislados, y de cualquier manera su plan alimenticio no tendría por qué ser diferente en su totalidad al del resto de personas que no poseen estos padecimientos, salvo en aquellos platos en los que esté presente la alergia o incompatibilidad considerada.

Finalmente, una vez completados todos los requisitos para la creación de un plan alimenticio solo falta indicar a la aplicación que lo genere mediante el botón correspondiente situado en la parte inferior derecha.

3.3.5. Plan alimenticio

Desde esta ventana el usuario podrá visualizar los planes alimenticios recomendados por la aplicación según los parámetros anteriores elegidos por el usuario (ver Figura 3.6). La ventana se separa en dos mitades, en la mitad superior estarán los planes alimenticios recomendados sin restricción de alergias o incompatibilidades alimenticias. En la mitad inferior estarán los que sí poseen estas restricciones, siempre y cuando el usuario lo haya indicado en la ventana anterior de planificación. La visualización será mostrada en una tabla en la que cada fila corresponde al menú del día.

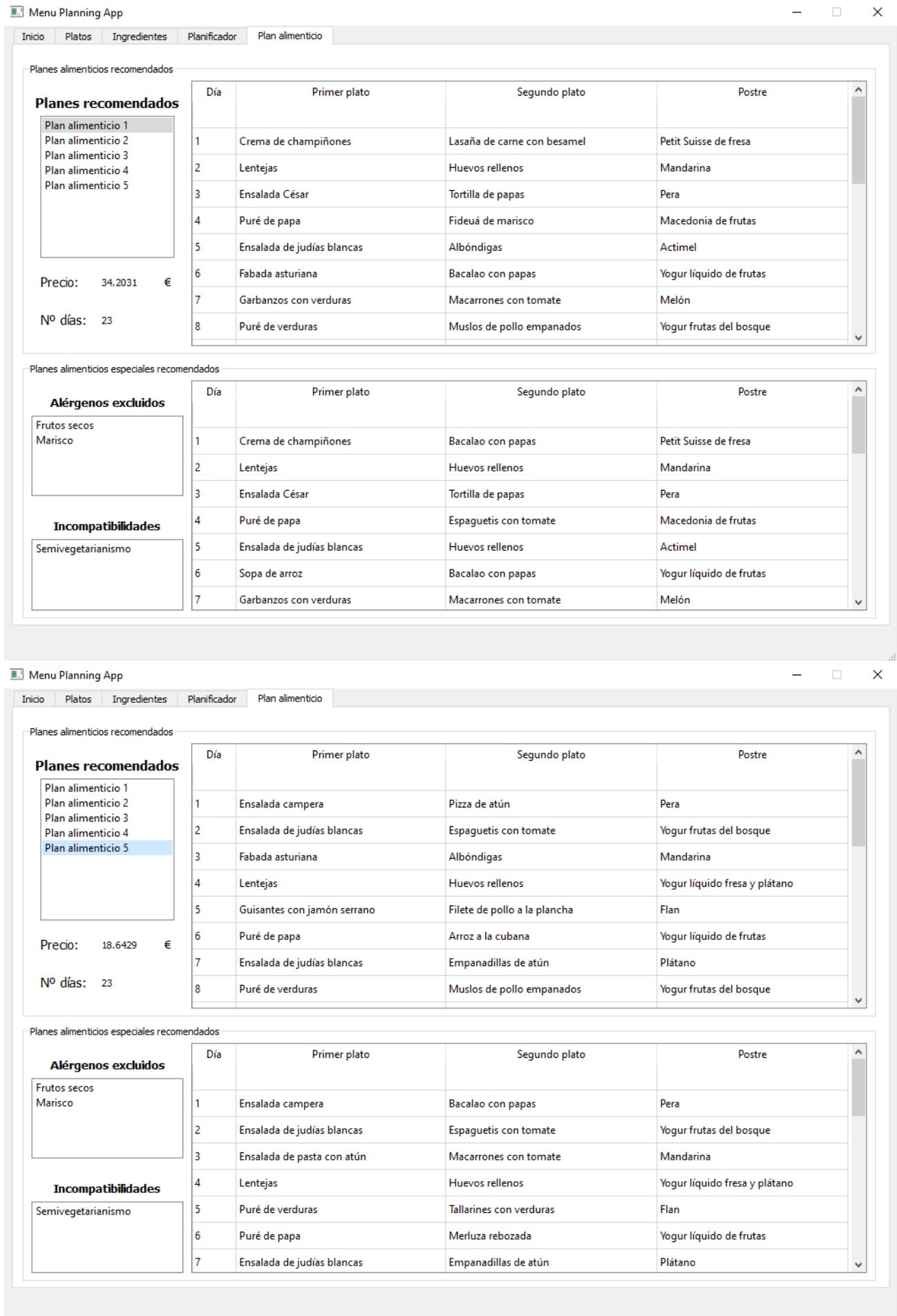


Figura 3.6: App - Ventana de plan alimenticio

Cabe destacar que al generarse un plan alimenticio, la aplicación trabaja con múltiples posibles soluciones, cada una de ellas es un plan alimenticio diferente. Al final de la ejecución del algoritmo se considerarán algunas de las mejores soluciones generadas, las no dominadas, y se mostrarán en esta ventana, con el objetivo de que el usuario pueda tener más de un plan alimenticio a elegir.

Capítulo 4

Resolución del problema

4.1. Optimización Multi-objetivo

El problema a tratar en este trabajo, el *Menu Planning*, es, como se ha comentado anteriormente, un problema de optimización multi-objetivo. Formalmente, se entiende por Problema de Optimización Multi-objetivo (MOP) aquellos problemas de la forma:

$$\text{maximizar o minimizar } F(x) = (f_1, f_2, \dots, f_n)$$

sujeto a m restricciones de desigualdad:

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, m$$

y p restricciones de igualdad:

$$h_j(x) = 0, \quad j = 1, 2, \dots, p$$

en donde n es el número de funciones objetivo y r el número de variables de decisión del problema contenido en un conjunto $x = x_1, x_2, \dots, x_r$.

En un problema de optimización multi-objetivo los objetivos del problema normalmente entran en conflicto unos con otros. Esto quiere decir que no se pueden minimizar o maximizar todos los objetivos de forma simultánea. Por este motivo en este tipo de problemas no se suele buscar una única solución, sino un conjunto que represente algunas de las mejores soluciones. El concepto de optimalidad es, por tanto, diferente al de otro tipo de problemas, como los problemas de optimización mono-objetivo. El término más comúnmente aceptado para estos problemas es el de *optimalidad de Pareto*.

Se dice que un vector de variables de decisión $x^* \in \mathcal{F}$ es un *óptimo de Pareto* si no existe otro $x \in \mathcal{F}$ tal que $f_i(x) \leq f_i(x^*)$ para todo $i = 1, \dots, k$ y $f_j < f_j(x^*)$ para al menos un j . Es decir, x^* es un óptimo de Pareto si no existe otro punto factible $x \in \mathcal{F}$ que mejore alguno de los objetivos sin causar un empeoramiento en alguno de los otros al mismo tiempo.

Esto, como se ha mencionado anteriormente, generalmente proporciona más de una solución. A este conjunto se le denomina *óptimos de Pareto* o soluciones no dominadas. La imagen formada por este conjunto se denomina *frente de Pareto*.

4.2. Algoritmos Evolutivos Multi-Objetivo

Los algoritmos evolutivos son una familia de técnicas metaheurísticas basadas en población e inspiradas en la evolución de las especies. Estas técnicas han sido utilizadas satisfactoriamente en numerosos problemas de gran complejidad en diferentes campos científicos y tecnológicos.

En la evolución natural, un entorno se encuentra habitado por una población limitada de individuos los cuales compiten por su supervivencia y reproducción. La selección natural, por tanto, es un mecanismo que evita el crecimiento descontrolado de las poblaciones y que favorece a aquellos individuos que mejor se adaptan al entorno que les rodea. Este fenómeno es también conocido como *la supervivencia del más apto*. Los supervivientes, por tanto, serán capaces de reproducirse con el objetivo de perpetuar la especie, dando origen a individuos igual o mejor adaptados al entorno.

Trasladando este concepto a la resolución de problemas de optimización, en este tipo de problemas tenemos funciones objetivo a optimizar sujetas a unas determinadas restricciones. En los algoritmos evolutivos se hace uso de funciones de aptitud, o *fitness*, para averiguar cuan buena es una solución con respecto a los objetivos del problema. La calidad de este valor de aptitud determinará si un individuo es apto para la supervivencia en la población y su reproducción.

El pseudocódigo general de estos algoritmos se muestra en el Algoritmo 1.

Algoritmo 1 Pseudocódigo genérico de un algoritmo evolutivo

- 1: **Inicialización.** Generación de la población inicial.
 - 2: **Evaluación.** Evaluar todos los individuos de la población inicial aplicando la función objetivo para obtener el valor de *fitness* de cada individuo.
 - 3: **mientras** (no se cumpla el criterio de parada) **hacer**
 - 4: **Selección de padres.** Selección de individuos de la población de padres para la construcción del *mating pool*, o conjunto de individuos más aptos.
 - 5: **Cruce y mutación.** Aplicar el operador de cruce entre los individuos del *mating pool* para generar la nueva población, denominada *offspring*, y utilizar en ella el operador de mutación con la intención de dotar al individuo de características que no estaban presentes en los padres.
 - 6: **Evaluación.** Evaluar los nuevos individuos para la asignación de su valor de *fitness*.
 - 7: **Selección de supervivientes.** Seleccionar los mejores individuos, entre padres e hijos, como supervivientes a la siguiente generación.
 - 8: **fin mientras**
-

Los algoritmos evolutivos de optimización multi-objetivo (MOEA) [10] son algoritmos evolutivos diseñados específicamente para dar solución a problemas de optimización mutli-objetivo. Son capaces de lidiar con un conjunto de soluciones y obtener una aproximación del frente de Pareto en una sola ejecución, incluso para diferentes formas del frente. A pesar de que obtener el frente de Pareto exacto así como soluciones óptimas de un problema de optimización multi-objetivo concreto es tarea difícil, estos algoritmos ofrecen buenas aproximaciones en tiempos de ejecución razonables.

Los objetivos principales de un MOEA son los siguientes:

- Preservar un conjunto de vectores no dominados en el espacio objetivo con sus correspondientes soluciones no dominadas en el espacio de decisión.
- Guiar el proceso de búsqueda a través del frente de Pareto en el espacio objetivo.
- Mantener la diversidad en el frente de Pareto (espacio objetivo), y/o las soluciones óptimas de Pareto (espacio de decisión).
- Proveer un conjunto de soluciones no dominadas, consideradas las mejores soluciones del problema.

Dentro de la familia de algoritmos evolutivos de optimización multi-objetivo se encuentra el NSGA-II (*Non-dominated Sorting Genetic Algorithm*) [13], el algoritmo que será utilizado en este trabajo para la resolución del problema.

El NSGA-II es uno de los MOEA más utilizados. Una sus principales características es que utiliza un método denominado *fast non-dominates sorting* el cual reduce la complejidad computacional del mismo además del carácter elitista en el operador de selección.

También cuenta con un operador denominado *crowded comparison* utilizado para ordenar los individuos. Para ello se asignan dos atributos a cada uno de los individuos: el *non-dominated rank* y el valor de *crowding distance* local.

El *non-dominated rank* es un valor que determina a cual de los frentes pertenece una solución dentro del espacio de soluciones factibles. El valor de *crowding distance* de un individuo es un valor de estimación de densidad de soluciones alrededor de dicho individuo.

Teniendo en cuenta estos conceptos, el operador de comparación \prec_c actúa de la siguiente manera:

$$i \prec_c j \text{ si } \begin{cases} (rank_i < rank_j) \\ 0 \\ ((rank_i = rank_j) \text{ y } (distance_i > distance_j)) \end{cases} \quad (4.1)$$

El pseudocódigo del NSGA-II se muestra en el Algoritmo 2.

Algoritmo 2 Pseudocódigo del NSGA-II

- 1: **Inicialización.** Generación de forma aleatoria de la población inicial P_0 con N individuos. Asignar $t = 0$.
 - 2: **Evaluación.** Evaluar todos los individuos de la población inicial calculando su función objetivo.
 - 3: **mientras** (no se cumpla el criterio de parada) **hacer**
 - 4: **Asignación de *fitness*.** Calcular el valor de *fitness* para los individuos en P_t . Hacer uso del *non-domination rank* en la primera generación, y el operador *crowded comparison* en las generaciones restantes.
 - 5: **Selección de padres.** Realizar una selección basada en un torneo binario determinista con remplazamiento en P_t con el objetivo de rellenar el *mating pool* de N padres.
 - 6: **Variación.** Aplicar los operadores de cruce y mutación, con probabilidad p_c y p_m respectivamente, a los individuos del *mating pool* para crear la nueva población CP con $M = N$ nuevos individuos.
 - 7: **Evaluación.** Evaluar cada nuevo individuo en CP mediante el cálculo de la función objetivo.
 - 8: **Selección de supervivientes.** Seleccionar los N individuos más aptos del conjunto de N padres y M hijos usando el operador *crowded comparison*, de manera que se constituye P_{t+1} .
 - 9: $t = t + 1$
 - 10: **fin mientras**
-

Otro algoritmo evolutivo considerado que también podría ser utilizado en este trabajo para la resolución del problema es el SPEA2 (*Strength Pareto Evolutionary Algorithm*) [42], el cual también ha demostrado ofrecer buenos resultados y sería ideal para comparar en obtención de resultados con el NSGA-II.

4.3. Algoritmo implementado

La creación de un plan alimenticio consiste en la generación de un número determinado de menús igual al número de días que el usuario seleccione en la aplicación. Un menú para un día estará compuesto por tres platos: primer plato, segundo plato y postre. Por ello es necesario que la aplicación contenga dichos platos en su base de datos. Los platos están compuestos por ingredientes, que a su vez están compuestos por una serie de datos que lo definen y caracterizan. Para almacenar todo esto, la base de datos está diseñada y organizada de la siguiente forma.

El algoritmo para la generación de un plan alimenticio, una vez ejecutado, se divide en seis procedimientos bien diferenciados.

Generación de un vector de platos y grupos alimenticios

En esta primera parte el algoritmo se encargará de detectar los platos que el usuario desea considerar para la generación del plan alimenticio y que debe haber especificado previamente en la sección del planificador. Una vez detectados dichos platos se crearán tres vectores, correspondientes a los tres tipos de platos (primer plato, segundo plato y postre), en los que, haciendo uso de consultas a la base de datos se almacenarán cada uno de los platos seleccionados por el usuario así como toda la información relativa a los mismos. También se almacenará en estos vectores otro dato, inicialmente asignado como infinito, referente al número de días desde que el plato fue seleccionado por última vez en el plan alimenticio. Este dato servirá posteriormente para el cálculo de uno de los objetivos del problema, el grado de repetición de los platos, y los vectores serán utilizados para saber qué platos de cada tipo pueden ser añadidos al plan alimenticio.

Un cuarto vector será construido para almacenar cada uno de los grupos alimenticios disponibles (carne, cereal, fruta, lácteo, legumbre, marisco, pasta, pescado, verdura y otro), así como otros dos datos numéricos que al igual que en los vectores anteriores servirán para hacer el cálculo del grado de repetición de los platos en el plan, estos son el número de días desde que se repitió un grupo alimenticio por última vez en el plan, y el número de veces que se ha repetido un grupo alimenticio el mismo día en el menú.

Generación de tabla de compatibilidad entre los platos

Para ofrecer una mayor calidad en la selección de menús para el plan alimenticio se ha decidido realizar una tabla en la que se determine el grado de compatibilidad de un plato con respecto a otro. En esta tabla están distribuidos todos los platos creados en la aplicación tanto en la primera fila como en la primera columna de la misma, de tal forma de que en las celdas restantes se asigna un valor numérico correspondiente al grado de compatibilidad entre dos platos. Cabe destacar que este grado de compatibilidad sólo se realizará entre los primeros y segundos platos, ya que los postres suelen ser compatibles con cualquier tipo de plato. Los valores numéricos en esta tabla pueden ser los siguientes:

- Se le dará un valor de 0 a una celda (i,j) si el plato i no es compatible con el plato j . Esto puede suceder si $i = j$, o bien si alguno de los platos es un postre, ya que como se comentó anteriormente, no se considerará la compatibilidad con estos platos. También tendrá un valor de 0 si i y j pertenecen al mismo tipo, es decir, ambos son primeros o segundos platos, ya que en un menú diario será imposible que se seleccionen dos platos

del mismo tipo.

- Se le dará un valor de 5 a una celda (i,j) si el plato i es poco compatible con el plato j . Esto sucederá si i comparte grupo/s alimenticio/s con j y los ingredientes de dicho grupo alimenticio son ingredientes principales. De esta forma se pretende evitar que se genere un menú en el que tanto en el primer plato como en el segundo haya, por ejemplo, carne como ingrediente principal. Favoreciendo así la variedad de los menús.
- Se le dará un valor de 10 a una celda (i,j) si el plato i es compatible con el plato j . Esto sucederá si entre ambos platos no se comparte ningún grupo alimenticio de ingredientes principales.

Cálculo del número de días del plan

El algoritmo comprobará las fechas de inicio y fin del plan seleccionados por el usuario y calculará el número de días correspondientes, que coincidirá con el número de menús a realizar. Como se ha comentado en una sección anterior, los fines de semana quedarán excluidos.

Cálculo de la ingesta recomendada de nutrientes

Se utilizarán los datos de ingesta diaria recomendada y el número de días para los que se debe realizar el plan para hacer un cálculo de la ingesta recomendada total de todo el plan.

Debido a que es muy poco probable que un plan alimenticio compuesto por gran cantidad de menús diarios cumpla de forma exacta con todos y cada uno de los requisitos nutricionales recomendados en la totalidad del plan, se proporcionará un margen en porcentaje, tanto por encima como por debajo de cada una de las cifras de nutrientes recomendadas. De esta forma se tiene una mayor flexibilidad a la hora de evaluar los menús por su cantidad de nutrientes.

Estos datos de ingesta de nutrientes recomendados compondrán las restricciones del problema.

Generación del plan alimenticio

Esta es la parte clave del trabajo, la generación del plan haciendo uso del algoritmo evolutivo NSGA-II así como los diferentes operadores de cruce, mutación, selección y evaluación de individuos.

En primer lugar se creará la primera población de individuos, para ello se hará uso de una clase llamada **Individuo** la cual contendrá toda la información necesaria de un plan alimenticio. Lo más importante a destacar, un vector cuyos elementos serán cada uno de los menús diarios compuestos por cada tipo de plato, una variable que recoge el precio total del plan, el que corresponderá al primer objetivo del problema, y una segunda variable que almacenará el grado de repetición de los platos en los diferentes menús, segundo objetivo del problema.

Esta primera población será generada de forma aleatoria, ya que no se dispone de una población anterior de la cual heredar. Se crearán tantos individuos que compondrán la población como se haya especificado en el algoritmo. Para la creación de un individuo se consultarán los vectores correspondientes a cada uno de los tipos de platos creados en la primera parte de este algoritmo y se seleccionarán platos de forma aleatoria hasta completar la totalidad de los menús requeridos.

Posteriormente se evaluará el plan generado, calculando el valor total de sus nutrientes, el precio, la cantidad de alérgenos e incompatibilidades alimenticias disponibles y el grado de repetición de los platos.

El valor del precio total, a minimizar, del plan alimenticio se calculará según la siguiente fórmula:

$$p_{total} = \sum_{i=1}^{n_D} p_{pp_i} + p_{sp_i} + p_{p_i}$$

En donde:

- p_{total} : precio total del plan alimenticio.
- n_D : número de días para los que está previsto realizar el plan.
- p_{pp_i} : precio del primer plato del día o menú i .
- p_{sp_i} : precio del segundo plato del día o menú i .
- p_{p_i} : precio del postre plato del día o menú i .

El valor del grado de repetición de los platos en el plan alimenticio, a minimizar, se calculará en base a la siguiente fórmula:

$$g_{Rep} = \sum_1^{n_D} \frac{1}{v_{Tabla} + v_{PP} + v_{SP} + v_P} + \frac{1}{v_{GA}} + v_{GA_d}$$

En donde:

- g_{Rep} : valor de repetición de platos en el plan alimenticio.
- n_D : número de días para los que está previsto realizar el plan.
- v_{Tabla} : valor de compatibilidad entre primeros y segundos platos recogidos en la tabla anteriormente mencionada.
- v_{PP} : número de días desde que se repitió el primer plato por última vez en el plan.
- v_{SP} : número de días desde que se repitió el segundo plato por última vez en el plan.
- v_P : número de días desde que se repitió el postre por última vez en el plan.
- v_{GA} : número de días desde que se repitió un grupo alimenticio por última vez en el plan.
- v_{GA_d} : número de veces que se ha repetido un grupo alimenticio el mismo día, en el mismo menú.

Estos pasos se repetirán en bucle hasta generar el número total de individuos que proceda. A partir de aquí los cálculos se harán sobre el vector que contiene a todos los individuos de la población.

A continuación se llamará a un método que comparará el valor total de los nutrientes con los valores de ingesta recomendados y se determinará, para cada uno de los individuos, si estos cumplen o no con los requisitos nutritivos. Sea cual sea el resultado se guardará en una variable de la clase de cada individuo.

Después de esto se pasará a ejecutarse el método ***fast non dominated sort***, propio del NSGA-II, en donde se calculará el rango de los individuos.

Posteriormente al cálculo del rango se procederá al cálculo de valor de *crowding distance*, también propio del NSGA-II. Este método requiere como parámetro un conjunto de soluciones no dominadas, por lo que se irán seleccionando las soluciones del problema con igual rango, en orden ascendente de rango, y se irán enviando al método de ***crowding distance*** hasta completar dicho cálculo en todas las soluciones contenidas en la población.

Estos cuatro últimos métodos, el cálculo total de los valores de cada individuo o solución, la comprobación del cumplimiento de las restricciones del problema,

el cálculo del rango y la *crowding distance* componen la fase de evaluación de las soluciones.

Llegados a este punto habrá finalizado la creación y evaluación de la población de la primera generación. A partir de aquí los pasos posteriores se repetirán en bucle hasta que se cumpla el número de generaciones especificado en el algoritmo.

Una contribución de este trabajo ha sido el método correspondiente a la fase de selección de los candidatos a la reproducción, el cual se ha adaptado a las características del problema, diferenciándolo así de la implementación original del NSGA-II. En este método se utilizará un vector de individuos denominado *mating pool* en el que se almacenarán los mejores padres. El procedimiento comienza con la división en dos vectores del total de la población. En uno de los vectores denominando **población_recomendada** se almacenarán los individuos que cumplen con las restricciones del problema, es decir, los que cumplen los requisitos nutricionales. En el segundo vector llamado **población_no_recomendada** se almacenarán aquellos que no cumplen estas restricciones. Posteriormente se ordenarán ambos vectores atendiendo al rango y al valor de *crowding distance* de cada uno de ellos. En primer lugar se ordenará de forma ascendente los individuos por el rango, y dentro de cada grupo de individuos con el mismo rango éstos serán ordenados descendientemente por el valor de *crowding distance*.

Una vez ordenados los vectores, el número de individuos que se seleccionará como candidatos a padres será de $N/2$, siendo N el tamaño de la población. La selección se hará, por tanto, eligiendo los $N/2$ primeros individuos existentes en el vector **población_recomendada**. Si el tamaño de este vector es inferior a $N/2$ se completará seleccionando individuos del vector **población_no_recomendada**. Estos individuos formarán parte del vector llamado *mating pool*.

Otro aspecto a tener en cuenta a la hora de seleccionar un individuo para su aceptación en el *mating pool* es el valor de *crowding distance*. El valor de *crowding distance* pretende reflejar el grado de dispersión en el que se encuentra un individuo con respecto a sus vecinos, de tal forma que cuanto mayor sea este valor menos soluciones habrá a su alrededor, o dicho de otra forma, más lejos se encuentran el resto de soluciones, por lo tanto es interesante que las soluciones escogidas tengan el mayor valor posible de *crowding distance*, ya que una solución con un valor bajo de este parámetro será una solución muy parecida a otra ya existente. Debido a esto, el método de selección excluirá aquellas soluciones que presenten un valor de *crowding distance* reducido, independientemente de su rango y posición en el vector ordenado. El límite aceptable de este valor será denominado como **minCrowDist**. Además, existe una debilidad en el diseño del método de *crowding distance* del NSGA-II advertido en las pruebas

realizadas y corregido mediante un procedimiento alternativo propuesto por F. Fortin *et al.* [19]. Este fallo, o debilidad en el diseño de este método permite que dos soluciones con idéntico valor de *fitness* puedan tener diferente valor de *crowding distance*. Además, si una solución posee el mismo valor de *fitness* que otra solución perteneciente a alguno de los extremos del frente de Pareto puede dar como resultado múltiples soluciones, más de dos, con asignación infinita de *crowding distance*.

Para evitar este problema se ha controlado que el operador de selección tampoco permita la elección de individuos cuyo valor de *fitness*, el correspondiente a los objetivos de precio y grado de repetición de platos, sea igual al de otro individuo ya introducido previamente en el vector de *mating pool*.

Posteriormente se dará paso al operador de reproducción o cruce. Para este proceso se ha decidido hacer un cruce por cada individuo i con otro individuo j escogido aleatoriamente atendiendo a una probabilidad de cruce determinada. De esta forma nos aseguramos de que todos los individuos puedan reproducirse al menos una vez, siempre y cuando la probabilidad de cruce les sea favorable. En caso de que una pareja de padres no pueda reproducirse debido a que la probabilidad de cruce asignada lo ha impedido, se crearán dos nuevos hijos exactamente iguales a los padres.

Los cruces entre individuos se realizarán partiendo el plan alimenticio de cada pareja en un punto intermedio del mismo, sin que este corte pueda producirse en los extremos, para posteriormente combinar dichas particiones y dar lugar a dos nuevos individuos. De esta forma se realizarán como máximo $N/2$ cruces, en caso de que la probabilidad de cruce sea favorable en todas las iteraciones anteriores. En cualquier caso se obtendrá un conjunto de N nuevos individuos.

Una vez obtenido el nuevo conjunto de individuos, llamados también hijos o conjunto *offspring*, se procederá al proceso de mutación. En este método, el algoritmo recorrerá el plan alimenticio de los nuevos individuos y para cada menú del día existirá una probabilidad definida en el método de que dicho menú cambie la totalidad de sus platos.

Una vez finalizado este procedimiento con todos los nuevos individuos se realizará con los hijos el mismo procedimiento de evaluación al que fueron sometidos los padres. Esto es, en primer lugar se calculará el valor total de todos los datos característicos del individuo así como el de los objetivos del problema, en segundo lugar se comprobará si el individuo cumple o no con las restricciones del problema, en tercer lugar se calculará el rango de cada uno de ellos y finalmente su valor de *crowding distance*.

Para finalizar se volverá a hacer una llamada al método de selección y se redefinirá el vector *mating pool* en el que se elegirán a los individuos supervivientes que pasarán a la siguiente generación, esta vez teniendo en consideración

tanto a los $N/2$ padres seleccionados para la reproducción como a los N hijos, haciendo una selección entre ellos de los N mejores candidatos.

Una vez el algoritmo haya terminado de ejecutar todas las generaciones programadas, del vector de individuos resultante se seleccionarán aquellos de menor rango, preferentemente los individuos no dominados, como candidatos a solución del problema. Estas soluciones se le presentarán al usuario en la aplicación (ver Figura 3.6) quedando a su elección la solución óptima.

Generación del plan alimenticio especial

El llamado plan alimenticio especial es aquel en el que a petición del usuario mediante la aplicación se requiere la exclusión de determinados platos que contengan ciertos alérgenos o incompatibilidades alimenticias. Este procedimiento sólo se realizará si el usuario así lo especifica en la aplicación. En ese caso se partirá de las soluciones finales generadas en el procedimiento anterior.

En primer lugar se crearán tres vectores correspondientes a los tres tipos de platos, exactamente iguales a los realizados en fases anteriores del algoritmo, en los que se almacenarán los platos seleccionados para ser incluidos en el plan alimenticio. La diferencia de estos tres vectores con los anteriores es que se excluirán aquellos platos en los que entre sus ingredientes estén presentes los alérgenos o incompatibilidades que el usuario desea excluir.

En segundo lugar se analizará el plan alimenticio de las soluciones seleccionadas en la fase final del procedimiento anterior y se sustituirán todos aquellos platos que presenten algún alérgeno o incompatibilidad a excluir por uno de los platos permitidos. De esta forma cada plan alimenticio calculado anteriormente y presentado al usuario en la aplicación como solución al problema contará con su versión libre de alérgenos e incompatibilidades especificadas por el usuario.

Finalmente estas soluciones serán debidamente presentadas junto con las demás en la ventana correspondiente de la aplicación (ver Figura 3.6).

Capítulo 5

Resultados computacionales

Una vez finalizado el diseño y la implementación del programa se procederá a la realización de un conjunto de pruebas para la evaluación de los resultados ofrecidos por el algoritmo y el rendimiento general del mismo.

5.1. Configuración de las pruebas

Una de las características del trabajo con algoritmos evolutivos es su particular sensibilidad a los resultados obtenidos debido a la experimentación con diferentes parámetros de entrada que se le administra. Incluso para los mismos parámetros de entrada, el carácter estocástico de este tipo de algoritmos puede ofrecer resultados con un rango variado en cuanto a la calidad de las soluciones.

Debido a esto se han realizado numerosas pruebas tanto con los mismos parámetros de entrada como con otros diferentes para comprobar la tendencia general de los resultados según el caso.

Los parámetros de entrada que se han utilizado son los siguientes:

- N = tamaño de la población.
- n_{Gen} = número de generaciones utilizadas.
- p_C = probabilidad de cruce.
- p_M = probabilidad de mutación.
- m_{CDist} = mínimo valor de *crowding distance* aceptado por el operador de selección para cada individuo.

Dato de instancia a destacar:

- n_D = número de días para los que se ha desarrollado el plan alimenticio. El número de días es proporcional al valor del grado de repetición de los alimentos, ya que afecta al número de menús que contendrá el plan alimenticio.

Dato de salida:

- t_e = tiempo de ejecución, en segundos, invertido por el algoritmo.

5.2. Resultados obtenidos

Para evaluar los resultados obtenidos por el algoritmo se ha decidido realizar un conjunto de pruebas separados en tres bloques que serán descritos a continuación. Es importante aclarar que las soluciones marcadas en color rojo representan soluciones no factibles, es decir, que no cumplen con las restricciones del problema. Los resultados obtenidos son los siguientes:

Elección de platos

Las siguientes pruebas reflejan el conjunto de soluciones general obtenido al generar un plan alimenticio prestando especial atención al número de platos seleccionado del total de platos disponibles. El número de platos disponibles en la aplicación actualmente según su tipo son los siguientes: primeros plato (19), segundos plato (34), postres (14).

Prueba	N	n_{Gen}	p_C	p_M	m_{CDist}	n_D	t_e
1.1	50	200	0.9	0.1	0.4	23	36.61
1.2	50	200	0.9	0.1	0.4	23	32.35

Tabla 5.1: Parámetros - Elección de platos

Como se puede observar en la Figura 5.1, cuando se hace una selección de platos suficientemente grande los resultados en cuanto a variedad son mucho más favorables. Tanto es así que un número muy bajo de platos seleccionados no solo ofrece unos valores de repetición de menús mucho más altos sino que puede no ser capaz ni siquiera de ofrecer soluciones factibles al problema, como puede observarse en los resultados de la segunda instancia.

De estos resultados se puede deducir la importancia de contar no solo con numerosos platos disponibles en la aplicación sino también de gran variedad de platos en cuanto a grupos alimenticios.

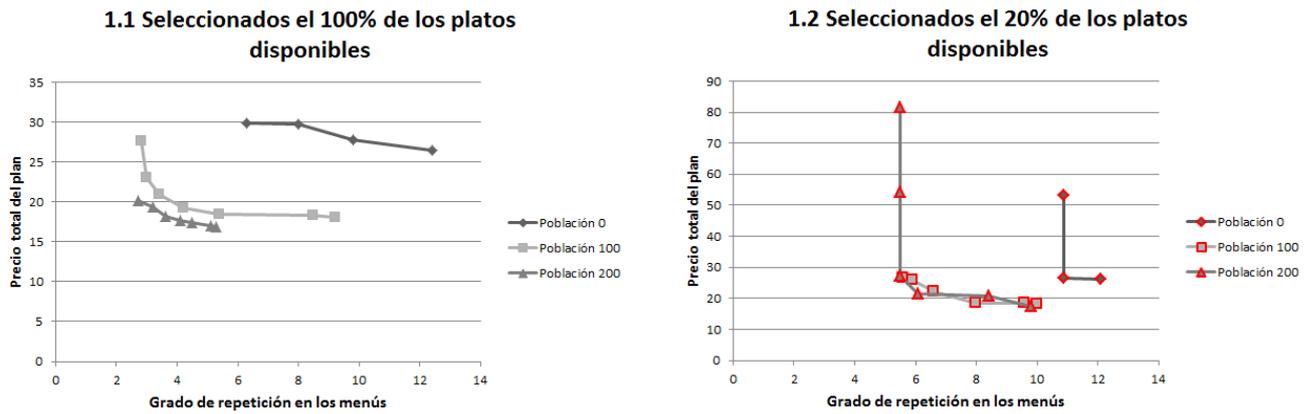


Figura 5.1: Gráficas - Elección de platos

Tamaño de población

Las siguientes pruebas pretenden mostrar el conjunto de soluciones general obtenido principalmente cuando se modifica el parámetro de tamaño de población.

Prueba	N	n_{Gen}	p_C	p_M	m_{CDist}	t_e
2.1	10	200	0.9	0.1	0.4	5.51
2.2	20	200	0.9	0.1	0.4	11.99
2.3	50	200	0.9	0.1	0.4	37.12
2.4	100	200	0.9	0.1	0.4	92.93

Tabla 5.2: Parámetros - Tamaño de población

La principal conclusión que se puede obtener de los resultados mostrados en la Figura 5.2, es que este algoritmo, para este problema suele generar mejores soluciones con tamaños medios de población, dentro del rango de las pruebas aquí realizadas. En este caso, con un valor de $N = 50$ se ha observado que se obtienen las mejores soluciones, no solo en los valores de los objetivos representados en las gráficas sino en los menús resultantes.

En tamaños grandes de población, desde $N = 100$ en adelante, ocurre que el operador de selección del algoritmo necesita encontrar una mayor cantidad de buenos individuos, desde el punto de vista de su valor de *crowding distance*. Esto es un problema en las primeras generaciones, en las que la cantidad de individuos de buena calidad es inferior, por lo que en muchas ocasiones el algoritmo no es capaz de encontrar los individuos necesarios. Para ello, se tendría que rebajar el nivel mínimo aceptable de *crowding distance* (m_{CDist}), con la consecuente selección de individuos de peor calidad candidatos posteriormente a la reproducción. Esta falta de buenos individuos puede resultar en la obtención de soluciones no factibles.

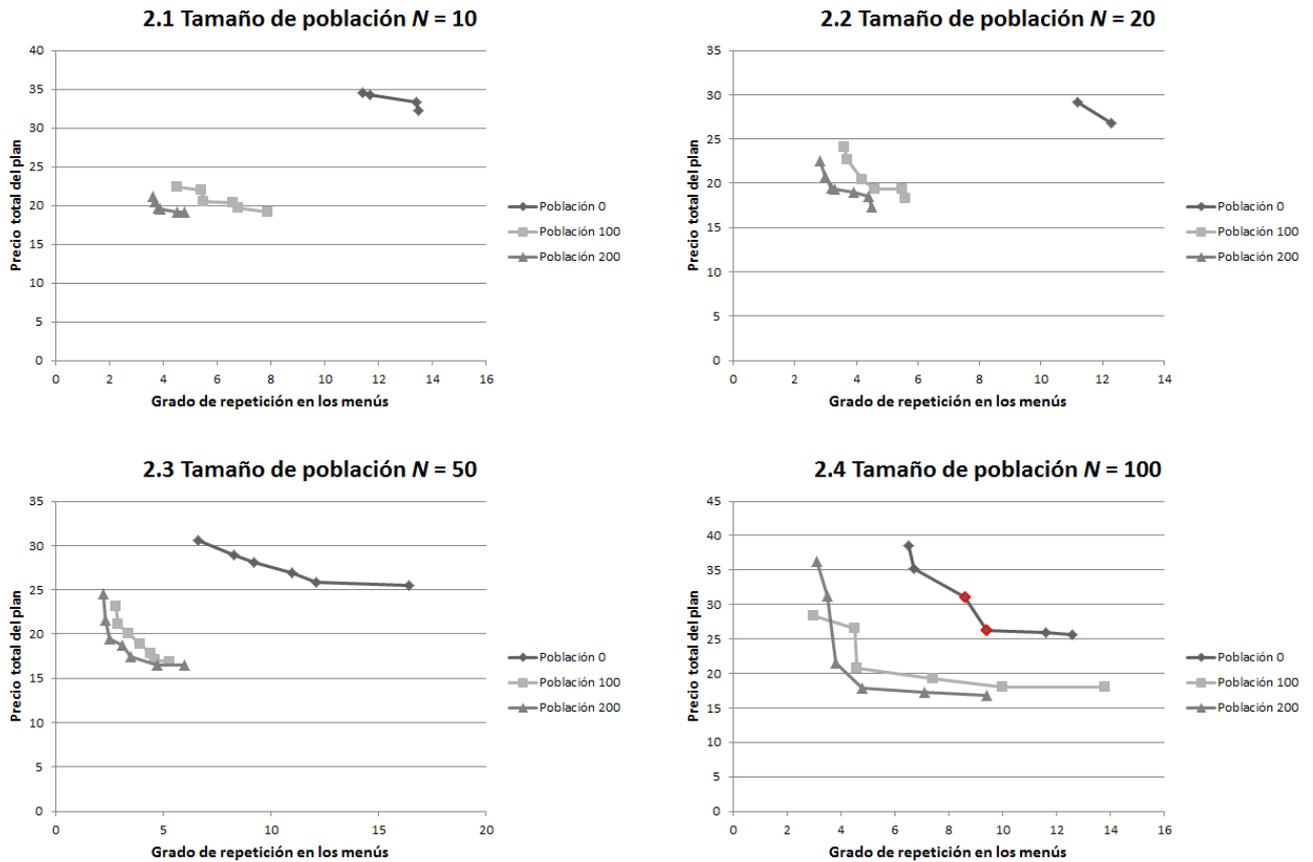


Figura 5.2: Gráficas - Tamaño de población

Probabilidad de cruce y mutación

En estas pruebas se evaluarán diferentes parámetros tanto de probabilidad de cruce como de mutación.

Prueba	N	n_{Gen}	p_C	p_M	m_{CDist}	t_e
3.1	50	200	0.9	0.2	0.4	37.34
3.2	50	200	0.8	0.1	0.4	36.83
3.3	50	200	0.8	0.2	0.4	38.72

Tabla 5.3: Parámetros - Probabilidad de cruce y mutación

En este caso podemos observar en la Figura 5.3, que el cambio entre una probabilidad de mutación del 10% y 20% no afecta significativamente a los resultados obtenidos. Sin embargo sí que se hace más notable el cambio entre una probabilidad de cruce del 80% y el 90%. En una probabilidad de cruce del 80% los resultados generales suelen ser peores, dando lugar a soluciones no factibles en generaciones avanzadas, así como menor número de soluciones en el frente de Pareto en determinadas generaciones con respecto a generaciones anteriores.

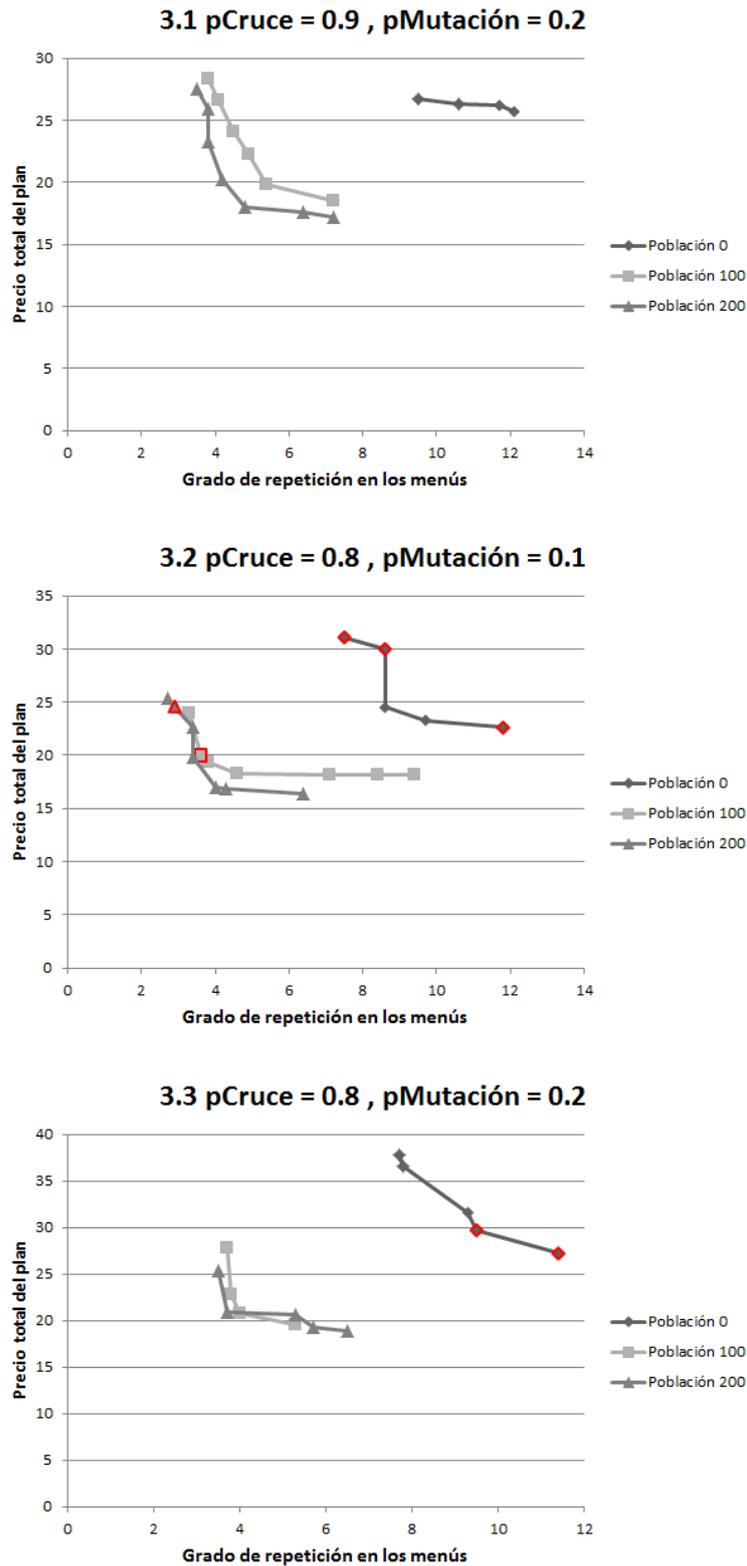


Figura 5.3: Gráficas - Probabilidad de cruce y mutación

Esta es una tendencia que se ha observado en numerosas realizaciones de pruebas con estos mismos parámetros, por lo que se puede deducir que es adecuado un mayor valor de probabilidad de cruce.

5.3. Conclusiones de las pruebas

Como resultado de estas y otras pruebas anteriormente realizadas, y no solo observando y evaluando los valores de los objetivos del problema en el frente de Pareto sino también cada uno de los menús generados correspondientes a cada solución, puede concluirse que los parámetros de entrada para el algoritmo que ofrecen mayor calidad de las soluciones son los siguientes:

- $N = 50$
- $n_{Gen} = 200$
- $p_C = 0.9$
- $p_M = 0.1$
- $m_{CDist} = 0.4$

Por este motivo éstos son los parámetros más utilizados en las pruebas realizadas anteriormente, y los que mejores resultados ofrecen.

Se ha querido comprobar si realmente estos parámetros ofrecen buenos resultados y estas conclusiones no obedecen a la naturaleza estocástica del algoritmo. Para ello se han realizado 50 ejecuciones del algoritmo con estos mismos parámetros y se ha recabado la siguiente información:

- El número medio de soluciones no dominadas por ejecución es de 6,36 soluciones no dominadas, 4 soluciones no dominadas en el peor caso y 8 soluciones no dominadas en el mejor caso.
- En cuanto a la tendencia de soluciones situadas en los extremos del frente de Pareto suelen ser más comunes aquellas en las que se tiene un precio muy bajo a costa de un grado de repetición de menús muy alto.
- En general todas las ejecuciones cuentan con al menos una o dos soluciones con un precio medio y alta variedad de menús y con un buen plan alimenticio.
- Los mejores planes alimenticios observados cuentan en la mayoría de sus días con menús variados y poca repetición de platos con respecto a los días anteriores y posteriores. No obstante, de entre todos los días que componen el plan alimenticio, suele haber alguno en el que se repite el mismo plato, bien sea primero, segundo plato o postre con respecto al día anterior, por lo que la solución no llega a ser óptima en su totalidad. Esto podría solucionarse mediante algún tipo de operador de reparación.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Existe un problema en la sociedad actual de gran importancia y que no hace distinción entre las diferentes poblaciones del mundo, ni siquiera en aquellas en los que el nivel de desarrollo es más elevado. Se trata de la malnutrición de la población, una nutrición deficiente o desequilibrada en las personas debida a una dieta pobre o excesiva en determinados nutrientes. El desarrollo y la evolución de las tecnologías de la información han supuesto y siguen siendo un tremendo avance en la evolución de la humanidad. Gracias a ello actualmente podemos hacer uso de estas tecnologías para dar solución, o al menos tratar de mejorar muchos de los problemas que posee la sociedad actual.

El problema de la malnutrición de la población, junto con la capacidad de la tecnología actual para interactuar con la sociedad y servir de apoyo a sus necesidades han impulsado la realización de este trabajo, consistente en el desarrollo de una aplicación capaz de realizar planes alimenticios compuestos por menús equilibrados desde el punto de vista nutricional. Esta aplicación está dedicada especialmente a realizar planes alimenticios en centros escolares para los niños, en donde el problema de la malnutrición es aún más importante. Es por ello que, aparte de conseguir menús de un nivel nutricional adecuado también se persiguen los objetivos de máxima variedad en los menús y precios reducidos, los cuales representan los objetivos del problema, teniendo en consideración, adicionalmente, los principales alérgenos e incompatibilidades alimenticias más comunes que puedan poseer determinadas personas, pretendiendo así obtener planes alimenticios de calidad y adaptables a personas con requerimientos nutricionales especiales.

El problema del que se está tratando es el llamado *Menu Planning*, o planificador de menús, un problema de optimización multi-objetivo para el que

se ha hecho uso de técnicas metaheurísticas para su resolución. Más concretamente se ha utilizado el algoritmo NSGA-II (*Non-dominated Sorting Algorithm II*), un algoritmo perteneciente a la familia de los algoritmos evolutivos que ha demostrado ofrecer buenos resultados a este tipo de problemas. También se ha realizado una revisión bibliográfica con algunas formulaciones del problema similares a la que se ha realizado.

Gracias a la aplicación realizada, el usuario de un centro escolar podrá añadir en la aplicación los platos que desee con sus correspondientes ingredientes y la información nutricional de los mismos. Posteriormente a esto, la aplicación se encargará de generar planes alimenticios según los parámetros escogidos por el usuario y atendiendo a los objetivos antes mencionados de mínimo coste y máxima variedad de los menús.

6.2. Trabajo futuro

Las líneas de trabajo futuro en esta aplicación están enfocadas a algunos aspectos que podrían llegar a mejorar la calidad de las soluciones, así como la posibilidad de ofrecer planes alimenticios más personalizados a los intereses del usuario. Las propuestas son las siguientes:

- **Mejora de los resultados:** a pesar de que la implementación actual del algoritmo ofrece buenos resultados, sería conveniente investigar posibles pequeñas modificaciones en el método de generación de planes, u otros parámetros de entrada para el algoritmo, con el objetivo de obtener soluciones aún mejores. También sería conveniente hacer estudios estadísticos para obtener mejores conclusiones sobre los resultados.
- **Modificación de plan alimenticio generado:** se podría dotar a la aplicación, una vez generado un plan alimenticio, de la posibilidad de modificar manualmente los menús establecidos por otros que prefiera el usuario.
- **Inclusión de otros alérgenos e incompatibilidades alimenticias a evitar:** aunque los alérgenos e incompatibilidades a evitar en un plan alimenticio que se han añadido en la aplicación son los más comunes, sería interesante añadir algunos más para dotar de mayores posibilidades a la creación de menús para personas con necesidades especiales.
- **Inclusión de un mayor número de platos:** una de los principales objetivos del problema es la obtención de menús variados. Añadiendo tantos platos de diferentes grupos alimenticios como sea posible, las probabilidades de obtener una mayor variedad en los planes alimenticios aumentan.

- **Automatización de los datos de ingredientes:** se podría hacer una herramienta que importara los datos referentes a la información nutricional, alérgenos, incompatibilidades alimenticias, grupo alimenticio, periodo de temporada y precio de un ingrediente de forma automática desde una fuente que contenga dichos datos, sin que el usuario tenga que añadir estos datos de forma manual.
- **Aplicación online:** también se podría realizar una versión online de la aplicación, de esta forma sería más cómoda y accesible al público.

Capítulo 7

Conclusions and future work

7.1. Conclusions

There is a very important problem in today's society, which makes no distinction between the different populations in the world, even in those where the level of development is higher. This is malnutrition of the population, poor or unbalanced nutrition in people due to poor diet or excessive in certain nutrients. The development and evolution of information technologies have resulted and continue to be a tremendous advance in the evolution of humanity. As a result we can now use these technologies to solve, or at least try to improve many of the problems that has today's society.

The problem of malnutrition of the population, along with the ability of current technology to interact with society and to support their needs have encouraged the realization of this work, consisting in the development of an application capable of performing meal plans with menu-balanced from a nutritional point of view. This application is specially dedicated to making meal plans in school centers for children, where the problem of malnutrition is even more important. That is why, apart from getting menus with an adequate nutritional level, also the goals of maximum variety in menus and discounted prices are pursued. This goals represent the objectives of the problem, taking into consideration, the major allergens and most common food incompatibilities that may possess certain persons. The aim is to obtain high quality meal plans and adaptable to people with special nutritional requirements.

The problem being treated is called *Menu Planning*, a problem of multi-objective optimization for which it made use of metaheuristic techniques for resolution. More specifically has been used the NSGA-II (*Non-dominated Sorting Algorithm II*) algorithm, an algorithm belonging to the family of evolutionary algorithms that have shown to provide good results to these sorts of problems.

It has also been performed a literature review with some formulations of the problem similar to that has been made.

Thanks to the application developed, the user can add in the application the dishes desired with their ingredients and nutritional information. Then, the application will generate meal plans according to the parameters chosen by the user and according to the above objectives of minimum cost and maximum variety of menus.

7.2. Future work

The future work lines in this application are focused on some aspects that could potentially improve the quality of the solutions and the ability to offer more customized solutions to the user's interests. The proposals are:

- **Improved results:** although the current implementation of the algorithm provides good results, it would be desirable to investigate possible minor changes in the method of plans generation, or other input parameters for the algorithm, in order to obtain even better solutions. It would also be desirable to make statistical studies to obtain better conclusions about the results.
- **Modification of generated meal plan:** the application could incorporate the possibility of manually modifying the plans generated by taking into account the user preferences.
- **Inclusion of other allergens and food incompatibilities to avoid:** although allergens and incompatibilities to avoid a meal plan that have been added in the application are the most common, it would be interesting to add some more to provide greater opportunities for creating menus for people with special needs.
- **Inclusion of a greater number of dishes:** one of the main objectives of the problem is obtaining a high variety of menus. By adding as many dishes of different food groups as possible, the odds of getting a greater variety in meal plans increase.
- **Data automation of ingredients:** it could make a tool that import data concerning of nutritional information, allergen, food incompatibilities, food group, period seasonal and price of an ingredient automatically from a source containing these data without the user having to add these data manually.

- **Online application:** it could also make an online version of the application, in this way would be more comfortable and accessible to the public.

Capítulo 8

Presupuesto

8.1. Presupuesto

La realización de este trabajo ha consistido en diferentes etapas las cuales a continuación serán nombradas y descritas brevemente. Cada una de estas etapas ha requerido de un número determinado de horas para su correcta realización, a las que se les asignará un presupuesto.

Las etapas de este trabajo han sido:

- **Etapas de selección y definición del problema:** la idea principal en este Trabajo de Fin de Grado era la resolución de un problema de optimización multi-objetivo mediante técnicas heurísticas y metaheurísticas. El problema a resolver no estaba definido en un principio, por lo que se tuvo que realizar una búsqueda de posibles problemas a tratar. Finalmente se decidió optar por el problema del planificador de menús.
- **Etapas de búsqueda de información y estado del arte:** una vez decidido el problema a tratar se realizó una extensa búsqueda sobre información relativa al problema, así como otras formulaciones del mismo realizadas por otros autores y aplicaciones que tratan el mismo problema.
- **Etapas de diseño y desarrollo de la aplicación:** se decidió utilizar el software Qt y el lenguaje C++ para la realización de la aplicación.
- **Etapas de diseño y desarrollo de la base de datos:** en este caso se decidió utilizar MySQL para almacenar el gran conjunto de datos requeridos para el correcto funcionamiento de la aplicación.
- **Etapas de implementación del algoritmo:** se definió, diseñó e implementó el algoritmo para la generación de planes alimenticios.

- **Etapa de realización de pruebas:** para evaluar el funcionamiento del algoritmo y la calidad de las soluciones ofrecidas se realizaron numerosas pruebas de testeo.
- **Etapa de realización de la memoria del trabajo:** se redactó la memoria de este Trabajo de Fin de Grado.

Estableciendo un precio total de 14 euros por hora de trabajo, el presupuesto final de este trabajo se resume en la siguiente tabla:

Etapas	Número de horas
1- Etapa de selección y definición del problema	20
2 - Etapa de búsqueda de información y estado del arte	60
3 - Etapa de diseño y desarrollo de la aplicación	180
4 - Etapa de diseño y desarrollo de la base de datos	12
5 - Etapa de implementación del algoritmo	40
6 - Etapa de realización de pruebas	20
7 - Etapa de realización de la memoria del trabajo	60
Número total de horas de trabajo	392

Presupuesto	5488 €
--------------------	---------------

Tabla 8.1: Tiempo dedicado y presupuesto

Bibliografía

- [1] Kale A. and Auti N. Automated menu planning algorithm for children: food recommendation by dietary management system using id3 for indian food database. volume 50, pages 197–202, 2015.
- [2] AEP. Libro blanco de la nutrición infantil. http://www.aeped.es/sites/default/files/documentos/libro_blanco_de_la_nutricion_infantil.pdf. [Online; accedida 5-Junio-2016].
- [3] AEPNAA. Alergia a los alimentos. <http://www.aepnaa.org/alergia/alergia-a-los-alimentos-40>. [Online; accedida 5-Junio-2016].
- [4] AESAN. La alimentación en los centros educativos. http://www.aecosan.msssi.gob.es/AECOSAN/docs/documentos/nutricion/educanaos/documento_consenso.pdf. [Online; accedida 5-Junio-2016].
- [5] Khan A.S. and Hoffmann A. Building a case-based diet recommendation system without a knowledge engineer. *Artificial Intelligence in Medicine*, 27(2):155 – 179, 2003.
- [6] Gaál B., Vassányi I., and Kozmann G. A novel artificial intelligence method for weekly dietary menu planning. *Methods of Information in Medicine*, 44(5):655–664, 2005.
- [7] Seijak B.K. Dietary menu planning using an evolutionary method. *Elektrotehnicki Vestnik/Electrotechnical Review*, 74(5):285–290, 2007.
- [8] Lee C-S., Wang M-H., and Hagraas H. A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. *Fuzzy Systems, IEEE Transactions on*, 18(2):374–395, April 2010.
- [9] Carrefour. Carrefour. <http://www.carrefour.es/>. [Online; accedida 5-Junio-2016].
- [10] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer US, 2007.

- [11] Osthus D. *A Genetic Algorithm Approach to Optimize Planning of Food Fortification*. PhD thesis, Department of Statistics, ISU, 2011.
- [12] Servicio Madrileño de Salud. Asesoramiento Nutricional del Menú Escolar. http://www.madrid.org/cs/Satellite?cid=1265098875455&pagename=PortalSalud/Page/PTSA_pintarContenidoFinal. [Online; accedida 5-Junio-2016].
- [13] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
- [14] Dietas.net. Tabla de composición nutricional por platos. <http://www.dietas.net/tablas-y-calculadoras/tabla-de-composicion-nutricional-por-platos/>. [Online; accedida 5-Junio-2016].
- [15] Kaldirim E. and Köse Z. *Application Of A Multi-Objective Genetic Algorithm To The Modified Diet Problem*. PhD thesis, Istanbul Technical University, Turkey, 2006.
- [16] Wang F., Yuan Y., Pan Y., and Hu B. Study on the principles of the intelligent diet arrangement system based on multi-agent. volume 1, pages 264–268, 2008.
- [17] FACE. La dieta sin gluten. <http://www.celiacos.org/la-dieta-sin-gluten/clasificacion-de-los-alimentos.html>. [Online; accedida 5-Junio-2016].
- [18] FD. Fundación para la Diabetes. <http://www.fundaciondiabetes.org/sabercomer>. [Online; accedida 5-Junio-2016].
- [19] Félix-Antoine Fortin and Marc Parizeau. Revisiting the nsga-ii crowding-distance computation. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13*, pages 623–630, New York, NY, USA, 2013. ACM.
- [20] Petot G.J., Marling C., and Sterling L. An artificial intelligence system for computer-assisted menu planning. *Journal of the American Dietetic Association*, 98(9):1009–1014, 1998.
- [21] Alfaro H.M. Menu planning using the exchange diet system. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 3, pages 3044–3049 vol.3, Oct 2003.

- [22] Aberg J. An evaluation of a meal planning system: Ease of use and perceived usefulness. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 278–287, Swinton, UK, UK, 2009. British Computer Society.
- [23] Bulka J., Izvorski A., Koleszynska J., Lis J., and Wochlik I. Automatic meal planning using artificial intelligence algorithms in computer aided diabetes therapy. pages 393–397, 2009.
- [24] Hsiao J-H. and Chang H. Smartdiet: A personal diet consultant for healthy meal planning. pages 421–425, 2010.
- [25] Makro. Makro. <http://www.makro.es/>. [Online; accedida 5-Junio-2016].
- [26] Mercalaspalmas. Mercalaspalmas. <http://www.mercalaspalmas.es/>. [Online; accedida 5-Junio-2016].
- [27] Mercatenerife. Mercatenerife. <http://mercatenerife.com/>. [Online; accedida 5-Junio-2016].
- [28] Funabiki N., Taniguchi S., Matsushima Y., and Nakanishi T. A proposal of a menu planning algorithm for two-phase cooking by busy persons. pages 668–673, 2011.
- [29] Jothi N., Husain W., and Damanhoori F. Myhealthcentral.com: Integrating knowledge based system in personalized diet plan and menu construction. 2011.
- [30] Chávez O.B., Marchi J., and Pozos P.P. Nutritional menu planning: A hybrid approach and preliminary tests. *Research in Computing Science*, 82:93–104, 2014.
- [31] OCU. Calendario de frutas y verduras. <http://www.ocu.org/alimentacion/alimentos/calculadora/calendario-de-frutas-y-verduras#>. [Online; accedida 5-Junio-2016].
- [32] Programa Perseo. Guía de comedores escolares. http://www.sennutricion.org/media/guia08_COMEDOR_ESCOLAR_txt.pdf. [Online; accedida 5-Junio-2016].
- [33] Gallardo P.R. and Pradenas L.R. *Modeling and solving with OR menus for healthy meals*. PhD thesis, Universidad de Concepción, Concepción, Chile, 2014.

- [34] Christian Pérez. Calendario de alimentos de temporada. <http://www.natursan.net/calendario-de-alimentos-de-temporada/>. [Online; accedida 5-Junio-2016].
- [35] Qt. Qt Documentation. <http://doc.qt.io/qt-5/>. [Online; accedida 5-Junio-2016].
- [36] AESAN RedBEDCA. BEDCA. <http://www.bedca.net/bdpub/>. [Online; accedida 5-Junio-2016].
- [37] Gumustekin S., Senel T., and Cengiz M.A. A comparative study on bayesian optimization algorithm for nutrition problem. *Journal of Food and Nutrition Research*, 2(12):952–958, 2014.
- [38] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 2000.
- [39] Kashima T., Matsumoto S., and Ishii H. Evaluation of menu planning capability based on multi-dimensional 0/1 knapsack problem of nutritional management system. *IAENG International Journal of Applied Mathematics*, 39(3), 2009.
- [40] UNED. Recomendaciones RDA. Cuadros y tablas. http://www2.uned.es/pea-nutricion-y-dietetica-I/guia/guia_nutricion/recomendaciones_rda.htm?ca=n0. [Online; accedida 5-Junio-2016].
- [41] Michael Widenius and Davis Axmark. *Mysql Reference Manual*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1st edition, 2002.
- [42] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimization and Control*, pages 19–26, Barcelona, Spain, 2002. CIMNE.