

Planificación de dietas con Answer Set Programming para pacientes con Diabetes Mellitus 2

Rosa Elena Pérez Ramírez¹, Claudia Zepeda Cortés¹
e Irma del Carmen Zamora Ginez²

¹Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
México

²Facultad de Medicina, Benemérita Universidad Autónoma de Puebla,
México

rosa.perez@solarium.cs.buap.mx, czepedac@gmail.com,
zamoraginezi@yahoo.es

Resumen La Diabetes Mellitus tipo II (DM2) es una enfermedad incurable y como parte del tratamiento de control los especialistas de la salud recomiendan una dieta personalizada, equilibrada y completa. La planificación de dietas requiere de procesar la información del paciente para obtener menús en los diferentes eventos alimenticios. En este trabajo se modela el problema de generar planes dietéticos para pacientes con DM2 basado en un enfoque declarativo como lo es *Answer Set Programming* (ASP).

Palabras clave: planificación, *Answer Set Programming*, representación del conocimiento, diabetes mellitus, planes dietéticos.

1. Introducción

La Inteligencia Artificial Médica se basa en la construcción de programas que ejecutan tipos de razonamiento como diagnóstico, planificación, toma de decisiones, etc. Algunas aplicaciones basadas en modelos simbólicos describen el problema con reglas y hechos para contestar preguntas sobre los objetos descritos y hallar soluciones a problemas particulares.

Por otro lado, tras el rezago de pacientes con DM2 con control metabólico positivo de la enfermedad, el Sistema de Salud Mexicano plasma un tratamiento para el control de la DM2 en la Norma Oficial Mexicana NOM-015-SSA2-2010 [1] con el apoyo del Sistema Mexicano Equivalentes de Alimentos (SMEA) [2].

La contribución de este trabajo es modelar el problema de agrupar un conjunto de alimentos que cumplan con ciertas restricciones para cada evento alimenticio (desayuno, comida, cena y dos colaciones), denominados en este trabajo planes dietéticos. La propuesta dirigida a pacientes con DM2 se basada en el SMEA y se desarrolla con un lenguaje de programación declarativo llamado ASP. ASP soluciona problemas de cálculo en términos de restricciones, excepciones y conocimiento incompleto y tiene aplicaciones tales como generar de planes de evacuación [3],

planificación de movimientos robóticos [4], planes de seguridad ante ataques [5], etc. La propuesta hace uso de DLV^k (implementación del paradigma de ASP) para modelar dicho problema.

El trabajo tiene 4 secciones. La sección 2 tiene una reseña de la operación de DLV^k en el modelado del problema de la planificación y bases de nutrición para generar planes dietéticos. En la sección 3 se desarrolla la propuesta: el modelo e implementación para generar planes dietéticos con DLV^k . Y en la sección 4 se presentan las conclusiones.

2. Marco teórico

Los fundamentos teóricos de este trabajo incluyen el funcionamiento de la herramienta DLV^k y el proceso detrás de generar planes dietéticos.

2.1. Planificación con DLV^k

DLV^k ¹ basado en la semántica estable; los programas en DLV^k se conforman del Conocimiento de Fondo Estático del Dominio (CFED) y la Descripción del Dominio (DD). La DD se compone de fluentes, acciones y reglas que procesan la información del CFED, con la cual generará un plan. La estructura general de DD es como en la figura 1.

```
fluents: <declaración de fluentes>
actions: <declaración de acciones>
always: <reglas>
initially: <inicialización de reglas>
[noConcurrency.]
goal: <consulta>?(i)
```

Figura 1. Estructura de la descripción del dominio.

Los fluentes representan las propiedades básicas del sistema que pueden cambiar en el tiempo, las acciones representan las operaciones que permiten una transición de estados; la ejecución de una acción se precondiciona en el bloque iniciado con *executable* y causa efectos indicados en el bloque iniciado con *caused*. El recorrido a través de los estados arranca desde un estado inicial y se plantea llegar al estado final, a través de un plan que corresponde a una secuencia de acciones.

2.2. Cálculo de necesidades energéticas en la dieta

La dieta se forma por alimentos que aportan cantidades adecuadas de nutrientes, distribuidos en 5 eventos alimenticios al día. Los macronutrientes aportan energía (en kilocalorías, kcal) por cada gramo (g) del modo siguiente: 1 g de proteínas aporta 4 kcal, 1 g de lípidos aporta 9 kcal y 1 g de carbohidratos aporta 4 kcal [6].

¹ <http://www.dlvsystem.com/dlv/>

Para realizar el cálculo de necesidades energéticas del paciente de inicio se registran sus datos antropométricos y clínicos: nombre, edad, género, compleción, peso, estatura, niveles de glucosa registrados ya sea, por la prueba de Hemoglobina Glucosilada (HbA1c) ó la Prueba de Glucosa en Ayuno (PGA), y el nivel de actividad física determinado por el número de pasos promedio que el paciente camina al día. Luego se calcula el Índice de Masa Corporal (IMC) con la fórmula (1).

$$IMC = \frac{\text{peso}}{\text{estatura}^2} \quad (1)$$

Inmediatamente de acuerdo a [1] se analiza si el paciente tiene problemas de peso y en seguida se analiza el control de la enfermedad analizando los niveles de glucosa X según el tipo de la prueba realizada: si es la prueba es GPA $X \leq 126\text{mg/dL}$ indican control, y si la prueba es HbA1c $X \leq 7\%$ indican control.

Seguidamente de acuerdo a [7] se obtiene el peso ideal del paciente $[p_{min}, p_{max}]$, y se calcula el total de kcal recomendadas por día $[GET_{min}, GET_{max}]$ también llamado Gasto Energético Total (GET) con la fórmula (2), para ello se calcula el Gasto Energético en Reposo (GER) con la fórmula de Harris –Benedict (3) y (4) según el género del paciente y el Factor de Actividad (FA) física de acuerdo a [6].

$$GET = GER * FA \quad (2)$$

$$GER_H = 66.5 + (13.7 * \text{peso}) + (5 * \text{estatura}) - (6.8 * \text{edad}) \quad (3)$$

$$GER_M = 665.1 + (9.5 * \text{peso}) + (1.7 * \text{estatura}) - (4.7 * \text{edad}) \quad (4)$$

Las kcal recomendadas por día se distribuyen en los eventos alimenticios en proporciones mostradas en la tabla 1 del modo siguiente: para cada evento alimenticio E_i se calcula el intervalo de kcal recomendadas $[C_{min}, C_{max}]_i$ en la proporción indicada por los límites porcentuales $[e_{min}, e_{max}]_i$, con las fórmulas (5) y (6).

$$C_{min} = \frac{e_{min} * GET_{min}}{100} \quad (5)$$

$$C_{max} = \frac{e_{max} * GET_{max}}{100} \quad (6)$$

Tabla 1. Distribución de kilocalorías por ingesta diaria en cada evento alimenticio

Índice del Evento	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Evento alimenticio E_i	Desayuno	Colación	Comida	Colación	Cena
Límites calóricos recomendados (%) $[e_{min}, e_{max}]_i$	25–30	10–15	25–35	10–15	15–20

Dado el estado de control de la enfermedad se establece la distribución de macronutrientes adecuada según la tabla 2 como sigue: se distribuye cada $[C_{min}, C_{max}]_i$ en cada macronutriente M_j se calcula el intervalo de kcal

recomendadas $[K_{min}, K_{max}]_{i,j}$ en la proporción indicada por los límites porcentuales $[m_{min}, m_{max}]_j$ con la fórmulas (7) y (8).

$$K_{min} = \frac{m_{min} * C_{min}}{100} \quad (7)$$

$$K_{max} = \frac{m_{max} * C_{max}}{100} \quad (8)$$

Tabla 2. Distribución de kilocalorías de macronutrientes en cada evento alimenticio

Índice de Macronutriente	Macronutrientes M_j	Límites recomendados para DM2 controlada (%) $[m_{min}, m_{max}]_j$	Límites recomendados para DM2 no controlada (%) $[m_{min}, m_{max}]_j$
$j=1$	Proteínas	13-15	15-20
$j=2$	Lípidos	25-35	35-45
$j=3$	Carbohidratos	50-60	40-50

Una vez obtenida una matriz de 5 x 3, donde cada elemento $[K_{min}, K_{max}]_{i,j}$ (límites de kcal recomendados para el evento E_i y el macronutriente M_j) en seguida se convierte a unidades en gramos $[g_{min}, g_{max}]_{i,j}$ con el uso de la fórmula (9) y la conversión de cada macronutriente (X) arriba especificados.

$$K_m/X = g_m \quad (9)$$

2.3. Uso del SMEA en los planes dietéticos

El SMEA contiene una base de datos de alimentos, donde cada alimento A_z tiene valores nutritivos y cantidades sugeridas, clasificados en 11 grupos denotado por S_k , cada elemento S_k con macronutrientes denotados por M'_{kj} , algunos se observan en la tabla 3. Un plan dietético denotado por P_i del evento alimenticio E_i se forma al agregar S_l elementos que en conjunto cumplen las restricciones $[g_{min}, g_{max}]_{i,j}$ y un máximo de repeticiones del elementos S_k en P_i denotado por n_k , luego se asocia cada elemento $S_l \in P_i$ a un elemento $A_z \in S_l$ de acuerdo a las preferencias del paciente.

Tabla 3. Extracto de grupos considerados por el SMEA

Elemento del SMEA S_k	Clasificación	Proteínas (g) M'_{k1}	Lípidos (g) M'_{k2}	Carbohidratos (g) M'_{k3}
S_1	Verduras	2	0	4
S_2	Frutas	0	0	15
S_3	Cereal sin grasa	2	0	15

Ejemplo. Sea un hombre adulto de 35 años, 70 kg, 170 cm, de complejión pequeña, 8% de la prueba HbA1c y sedentario. El análisis arroja que tiene IMC de 24.221, peso clasificado normal, su estado de DM2 es controlado y las necesidades

energéticas calóricas son $[GET_{min}, GET_{max}] = [2022, 2129]$. La distribución (g) en cada E_i y a su vez en cada M_j es como se muestra en la tabla 4.

Tabla 4. Resultado del cálculo de necesidades calóricas en un paciente con DM2

	Desayuno	Colación ₁	Comida	Colación ₂	Cena
Proteínas(g)	[15,28]	[6,12]	[15,28]	[6,12]	[9,20]
Lípidos(g)	[13,29]	[5,12]	[13,29]	[5,12]	[8,21]
Carbohidratos(g)	[70,112]	[28,48]	[70,112]	[28,48]	[42,80]

Un plan dietético P_2 para la primera colación podrían estar formado de 3 grupos S_i del modo siguiente: S_1 es 1 ración de fruta, S_2 es 1 ración de lácteo semidescremado, S_3 es 1 cereal con grasa, cumpliendo así con las restricciones de la tabla 4 para la primera colación. Luego se asocia cada S_i con los alimentos pertenecientes a dicho grupo y que sean de la preferencia del paciente, indicando las porciones recomendadas, como por ejemplo la siguiente asociación: 1 taza de fresa, 1 taza de yogur semidescremado y $\frac{3}{4}$ de barrita de granola.

En conclusión en la figura 2 se presenta el pseudocódigo que abstrae los pasos para generar planes dietéticos.

```

1. Inicio
2. Obtener_IMC()
3. Identificar_problemas_peso()
4. estado_control = Identificar_estado_control()
5. [pmin, pmax] = Buscar_peso_ideal()
6. [GETmin, GETmax] = Calcular_Getmax_y_Getmin ([pmin, pmax])
7. [mmin, mmax]j = Obtener_distribución_macronutrientes(estado_control)
8. Para cada evento alimenticio Ei hacer
9.     [Cmin, Cmax]i = Obtener_porcentaje ([emin, emax]i, [GETmin, GETmax])
10.    Para cada macronutriente Mj hacer
11.        [Kmin, Kmax]i,j = Obtener_porcentaje ([mmin, mmax]i, [Cmin, Cmax])
12.        [gmin, gmax]i,j = Convertir_kcal_gramos([Kmin, Kmax]i,j)
13.    Fin_Para
14.    Pi = Conjunto de Sk = Obtener_plan([gmin, gmax]i,1, [gmin, gmax]i,2, [gmin, gmax]i,3)
15.    Conjuntos de Az = Asociar_según_preferencias(Pi)
16.    Fin_Para
17. Fin

```

Figura 2. Estructura de la descripción del dominio

3. Implementación del problema de planificación de dieta

En esta sección se trata el modelado para generar planes dietéticos mediante el uso de DLV^k tras haber calculado las restricciones. Para ello, y por cuestiones de espacio, se muestra la implementación de un ejemplo sencillo con datos e información simplificada que nos ayude a comprenderlo fácilmente. De la figura 2 en la línea 14 se describe la rutina llamada `Obtener_plan` que recibe como parámetros las

restricciones $[g_{min}, g_{max}]_{i,j}$ para cada evento alimenticio E_i , obteniendo como resultado un plan P_i , que es ejecutada por el entorno DLV^k .

Para realizar esta implementación, se supone se tiene un plato vacío al cual agregaremos elementos S_k y al realizar esta acción, también se incrementa el número de S_k agregados al plato y las cantidades de macronutrientes M'_{k1} (propiedades de S_k), ejecutando esta operación mientras se cumplen las restricciones de cada macronutriente $[g_{min}, g_{max}]_{i,j}$ y el máximo de posibles repeticiones del elemento S_k denotado por n_k , hasta tener un plato completo.

El CFED describe los elementos S_k que pueden conformar un plan dietético P_i para E_i , las restricciones $[g_{min}, g_{max}]_{i,j}$ y el valor máximo de veces que puede repetirse S_k en P_i como se muestra en la tabla 5.

Tabla 5. Elementos del Conocimiento del Fondo Estático del Dominio (CFED)

Descripción del elemento	Formato de la descripción	Ejemplo
Grupo S_k y valores respectivos de macronutrientes .	$grupo(nombre_del_S_k,$ $n\acute{u}m_gramos_prote\acute{e}inas_del_S_k,$ $n\acute{u}m_gramos_l\acute{a}pidos_del_S_k,$ $n\acute{u}m_gramos_carbohidratos_del_S_k).$	De la tabla 3, elemento S_2 . $grupo(fruta, 0, 0, 5).$
Límites de cada macronutriente para cada E_i	$x_macronutriente_l\acute{m}(l\acute{m}ite_infe_rior, l\acute{m}ite_superior).$	De la tabla 2, los límites de las proteínas. $proteinas_lim(3, 6).$
Máximo de posibles repeticiones del elemento S_k en plan P_i	$rac_max(nombre_S_k,$ $l\acute{m}ite_superior).$	En un plan P_i como máximo se pueden agregar 5 verduras. $rac_max(verdura, 5).$

3.1. Descripción del dominio (DD)

Este modelo considera los fluentes siguientes: la cantidad de cada macronutriente contenida en el plato, el número de cada S_k contenidos en el plato y el estado que describe si el plato está completo. Además la acción considerada es agregar un S_k al plato, instanciando los parámetros del elemento S_k , codificado como sigue:

```
actions:
  agregar(A, P, L, CH) requires grupo(A, P, L, CH).
```

Los efectos al ejecutar la acción `agregar` son el incremento de cada S_k que se agregan al plato y el incremento de los valores de cada macronutriente del S_k que se agrega. En el bloque de abajo se muestra el conteo de cada S_k que se agrega al plato tras ejecutar la acción `agregar`, proceso análogo para cada macronutriente.

```

always:
  caused enPlato(A,C_new,M)
  if C_new = C_old+1
  after agregar(A,P,L,CH), enPlato(A,C_old,M).
  caused -enPlato(A,C,M)
  after enPlato(A,C,M), agregar(A,P,L,H),
  rac_max(A,M), C <= M.

```

Un plato está completo si se han cumplido las restricciones de los macronutrientes $[g_{min}, g_{max}]_{i,j}$, instanciando variables en el bloque *after* y verificando los límites de cada variable en el bloque *caused*.

```

caused platoCompleto if TP = P+P1, TP >= MinP, TP <= MaxP,
TCH = CH+CH1, TCH <= MaxCH, TCH >= MinCH, TLip = Lip+Lip1,
TLip >= MinLip, TLip <= MaxLip

after agregar(A,P1,Lip1,CH1),
  proteinas_lim(MinP,MaxP), proteinas_enPlato(P),
  lipidos_lim(MinLip,MaxLip), lipidos_enPlato(Lip),
  chidratos_lim(MinCH, MaxCH), chidratos_enPlato(CH).

```

La acción *agregar* se ejecuta mientras no se cumpla que el plato está completo y se cumplen las restricciones de raciones de cada S_k

```

executable agregar(A,P,L,CH)
if not platoCompleto, enPlato(A,R,M), R <= M.

```

El estado inicial describe las cantidades nulas de cada macronutriente presentes en el plato y la cantidad nula de S_k contenido en el plato. Además hace el uso de la negación como falla indicando que no tenemos evidencia de si el plato este completo hecho que mientras no sea verdadero se ejecuta la acción *agregar* como se muestra a continuación:

-platoCompleto.

3.2. Ejecución de la implementación

Se realiza la ejecución del código mediante el siguiente comando:

```
C:/dlv.exe genera_plan.plan evento_i.dl -FP -N=17
```

Obteniendo resultados en la tabla 6, donde tras una acción se cambia de estado, siendo el plan el conjunto de acciones realizadas para obtener un estado final.

Tabla 6. Ejecución de la implementación con DLV^k

Estado 0 ó estado inicial	Estado 1 Acción: agregar(lact_sem,5,4,4)
-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 0, 4) enPlato(lact_sem, 0, 1) chidratos_enPlato(0) lipidos_enPlato(0) proteinas_enPlato(0)	-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 0, 4) enPlato(lact_sem, 1, 1) -enPlato(lact_sem, 0, 1) chidratos_enPlato(4) lipidos_enPlato(4) proteinas_enPlato(5) -chidratos_enPlato(0) -lipidos_enPlato(0) -proteinas_enPlato(0)
Estado 2 Acción: agregar(azucar_conG,0,5,3)	Estado 3 Acción: agregar(cereal_sinG,1,0,5)
-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 1, 4) enPlato(lact_sem, 1, 1) -enPlato(azucar_conG, 0, 4) chidratos_enPlato(7) lipidos_enPlato(9) proteinas_enPlato(5) -chidratos_enPlato(4) -lipidos_enPlato(4)	-platoCompleto, enPlato(fruta, 0, 5) enPlato(cereal_sinG, 1, 6) enPlato(azucar_conG, 1, 4) enPlato(lact_sem, 1, 1) -enPlato(cereal_sinG, 0, 6) chidratos_enPlato(12) lipidos_enPlato(9) proteinas_enPlato(6), -chidratos_enPlato(7) -proteinas_enPlato(5)

Logrando un plan del modo siguiente:

```
agregar(lact_sem, 5, 4, 4);  

agregar(azucar_conG, 0, 5, 3);  

agregar(cereal_sinG, 1, 0, 5);
```

Indicando agregar un elemento de los siguientes grupos: S_1 es 1 lácteo semidescremado, S_2 es 1 ración azúcar con grasa, S_3 es cereal sin grasa que se asocia a alimentos dadas las preferencias del paciente.

4. Conclusiones y trabajo a futuro

El propósito de este trabajo es realizar una aplicación con fundamentos de la Inteligencia Artificial y Nutrición, por medio de la planificación con un lenguaje declarativo para obtener planes dietéticos para pacientes con DM2 que requieren un control y distribución adecuada de macronutrientes en la ingesta calórica. Dada la clasificación de grupos alimenticios del SMEA, este trabajo presenta como haciendo

uso de dicha clasificación se plante el problema de forma discreta y se resuelve con una implementación mediante descripción de reglas y hechos con DLV^k , obteniendo buenos resultados, además de ser una codificación relativamente fácil. Sin embargo para lograr una dieta se sugiere automatizar el pre-procesamiento de la información y analizar el estudio de preferencias del paciente, para presentar al paciente menús específicos de alimentos y no solo grupos, prototipo de un sistema en el cual actualmente se trabaja.

Referencias

1. Secretaría de Salud, Norma Oficial Mexicana NOM-015-SSA2-2010, Diario Oficial de la Federación. Noviembre 23 de 2010, Versión online: http://dof.gob.mx/nota_detalle.php?codigo=5168074&fecha=23/11/2010
2. Ana B. Pérez L., Berenice Palacio G., Ana L. Castro Becerra, Sistema Mexicano Equivalente de Alimentos, Fomento de nutrición y salud (2001)
3. Claudia Zepeda C. Evacuation Planning using Answer Set Programming. Tesis de Doctorado. Universidad de las Américas Puebla, 2005. Online <http://theses.insa-lyon.fr/publication/2005ISAL0117/thesis.pdf> (última fecha de verificación abril 2013).
4. Julián J. Portillo, Carmen L. García M., Pedro R. Márquez G. y Rogelio Baray A. Robot Platform Motion Planning using Answer Set Programming, Instituto Tecnológico de Chihuahua, México.
5. Chitta Baral, Michael Gelfond, Richard Scherl, Answer set programming as the basis for a Homeland Security QASk.
6. Daniel L. Román, Diego Bellido G., Pedro P. García L., Dietoterapia, nutrición clínica y metabolismo, 109 p. (2010)
7. José Mataix V., Nutrición y Alimentación Humana, Océano Ergon (2002)