**Dibujo en blanco y negro

Descripción generada automáticamente con confianza bajaINSTITUTO POLITÉCNICO NACIONAL**

ESCUELA SUPERIOR DE CÓMPUTO

**NOMBRES:**

PADILLA MATIAS CRISTIAN MICHEL

SAUCILLO GONZÁLEZ JESSE OBED

**GRUPO:**

4BM1

**TRABAJO:**

Practica 4

"Lenguaje de Predicados"

**MATERIA:**

Fundamentos de Inteligencia Artificial

**FECHA**

25 de octubre del 2023

Logotipo

Descripción generada automáticamente

**PRACTICA 2. Lenguaje de Predicados**

**RESUMEN**

En la práctica 5, se empleará el enfoque de programación lógica utilizando PROLOG para crear un sistema de predicados. Inicialmente, se desarrollará un código que presentará un menú interactivo para calcular el Índice de Masa Corporal (IMC) y ofrecer recomendaciones de dieta saludable. Para llevar a cabo estos cálculos, se utilizarán fórmulas específicas y valores de referencia.

Es esencial establecer una base de conocimientos para cada uno de los códigos, definiendo atributos, relaciones y reglas necesarias para su correcto funcionamiento.

**INTRODUCCIÓN**

La búsqueda no informada, a veces llamada búsqueda ciega o búsqueda a ciegas, es un enfoque de búsqueda en la inteligencia artificial donde el algoritmo no tiene información previa sobre la ubicación o la estructura del objetivo que está tratando de encontrar. La lógica proposicional proporciona los principios fundamentales para las expresiones booleanas en la programación, basándose en la lógica de predicados y reglas de inferencia. En el contexto de la programación lógica, se pueden formalizar hechos del mundo real. PROLOG se basa en la lógica de primer orden, una extensión de la lógica de predicados.

PROLOG, siendo el lenguaje principal en programación lógica, se sustenta en dos conceptos poderosos: resolución y unificación. La resolución implica hacer inferencias simples a partir de cláusulas, similar a la transitividad en álgebra, permitiendo llegar a conclusiones.

Por otro lado, la unificación es un proceso que encuentra coincidencias en patrones y determina cómo se pueden utilizar las variables durante la creación de resoluciones simultáneas. Este proceso es recurrente, encontrando todas las instanciaciones posibles para las cuales pueden hacerse resoluciones.

Los programas lógicos aprovechan el no determinismo y la búsqueda de retroceso en sus aplicaciones. Un programa no determinista puede tener múltiples soluciones para un problema en lugar de una única solución. El mecanismo de búsqueda de retroceso, que permite este no determinismo, está integrado internamente en el intérprete de PROLOG y está disponible para todos los programas escritos en PROLOG.

**DESARROLLO**

El código de la práctica se realizó en lenguaje Prolog.

Para este ejercicio, primero se crea un ciclo principal con la regla llamada main, con cual será posible mostrar el menú, y leer la opción que el usuario indique dentro del menú, cada una con su respectiva regla que serán 1.-Calcular índice de masa corporal (doImc) y 2.-Recomendar una dieta saludable (doDieta), además de la opción 3.- que es para salir del programa. Igualmente, cuando se termine de ejecutar la opción 1 o 2, mostrará de nuevo el menú, no lo mostrará más cuando el usuario seleccione la opción 3.

Texto, Carta

Descripción generada automáticamente

Posteriormente, en la regla llamada pinta\_menu, únicamente se mostrará en pantalla el menú, así el usuario podrá observar las indicaciones y las opciones a seleccionar.

Texto, Carta

Descripción generada automáticamente

En el caso de que el usuario seleccione la opción 1, se ejecutará la regla llamada doImc, dicha regla requiere datos del usuario, por lo que se usará la sentencia read para leer los valores ingresados, que son peso, estatura y sexo. A la variable IND se le asignará el resultado de dividir peso entre la estatura al cuadrado, para obtener el índice de masa corporal.

También, se mostrará un diagnóstico, el cual mostrará un mensaje de acuerdo con el IMC de la persona, que al mismo tiempo depende de su sexo. Para esto se utilizó la siguiente tabla.

Texto

Descripción generada automáticamente

En el caso de que el usuario seleccione la opción 2, se ejecutará la regla llamada doDieta

Texto, Carta

Descripción generada automáticamente

Se crea la base de conocimiento de diferentes alimentos que se encuentran dentro del plato del buen comer, cada cual puede ser consumido durante el desayuno, comida y cena de acuerdo a la aportación nutrimental necesaria a una hora específica, y cada alimento perteneciente a un grupo se catalogó con una etiqueta específica, donde se especifica el nombre del alimento y su aportación alimentaria medida en calorías.

Texto

Descripción generada automáticamenteTexto, Carta

Descripción generada automáticamente Texto

Descripción generada automáticamente

Esta base de conocimientos será necesaria para recetar una dieta saludable basándose en el peso y la edad.

Una vez creada la base de conocimientos, será necesaria escoger la opción 2 del menú principal descrita anteriormente, después de seleccionar esta opción, se preguntará al usuario su sexo, peso y edad, necesarios para calcular el consumo de calorías diarias.

Texto, Carta

Descripción generada automáticamente

Después de obtener los datos del paciente, el programa utilizará la función calcular\_GAST (Peso, Edad, Sexo, GAST). Donde evaluará las condiciones de peso, edad y sexo, utilizando la función donde los parámetros cumplan con los rangos de valores necesarios para calcular el gasto de calorías, utilizando las condiciones que:

Sexo:

• Si es hombre, el peso se calculará multiplicando el peso en Kg por 24.

• Si es mujer, el peso se calculará multiplicando el peso en Kg por 21.6.

Edad:

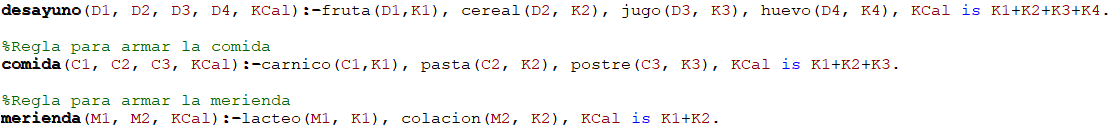
• Si tiene menos de 25 años, se añadirán 300 calorías al gasto de calorías.

• Si tiene entre 25 y 45 años, no se añade ni se quita.

• Si tiene más de 45 años, se restan 100 calorías por cada 10 años de más.

Después de evaluar los parámetros, la función devolverá el valor de gasto de calorías del paciente.

Finalmente, después de haber calculado el gasto de calorías del paciente, se evaluarán todas las posibles dietas que pueden ser recomendadas al paciente.



Utilizando 3 funciones, se evaluarán todos los posibles desayunos, comidas y cenas que puedan ser recomendadas, cada una teniendo un diferente alimento:

• Desayuno: Compuesto de una fruta, un tipo de cereal, un vaso de jugo y una preparación de huevo.

• Comida: Una ración de carne, una ración de pasta y un postre.

• Cena/merienda: un alimento o bebida láctea y una colación o postre ligero.

Cada función evaluará los alimentos disponibles en la base de conocimiento creada anteriormente, y tomará el valor calórico de cada uno para sumarlos y obtener la cantidad total de calorías que se consumirían en esos alimentos.

Texto, Carta

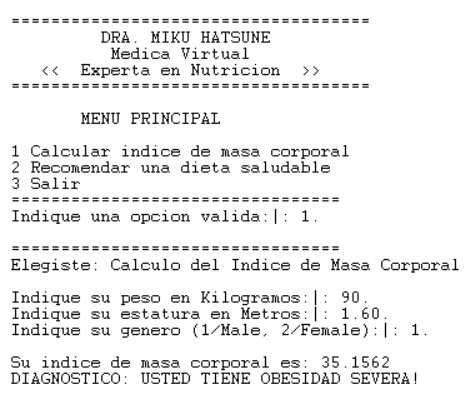
Descripción generada automáticamente

Finalmente, se utilizará la función dieta (Gasto) para evaluar las posibles combinaciones para satisfacer las necesidades calóricas del paciente; esta función es utilizada después de calcular el gasto de calorías, con lo que podrá recomendar una dieta saludable en función a los datos del paciente, para ello evalúa los posibles desayunos, comidas y cenas diarias, suma el total de calorías consumidas con las 3 comidas y las evalúa en función del gasto diario de calorías, con un margen aceptable de ±10% del consumo de calorías, logrando recomendar diferentes comidas diarias que satisfacen el consumo necesario de calorías diarias del paciente.

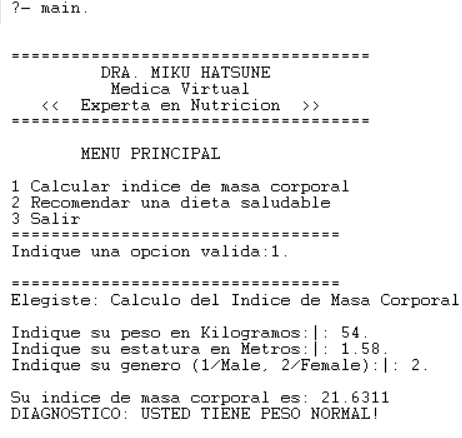
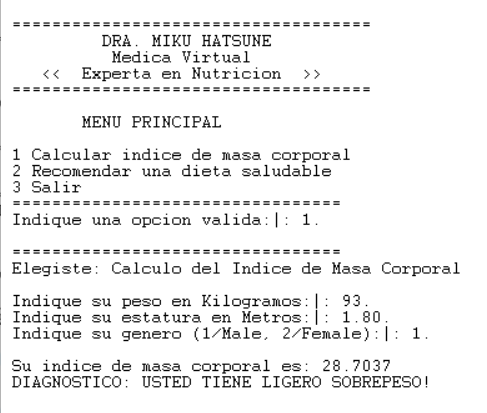
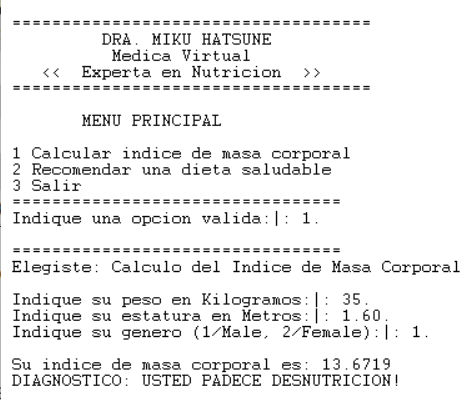
**DEMOSTRACIÓN DE FUNCIONAMIENTO**

Ejecutando el programa:

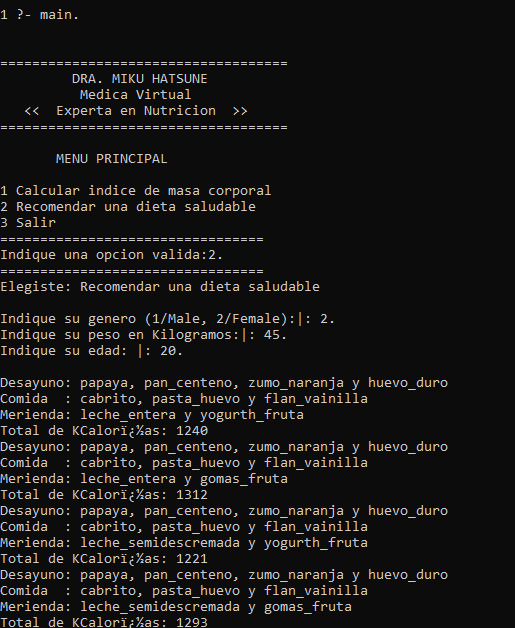
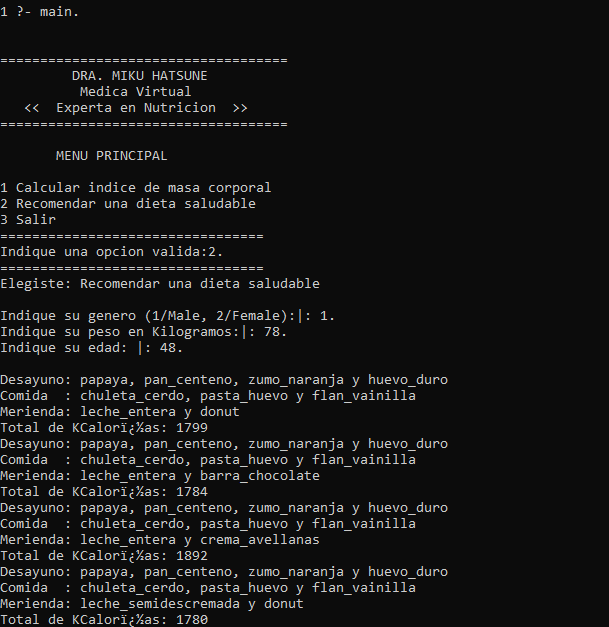
Opción 1:

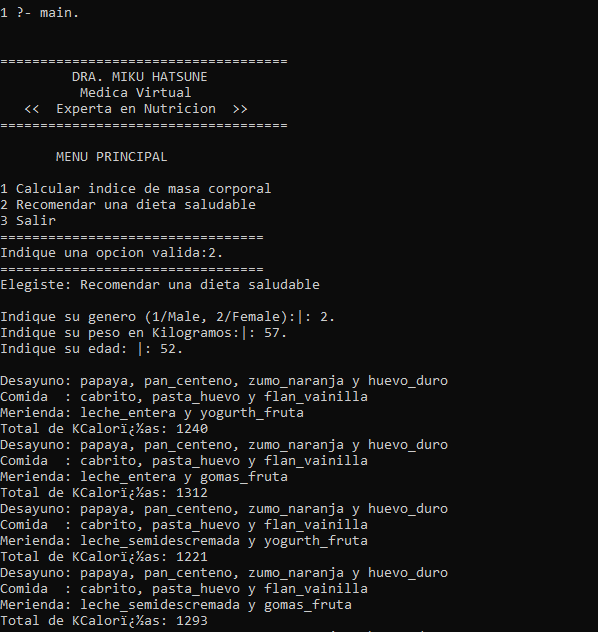
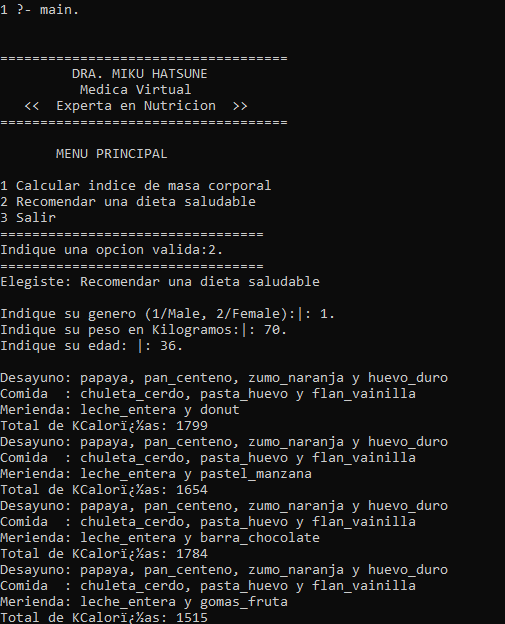


En estos cuatro ejemplos, se muestran casos en los que una mujer tiene un peso normal ya que (20 <=IMC<24), un hombre tiene ligero sobrepeso (25 <=IMC<30), otro hombre padece desnutrición (IMC<17) y un último hombre tiene obesidad severa (30 <=IMC<40).



Opción 2

Como se puede observar en estos 4 ejemplos, después de calcular el gasto de calorías, el programa evalúa cada comida posible que pueda satisfacer el consumo de calorías necesarias diarias para el paciente, como se había mencionado, con un margen de cambio de ±10% del valor de gasto de calorías, con lo que es posible crear miles de combinaciones con los alimentos de la base de conocimientos mencionada anteriormente, satisfaciendo las necesidades del paciente de una manera eficiente.

**CONCLUSIONES**  
PROLOG se emplea principalmente en campos como demostración de teoremas, sistemas expertos, reescritura de términos, sistemas de tipo y planificación automatizada. Sin embargo, en esta práctica, su aplicación principal se centró en el procesamiento del lenguaje natural.

Para llevar a cabo esta práctica, utilizamos conocimientos previamente adquiridos en el curso de Paradigmas de Programación. La implementación del árbol genealógico no presentó dificultades significativas, ya que previamente habíamos trabajado en una práctica similar en Paradigmas. Solo fue necesario realizar ajustes para incorporar nuevas reglas y atributos a las relaciones.

Es importante mencionar que durante esta práctica, también exploramos conceptos de funciones y ciclos, los cuales fueron aplicados en el segundo código desarrollado. La utilidad clave de este paradigma de programación radica en la capacidad para crear un menú, como el del segundo código, que permite a los usuarios seleccionar opciones, desencadenando así la ejecución de reglas y líneas de código específicas.**BIBLIOGRAFÍA**

Lukegarrigan. (2021, 5 Junio). *What is Breadth-First Search (BFS) - Codeheir*. Codeheir. https://codeheir.com/2021/06/05/what-is-breadth-first-search-bfs/

Alfonso López. (2020, 11 mayo). *Algoritmo primero el mejor (BFS) C++* [Vídeo]. YouTube. https://www.youtube.com/watch?v=MQ8ZKqE4S2c

‌