



Lenguajes, Compiladores e Intérpretes CE-3104

Tarea 2 Lenguajes

NutriTec

Estudiantes

Juan Peña Rostrán 2018080231

Ignacio Morales Chang 2020096499

Anthony Montero Román 2019275097

Profesor

Marco Rivera Meneses

3 de noviembre 2021

## **Tabla de contenidos**

<b>Manual de usuario</b>	<b>3</b>
<b>Descripción de los hechos y reglas implementadas</b>	<b>3</b>
<b>Descripción de las estructuras de datos desarrolladas</b>	<b>6</b>
<b>Descripción detallada de los algoritmos desarrollados</b>	<b>7</b>
<b>Problemas sin Solución</b>	<b>8</b>
<b>Actividades realizadas por estudiante</b>	<b>8</b>
<b>Problemas solucionados</b>	<b>9</b>
<b>Conclusiones</b>	<b>11</b>
<b>Recomendaciones</b>	<b>12</b>
<b>Bibliografía</b>	<b>12</b>
<b>Repositorio del proyecto</b>	<b>12</b>
<b>Anexos</b>	<b>13</b>

## Manual de usuario

Para el desarrollo del programa se utiliza el lenguaje de programación Prolog en su versión 8.4.0, por lo que se recomienda tener este lenguaje y preferiblemente usar esta misma versión para la ejecución del mismo.

Para el funcionamiento del programa, se debe ejecutar el archivo con nombre “guiNutriTec” en la carpeta Lógica del repositorio de gitHub. Por haber ejecutado el programa, posteriormente se debe escribir en la consola de Prolog el comando “play.” el cual iniciará la principal función del programa, para que de este modo pueda interactuar con NutriTec. Ya por haber iniciado el programa y ejecutado la función principal, solamente debe saludar a NutriTec, esperar que este le conteste e ir respondiendo en consola cada una de las preguntas que el programa le efectúe.

Cabe recalcar que cada oración que se escriba en la consola debe de terminar con un punto y no pueden haber palabras en mayúscula, o llevar caracteres como comillas, comas o puntos con la excepción del que termina la oración.

## Descripción de los hechos y reglas implementadas

### GUI

**leer\_frase**(Palabras)

Ejemplo: leer\_frase().

|: yo quiero comer saludable.

Palabras = [yo, quiero, comer, saludable]

El objetivo de esta función es ser llamada por la función principal al momento de querer escribir una oración en consola y leer la misma para separarla en una lista de palabras.

La función recibe la entrada 'Palabras' la cual es la petición de la lista de palabras de la oración escrita.

## **BNF**

**oracion2**(S0, S, Clave)

Ejemplo: oracion2([yo,quiero,una,dieta,keto],[],keto) .

True.

“oracion2” recibe una oración descompuesta en una lista de las palabras, una lista vacía y la palabra clave de la oración. Después con estos parámetros se revisa qué tipo de estructura tiene la oración y verificar si es una estructura válida.

## **Base de datos nutricional:**

**tiposdietas**(Codigo, Tipo):

Tabla de tipos de dietas en la base de datos.

Codigo: codigo de la dieta

Tipo: tipo de la dieta.

Ejemplo: tiposdietas(d1, keto).

**nombredieta**(Codigo, Nombre):

Tabla de nombres de dietas en la base de datos.

Codigo: codigo de la dieta

Nombre: Nombre de la dieta.

Ejemplo: nombredieta(s1,'dieta baja en carbohidratos').

**padecimientos**(Codigo, Nombre, Causa, Recomendacion):

Tabla de padecimientos en la base de datos.

Codigo: codigo del padecimiento

Nombre: Nombre de padecimientos.

Causa: string con breve causa del padecimiento.

Recomendación: codigo del nombre de una dieta recomendada.

Ejemplo:

padecimientos(p1, dislipemia, 'problemas de control de colesterol', s1).

**dietas**(codigo de dieta, codigo nombre dieta, lista padecimientos no recomendados, lista padecimientos recomendados, actividades no recomendadas, actividades recomendadas).

Tabla de datos para la unión de las columnas de datos nutricionales anteriormente comentados.

Ejemplo:

dietas(s1, d1, 1800, [p3,p4, p5], [p1, p2],[no hacer fondo'], ['trotar 15min. 3 días por semana'],'hacer 15min campo traviesa'], detalle1).

**consultanombredieta(Codigodieta, Nombre):**

Predicado para obtener el nombre de una dieta en la base de datos.

Codigodieta: es un parámetro asociado a la tabla de nombres de dietas en la base de datos, sus valores pueden ser s1, s2,..., s6.

Nombre: resultado de la búsqueda que hace match con el código

Ejemplo: consultanombredieta(s1, Nombre).

Nombre = 'dieta baja en carbohidratos'.

**consultapadecimiento(Nombre, Codigo):**

Predicado para obtener el codigo de un padecimiento en la base de datos.

Codigo: es un parametro asociado a la tabla de tipos de dietas en la base de datos, sus valores pueden ser p1, p2,..., p5.

Nombre: valor que pertenece al nombre de la enfermedad en la base de datos que coinciden con el codigo.

Ejemplo:

consultapadecimiento(dislipemia, Codigopade).

Codigopade = p1.

**validarDieta(Padecimiento, Calorias, Actividad, Tipodieta):**

Predicado que comprueba que los parametros de salud del cliente coincidan con la base de datos y le recomienda una dieta.

Padecimiento: enfermedad indicada por el usuario

Calorías: Número entero entre 1000 y 3000.

Actividad: Número entero mayor que 0.

Tipodieta: dieta preferida por el usuario.

Ejemplo:

validarDieta(Diabetes, 1200, 3, proteica)

Descripción de las estructuras de datos desarrolladas

**Listas:**

**miembroPadecimiento(Padecimiento, Lista):**

Predicado para conocer si un codigo de padecimiento está en la lista de padecimientos, ya sea recomendado o no recomendados.

Padecimiento: codigo de un padecimiento de la base de datos.

Lista: Lista de codigos de padecimientos.

La lista se recorre recursivamente comparando el padecimiento con la cabeza actual de la lista, una vez realizada la comparacion, se procede a cortar la lista y hacer la llamada recursiva con el resto de la lista y el mismo padecimiento, cuando la lista queda vacía se retorna el resultado true o false.

Ejemplo:

miembroPadecimiento(p1, [p4, p2, p3]):-

miembroPadecimiento(p1, [p4|[p2, p3]])

miembroPadecimiento(p1, [p2|[p3]])

miembroPadecimiento(p1, [p3|[ ]]).

miembroPadecimiento(p1, [ ]): false.

**escribirActividades(Lista):**

Predicado que recorre la lista y escribe los elementos de la misma, escribe la cabeza de la lista, luego corta la lista con el resto de datos y repite el proceso hasta que la lista esté vacía.

Lista: lista de actividades o padecimientos.

escribirActividades([ ]).

escribirActividades([Act|Actividades]):-write(Act),

nl, escribirActividades(Actividades).

## Descripción detallada de los algoritmos desarrollados

- Se crea un algoritmo para verificar si una oración está bien escrita estructuralmente y recuperar la palabra clave de esa oración. Esta tiene como nombre oracion2 y recibe como parámetros la oración mostrada como una lista de palabras (S0), una lista vacía (S) y la palabra clave de la oración

(Clave). En caso de querer averiguar la palabra clave de la oración, simplemente se ingresa una “X” como parámetro “Clave”.

Ejemplos de uso:

oracion2([yo,quiero,una,dieta,keto],[],X).

X = keto.

oracion2([yo,tengo,sobrepeso],[],X).

X = sobrepeso.

oracion2([no],[],X).

X = no.

oracion2([si,1200],[],X).

X = 1200.

## Problemas sin Solución

- Las oraciones escritas por el usuario deben ser escritas en minúscula y sin signos de puntuación en media oración.
- Las oraciones sobre preferencias en forma de negación del usuario no se implementaron, por ejemplo, Nutritec no podría reaccionar ante las respuestas a las preguntas: ¿cuál dieta NO prefiere consumir?, ¿cuál alimento prefiere NO consumir?

## Actividades realizadas por estudiante

Anthony Montero:

Investigar el funcionamiento de los Sistemas Expertos.

1.1 Inicio de una conversación

1.2. Preguntas

1.3. Fin de una conversación



## 1.4 Manejo de interfaz con el sistema: Lectura y escritura de información

Juan Peña

## 2.2 Definición de Base de datos que contiene las siguientes cuatro Tablas:

- a. TiposDietas
- b. Padecimientos: [[Dislipidemia “Problemas del control del colesterol” “Dieta baja en grasas”] [Hipercolesterolemia “aumento de los niveles considerados normales de colesterol en la sangre” “Dieta vegana”]...]
- c. Niveles de Actividad física: inicial(0-2 v/semana) intermedio(3-4), Avanzado(>=5)
- d. Objeto Dieta: [Nombre Tipo Calorías [padecimientos no recomendados] [padecimiento recomendado] [actividad no recomendadas] [actividades recomendadas] [Detalle Dieta]] (6 dietas diferentes)  
  
¿Qué es actividad recomendada?
- e. Reglas y hechos para realizar consultas a la base de datos.

Ignacio Morales

## 3.1 Hacer la gramática libre de contexto BNF para NutriTec

3.2 Realizar la revisión de las oraciones por Palabras clave, sintagmas nominales, verbales, entre otros.

## 3.3 Identificación de palabras clave en cada oración.

## Problemas solucionados

## Lectura de los datos escritos en la consola:

En un principio se tenía cierto problema con la lectura de las frases escritas en la consola, ya que al querer convertir la oración escrita en una lista

no era posible amenos de que fuera escrita entre comillas, sin embargo la oración al estar entre comillas era convertida en una lista de strings, lo que significaba que se producirían conflictos con los distintos hechos que habían como por ejemplo los números, ya que estos estarían igualmente en forma de string.

Se prueban distintas maneras de arreglar esto como por ejemplo, escribir todos los hechos entre comillas, sin embargo para el caso de los números esto no era una solución viable. Se prueban también distintas formas de leer lo escrito en consola, viendo distintos proyectos, videos y foros, sin embargo al hacer pruebas de todas estas alternativas, no se encuentra la solución requerida.

Este problema fue solucionado con ayuda de un documento en la web subido por la Universidad de Córdoba titulado “Programación Declarativa”, el cual en uno de sus temas explica la lectura y escritura de términos y frases y su transformación en átomos en el lenguaje de Prolog (Escuela Politécnica Superior de Córdoba, s.f).

### Creación de base de datos para el plan nutricional

Para conocer la opinión de nutricionistas se consultó la página web (Eenfeldt, 2021), donde existen una serie de planes nutricionales para dietas según las condiciones de una persona. Una vez que se conoció esta información se hizo la implementación de una base de datos relacional basada en el modelo presentado en (Escrig, Toledo y Pacheco, 2000).

### Extracción de datos según información brindada por usuario.

Se tuvo problemas a la hora de conectar lo que decía el usuario con el método para escoger una dieta. El problema radicaba en que no se guardaban

las palabras clave en cada oración para ser utilizadas después. Esto se soluciona agregando un tercer parámetro a “oracion2”, la cual sería la palabra clave “Clave”. Este guarda la palabra clave ya sea una negación, enfermedad, dieta, número de calorías o cantidad de actividad física para así poder escoger una dieta según lo que diga el usuario.

## Conclusiones

Al crear una aplicación que se comporta como sistema experto en nutrición se permite comprender la mecánica que hay detrás de la detección automatizada de información personal que ofrecen aplicaciones similares, donde gran parte de la solución a este problema se logra mediante la detección correcta de palabras clave que van dirigiendo al sistema experto hacia una respuesta adecuada con respecto a su conversación con el usuario.

Como parte de la creación de una base de datos relacional, fue necesario estructurar la información de forma que el paradigma lógico lograra ofrecer datos cuando fueran consultados, lo cual se logró gracias al manejo y creación de hechos y reglas bajo predicados con nombres significativos para el tipo de datos que se representaron.

Finalmente, las listas como estructura de datos primitiva para el paradigma lógico se presentaron como la mejor opción en el manejo de constantes y variables a lo largo del desarrollo de NutriTec, por ejemplo, el recorrido recursivo de la lista de palabras ingresadas por el usuario fue la forma en que la gramática libre de contexto identifica oraciones del lenguaje español, si tienen sentido o no.

## Recomendaciones

La eficiencia del sistema experto puede depender del alcance que tenga su base de conocimiento, por ello, una de las mejoras que NutriTec podría tener es ampliar las palabras del idioma español que sabe, de manera que sea sencillo ingresar sinónimos de palabras durante una conversación sin incurrir en un fallo del sistema.

En cuanto al poder de respuesta de NutriTec se recomienda la implementación de más preguntas de índole nutricional y su respectivo análisis sintáctico para que los planes nutricionales generados sean más específicos.

## Bibliografía

Eenfeldt, D., 2021. *Términos y condiciones - Diet Doctor*. [online] Diet Doctor. Recuperado de: <<https://www.dietdoctor.com/es/condiciones#copyright>> [Consultado 27 octubre, 2021].

Escrig, M., Pacheco, J., & Toledo, F. (2000). *El Lenguaje de Programación Prolog*.

Escuela Politécnica Superior de Córdoba. (s. f.). *Programación Declarativa*.

<http://www.uco.es/users/ma1fegan/2013-2014/pd/temas/Tema-12/PD-Tema-12.pdf>

## Repositorio del proyecto

<https://github.com/AnthonyHMR/ITCR.CE.Lenguajes.TareaII.NutriTec.git>

## Anexos

```
?-
% c:/users/hp/desktop/tec/viii semestre/lenguajes/itcr.ce.lenguajes.tarea11.nutritec/lógica/guinuec compiled 0.02 sec, 3 clauses
?- play.
|: hola nutritec.

Hola encantado de verlo mejorar su estilo de vida. Cuénteme ¿en qué lo puedo ayudar?
|: quiero comer saludable.

Excelente iniciativa. Estamos para asesorarte en todo lo que necesites.
¿Tienes alguna enfermedad por la que ha iniciado este proceso?, ¿cual?
|: si diabetes.

¿Tienes pensado una cantidad específica de calorías diarias por consumir?
|: yo quiero una dieta de 1550 calorías.

¿Cuántas veces a la semana hace actividad física?
|: yo corro 4 veces a la semana.

¿Tienes un tipo de dieta que te gustaría realizar?
|: quiero una dieta proteica.

¿Qué alimentos preferirías no consumir?
|: mariscos.

Para empezar le recomiendo la dieta: dieta alta en proteínas

Desayuno:
Yogur con semillas de linaza, acompañado con un rollito de queso y jamón de pavo.

Almuerzo:
Pollo a la plancha cortado en tiras con ensalada de pepino, lechuga y tomate aderezada con aceite de oliva y limón.

Merienda:
Huevo cocido con unos palitos de zanahoria.

Cena:
Salmón a la plancha con ensalada de brócoli, zanahoria y tomate aderezada con limón y aceite de linaza.

También le recomiendo hacer algunas de estas actividades:
Además le recomiendo hacer estas actividades físicas:
Practicar natación
Hacer Senderismo
|: ■
```

---

Figura 1. Ejemplo de uso del bot NutriTec.