

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA SUGERENCIA DE
DIETAS BALANCEADAS ORIENTADAS A EQUILIBRAR Y CONTROLAR EL
PESO DE LAS PERSONAS**

Tesis para optar por el Título de **Ingeniero Informático**, que presenta el bachiller:

Jesus Humberto Cano Cauzo

ASESOR: Ing. Rony Cueva Moscoso

Lima, Septiembre del 2013

Resumen

Uno de los problemas que afecta la salud de las personas hoy en día es el sobrepeso, el cual es producto de una suma de factores, uno de los cuales es el mal hábito de las personas al momento de ingerir alimentos, esto trae consigo una serie de enfermedades las cuales son potencialmente mortales, de ahí es que nace la necesidad de mantener a las personas en su peso ideal y evitar de esta manera dichas enfermedades.

Con la finalidad de que se logre este objetivo es que se implementa este proyecto de fin de carrera, el cual está dotado de una base de conocimiento de todos los alimentos del medio y sus respectivos aportes nutricionales además de una serie de dolencias alimenticias las cuales serán diagnosticadas durante el proceso de sugerencia de dieta semanal. Cuenta también con toda la lógica seguida por los nutricionistas para sugerir una dieta a partir de una lista de alimentos la cual es indicada por el sujeto a ser atendido según su consumo normal. Es en este último punto donde el proyecto difiere de otros, ya que la sugerencia estará sujeta a las posibilidades económicas del individuo a ser tratado, siempre manteniendo la vista en el objetivo el cual es hacer que la persona llegue a su peso ideal y mantenerla.

Debido al alto grado de parametrización con la que se ha dotado al sistema, el usuario podrá mantener tanto la base de conocimiento como las reglas de diagnóstico así como también todos los conceptos intermedios (fórmulas especializadas) que son usadas en el sistema para los cálculos pertinentes, registrar pacientes, generar reportes y consultar los mismos para llevar un control más exacto del progreso del individuo a ser tratado.

FACULTAD DE
**CIENCIAS E
INGENIERÍA**
ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA SUGERENCIA DE DIETAS BALANCEADAS ORIENTADAS A EQUILIBRAR Y CONTROLAR EL PESO DE LAS PERSONAS

ÁREA: SISTEMAS DE INFORMACIÓN

PROPONENTE: Ing. Rony Cueva Moscoso

ASESOR: Ing. Rony Cueva Moscoso

ALUMNO: Jesus Humberto Cano Cauzo

CÓDIGO: 20047005

TEMA N°: 467

FECHA: San Miguel, 26 de agosto de 2012



DESCRIPCIÓN

El sobrepeso se define como una acumulación anormal y excesiva de grasa que puede ser perjudicial para la salud. El sobrepeso se ha convertido en el quinto factor de principal riesgo de muerte en el mundo, según la OMS (Organización Mundial de la Salud en el documento Obesidad y Sobrepeso – Nota Descriptiva N°311 publicada en la web), debido básicamente a los malos hábitos alimenticios y al desequilibrio energético entre calorías consumidas y gastadas, siendo estas, consecuencias de un aumento en la ingesta de alimentos hipocalóricos que son ricos en grasa, sal y azúcares pero pobres en vitaminas, minerales y otros micronutrientes y un considerable descenso en la actividad física como resultado de la naturaleza sedentaria de muchas formas de trabajo, nuevos modos de desplazamiento y de una creciente urbanización.

Cada año fallecen por lo menos 2,8 millones de personas adultas por consecuencia del sobrepeso, siendo este, motivo de enfermedades como, diabetes, cardiomiopatías isquémicas y algunos cánceres.

Mantenerse en el peso ideal ayuda a combatir las enfermedades antes mencionadas, este peso ideal puede ser medido mediante el índice de masa corporal (IMC), la cual es una medida de asociación entre el peso y la talla de una persona, esta medida fue ideada por el estadístico belga Lambert Adolph Jacques Quételet. El valor obtenido no es constante sino que varía con la edad y el sexo y otros factores de índole muscular. En el caso de los adultos, este índice es utilizado como un recurso para evaluar el estado nutricional, todo esto de acuerdo con los valores propuestos por la OMS (Organización Mundial de la Salud: Índice de masa corporal apropiado. Ginebra (Suiza))

Para ayudar a llegar al peso ideal, actualmente los nutricionistas elaboran dietas que en algunos casos, no van de acuerdo al presupuesto de las personas que acuden a estas consultas, por lo que o bien se deja de lado la sugerencia médica o no se cumple del todo, motivo por el cual, este proyecto cubrirá, dada una casuística similar y basándose en el presupuesto de esta persona, la sugerencia de una dieta la cual cumpla con ayudar a llegar al peso ideal y que sea una alternativa factible para dicha persona, factible en el sentido económico.

Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono:
(511) 626 2000 Anexo 4801

FACULTAD DE
**CIENCIAS E
INGENIERÍA**
ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

OBJETIVO GENERAL

Análisis, diseño e implementación de un sistema basado en conocimiento para la sugerencia de dietas balanceadas orientadas a equilibrar y controlar el peso de las personas.

OBJETIVOS ESPECÍFICOS

OE1. Identificar tanto los aportes nutricionales como el costo de los distintos productos alimenticios de mayor comercialización en el país.

OE2. Identificar los productos alimenticios más comercializados dependiendo de la estación del año.

OE3. Identificar los presupuestos promedio diarios en los distintos niveles socioeconómicos del medio con la finalidad de definir intervalos para categorizar a los usuarios y en base a esto poder definir una sugerencia mas sujeta a su condición socioeconómica.

OE4. Representar el conocimiento sobre la información con la que se irá alimentando el sistema.

OE5. Establecer un algoritmo parametrizable que genere un listado diferente de productos alimenticios con su respectiva equivalencia en gramos por día, para ser usada en la dieta, la cual en conjunto cumpla con el suministro necesario tanto de calorías como nutrientes que permitan llegar al objetivo, el cual es controlar el peso del individuo.

OE6. Elaborar un prototipo de la solución que permita validar la solución propuesta.

ALCANCE

El sistema comprenderá, básicamente, una carga inicial de conocimiento, el cual alimentará de data al sistema, la misma que se irá actualizando con el paso del tiempo. Además, el sistema permitirá registrar los parámetros necesarios para la sugerencia de los productos y las cantidades de estos (expresado en gramos) que la persona debe consumir para controlar su peso, el cual será obtenido después de la ejecución del algoritmo de obtención de índices (masa muscular y necesidad calórica) y del mecanismo de la base de casos.

También este sistema permitirá hacer un seguimiento de la evolución de las personas que son tratadas con el uso de este sistema, además de la generación y consulta de reportes.

Máximo: 100 páginas




Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono:
(511) 626 2000 Anexo 4801



Agradecimientos

A mis padres, Edi y Jesús
por su apoyo, paciencia y comprensión durante el desarrollo de este proyecto.

A mi asesor,
por su buena disposición, tiempo y consejos a lo largo del desarrollo de este proyecto.



Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	17/04/11	1.0	Versión inicial.	Jesús Humberto Cano Cauzo
2	04/05/11	2.0	Plan de Tesis	Jesús Humberto Cano Cauzo
3	13/06/11	3.0	Capítulo 2	Jesús Humberto Cano Cauzo
4	24/08/12	4.0	Capítulo 2	Jesús Humberto Cano Cauzo
5	08/10/12	5.0	Capítulo 3	Jesús Humberto Cano Cauzo
6	26/10/12	6.0	Capítulo 4	Jesús Humberto Cano Cauzo
7	16/11/12	7.0	Capítulo 5	Jesús Humberto Cano Cauzo



Índice de Contenido

1.	Generalidades	7
1.1.	Definición del Problema.....	7
1.2.	Marco Conceptual.....	7
1.3.	Estado del Arte	16
1.4.	Plan de proyecto.....	19
1.5.	Planificación.....	20
1.6.	Descripción y sustentación de la solución.....	23
2.	Análisis.....	24
2.1.	Metodología aplicada para el desarrollo de la solución	24
2.2.	Identificación de requerimientos.....	25
2.3.	Especificación de casos de uso	26
2.4.	Definición de la base de conocimiento	32
2.5.	Definición del objetivo de la inferencia que debe realizar el sistema.....	34
2.6.	Conceptualización del método de inferencia en el que se basará el sistema.....	34
2.7.	Análisis del Sistema.....	34
3.	Diseño.....	38
3.1.	Arquitectura de la solución	38
3.2.	Diseño de interfaz gráfica.....	39
3.3.	Arquitectura de información	47
4.	Construcción y pruebas.....	49
4.1.	Construcción.....	49
4.2.	Pruebas.....	52
5.	Observaciones, conclusiones y recomendaciones.	63
5.1.	Observaciones.....	64
5.2.	Conclusiones.....	64
5.3.	Recomendaciones.....	65
6.	Referencias.....	66

Índice de Imágenes

Ilustración 1.1 Arquitectura básica de los sistemas Expertos	11
Ilustración 1.2 Forma de trabajo de los motores de inferencia	12
Ilustración 1.3 Forma de trabajo del encadenamiento regresivo	13
Ilustración 1.4 Forma de trabajo del encadenamiento progresivo	13
Ilustración 1.5 Sistema BMW 2.0 Diet Manager	17
Ilustración 1.6 Sistema EquiLibra	18
Ilustración 1.7 Sistema EquiLibra	18
Ilustración 1.8 Sistema experto SEFDA	19
Ilustración 1.9 Estructura de Descomposición de Trabajo	21
Ilustración 1.10 Cronograma del proyecto	22
Ilustración 2.1 Diagrama de casos de uso	27
Ilustración 3.1 Estructura de interacción	39
Ilustración 3.2 Estándar de pantallas	40
Ilustración 3.3 Pantalla de logueo	43
Ilustración 3.4 Pantalla principal	43
Ilustración 3.5 Pantalla de registro de usuario	44
Ilustración 3.6 Pantalla de edición de ontología	45
Ilustración 3.7 Pantalla sugerencia estándar de dieta	46
Ilustración 3.8 Pantalla de sugerencia personalizada de dieta	47

Índice de Tablas

Tabla 1.1 Dolencias alimenticias	15
Tabla 1.2 Distribución calórica	15
Tabla 1.3 Distribución calórica sobre propiedades	15
Tabla 1.4 Distribución de proteínas	15
Tabla 1.5 Distribución proteínas vegetales	16
Tabla 1.6 Comparación de Software	19
Tabla 2.1 Requerimientos Funcionales	26
Tabla 2.2 Requerimientos no Funcionales	26
Tabla 2.3 Prioridades	26
Tabla 2.4 Fórmulas usadas	33
Tabla 2.5 Comparación de lenguajes de programación	35
Tabla 2.6 Comparación de motor de base de datos	36
Tabla 2.7 Estructura del plan de trabajo	37
Tabla 2.8 Egresos y estimación de costos	37
Tabla 3.1 Etapas del diagnóstico	39
Tabla 3.2 Botones del sistema	41
Tabla 3.3 Mensajes del sistema	42
Tabla 4.1 Librerías usadas en el sistema	50
Tabla 4.2 Clases del sistema	51
Tabla 4.3 Métodos del sistema	51
Tabla 4.4 Objetos del sistema	52
Tabla 4.5 Prefijos para componentes del sistema	52
Tabla 4.6 Pruebas unitarias	53
Tabla 4.7 Pruebas del sistema	53
Tabla 4.8 Pruebas con usuario experto	53
Tabla 4.9 Casos de prueba PU001	55
Tabla 4.10 Casos de prueba PU002	57
Tabla 4.11 Casos de prueba PU003	59
Tabla 4.12 Casos de prueba PU004	60
Tabla 4.13 Casos de prueba PU005	62
Tabla 4.14 Casos de prueba PU009	63

1. Generalidades

1.1. Definición del Problema

El sobrepeso se define como una acumulación anormal y excesiva de grasa que puede ser perjudicial para la salud. El sobrepeso se ha convertido en el quinto factor de principal riesgo de muerte en el mundo, según la OMS (Organización Mundial de la Salud en el documento *Obesidad y Sobrepeso – Nota Descriptiva N°311* publicada en la web), debido básicamente a los malos hábitos alimenticios y al desequilibrio energético entre calorías consumidas y gastadas, siendo estas, consecuencias de un aumento en la ingesta de alimentos hipocalóricos que son ricos en grasa, sal y azúcares pero pobres en vitaminas, minerales y otros micronutrientes y un considerable descenso en la actividad física como resultado de la naturaleza sedentaria de muchas formas de trabajo, nuevos modos de desplazamiento y de una creciente urbanización.

Cada año fallecen por lo menos 2,8 millones de personas adultas por consecuencia del sobrepeso, siendo este, motivo de enfermedades como, diabetes, cardiomiopatías isquémicas y algunos cánceres.

Mantenerse en el peso ideal ayuda a combatir las enfermedades antes mencionadas, este peso ideal puede ser medido mediante el índice de masa corporal (IMC), la cual es una medida de asociación entre el peso y la talla de una persona, esta medida fue ideada por el estadístico belga Lambert Adolph Jacques Quételet. El valor obtenido no es constante sino que varía con la edad y el sexo y otros factores de índole muscular. En el caso de los adultos, este índice es utilizado como un recurso para evaluar el estado nutricional, todo esto de acuerdo con los valores propuestos por la OMS (Organización Mundial de la Salud: Índice de masa corporal apropiado. Ginebra (Suiza))

Para ayudar a llegar al peso ideal, actualmente los nutricionistas elaboran dietas que en algunos casos, no van de acuerdo al presupuesto de las personas que acuden a estas consultas, por lo que o bien se deja de lado la sugerencia médica o no se cumple del todo, motivo por el cual, este proyecto cubrirá, dada una casuística similar y basándose en el presupuesto de esta persona, la sugerencia de una dieta la cual cumpla con ayudar a llegar al peso ideal y que sea una alternativa factible para dicha persona, factible en el sentido económico.

[OMS11]

1.2. Marco Conceptual

1. Dieta Balanceada

Es aquella manera de alimentarse que aporta alimentos variados en cantidades adaptadas a nuestros requerimientos y condiciones personales. Llevar una dieta balanceada no es ingerir mucha comida, ya que es tan importante la cantidad como la calidad de la misma. Es importante pensar que no se puede disfrutar de una vida saludable comiendo unos pocos alimentos. La variación es lo idóneo desde el punto de vista del bienestar. De hecho, comer y beber forma parte de la alegría de vivir y, desde luego, la gastronomía no está reñida con las recomendaciones dietéticas ni con una alimentación saludable. Existen variantes de dietas dependiendo de la etapa en la cual se encuentra el individuo, por ejemplo, existen dietas para la etapa de embarazo, para la niñez y la vejez.

[LDE11]

2. Grasa

Son compuestos orgánicos que se componen de carbono, hidrogeno y oxígeno y son la fuente de energía en los alimentos. Las grasas pertenecen al grupo de las sustancias llamadas lípidos y vienen en forma líquida o sólida. Todas las grasas son combinaciones de los ácidos grasos saturados e insaturados. Una de sus principales funciones es la de proporcionar calorías al cuerpo, al igual que las proteínas y carbohidratos, proporcionando 9 calorías por gramo, siendo esta cantidad casi el doble de lo que proporcionan las proteínas y carbohidratos, además las grasas son esenciales para el funcionamiento del cuerpo, debido a que proporcionan los ácidos grasos esenciales que no son elaborados por el cuerpo y deben obtenerse de los alimentos, estos ácidos grasos son importantes para controlar la inflamación, la coagulación de la sangre y el desarrollo del cerebro. También sirve como sustancia de almacenamiento para las calorías extras del cuerpo y llena las células adiposas que ayudan a aislar el cuerpo.

[UOMMC11]

3. Sal

Es un componente químico formado por cationes enlazados a aniones, el cual es producto de una reacción química entre ácidos y bases, siendo el ácido quien proporciona el anión y la base el catión.

[UOMMC11]

4. Carbohidratos

Uno de los principales componentes de la dieta, la cual tiene como principal función la de llevar energía al cuerpo, especialmente al cerebro y al sistema nervioso, una enzima ayuda a descomponer los carbohidratos en glucosa (azúcar en la sangre) la cual se usa como fuente de energía por parte del cuerpo.

[UOMMC11]

5. Caloría

Es la unidad de energía más utilizada en sistemas biológicos, o sea, la cantidad de calor necesario para elevar un kilogramo de agua en un grado centígrado, la cual es utilizada para calcular el contenido energético de los alimento.

[VCA64]

6. Proteína

Complejo de sustancias nitrogenadas halladas en tejidos de animales y plantas. Estas constituyen las tres cuartas partes del cuerpo animal, constituidas por aminoácidos que es la unidad estructural. Existen de varios tipos, tales como proteínas simples, derivadas y conjugadas.

[VCA64]

7. Sistemas basados en conocimientos

Los Sistemas basados en Conocimiento representan un paso delante de los sistemas de información convencionales al pretender representar funciones cognitivas del ser humano como el aprendizaje y el razonamiento. Esta clase de aplicaciones descansan en las

contribuciones de la Inteligencia Artificial en lo general y en la Ingeniería del Conocimiento en lo particular. Su orientación es la automatización del análisis de problemas, la búsqueda de soluciones, la toma de decisiones y el empleo de conocimiento especializado en un campo específico de aplicación. Entre los productos más significativos de los Sistemas basados en Conocimiento se encuentran los Sistemas Expertos, los cuales están encargados de representar el conocimiento de los especialistas de una rama en la procura de su aprovechamiento para tareas de diagnóstico, enseñanza y control. La composición de los Sistemas basados en Conocimiento consta de: Un mecanismo de aprendizaje, una base de conocimientos, un motor de razonamiento, y medios de comunicación hombre-máquina.

[PAA06]

8. Características de un sistema basado en conocimientos

Cuando se modelan este tipo de sistemas se busca que tengan las siguientes características:

- **Habilidad de llegar a la solución de un problema en forma rápida**
Esta es la habilidad principal que se espera que estos sistemas puedan llevar a cabo. Al mencionar “en forma rápida” obliga a que el sistema no solo tenga conocimiento del campo en el que va a diagnosticar, sino que además tenga experiencia tomando decisiones en él.
- **Habilidad para explicar los resultados a las personas que no cuentan con ese conocimiento**
Esto significa que el sistema debe de poder responder de forma clara las preguntas concernientes a las razones de los resultados, el razonamiento derivado de los mismos y las implicaciones subsecuentes. Las personas que no cuentan con estos conocimientos esperan tener una respuesta clara que se acerque a algo que ellos pueden entender.
- **Habilidad para aprender de las experiencias**
El sistema debe de aprender tanto de sus experiencias como de las experiencias de los demás. Deben de estar al día en cuanto a la base de sus conocimientos así como a modificar el proceso de su razonamiento.
- **Habilidad de reestructurar el conocimiento para que se adapte al ambiente**
Esto se refiere a que el sistema pueda subdividir la base de su conocimiento y usar la porción útil de la misma en la resolución del problema, reduciendo así el tiempo de respuesta. También se refiere a visualizar el conocimiento desde distintas perspectivas usando varias partes del conocimiento.
- **Conciencia de sus limitaciones**
El sistema puede evaluar su capacidad para resolver un problema dado y determinar si el mismo se encuentra dentro de sus posibilidades de resolución.

[PAA06]

9. Componentes de un sistema basado en conocimientos

Los componentes de estos sistemas son los siguientes:

- **Base de conocimientos**
Existen tres tipos de conocimientos para construir un sistema basado en conocimientos:
 - Reglas

- Hechos y relaciones entre los componentes
- Afirmaciones y preguntas

Para representar estos tipos de conocimientos en la base de conocimientos, se utilizan tres métodos:

- Reglas. Son una serie de declaraciones estructuradas en forma de condicionales y están expresadas a través de las estructuras condicionales IF-THEN-ELSE. Con ellas se pueden hacer reglas que regulen el resultado de determinada información y proporcionan distintos caminos que nos lleven a un proceso más eficiente.
- Estructuras. Contienen una jerarquía de componentes y de atributos de objetos que pueden ser asignados o heredados de otras estructuras a través de diversos procedimientos. Los atributos se encuentran en las casillas de las estructuras. Una estructura, ya sea como un arreglo, como una estructura o como una casilla de atributo de una estructura puede ser llamada a través de reglas o expresiones lógicas.

La diferencia entre una estructura y una regla es que una estructura puede representar valores iniciales, punteros a otras estructuras o reglas y procedimientos para los cuales se han especificado valores, términos y condiciones de cualquier acción que necesite ser tomada. La capacidad de representar procedimientos, términos y condiciones con valores o acciones que son requeridas, es útil para conectar varios componentes de información en un sistema experto. En una estructura podemos tener representaciones descriptivas así como representaciones procedurales.

- Lógica. Las expresiones lógicas cuentan con predicados, valores y átomos para evaluar hechos del mundo real. El objeto puede ser una constante o una variable que puede cambiar a través del tiempo. El predicado puede tener uno o más argumentos que son los objetos que describe.

- **Mecanismo de inferencia**

Cuando la base de conocimientos esté completa, esta necesita ser ejecutada por un mecanismo de razonamiento y un control de búsqueda para resolver problemas. El mecanismo de inferencia es la unidad lógica con la que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas que está configurado imitando el procedimiento humano para solucionar problemas. El método más común de razonamiento en los sistemas basados en conocimientos es la aplicación del modus ponens.

- Modus ponens. Es la regla de inferencia más comúnmente utilizada. Se utiliza para obtener conclusiones simples. En ella, se examina la premisa de la regla, y si es cierta, la conclusión pasa a formar parte del conocimiento. Por ejemplo, si se tiene la regla, “Si A es cierto, entonces B es cierto” y que se sabe además que “A es cierto”. La regla Modus Ponens concluye que “B es cierto.” Esta regla de inferencia, que parece trivial, es la base de un gran número de sistemas basados en conocimientos. En un sistema basado en conocimiento existirá un hecho solo cuando esté contenido en la base de conocimiento. Para minimizar tiempos de razonamiento, los métodos de control de búsqueda son usados para determinar dónde empezar el proceso de razonamiento y elegir que regla examinar después cuando varias reglas lleguen a la misma conclusión. Existen dos formas para lograr esto:

- Usar las premisas que se ajusten a la situación para obtener otras premisas a través de las reglas y continuar así hasta llegar a las respuestas.
- Hacer uso de los hechos para llegar a la hipótesis.

Las funciones del mecanismo de inferencia son:

- Determinar qué acciones tendrán lugar, el orden en que lo harán y como lo harán.
- Determinar cómo y cuándo se procesaran las reglas, y dado el caso también la elección de que reglas deberán procesarse.
- Control del dialogo con el usuario.

La Ilustración 1.1 y la Ilustración 1.2 muestran los componentes de un sistema experto y la forma de trabajo de un motor de inferencia respectivamente.

[GJM02]

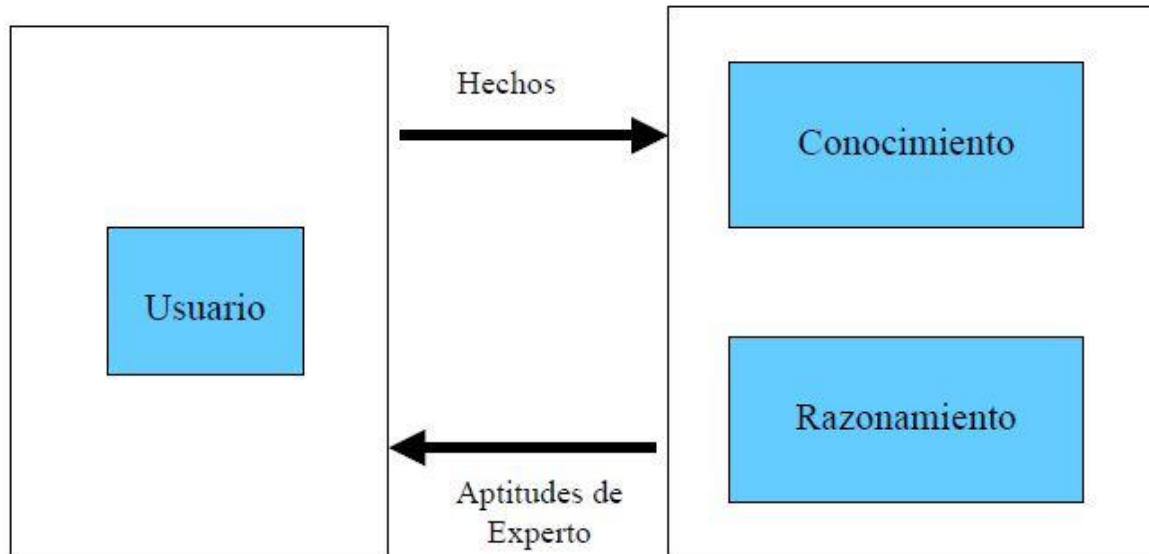


Ilustración 1.1 Arquitectura básica de los sistemas Expertos

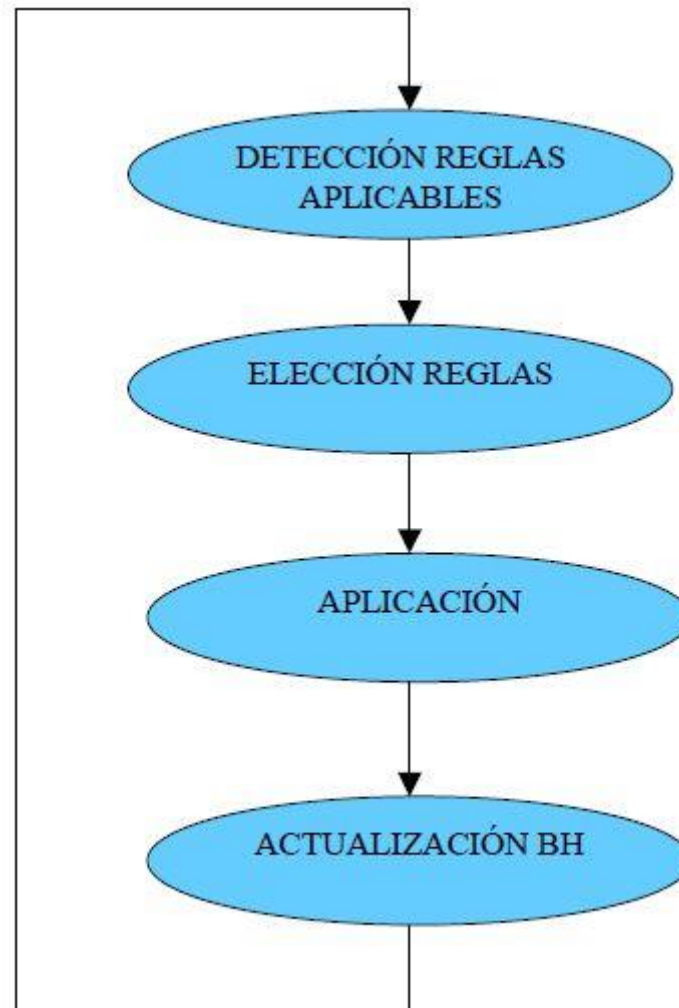


Ilustración 1.2 Forma de trabajo de los motores de inferencia

10. Métodos usados en los mecanismos de inferencia

Los métodos que utilizan los mecanismos de inferencia son los siguientes:

- Encadenamiento regresivo (Backward chaining)
 Mecanismo de inferencia que empieza con la conclusión deseada y decide si los hechos que existen pueden dar lugar a la obtención de un valor para esta conclusión. Sigue un funcionamiento similar a la búsqueda primero en profundidad. El sistema empieza con un conjunto de hechos conocidos que típicamente está vacío, se proporciona una lista ordenada de objetivos (o conclusiones), para las cuales el sistema trata de derivar valores. El proceso de razonamiento regresivo utiliza una lista de objetivos para coordinar su búsqueda a través de las reglas de la base de conocimientos. La Ilustración 1.3 muestra la forma de trabajo del encadenamiento regresivo.

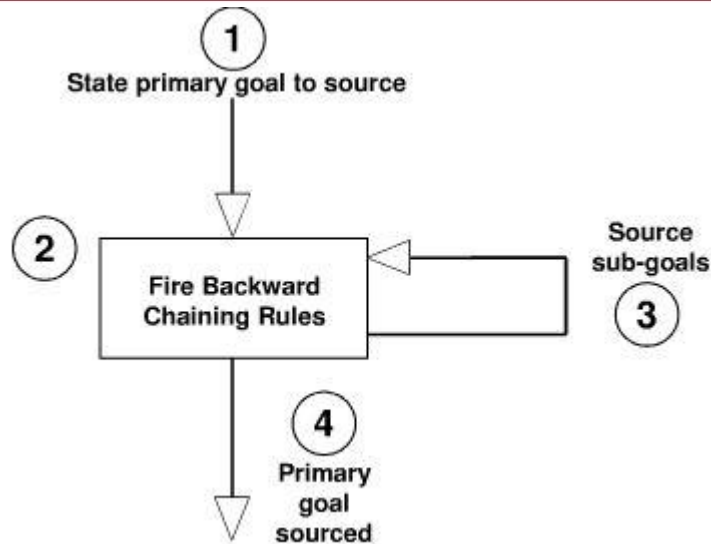


Ilustración 1.3 Forma de trabajo del encadenamiento regresivo

- Encadenamiento progresivo (Forward chaining)
Mecanismo de inferencia el cual empieza a partir de un conjunto de datos colectados y se evoluciona hacia una conclusión. Se chequea cada una de las reglas para ver si los datos observados satisfacen las premisas de alguna de las reglas. Si una regla es satisfecha, es ejecutada derivando nuevos hechos que pueden ser utilizados por otras reglas para derivar hechos adicionales. La Ilustración 1.4 muestra la forma de trabajo del encadenamiento progresivo.

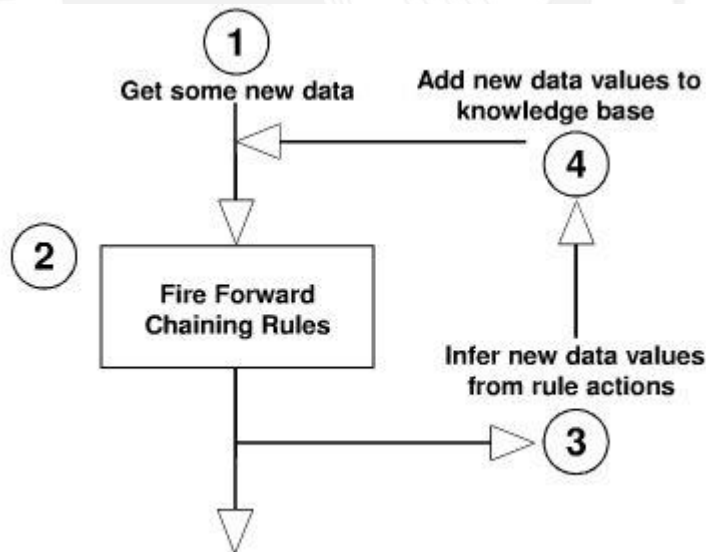


Ilustración 1.4 Forma de trabajo del encadenamiento progresivo

[IFBC12]

11. Ontología

La literatura de inteligencia artificial contiene varias definiciones de ontología; muchas de ellas contradicen otras. Para propósitos de este proyecto, una ontología es una descripción explícita y formal de conceptos en un dominio de discurso (clases o

conceptos), propiedades de cada concepto describiendo varias características y atributos del concepto (slots, a veces llamados roles o propiedades), y restricciones sobre los slots (facetas, a veces llamado restricciones de rol).

Una ontología junto con un conjunto de individuos de clases constituye una base de conocimiento.

[NNFMD05]

12. Diagnóstico de dolencias alimenticias

Para poder definir como se realizan los diagnósticos de este tipo de enfermedades, explicaremos en primera instancia, que es diagnóstico y las enfermedades que serán tomadas en cuenta para el diagnóstico del sistema.

Según el diccionario de la real academia de la lengua española, el diagnóstico es determinar el carácter de una enfermedad mediante el examen de sus signos, para ello, los expertos en esta materia revisan una inspección general del paciente en primera instancia, tomando posibles candidatos de diagnóstico, los cuales, en base a una serie de exámenes clínicos y/o de sintomatologías, van descartando que enfermedades no cumplen con las características que ellos manejan para luego obtener el diagnóstico final. [RAE12]

Las dolencias alimenticias que son tomadas en cuenta para este proyecto son las que se detallan en la Tabla 1.1:

Dolencia	Características
Hipercolesterolemia	Aumento del colesterol en la sangre por encima de los 2.5 g/l la cual puede ser congénita o adquirida. Provoca enfermedades coronarias que pueden llegar a ser graves, como el infarto cardiaco.
Hipoglucemia	Reducción de los niveles de glucosa en sangre. las causas pueden variar entre un ayuno muy prolongado, una administración excesiva de insulina o a trastornos metabólicos
Hiperglucemia	Aumento de los niveles de glucosa en sangre. Se puede producir como un fenómeno natural y transitorio tras la ingestión de una comida o como síntoma de diabetes o de ciertas enfermedades apáticas.
Diabetes mellitus	Enfermedad que se caracteriza por un exceso de glucosa en sangre, aun mas que la hiperglucemia, causada por el déficit de insulina. Los síntomas de esta enfermedad pueden ser cansancio, debilidad general, predisposición a las infecciones, micciones muy frecuentes, ceguera, trastornos vasculares, cerebrales y nerviosos, mala circulación sanguínea, etc.

Infra peso	Nivel por debajo del normal. Concepto relacionado con el cálculo del índice de masa corporal, si este índice tiene un valor menor a 18.5 se considera que la persona tiene infra peso.
Obesidad	Acumulación excesiva de grasa en el cuerpo. Las causas pueden ser múltiples como la herencia, el exceso de aportes energéticos respecto al consumo energético, trastornos psicoafectivos, trastornos endocrinos y alteraciones en el sistema nervioso. La obesidad aumenta el riesgo de padecer enfermedades cardiovasculares. Dependiendo de las formulas con las que se obtengan, la obesidad puede ser clasificada como obesidad tipo I, II y III.

Tabla 1.1 Dolencias alimenticias

[DS12]

13. Proceso para el cálculo de los requerimientos nutricionales

Una vez calculada las calorías totales para las 24 horas, se debe distribuir esta cantidad entre el desayuno, almuerzo y cena. Esta distribución se rige bajo los siguientes aspectos teórico-prácticos que se muestran en la Tabla 1.2:

Objetivo	Distribución (%)
Desayuno	[25 .. 30]
Almuerzo	[40 .. 50]
Cena	[15 .. 20]

Tabla 1.2 Distribución calórica

Luego de esta distribución, se debe tener en cuenta que estas cantidades deben ser distribuidas entre proteínas, carbohidratos y grasas. Esta distribución se rige bajo los siguientes aspectos teórico-prácticos que se muestran en la Tabla 1.3:

Objetivo	Distribución (%)
Proteínas	[12 .. 15]
Carbohidratos	[50 .. 60]
Grasas	[25 .. 35]

Tabla 1.3 Distribución calórica sobre propiedades

Para calcular la cantidad de nutrientes es prioritario calcular en primera instancia las proteínas teniendo en cuenta los siguientes parámetros (aplicados en la práctica clínica) que se muestran en la Tabla 1.3:

Tipo	Distribución (%)
Proteína animal	70
Proteína vegetal	30

Tabla 1.4 Distribución de proteínas

En caso existan nutrientes de origen animal, la distribución de las proteínas debe hacerse de forma uniforme.

Los nutrientes de origen vegetal son diferenciados en dos categorías, las cuales se muestran en la Tabla 1.5:

Tipo	Distribución (%)
Frutas y verduras	30
Otros	70

Tabla 1.5 Distribución proteínas vegetales

La distribución anterior se fundamenta en el hecho de que la distribución equitativa de los distintos vegetales incrementa la cantidad en forma exagerada de las frutas y verduras las cuales no van acorde con una aplicación práctica culinaria.

Las cantidades de carbohidratos y grasas se obtienen a partir de la información de los distintos alimentos y el cálculo de las proteínas anteriormente expuestas. El cálculo del azúcar se usa para complementar los requerimientos de carbohidratos así como el aceite vegetal se usa para complementar los requerimientos de grasas siempre y cuando estos sean insuficientes.

Se debe tener en cuenta que los valores de calorías y proteínas son los que determinan los requerimientos nutricionales del paciente.

[EE12]

1.3. Estado del Arte

Los sistemas de información tradicionales son mecanizados, concebidos e implementados como instrumentos que procesan información con el propósito de apoyar a la toma de decisiones, contener “conocimiento técnico”, reducir a un grado mínimo el error humano, automatizar las operaciones repetitivas, bajar costos, facilitar la tarea al usuario, ofrecer información, manipular gran cantidad de datos y satisfacer las necesidades del usuario, mientras que los sistemas basados en conocimiento se consideran como una extensión de los sistemas de información, cuyos alcances y complejidad son mayores,. Estas tienen por propósito aprender, evolucionar, adaptar, razonar, tomar decisiones contener conocimiento empírico, mundano y del lenguaje, analizar problemas, generar alternativas de solución, emular al experto humano y generar conocimiento a partir del que ya se tiene.

Para la concepción de un sistema basado en conocimientos, podemos emplear la ingeniería de ontologías, la cual permite convertir información en conocimiento mediante unas estructuras de conocimiento formalizadas (Ontologías) que referencien los datos, por medio de metadatos, bajo un esquema común normalizado sobre algún dominio del conocimiento. Usando la ontología podemos tener beneficios como proporcionar una forma de representar y compartir el conocimiento utilizando un vocabulario común, permiten usar un formato de intercambio de conocimiento, proporcionan un protocolo específico de comunicación y permiten una reutilización del conocimiento. Este proyecto se apoyará en el uso de ontologías para poder representar el conocimiento.

Actualmente, existen algunos sistemas para el control del peso, de las cuales se puede mencionar:

- **BWM 2.0 diet manager:** Sistema administrador de dietas, multiplataforma que incluye una completa base de datos de productos alimenticios, menús de restaurantes y demás herramientas que se necesita para controlar el peso de las

personas. Este sistema proporciona reportes diarios de los progresos de las personas que usan esta herramienta.

Una vista del sistema se muestra en la Ilustración 1.5:

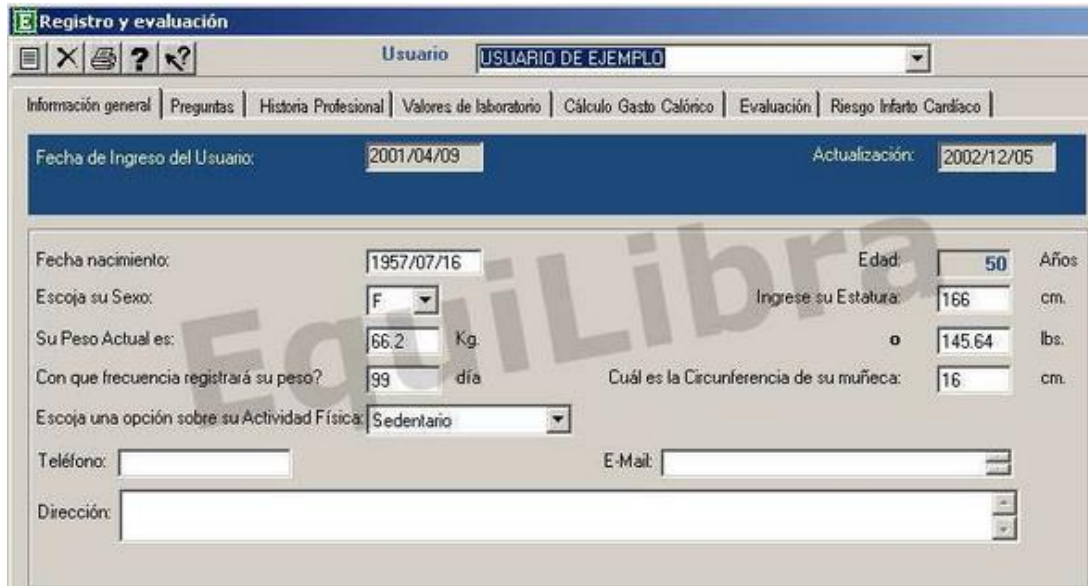


Ilustración 1.5 Sistema BWM 2.0 Diet Manager

[BDM11]

EquiLibra: Sistema desarrollado para profesionales de la nutrición que requieren de un asistente para realizar cálculos nutricionales y soporte informático para llevar un registro nutricional de cada paciente. Genera dietas balanceadas e imprime dichas dietas para los pacientes, genera nuevas dietas modificando las ya existentes en el sistema y realiza un control del costo por dieta.

Una vista del sistema se muestra en la Ilustración 1.6:



Registro y evaluación

Usuario: USUARIO DE EJEMPLO

Información general | Preguntas | Historia Profesional | Valores de laboratorio | Cálculo Gasto Calórico | Evaluación | Riesgo Infarto Cardíaco

Fecha de Ingreso del Usuario: 2001/04/09 Actualización: 2002/12/05

Fecha nacimiento: 1957/07/16 Edad: 50 Años

Escoja su Sexo: F Ingrese su Estatura: 166 cm.

Su Peso Actual es: 66.2 Kg. Ingrese su peso: 145.64 lbs.

Con que frecuencia registrará su peso? 99 día Cuál es la Circunferencia de su muñeca: 16 cm.

Escoja una opción sobre su Actividad Física: Sedentario

Teléfono: E-Mail:

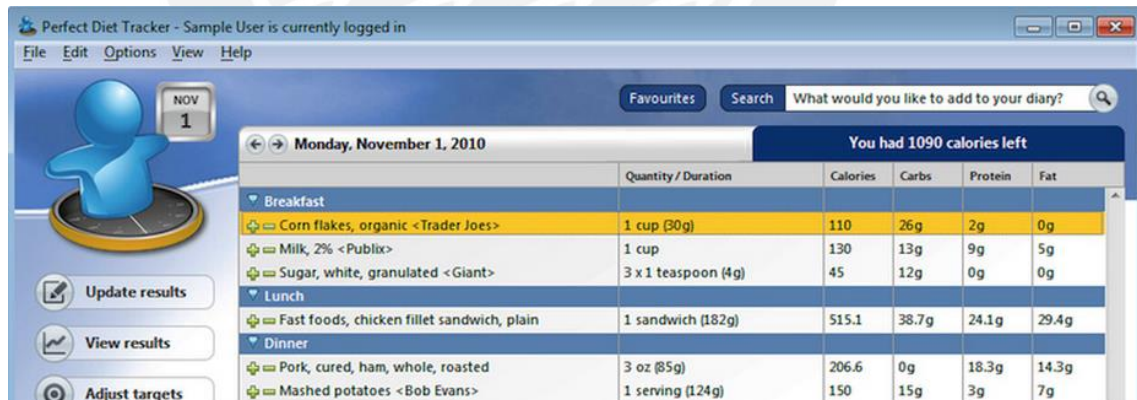
Dirección:

Ilustración 1.6 Sistema EquiLibra

[ESN11]

- **Perfect Diet Tracker:** Sistema multiplataforma basado en la investigación científica y usando la ecuación de Harris Benedict para calcular la tasa metabólica basal. Cuando se crea un nuevo usuario en el software, se le pide información como edad, altura, peso, sexo y el nivel actual de actividad, para luego ser usados en el cálculo del número de calorías que necesita consumir para mantener su peso, luego, fija metas semanales para bajar de peso, sugiriendo una cifra diaria de ingesta de calorías.

Una vista del sistema se muestra en la Ilustración 1.7.



Perfect Diet Tracker - Sample User is currently logged in

File Edit Options View Help

Search What would you like to add to your diary?

Monday, November 1, 2010 You had 1090 calories left

	Quantity / Duration	Calories	Carbs	Protein	Fat
Breakfast					
☛ Corn flakes, organic <Trader Joes>	1 cup (30g)	110	26g	2g	0g
☛ Milk, 2% <Publix>	1 cup	130	13g	9g	5g
☛ Sugar, white, granulated <Giant>	3 x 1 teaspoon (4g)	45	12g	0g	0g
Lunch					
☛ Fast foods, chicken fillet sandwich, plain	1 sandwich (182g)	515.1	38.7g	24.1g	29.4g
Dinner					
☛ Pork, cured, ham, whole, roasted	3 oz (85g)	206.6	0g	18.3g	14.3g
☛ Mashed potatoes <Bob Evans>	1 serving (124g)	150	15g	3g	7g

Update results View results Adjust targets

Ilustración 1.7 Sistema EquiLibra

[PDT11]

- **Sistema experto para la formulación de dietas alimenticias por vía enteral en el ser humano (SEFDA):** El sistema formula dietas alimenticias a personas que presentan las siguientes enfermedades: anemia, desnutrición proteica y calórica, diabetes, hiperlipidemias, hiperglucemia, hipertensión arterial, hiperuricemia, insuficiencia renal y obesidad. Desarrollado a partir de análisis y la recopilación de la información necesaria de parte de los nutricionistas, para entender el proceso de formulación manual que estos realizan.

Una vista del sistema se muestra en la Ilustración 1.8:



Ilustración 1.8 Sistema experto SEFDA

[RGM92]

A continuación, en la Tabla 1.2, se muestra una comparación entre dichos sistemas:

Características	BWM 2.0 Dieta Manager	EquiLibra	Perfecta Dieta Cracker	SEFDA
Generar Dietas Aleatorias	SI	SI	SI	SI
Registrar Usuario	SI	SI	SI	SI
Control Personalizado	SI	SI	SI	SI
Generar Reportes	SI	SI	SI	SI
Control de Costo por Dieta	NO	SI	NO	NO
Basados en Ontologías	NO	NO	NO	NO
Multiplataforma	SI	NO	SI	NO

Tabla 1.6 Comparación de Software

1.4. Plan de proyecto

Para este proyecto se tomará como base la estructura que tienen los sistemas basados en conocimiento y teniendo en consideración que el objetivo principal del proyecto es la construcción del sistema, se decide tomar la siguiente estructura:

- 1. Concepción del ámbito del conocimiento y toma de requerimientos.**
 - a. Definición de la base de conocimiento.
 - b. Formalización de la base de conocimiento.
 - c. Lista de requerimientos.
 - d. Identificación de casos de uso.
 - e. Especificación de requerimientos.
- 2. Concepción del mecanismo de inferencia.**
 - a. Definición del objetivo de la inferencia que debe realizar el sistema.
 - b. Conceptualización del método de inferencia en el que se basará el sistema.
 - c. Identificación de reglas a ser usadas por el sistema.
 - i. Definición de dolencias alimenticias a ser cubiertas por el sistema.
 - ii. Reglas a ser usadas para el diagnóstico
- 3. Arquitectura e Interfaz de interacción con usuario.**
 - a. Definición de arquitectura del sistema.
 - i. Descripción de la arquitectura del sistema.
 - b. Diseño de interfaz gráfica.
- 4. Construcción y pruebas del sistema.**
 - a. Tecnología a ser utilizada para la construcción.
 - b. Base de conocimiento formalizado.
 - c. Motor de diagnóstico.
 - d. Método de distribución de alimentos.
 - e. Catálogo de pruebas.

1.5. Planificación

Para la planificación del proyecto planteamos la siguiente Estructura de descomposición de trabajo (EDT) y el cronograma que se muestran en la Ilustración 1.9 y 1.10 respectivamente.

Estructura de Descomposición de Trabajo (EDT)

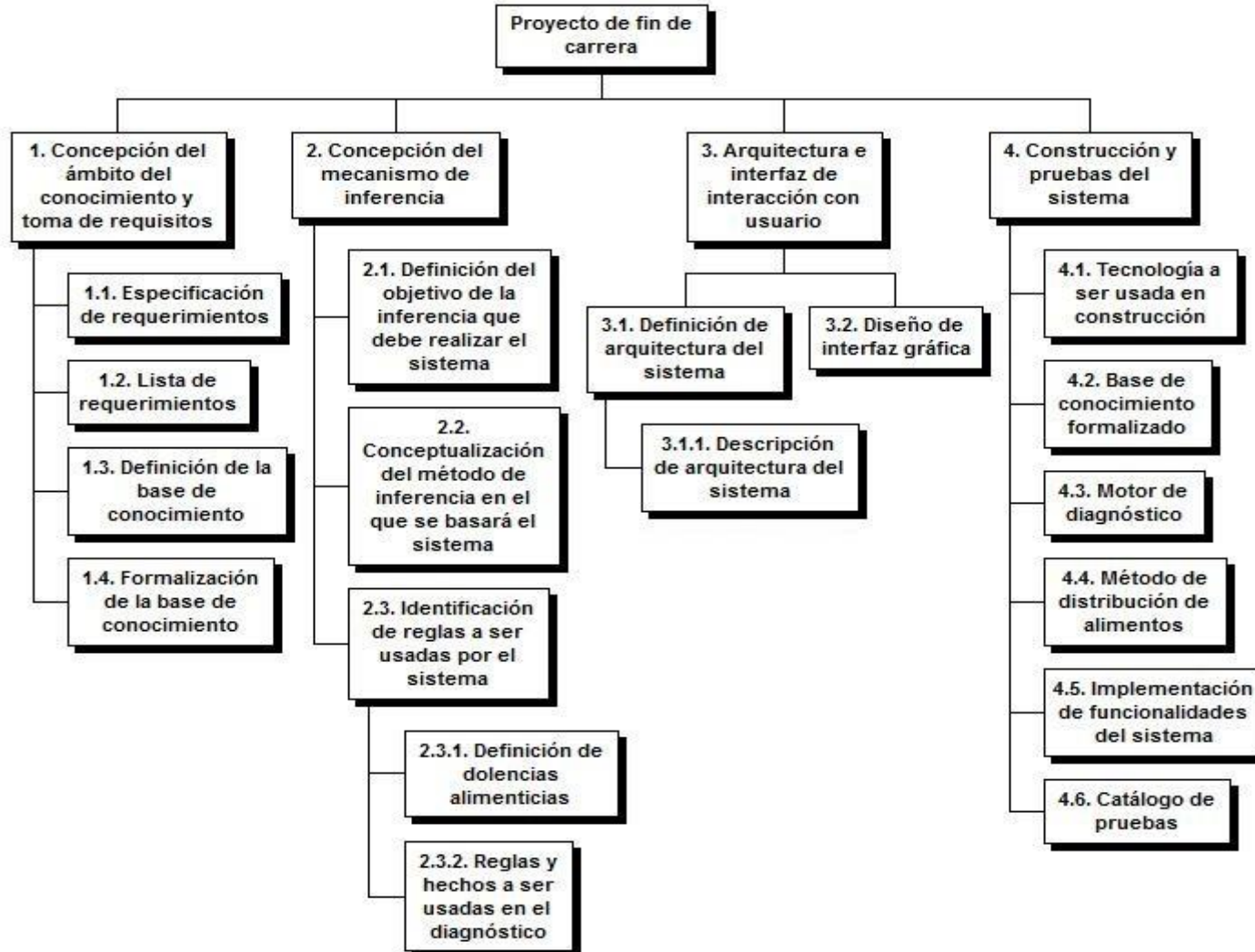


Ilustración 1.9 Estructura de Descomposición de Trabajo

Cronograma

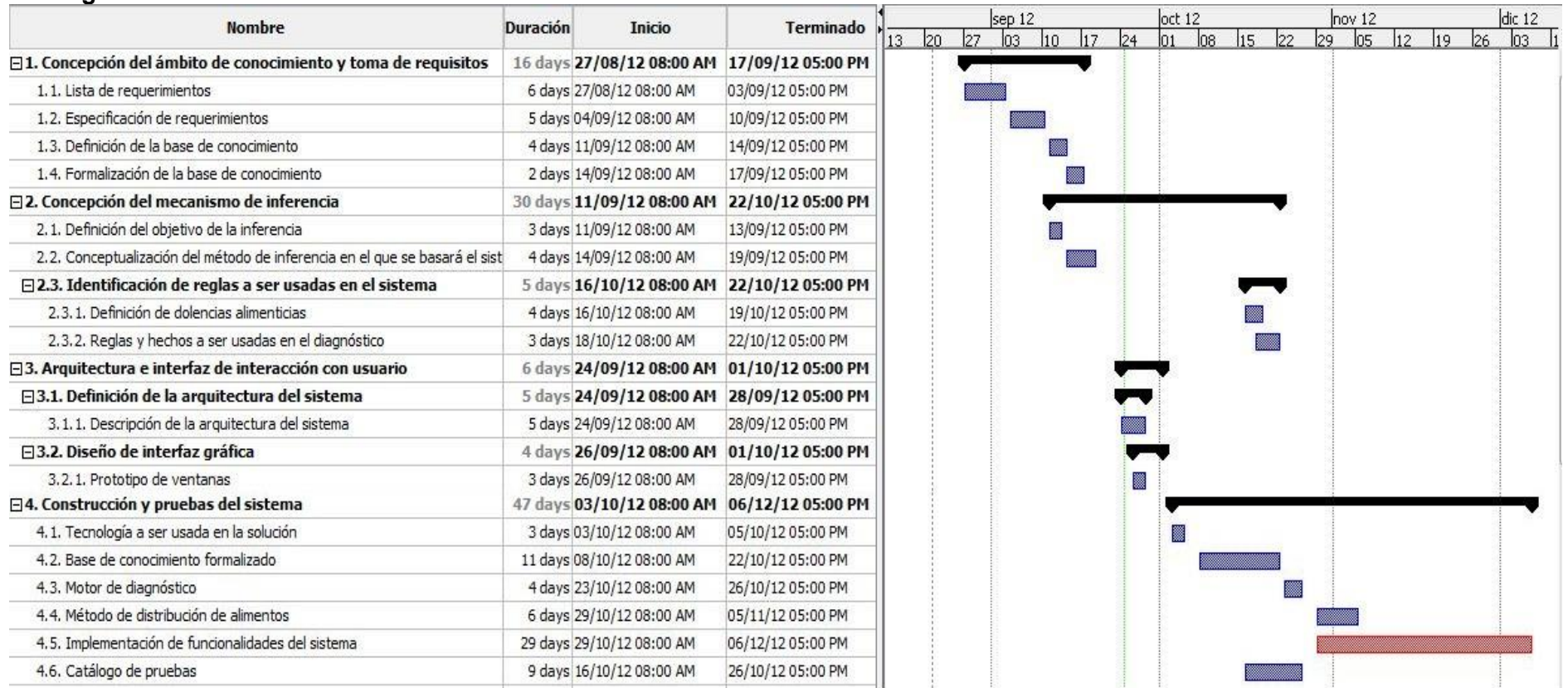
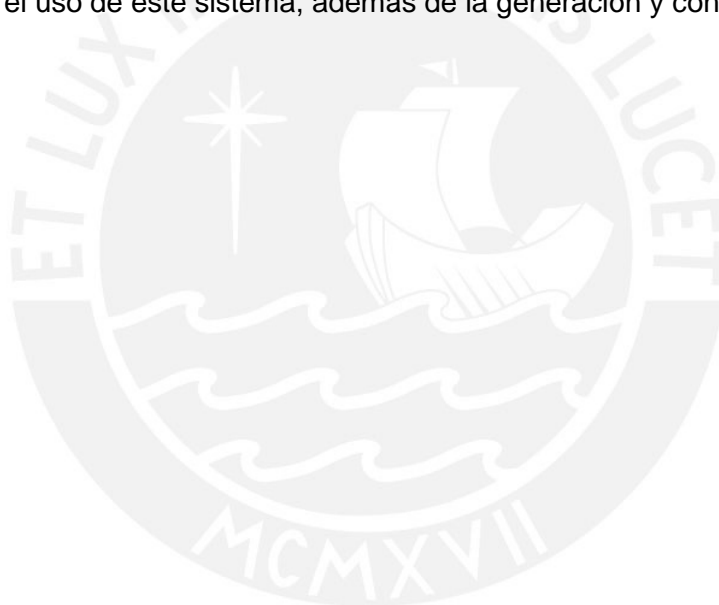


Ilustración 1.10 Cronograma del proyecto

1.6. Descripción y sustentación de la solución

El sistema comprenderá, básicamente, una carga inicial de conocimiento a nivel de la ontología, el cual alimentará de data al sistema, la misma que podrá ser actualizada desde la aplicación. Además, el sistema permitirá registrar los parámetros necesarios para la sugerencia de los productos y las cantidades de estos (expresado en gramos) que la persona debe consumir para controlar su peso, el cual será obtenido después de la ejecución del algoritmo de obtención de índices (masa muscular, necesidad calórica y peso ideal) y basado en los niveles de azúcar y colesterol en sangre, se diagnosticará que tipos de dolencias alimenticias padece el paciente a tratar, una vez realizado el diagnóstico, el cual se realizara usando el método de encadenamiento progresivo, se lanzarán las sugerencias estándares para cada tipo de patología, la cual puede ser modificada según el paciente lo indique (respetando que alimentos consume en su día a día), con este detalle se estará cubriendo el hecho de que la sugerencia generada vaya de acuerdo con la situación socio económica del paciente.

También este sistema permitirá hacer un seguimiento de la evolución de las personas que son tratadas con el uso de este sistema, además de la generación y consulta de reportes.



2. Análisis

En este capítulo se hará mención tanto de la metodología que se usará para el desarrollo de la solución, la identificación de requerimientos como el análisis de la solución propuesta.

2.1. Metodología aplicada para el desarrollo de la solución

La metodología a usar en este proyecto abarca los siguientes aspectos:

1. Concepción del ámbito del conocimiento y toma de requerimientos.

En este apartado, se realiza la investigación y documentación del dominio de la base de conocimiento del sistema, además de la identificación de requisitos del sistema, los entregables y donde serán usados se detallan a continuación:

- a. Definición de la base de conocimiento.
Parte del documento de análisis, en el se describirá el dominio y la forma de representación de la base de conocimientos.
- b. Formalización de la base de conocimiento.
Parte del documento de arquitectura, se define la estructura de la base de conocimientos.
- c. Lista de requerimientos.
Parte del documento de análisis, se definen los requerimientos funcionales y no funcionales del sistema.
- d. Identificación de casos de uso.
Parte del documento de análisis, se identifican y diagraman los mismos.
- e. Especificación de requerimientos.
Parte del documento de análisis, se realiza la especificación de las funcionalidades del sistema.

2. Concepción del mecanismo de inferencia.

En este apartado, se realiza la investigación de cómo es la mejor forma de conceptualizar la forma en la que se basa el sistema experto, los entregables y donde serán usados se detallan a continuación:

- a. Definición del objetivo de la inferencia que debe realizar el sistema.
Parte del documento de análisis. El sistema debe realizar un diagnóstico de dolencias previo a la ejecución de la sugerencia de dieta.
- b. Conceptualización del método de inferencia en el que se basará el sistema.
Parte del documento de análisis, se identifica que tipo de encadenamiento será el que se use para encadenar el conocimiento del experto.

3. Arquitectura e Interfaz de interacción con usuario.

En este apartado, se define concretamente la arquitectura que seguirá el sistema de este proyecto. Los entregables y donde serán usados se detallan a continuación:

- a. Definición de arquitectura del sistema.
Parte del documento de arquitectura, se indicarán los siguientes aspectos:
 - i. Descripción de la arquitectura del sistema.
- b. Diseño de interfaz gráfica.

4. Construcción y pruebas del sistema.

- a. Tecnología a ser utilizada para la construcción.
- b. Base de conocimiento formalizado.
- c. Motor de diagnóstico.
- d. Identificación de reglas a ser usadas por el sistema.
Parte de la fase de construcción y pruebas (documentación sobre el motor de diagnóstico), se indicarán los siguientes aspectos:
 - i. Definición de dolencias alimenticias a ser cubiertas por el sistema.
 - ii. Reglas y hechos a ser usadas para el diagnóstico
- e. Método de distribución de alimentos.
- f. Catálogo de pruebas.

2.2. Identificación de requerimientos.

La lista de requerimientos funcionales y no funcionales que el sistema deberá cumplir se muestra a continuación:

2.2.1. Requisitos Funcionales

Número	Requerimiento	Prioridad
R01	El sistema deberá permitir realizar el mantenimiento de usuarios	1
R02	El sistema deberá permitir hacer un seguimiento de los movimientos del usuario en el sistema	1
R03	El sistema permitirá registrar y modificar la contraseña de acceso al sistema del usuario	1
R04	El sistema deberá bloquear a un usuario por un determinado tiempo por el motivo de la acumulación de intentos fallidos de acceso al sistema	1
R05	El sistema deberá permitir el mantenimiento de pacientes	1
R06	El sistema deberá permitir hacer un seguimiento del progreso del paciente	1
R07	El sistema deberá permitir mantener la base de conocimiento en tiempo de ejecución.	1
R08	El sistema deberá permitir mantener la información relevante a los productos alimenticios	1
R09	El sistema deberá generar el nombre de usuario del sistema de manera automática.	1
R10	El sistema deberá permitir diagnosticar cual es la dolencia alimenticia del paciente a tratar, tomando como indicadores de diagnóstico, el nivel de colesterol, azúcar en la sangre e índice de masa corporal.	1
R11	El sistema deberá explicar al usuario como llego a una determinada sugerencia de dieta	2
R12	El sistema deberá permitir mantener el historial clínico del paciente	1
R13	El sistema deberá permitir la generación de reportes con los datos de los pacientes atendidos por usuario	1

R14	El sistema deberá permitir sugerir una dieta alimenticia para el paciente a tratar.	1
R15	El sistema deberá permitir la exportación de los reportes en formato pdf	2
R16	El sistema deberá permitir la generación de reporte de movimientos en el sistema de los usuarios registrado con el fin de llevar un control del uso del sistema	1
R17	El sistema deberá generar de forma automática la contraseña del usuario registrado	1
R18	El sistema deberá permitir mantener perfiles de usuario.	1
R19	El sistema deberá permitir modificar los alimentos sugeridos en una dieta a gusto del paciente (según su preferencia)	1
R20	El sistema deberá permitir generar reportes sobre que pacientes padecen algún tipo de dolencia alimenticia específica	1
R21	El sistema deberá permitir generar reportes sobre cuáles son los alimentos más solicitados en las dietas	2
R22	El sistema deberá encriptar la contraseña de los usuarios	1
R23	El sistema deberá validar que un usuario no inicie sesión paralelamente a una sesión ya activa.	3

Tabla 2.1 Requerimientos Funcionales

2.2.2. Requisitos No Funcionales

Número	Requerimiento	Prioridad
R01	El sistema deberá presentar una interfaz cliente servidor	1
R02	El sistema se deberá desarrollar en lenguaje java	1
R03	El sistema deberá ser multiplataforma	2
R04	El sistema deberá contar con una base de datos Mysql	2
R05	El sistema deberá soportar el acceso de múltiples usuarios al sistema y a la base de datos (máximo 25 usuarios)	2
R06	El sistema deberá contar con una base de conocimiento basado en el lenguaje OWL	1

Tabla 2.2 Requerimientos no Funcionales

Número	Descripción
1	Alto
2	Medio
3	Bajo

Tabla 2.3 Prioridades

2.3. Especificación de casos de uso

El diagrama de los principales casos de uso, así como la especificación de los mismos, se presenta a continuación.

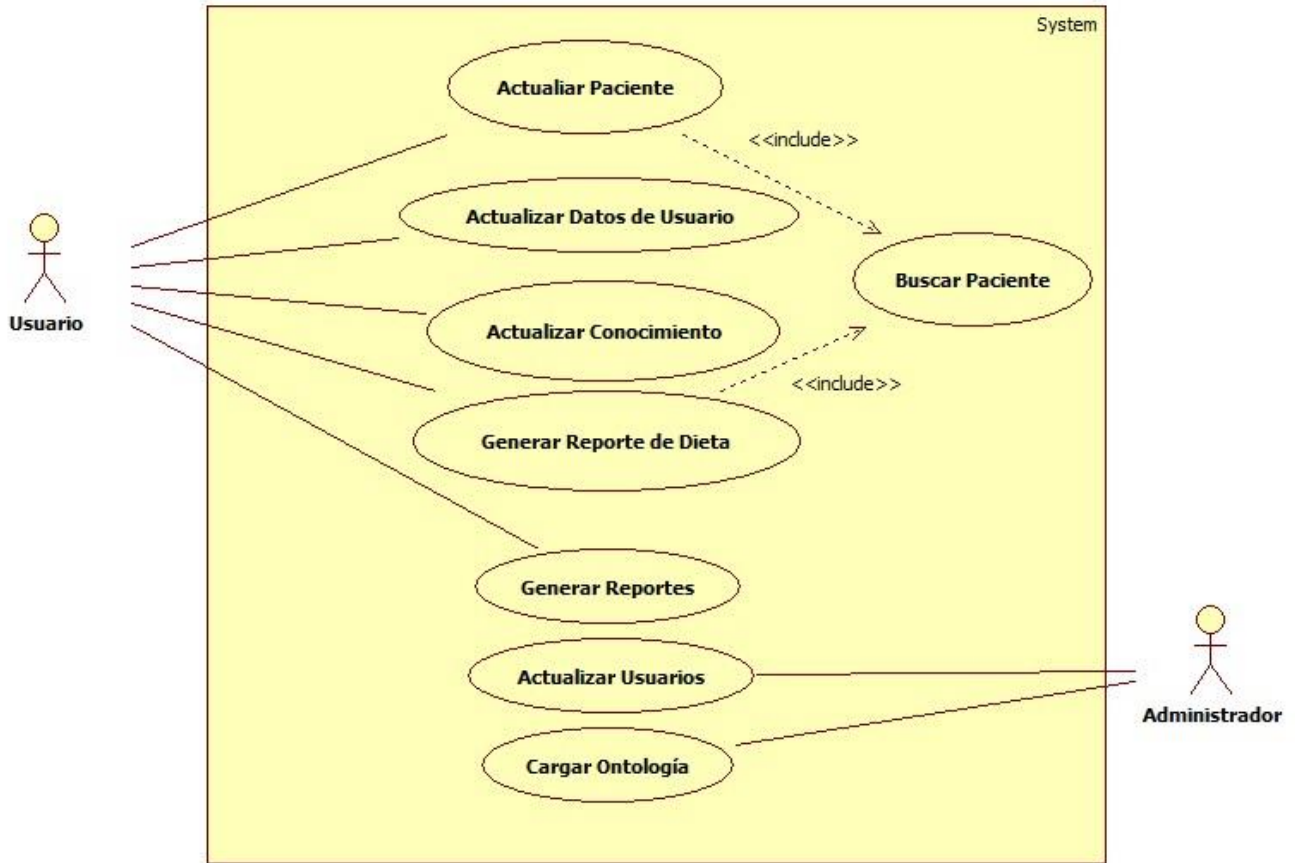


Ilustración 2.1 Diagrama de casos de uso

2.3.1. Actualizar paciente

ID	APP-01
Descripción	Este caso de uso se encarga del registro de pacientes en el sistema
Actor	Usuario
Precondición	El usuario debe haber ingresado al sistema.
Post-condición	El usuario registra al paciente en el sistema.
Flujo básico: “Registrar paciente”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Nuevo Paciente”. 2. El sistema muestra un formulario de la sección pacientes con los campos: Apellido paterno, apellido materno, nombres, dirección, DNI, fecha de nacimiento, peso, sexo, talla. 3. El usuario completa los datos y selecciona la opción “Registrar”. 4. El sistema muestra un mensaje de confirmación del registro y el caso de uso finaliza. 	
Flujo alternativo: “Modificar paciente”	

1. <<include Buscar paciente>>.
2. El sistema muestra un formulario en el que se pueden modificar los datos del paciente, tales como: Apellido paterno, apellido materno, nombres, dirección, DNI, fecha de nacimiento, peso, sexo, talla.
3. El usuario modifica los campos que desee y selecciona la opción “Modificar”.
4. El sistema muestra un mensaje de confirmación de la modificación de los datos del paciente y el caso de uso finaliza.

2.3.2. Generar sugerencia de dietas

ID	APP-02
Descripción	Este caso de uso se encarga de la generación del reporte de sugerencia de dietas para el paciente
Actor	Usuario
Precondición	El usuario debe haber ingresado al sistema.
Post-condición	El usuario genera el reporte de sugerencia de dietas para el paciente

Flujo básico: “Generar sugerencia de dietas”

1. El usuario selecciona la opción “Atención”.
2. El sistema muestra un formulario de la sección atención con los campos: sexo, edad, talla, peso actual, nivel de colesterol, nivel de azúcar en la sangre, índice de masa muscular, necesidad calórica y peso ideal para la zona de datos personales.
3. El usuario selecciona la opción “Buscar Paciente” para buscar al paciente que se le va a generar la sugerencia de dietas.
4. <<include Buscar paciente>>
5. El usuario ingresa los valores en los campos nivel de colesterol y nivel de azúcar en la sangre y selecciona la opción “Calcular”
6. El sistema muestra la información de la zona de datos personales cargadas con la del paciente buscado, además de la autogeneración de los campos índice de masa muscular, necesidad calórica y peso ideal.
7. El sistema inicia el proceso de diagnóstico de las dolencias que padece el paciente tratado.
8. El sistema muestra una plantilla predeterminada para el paciente dependiendo del diagnóstico generado.
9. El usuario puede modificar las plantillas predeterminadas por el sistema, adecuándolos a la realidad del paciente.
10. El usuario completa la información de la zona alimentos y selecciona la opción “Generar”.

11. El sistema muestra la lista de distribución de dietas para la semana y una leyenda indicando el por qué de esta conclusión y el caso de uso finaliza.

2.3.3. Buscar paciente

ID	APP-03
Descripción	Este caso de uso se encarga de la búsqueda de pacientes en el sistema.
Actor	Usuario
Precondición	El usuario debe haber ingresado al sistema.
Post-condición	El sistema devuelve la información del paciente buscado.

Flujo básico: "Buscar paciente"

1. El usuario selecciona la opción "Buscar paciente".
2. El sistema muestra un formulario con los campos: Apellido paterno, apellido materno, nombres, DNI.
3. El usuario completa los datos que desee y selecciona la opción "Buscar".
4. El sistema muestra los resultados de la búsqueda en una tabla.
5. El usuario selecciona el resultado que desee y elige la opción "Aceptar" y el caso de uso finaliza.

2.3.4. Diagnosticar dolencia

ID	APP-04
Descripción	Este caso de uso se encarga del diagnóstico de alguna dolencia alimenticia que presente el paciente a tratar.
Actor	Usuario
Precondición	El usuario debe haber ingresado al sistema y haber seleccionado a opción "Atención".
Post-condición	El sistema determina que dolencias alimenticias padece el paciente.

Flujo básico: "Diagnosticar dolencia"

1. El sistema, basándose en los valores ingresados por el usuario (colesterol y azúcar en la sangre) así como del valor calculado para obtener el índice de masa muscular, inicia el proceso de diagnóstico.
2. El sistema haciendo uso del motor de inferencia en el módulo de diagnóstico y haciendo uso del método de encadenamiento regresivo, determina los tipos de dolencias que padece el paciente y el caso de uso finaliza

2.3.5. Actualizar conocimiento

ID	ONTO-01
-----------	---------

Descripción	Este caso de uso se encarga del registro de instancias en la base de conocimiento del sistema
Actor	Usuario
Precondición	El usuario debe haber ingresado al sistema.
Post-condición	El usuario registra la instancia en la base de conocimientos del sistema.
Flujo básico: “Registrar conocimiento”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Registrar Instancia”. 2. El sistema muestra un formulario de la sección ontología con los campos: Seleccione la clase que desea editar, seleccione la subclase que desea editar, una tabla con las instancias actuales de una subclase, y en la zona de edición los campos instancia, valor energético, proteínas, carbohidratos, grasas, azúcar. 3. El usuario selecciona la clase y la subclase en la que desee agregar una instancia, y completa los campos de la zona de edición y elige la opción “Grabar”. 4. El sistema muestra un mensaje de confirmación del registro de la nueva instancia. 5. El usuario selecciona la opción “Aceptar” y el sistema actualiza la tabla, agregando la nueva instancia registrada y el caso de uso finaliza. 	
Flujo alternativo: “Modificar conocimiento”	
<ol style="list-style-type: none"> 1. El usuario selecciona una clase y una subclase. 2. El sistema muestra la tabla de la sección instancias con la información de todas las instancias que pertenecen a esa subclase, tales como nombre, valor energético, proteínas, carbohidratos, grasas, azúcar. 3. El usuario selecciona de la tabla una de las instancias que desee modificar. 4. El sistema completa la zona de edición con la información de la instancia seleccionada. 5. El usuario edita la información que desee relevante, tales como nombre, valor energético, proteínas, carbohidratos, grasas, azúcar y selecciona la opción “Modificar”. 6. El sistema muestra un mensaje de confirmación de la modificación de la instancia. 7. El usuario selecciona la opción “Aceptar” y el sistema actualiza la tabla de la zona de instancias y el caso de uso finaliza. 	
Flujo alternativo: “Eliminar conocimiento”	
<ol style="list-style-type: none"> 1. El usuario selecciona una clase. 2. El sistema muestra la tabla de la sección instancias con la información de todas las instancias que pertenecen a esa clase, tales como nombre valor energético, proteínas, carbohidratos, grasas, azúcar. 3. El usuario selecciona de la tabla una de las instancias que desee eliminar y selecciona la opción “Eliminar” 	

4. El sistema muestra un mensaje solicitando la confirmación de la eliminación de la instancia.
5. El usuario selecciona la opción “Si”.
6. El sistema muestra un mensaje de confirmación de la eliminación de la instancia y actualiza la tabla de la zona de instancias y el caso de uso finaliza.

2.3.6. Cargar ontología

ID	ONTO-02
Descripción	Este caso de uso se encarga de la carga inicial de la ontología que forma parte de la base de conocimiento del sistema.
Actor	Administrador
Precondición	El administrador debe haber ingresado al sistema.
Post-condición	El administrador carga la ontología del sistema.

Flujo básico: “Cargar ontología”

6. El administrador selecciona la opción “Cargar ontología”.
7. El sistema muestra un formulario de la sección ontología con el campo dirección el cual hace referencia a la ubicación del archivo en formato Excel que será usado para cargar la ontología.
8. El usuario selecciona la ubicación y elige la opción “Cargar”.
9. El sistema muestra un mensaje de confirmación de la carga de la ontología y el caso de uso finaliza.

2.3.7. Actualizar usuarios

ID	SIS-01
Descripción	Este caso de uso se encarga del registro de usuarios en el sistema.
Actor	Administrador
Precondición	El administrador debe haber ingresado al sistema.
Post-condición	El usuario queda registrado en el sistema.

Flujo básico: “Registrar usuario”

1. El usuario selecciona la opción “Registrar usuario”.
2. El sistema muestra un formulario de la sección usuarios con los campos: Apellido paterno, apellido materno, nombres, especialidad, dirección, teléfono, DNI.
3. El usuario completa los datos y selecciona la opción “Registrar”.
4. El sistema muestra un mensaje de confirmación del registro y el caso de uso finaliza.

Flujo alternativo: “Modificar usuario”

1. El usuario selecciona la opción “Modificar usuario”
2. El sistema muestra un formulario con una sección de búsqueda por los campos

- apellido paterno, apellido materno, nombres y DNI.
3. El usuario realiza la búsqueda.
 4. El sistema muestra las coincidencias de la búsqueda en una tabla.
 5. El usuario selecciona al usuario del sistema que desea modificar.
 6. El sistema completa la sección de datos de usuario.
 7. El usuario modifica los campos que crea convenientes, tales como Apellido paterno, apellido materno, nombres, especialidad, dirección, teléfono y selecciona la opción “Modificar”
 8. El sistema muestra un mensaje de confirmación de la modificación y el caso de uso finaliza.

2.3.8. Generar reportes	
ID	APP-05
Descripción	Este caso de uso se encarga de la generación de reportes.
Actor	Usuario y/o Administrador
Precondición	El usuario debe haber ingresado al sistema.
Post-condición	El usuario genera el reporte que necesita.
Flujo básico: “Generar reportes”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Reportes”. 2. El sistema muestra un formulario de la sección reportes con los campos: tipo de reporte, fecha inicio y fecha fin. Al elegir el tipo de reporte se mostrarán los reportes dependiendo de si se trata del usuario médico o usuario administrador del sistema. 3. El usuario selecciona el tipo de reporte los cuales pueden ser de pacientes por usuario, dietas por paciente para el usuario médico y movimientos de usuarios en el sistema para el usuario administrador, completa las fechas y selecciona la opción “Generar” 4. El sistema muestra el reporte solicitado y el caso de uso finaliza. 	

2.4. Definición de la base de conocimiento

Al tratarse de un sistema que debe conocer todas las propiedades de los distintos tipos de alimentos de forma exacta, rápida y sin errores (aspectos que el experto no maneja, es decir, para realizar esta tarea se debe consultar a una fuente escrita y realizar inspecciones ópticas) el dominio de la base de conocimientos abarca el cubrir todos (o por lo menos la gran mayoría) de los alimentos que forman parte de la alimentación diaria promedio, esto es, las grandes bloques de alimentos que formarán parte de la base de conocimientos son:

- Cereales
- Cereales y preparados
- Lácteos, quesos y huevos
- Azúcares y aceites
- Leguminosas y tubérculos
- Harinas y preparados
- Frutas
- Verduras
- Carnes
- Vísceras y preparados
- Pescados y mariscos

Cada uno de estos bloques posee internamente varias subdivisiones o tipos lo cual nos lleva a tomar la decisión de que una forma de representar el conocimiento sea mediante marcos o frames haciendo uso de una ontología estructurada en el lenguaje owl.

Para la obtención de la sugerencia de dietas, es necesario que estos alimentos presenten ciertas propiedades, las cuales, luego de la conversación con el experto, se indican a continuación:

- Valor energético.
- Proteínas.
- Carbohidratos.
- Grasas.
- Azúcar.

Como parte del conocimiento que se refleja en el sistema, contamos con una serie de ecuaciones que se utilizan para los distintos cálculos, las cuales se detallan en la Tabla 2.4:

Objetivo	Fórmula
Metabolismo basal (Masculino)	$66.47 + 13.75 * \text{peso} + 5 * \text{talla} - 6.75 * \text{edad}$
Metabolismo basal (Femenino)	$655 + 9.56 * \text{peso} + 1.85 * \text{talla} - 4.7 * \text{edad}$
Actividad física diaria	$0.65 * \text{metabolismo_basal}$
Acción dinámica específica	$0.09 * (\text{metabolismo_basal} + \text{actividad_fisica_diaria})$
Energía no absorbida	$0.1 * (\text{metabolismo_basal} + \text{acción_dinamica_especifica} + \text{actividad_fisica_diaria})$
Valor calórico total	$\text{Metabolismo_basal} + \text{actividad_fisica_diaria} + \text{acción_dinamica_especifica} + \text{no_absorvido}$
Índice masa corporal	$\text{Peso_actual} / (\text{talla}^2)$
Peso ideal	$(\text{Talla}^2) * 25$

Tabla 2.4 Fórmulas usadas

2.5. Definición del objetivo de la inferencia que debe realizar el sistema

El objetivo principal del sistema es la de sugerir dietas para los pacientes a tratar y que en base a estas sugerencias pueda llegar a su peso ideal, sin embargo, esto se realizará previo diagnóstico de las dolencias alimenticias que padezca el paciente. Es en este punto donde el sistema, tomando en cuenta los datos ingresados por el usuario, deberá inferir cuál o cuáles son las dolencias alimenticias que padece el paciente y en base a esto lanzará sugerencias predeterminadas a las cuales se les podrá realizar las modificaciones que el usuario requiera.

2.6. Conceptualización del método de inferencia en el que se basará el sistema

El proceso por el cual el sistema identificará cual es la dolencia alimenticia que padece el paciente a tratar pasará por las siguientes etapas:

- El usuario ingresa los datos del paciente a tratar, entre ellos, el nivel de colesterol y azúcar en sangre. Además del cálculo interno del índice de masa corporal.
- Luego, el sistema, usando esta información, y basándose en los objetivos a los cuales espera llegar (identificar que dolencias alimenticias padece el paciente) inferirá si estos datos satisfacen todo lo requerido para que se cumpla el objetivo, ingresando a revisar una serie de reglas IF – THEN – ELSE y tomando en cuenta esto, se obtendrá el diagnóstico.

Como se conceptualiza este proceso, podemos notar que la inferencia usa los datos ingresados por el usuario y con ello revisando cada conclusión (dolencia) se llega a una conclusión, esta es la forma en el cual se basa el método de encadenamiento regresivo, el cual empieza apilando todos los objetivos que se deben ir revisando con los datos ingresados y de esta forma determinar si cumplen o no con las reglas, aquellas conclusiones que sean verdaderas formarán parte del diagnóstico del paciente.

2.7. Análisis del Sistema

En esta parte se detalla la viabilidad del sistema y las restricciones de costo y tiempo.

2.7.1. Viabilidad del sistema

En este punto se detallan la viabilidad técnica y de gestión del tiempo.

- **Viabilidad técnica:**

El sistema será desarrollado utilizando el lenguaje de programación Java e implementando la base de conocimientos en el lenguaje OWL y un motor de base de datos Mysql, con lo cual notamos que estamos bajo el paradigma del Open Source. Al ser implementado en su totalidad en el lenguaje Java, esto nos indica que el sistema será multiplataforma, es decir, podrá ser ejecutado tanto en sistemas operativos Microsoft Windows así como también en sistemas operativos Linux.

En base a esto, podemos notar que no se necesita realizar una fuerte inversión económica para garantizar el desarrollo del proyecto, tan solo basta con cubrir la infraestructura mínima que se requiere para la realización de este proyecto.

Para profundizar más en este estudio, realizamos las siguientes evaluaciones basándonos en los siguientes aspectos:

1. Lenguaje de programación

El proceso de selección de un lenguaje de programación es muy importante debido a que si la elección es equivocada, se podría afectar al tiempo de procesamiento del sistema o puede darse el caso de que se trate de elegir un lenguaje que el desarrollador no conozca, la curva de aprendizaje puede ser muy alto y esto trae consigo el riesgo de no completar el proyecto.

Dada que la arquitectura de este proyecto es cliente – servidor, en la Tabla 2.5 se muestran algunos lenguajes que soportan esta arquitectura.

Característica	Java	Visual Basic	C#
Compatibilidad con desarrollo de ontologías	X		
Conocimiento del lenguaje	X	X	X
Rapidez de ejecución	X	X	X
Documentación disponible	X	X	X
Experiencia cotidiana de desarrollo	X	X	

Tabla 2.5 Comparación de lenguajes de programación

Luego de analizar el cuadro anterior, elegimos en lenguaje Java, debido a que es un lenguaje open source (aún cuando java sun fue últimamente adquirida por Oracle) y se han desarrollado librerías para el manejo de ontologías las cuales se basan en este lenguaje y por ello hacen más fácil es acceso a la información de las mismas. Este lenguaje tiene una curva de aprendizaje muy alta, todos aquellos familiarizados con el lenguaje C ven a java como un lenguaje más sencillo, orientado a objetos y distribuido, indiferente a la arquitectura donde se vaya a usar y portable (JVM – Java Virtual Machine).

Otro aspecto importante de esta elección, es que el ingeniero de conocimiento viene desarrollando proyectos en este lenguaje de manera muy seguida, por lo que su experticia en este lenguaje hace que se pueda desarrollar de manera rápida cualquier tipo de funcionalidad.

[J12]. [DN12]

2. Motor de base de datos

Dado que el lenguaje elegido muestra compatibilidad con la mayoría de motores de base de datos, la elección de alguno de estos pasa más por la experiencia del desarrollador. En la Tabla 2.6 se muestran los motores que pueden ser usados.

Característica	MySQL	SQL Server	Oracle
Sin costos asociados	X		
Fácil configuración	X		X
Fácil instalación	X	X	X
Mayor experiencia	X		X

de uso por parte del desarrollador			
------------------------------------	--	--	--

Tabla 2.6 Comparación de motor de base de datos

Por tal motivo, el motor de base de datos a usar será MySQL, debido a no conlleva costos asociados a su adquisición y posterior instalación. Además, este motor soporta clientes Java lo cual es idóneo para este proyecto, usa GNU autoconf para portabilidad, es capaz de trabajar en distintas plataformas y sistemas operativos distintos, encripta las palabras que viajan a través de la red y es capaz de optimizar y reparar tablas. Otro aspecto a tomar en cuenta es que el ingeniero de conocimiento viene realizando proyectos que amarran al lenguaje java y el motor de base de datos MySQL, por tal motivo, el manejo de este proyecto será más fácil y asegurará, en cierto modo, pocos retrasos en la implementación.

[MSQL12], [MSS12], [ORA12]

- **Viabilidad del tiempo de gestión:**

El proyecto en su totalidad está pensado y estimado para que tenga una duración aproximada de 100 días hábiles, en el cual se cubrirán todos los aspectos que conlleva la adquisición de conocimiento y la construcción del sistema. Dado que en trayecto del desarrollo del proyecto pueden darse problemas que retrasen la entrega final del proyecto, se deberán ir definiendo políticas de gestión y mitigación de riesgos con el fin de evitar que el retraso por estos motivos sea muy prolongado.

El proyecto está estructurado para cumplir el siguiente plan de trabajo mostrado en la Tabla 2.7.

S	Fecha	Compromiso
1	03Sep	Presentación de lista de requerimientos
		Presentación de casos de uso del sistema
2	10Sep	Presentación de especificación de requisitos (ERS)
3	17Sep	Presentación del documento de análisis
		Presentar capítulo 2
4	24Sep	Presentación de primer avance
5	01Oct	Presentación de la arquitectura del sistema
		Presentación de interfaz gráfica
6	08Oct	Feriado
7	15Oct	Semana de exámenes parciales
8	22Oct	Presentación de arquitectura de la base de conocimiento
		Presentación de base de casos
		Presentar capítulo 3
9	29Oct	Presentación del catálogo de pruebas
		Presentar capítulo 4
		Presentación de algoritmos de índices (masa muscular y necesidad calórica)
		Presentación de implementación de funcionalidades del módulo "Aplicación"
10	05Nov	Presentar Capitulo 5: Observaciones
		Presentar Capitulo 5: Conclusiones.
		Presentación de implementación de funcionalidades del

		módulo “Ontología”
11	12Nov	Presentar capítulo 5: Recomendaciones
		Presentar capítulo 5
		Presentación de implementación de funcionalidades del módulo “Sistema”
12	19Nov	Entrega final del documento
13	26Nov	Presentación de implementación de funcionalidades que fueron sugeridos como “Por corregir”
14	03Dic	Semana de exámenes finales
15	10Dic	Semana de exámenes finales
16	20Dic	Entrega final de resultados del proyecto

Tabla 2.7 Estructura del plan de trabajo

• **Viabilidad económica:**

Debido a que el proyecto se desarrollará enteramente bajo el paradigma del Open Source y el software libre, estamos de cierto modo liberándonos de los gastos que conllevan la adquisición de licencias, sin embargo, se tendrán costos por los dispositivos físicos a ser usados así como también el costo en tiempo que conlleva el desarrollo de este proyecto.

En la Tabla 2.8 se muestran los principales egresos que conllevará el desarrollo de este proyecto.

Concepto	Subtotal	Criterio
Mano de obra		
Levantamiento de información	1,800.00	12 días a 10 horas por día
Tiempo aplicado al proyecto	12,600.00	14 semanas, aproximadamente 6 días por semana a 10 horas por día.
Otros gastos		
Luz, internet	500.00	
Artículos de oficina	120.00	
Total aproximado	15,020.00	

Tabla 2.8 Egresos y estimación de costos

Actualmente, el costo de horas hombre del desarrollador está estimado en 15 soles por hora de trabajo.

Por todo lo expuesto anteriormente y luego del análisis de viabilidad, podemos decir que si es factible la implementación del sistema basado en conocimientos expuesto en este proyecto de tesis.

3. Diseño

En este capítulo se definirá la arquitectura de la solución, la formalización de la estructura de la base de conocimiento y el diseño de interfaz gráfica del sistema.

3.1. Arquitectura de la solución

En esta sección detallaremos el método que seguirá el motor de inferencia para obtener el diagnóstico de las dolencias que padece el paciente a tratar.

Como se mencionó en el capítulo anterior, el método a usar será la del encadenamiento regresivo, el cual, tomando como punto de partida una serie de datos ingresados por el usuario y los objetivos que se espera determinar si cumplen o no con sus reglas, recorrerá las reglas definidas e irá extrayendo las conclusiones válidas a partir de hechos válidos.

Los pasos que debe seguir el proceso de diagnóstico es el siguiente:

1. Se determina un objetivo de la lista de dolencias.
2. Se comparan sus reglas con los hechos ingresados por el usuario.
3. En caso se satisfagan todas las reglas, el objetivo se toma como verdad y pasa a formar parte del diagnóstico del paciente.
4. Se descarta este objetivo y se pasa al paso 1.

A continuación, un ejemplo del proceso de diagnóstico.

Se tienen las siguientes reglas:

- **SI** DIAGNÓSTICO = INFRAPESO **ENTONCES** SEXO = MASCULINO **O** SEXO = FEMENINO **Y** INDICE DE MASA CORPORAL (IMC) < 18.5
- **SI** DIAGNÓSTICO = PESO NORMAL **ENTONCES** SEXO = MASCULINO **O** SEXO = FEMENINO **Y** INDICE DE MASA CORPORAL (IMC) >= 18.5 **Y** INDICE DE MASA CORPORAL (IMC) <= 24.9

Los pasos para obtener el diagnóstico para este punto por ejemplo, se muestran en la Tabla 3.1 y serían los siguientes:

PASO	DESARROLLO
Se determina un objetivo de la lista de dolencias	En este punto, por ejemplo, tomamos el objetivo INFRAPESO, la lista consiste en todas las posibles dolencias que están en VERDAD, luego, se van a ir extrayendo una a una para determinar la verdad o falsedad, dependiendo de los hechos que ingrese el usuario y de la comparación con las reglas.

Se comparan las reglas con los hechos ingresados por el usuario	En este punto, y suponiendo que el usuario ingresó SEXO = MASCULINO y el IMC obtenido por el sistema fue 17, entonces se verifica la primera regla, dado que DIAGNOSTICO es verdadero, se pasa a verificar las condiciones, en este caso, todo es verdadero.
En caso se satisfaga la regla, el objetivo se toma como verdad y pasa a formar parte del diagnóstico del paciente	En este punto, se concluye que el paciente tiene INFRAPESO.
Se descarta el objetivo y se pasa al paso 1	En este punto, se pasará a evaluar el siguiente objetivo de la lista.

Tabla 3.1 Etapas del diagnóstico

Luego de que el proceso de diagnóstico esté completo, el sistema mostrará una lista de alimentos predeterminado dependiendo de cuáles son las dolencias que padece, la misma que podrá ser modificada de acuerdo a la realidad de consumo del paciente. La información de los aportes de cada alimento que será usado para la obtención de la cantidad en gramos que se debe ingerir por el paciente, se encuentra almacenada en la ontología de alimentos en forma de un archivo con extensión .owl la cual forma parte de la base de conocimientos. El acceso a esta información se hizo mediante librerías especializadas en el manejo de este tipo de archivos.

La estructura de interacción se muestra en la Ilustración 3.1:

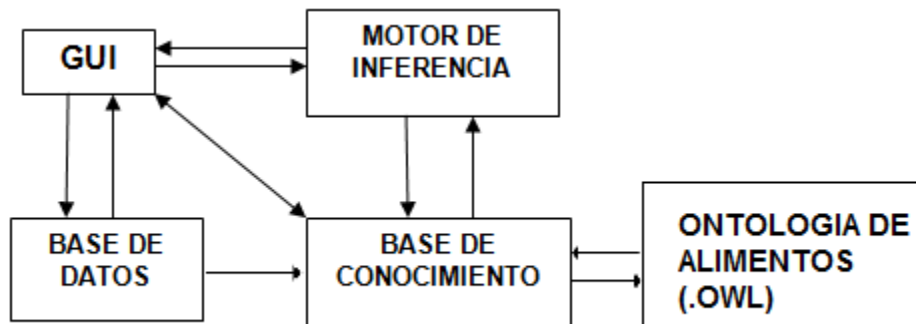


Ilustración 3.1 Estructura de interacción

3.2. Diseño de interfaz gráfica

A continuación, se indican los criterios que se tomaron en cuenta para el diseño de la interfaz gráfica del sistema:

3.2.1. Criterios utilizados

Dado que, la puerta de comunicación entre el usuario y el sistema es el entorno gráfico, este aspecto es muy importante. El sistema debe contar con un diseño gráfico que sea atractivo a la vista del usuario y fácil de entender, con ello estaríamos

garantizando, de alguna manera, que el usuario quiera usar el sistema. Los criterios que se toman en cuenta para el diseño son los siguientes:

1. **Agrupación:** Toda la información que se muestre en el sistema, debe estar correctamente agrupada, de esta forma, el usuario podrá acceder rápidamente a ella.
2. **Simetría:** Todos los componentes de las pantallas conservan una alineación tanto a nivel horizontal como vertical.
3. **Claridad:** Todas las funcionalidades que se exponen al usuario deben ser fáciles de comprender, de tal manera que no se preste a confusión.
4. **Secuencia:** El flujo de información mostrado al usuario debe ser ordenado, de esta forma, el usuario podrá llegar a la opción y/o información que necesite de forma rápida.

La mayoría de pantallas del sistema, cumplen con un estándar similar al que se muestra en la Ilustración 3.2:

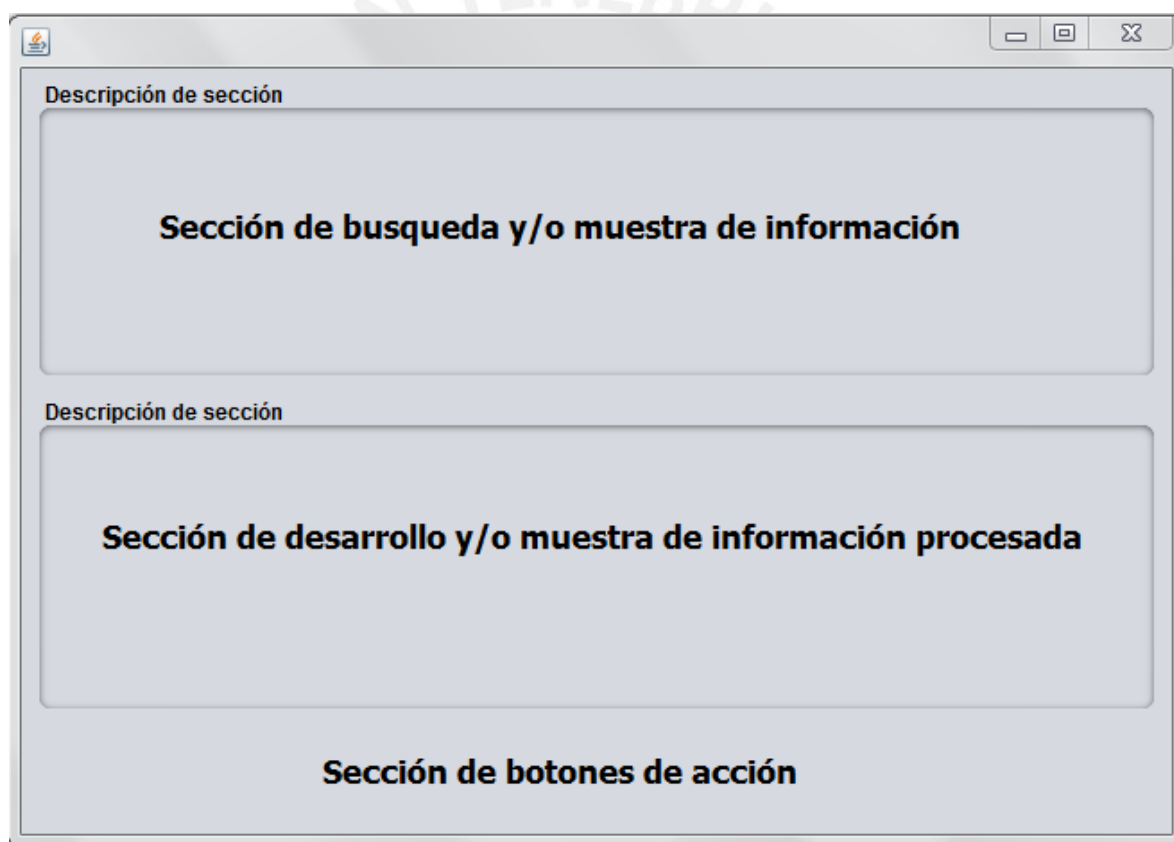


Ilustración 3.2 Estándar de pantallas

Algunos aspectos que forman parte del estándar de pantallas se mencionan a continuación:

- Tipo de letra: Tahoma.
- Tamaño de letra: 11.
- Color: Negro.

- Botones de acción: Ingresar, Grabar, Modificar, Exportar, Generar, Limpiar, Cancelar, Salir.




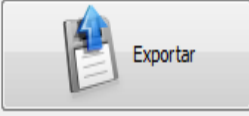

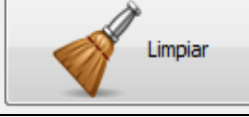

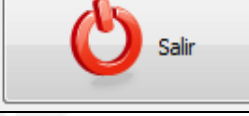
Botón	Imagen
Ingresar	 Ingresar
Grabar	 Grabar
Modificar	 Modificar
Exportar	 Exportar
Generar	 Generar
Limpiar	 Limpiar
Cancelar	 Cancelar
Salir	 Salir

Tabla 3.2 Botones del sistema

- Mensajes: Error, advertencia, confirmación.

Mensaje	Imagen
Error	
Advertencia	
Confirmación	

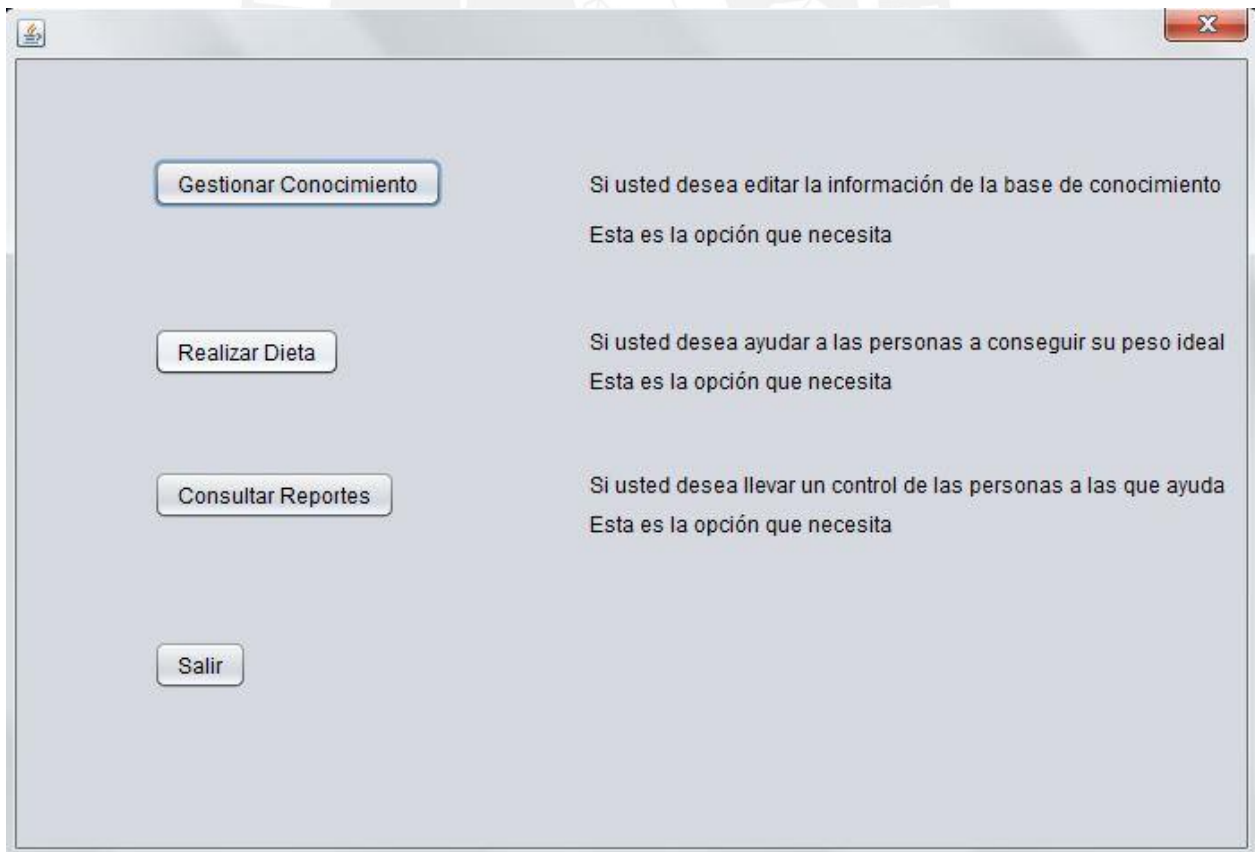
Tabla 3.3 Mensajes del sistema

A continuación, los principales prototipos de pantallas a ser usadas en el sistema:



The screenshot shows a login window titled "SMART NUTRITION". It features a central title in large, bold, black letters. Below the title, there are two input fields: "Usuario:" followed by a text box, and "Contraseña:" followed by a text box. At the bottom, there are three buttons: "Ingresar", "Limpiar", and "Salir".

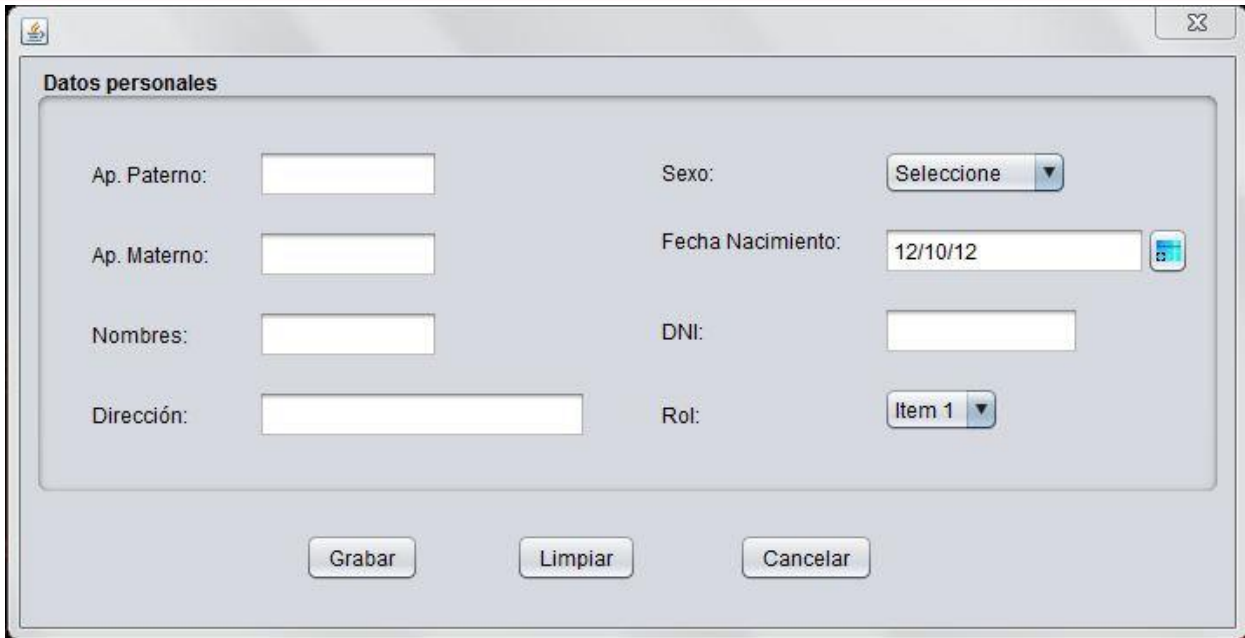
Ilustración 3.3 Pantalla de logueo



The screenshot shows the main menu of the "SMART NUTRITION" application. It contains four buttons on the left side, each with a corresponding description on the right:

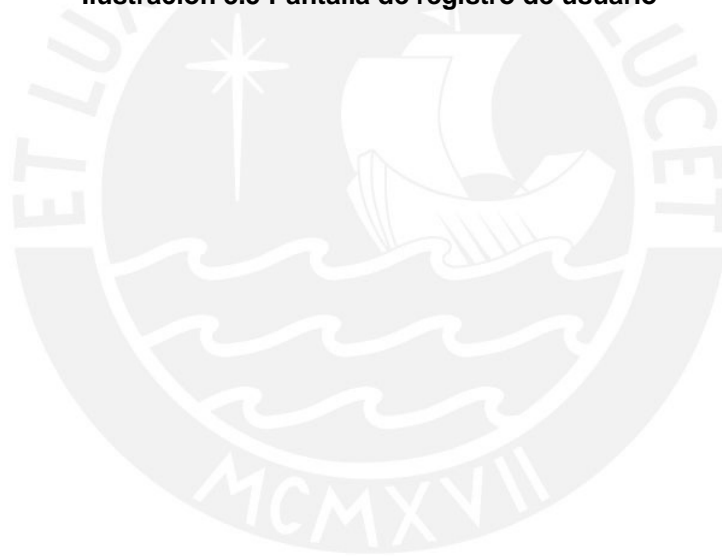
- Gestionar Conocimiento**: Si usted desea editar la información de la base de conocimiento Esta es la opción que necesita
- Realizar Dieta**: Si usted desea ayudar a las personas a conseguir su peso ideal Esta es la opción que necesita
- Consultar Reportes**: Si usted desea llevar un control de las personas a las que ayuda Esta es la opción que necesita
- Salir**: (No description provided)

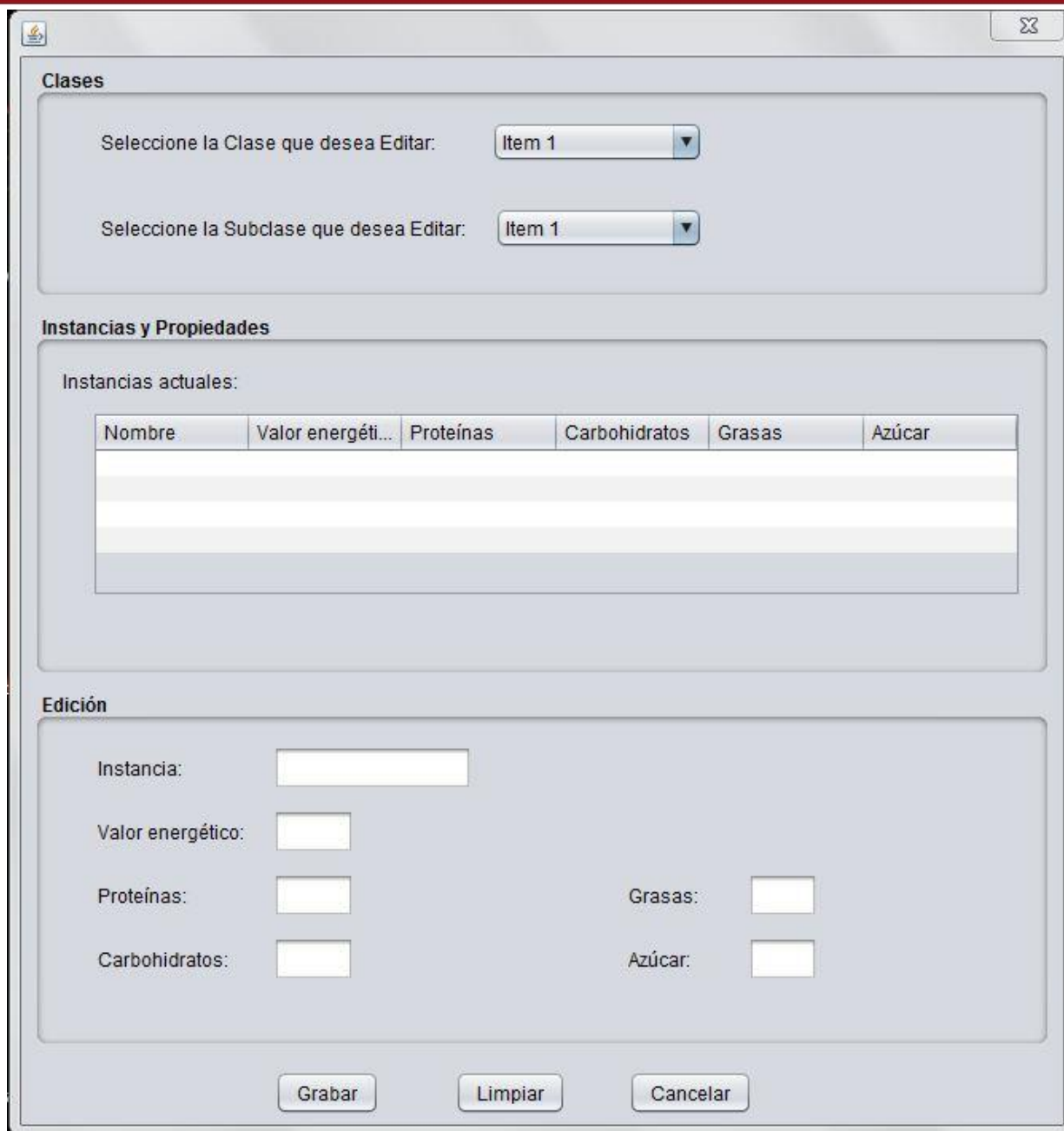
Ilustración 3.4 Pantalla principal



The image shows a screenshot of a web application window titled "Datos personales". The window contains several input fields and dropdown menus for user registration. The fields are arranged in two columns. The first column includes "Ap. Paterno:", "Ap. Materno:", "Nombres:", and "Dirección:". The second column includes "Sexo:" (with a dropdown menu showing "Seleccione"), "Fecha Nacimiento:" (with a date input field showing "12/10/12" and a calendar icon), "DNI:", and "Rol:" (with a dropdown menu showing "Item 1"). At the bottom of the window, there are three buttons: "Grabar", "Limpiar", and "Cancelar".

Ilustración 3.5 Pantalla de registro de usuario



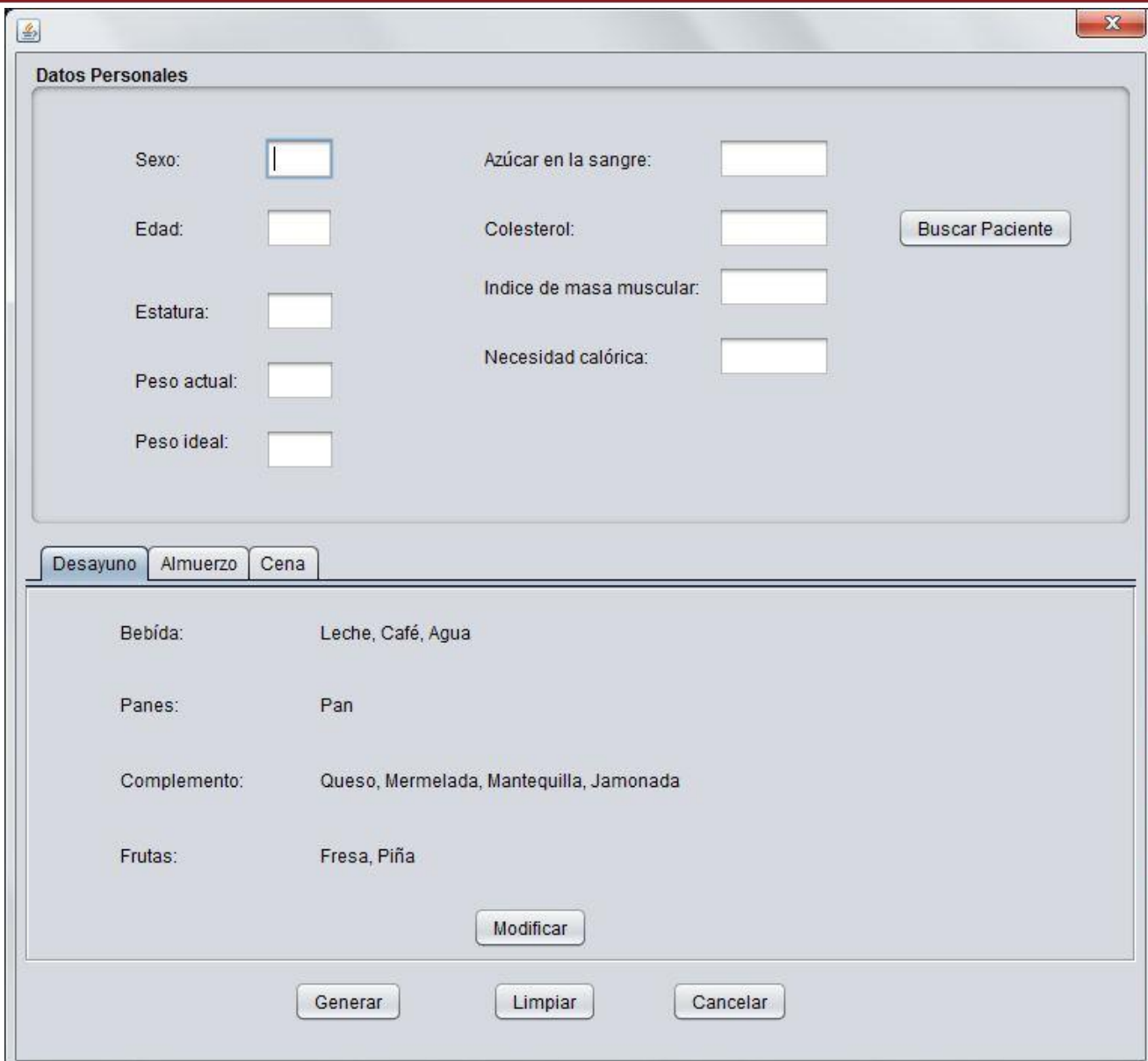


The screenshot shows a software window titled "Clases" with a close button in the top right corner. It is divided into three main sections:

- Clases:** Contains two dropdown menus. The first is labeled "Seleccione la Clase que desea Editar:" and has "Item 1" selected. The second is labeled "Seleccione la Subclase que desea Editar:" and also has "Item 1" selected.
- Instancias y Propiedades:** Labeled "Instancias actuales:", it contains a table with six columns: "Nombre", "Valor energéti...", "Proteínas", "Carbohidratos", "Grasas", and "Azúcar". The table body is currently empty.
- Edición:** Contains several input fields for editing an instance:
 - "Instancia:" followed by a text input field.
 - "Valor energético:" followed by a numeric input field.
 - "Proteínas:" followed by a numeric input field.
 - "Carbohidratos:" followed by a numeric input field.
 - "Grasas:" followed by a numeric input field.
 - "Azúcar:" followed by a numeric input field.

At the bottom of the window, there are three buttons: "Grabar", "Limpiar", and "Cancelar".

Ilustración 3.6 Pantalla de edición de ontología



Datos Personales

Sexo: Azúcar en la sangre:

Edad: Colesterol:

Estatura: Índice de masa muscular:

Peso actual: Necesidad calórica:

Peso ideal:

Bebida: Leche, Café, Agua

Panes: Pan

Complemento: Queso, Mermelada, Mantequilla, Jamonada

Frutas: Fresa, Piña

Ilustración 3.7 Pantalla sugerencia estándar de dieta

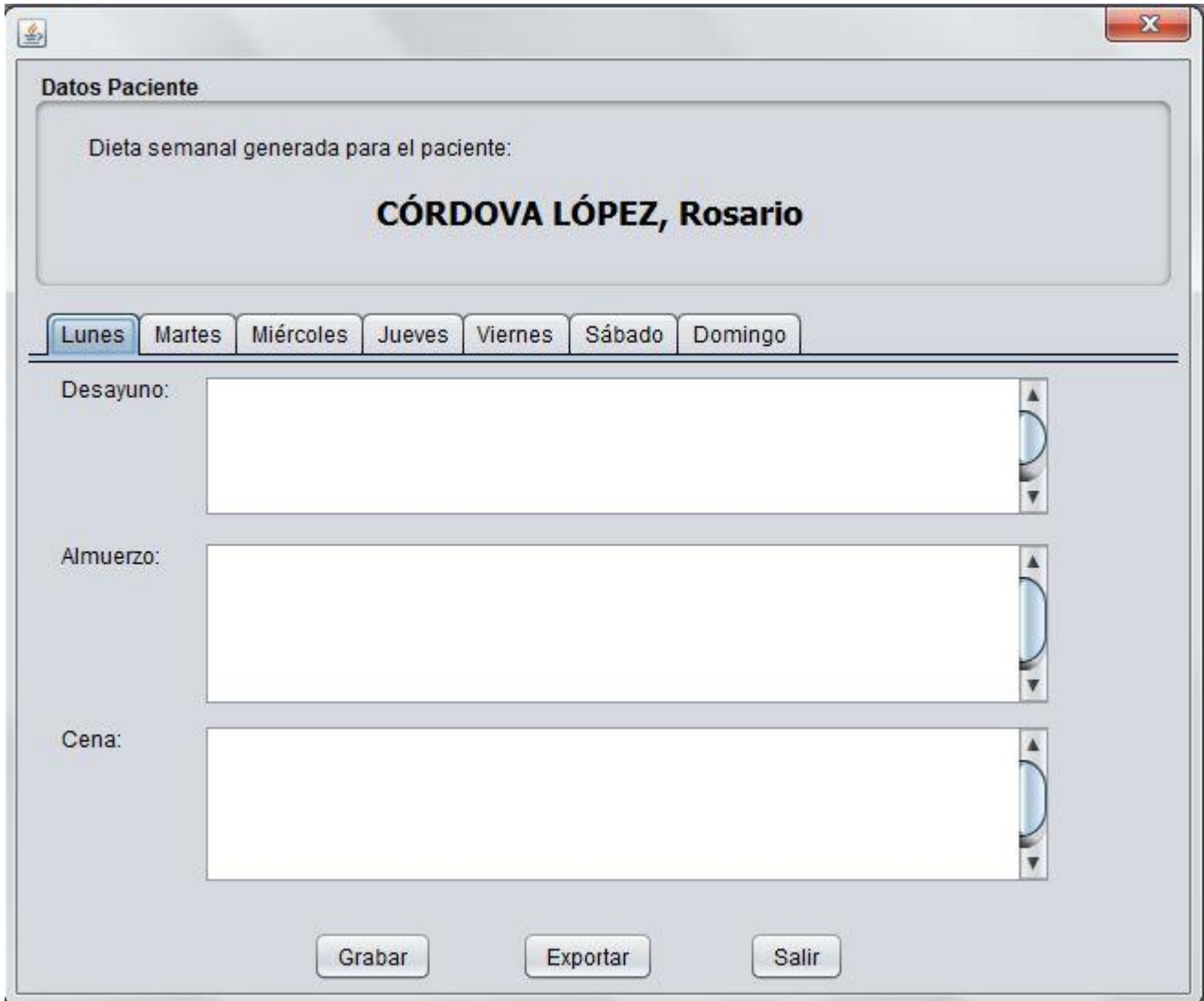


Ilustración 3.8 Pantalla de sugerencia personalizada de dieta

3.3. Arquitectura de información

En esta sección, indicaremos la arquitectura de información usada para la base de conocimiento y la base de datos del sistema.

La base de conocimiento contiene dos aspectos, la información para generar el diagnóstico de dolencias alimenticias y la ontología que contiene toda la estructura de alimentos y sus componentes nutricionales, relevantes para el proceso de generación de dietas.

3.3.1. Base de conocimiento

La arquitectura que se usa en este punto, se divide en dos partes:

1. Ontología

La ontología del dominio de los alimentos y sus clasificaciones, están estructuradas en el lenguaje OWL (Web Ontology Language), el cual posee una estructura similar a la del XML.

A continuación un ejemplo de la estructura de almacenamiento de información de alimentos mencionada:

```
<rdf:Description rdf:about="alimentos:cafe_instantaneo">
  <j.0:Azucar
rdf:datatype="http://www.w3.org/2001/XMLSchema#double">0.0</j.0:Azucar>
  <j.0:Carbohidratos
rdf:datatype="http://www.w3.org/2001/XMLSchema#double">41.3</j.0:Carbohidratos>
  <j.0:Grasas
rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.8</j.0:Grasas>
  <j.0:Proteinas
rdf:datatype="http://www.w3.org/2001/XMLSchema#double">12.5</j.0:Proteinas>
  <j.0:Calorias
rdf:datatype="http://www.w3.org/2001/XMLSchema#double">130.0</j.0:Calorias>
  <rdf:type rdf:resource="alimentos:varios"/>
</rdf:Description>
```

En la sección Anexos, se encuentra un extracto de la estructura completa de la ontología a usar.

2. Diagnóstico de dolencias alimenticias

Para esta sección, la estructura a utilizar será la del uso de reglas, con ellas, se indica, dada una dolencia a descartar, se deben cumplir con una serie de hechos, los cuales, de ser verdaderos, implicarán que dicha dolencia es padecida por el paciente. La estructura que siguen estas reglas es la siguiente:

SI conclusión_supuesta **ENTONCES** premisa **Y** premisa **Y** premisa

En la sección Anexos, se encuentran todas las reglas a ser usadas por el sistema para el proceso de diagnóstico.

3.3.2. Base de datos

La base de datos del sistema almacenará la información de los usuarios, pacientes, dolencias diagnosticadas a los pacientes y las dietas sugeridas por el sistema para cada paciente. Además, se cuenta con una tabla que contiene toda la información que será usada para el proceso de diagnóstico y para la actualización de los coeficientes de las fórmulas que dormán parte de la base de conocimiento. En la sección Anexos, se muestra la base de datos del sistema.

4. Construcción y pruebas

En este capítulo se dará a conocer la tecnología que será aplicada en el desarrollo del sistema así como las pruebas por las que se deberá someter el mismo para garantizar su correcto funcionamiento.

4.1. Construcción.

Para la construcción del sistema, se utilizarán las siguientes tecnologías:

4.1.1. Lenguaje de programación.

El lenguaje de programación seleccionado para este proyecto fue JAVA. Las características principales que llevaron a la elección de este lenguaje para el desarrollo del sistema se detallan a continuación:

- Java es un lenguaje indiferente a la arquitectura, las aplicaciones desarrolladas en este lenguaje pueden ser ejecutadas en distintos sistemas operativos ya que el compilador de java genera bytecodes, que son formatos intermedios independientes de la arquitectura que están diseñados para transportar código de manera eficiente a múltiples plataformas hardware y software.
- Java es un lenguaje robusto, diseñado para crear software altamente fiable, libera al programador de los errores provenientes de la aritmética de punteros y posee la denominada recolección de basura, la cual libera al programador de realizar liberación explícita de memoria.
- Existe mucha documentación en línea sobre este lenguaje, lo cual hace más simple su uso y consulta en caso de problemas en la etapa de desarrollo.

[J12]

4.1.2. Librerías a ser usadas.

Las librerías a ser usadas en este proyecto fueron desarrolladas por terceros en el lenguaje JAVA.

- Librerías JAVA: en la Tabla 4.1, las librerías usadas en el proyecto.

Librería	Descripción	Funcionalidad
com.hp.hpl.jena	Permite el manejo de ontologías mediante código en el lenguaje java.	Se usa para la sección de creación y mantenimiento de la ontología de alimentos.
com.hp.hpl.jena.ontology.OntModel	Permite crear el modelado de una ontología.	Se usa en la sección de creación y mantenimiento d la ontología

		de alimentos.
com.hp.hpl.jena.ontology.OntClass	Permite crear clases ontológicas.	Se usa en la sección de creación y mantenimiento d la ontología de alimentos.
com.hp.hpl.jena.ontology.Individual	Permite crear instancias de las clases ontológicas.	Se usa en la sección de creación y mantenimiento d la ontología de alimentos.
com.hp.hpl.jena.ontology.DatatypeProperty	Permite agregar propiedades a las clases ontológicas.	Se usa en la sección de creación y mantenimiento d la ontología de alimentos.
datechooser.beans.DateChooserCombo	Permite el uso de calendarios de fechas.	Se usa en las secciones donde se necesite capturar fechas en el sistema.
jxl	Permite la lectura de archivos en formato .xls	Se usa en la sección de carga inicial de la ontología.

Tabla 4.1 Librerías usadas en el sistema

4.1.3. Base de datos.

La base de datos seleccionada para este proyecto fue MySQL. Al usar este motor de base de datos, se tienen ventajas como la velocidad al realizar operaciones, bajo costo en requerimientos para la elaboración de bases de datos, esto debido a su bajo consumo, puede ser ejecutado en máquinas con recursos escasos.

MySQL es un gestor de base de datos relacional que provee acceso multiusuario a las bases de datos. Este motor de base de datos es uno de los más usados a nivel mundial.

Provee funcionalidades tales como procedimientos almacenados, disparadores, schemas, soporte de Unicode, transacciones y soporte multiplataforma.

Existe mucha documentación en línea sobre la instalación, configuración y funcionalidades, además de interacción entre diferentes aplicaciones que son desarrollados en lenguajes de programación diversos, lo que hace que su uso sea más sencillo.

[MSQL12]

4.1.4. Entorno de desarrollo.

El entorno de desarrollo seleccionado para este proyecto fue NetBeans IDE en su versión 7.0.1 el cual es una herramienta que permite escribir, depurar, compilar y ejecutar programas en diferentes lenguajes de programación.

Este entorno de desarrollo será usado para trabajar en el lenguaje Java, para el cual brinda facilidades al momento de iniciar un proyecto en este lenguaje, tales como auto compleción de código y la integración con MySQL.

Debido a su fácil instalación, su interfaz intuitiva, la experiencia de uso por parte del ingeniero de conocimiento y por ser un producto bastante estable, es que se eligió este entorno de desarrollo.

Existe mucha documentación en línea sobre el cómo usar este entorno, como realizar instalación de pluggins, asociar librerías externas al proyecto en curso así como también el fácil manejo de las herramientas gráficas al momento de diseñar las vistas del sistema. [NETB12]

4.1.5. Estándares de programación.

Para que durante el desarrollo del sistema se genere un código fuente claro y fácil de mantener, se definen una serie de estándares que garanticen que cualquier persona lo pueda entender.

A continuación se detallan los estándares de programación usados en este sistema:

1. Clases.

La definición de las clases será con la primera letra en mayúscula, el resto del cuerpo del nombre en minúsculas y seguidos del identificador del tipo de clase, así por ejemplo, en la Tabla 4.2:

Tipo de clase	Nombre de clase
Entidad	Usuario
Lógica de negocio	UsuarioBL
Acceso a datos	UsuarioDAO

Tabla 4.2 Clases del sistema

2. Métodos.

Para garantizar el entendimiento de la funcionalidad que se desea implementar con un método, este debe tener un nombre intuitivo escrito en minúsculas pero, en caso de que sea un nombre compuesto, la primera letra de la siguiente palabra debe ir en mayúscula y todo junto, así por ejemplo, en la Tabla 4.3:

Métodos
buscarPaciente()
cargarOntología()

Tabla 4.3 Métodos del sistema

3. Declaración de objetos.

Los objetos en el sistema serán declarados anteponiendo a su nombre la abreviatura en minúsculas obj, así por ejemplo, en la Tabla 4.4:

Declaración de objetos	
objUsuario	
objPaciente	

Tabla 4.4 Objetos del sistema

4. Componentes gráficos.

Estos componentes tendrán como nombre de variable la composición del prefijo del tipo de componente seguido del nombre, así por ejemplo:

txtNombreUsuario.

A continuación, en la Tabla 4.5, la lista de prefijos de los componentes usados en el sistema:

Componente	Prefijo
TextBox	txt
ComboBox	cmb
Table	tbl
Button	btn
Date time picker	dtp

Tabla 4.5 Prefijos para componentes del sistema

4.2. Pruebas.

Esta sección es importante para el sistema ya que con las pruebas podremos encontrar errores y corregirlos, de esta forma, logramos un producto funcionalmente correcto.

Una vez que la implementación de las funcionalidades este completa, se realizarán las pruebas y, en caso amerite, se definirán las correcciones sobre las mismas.

Las pruebas que se realizarán en el sistema son las siguientes:

4.2.1. Tipos de pruebas

Las pruebas que se realizaran en el sistema son las siguientes:

1. Pruebas unitarias.

La finalidad de este tipo de pruebas es la de garantizar que los flujos del sistema funcionen correctamente.

Las pruebas usadas se detallan en la Tabla 4.6:

Código	Descripción de la prueba
PU001	Verificar que se puede crear y modificar usuarios en el sistema
PU002	Verificar que se puede crear y modificar pacientes en el sistema
PU003	Verificar que se puede crear y modificar alimentos en la ontología usada en el sistema
PU004	Verificar que se puede iniciar sesión en el sistema

PU005	Verificar que se puede generar un diagnóstico para el paciente
PU006	Verificar que se puede generar y modificar la sugerencia de dieta obtenida para un paciente
PU007	Verificar que se puede generar reportes por paciente, medico, diagnóstico y sugerencias de dietas
PU008	Verificar que el sistema calcule los índices que servirán para el diagnóstico
PU009	Verificar que se puede realizar la carga inicial de la ontología

Tabla 4.6 Pruebas unitarias

2. Pruebas del sistema.

La finalidad de este tipo de pruebas es la de validar el desempeño a nivel tecnológico y funcional del sistema.

Las pruebas usadas se muestran a continuación en la Tabla 4.7:

Código	Descripción de prueba
PS001	Verificar que el sistema tenga un tiempo de respuesta no mayor a 15 segundos

Tabla 4.7 Pruebas del sistema

3. Pruebas con el usuario experto.

Para definir si los resultados finales que genera el sistema son correctos, lo sometemos a pruebas con resultados que obtiene el usuario experto, en caso la diferencia es muy notoria, se procede a hacer las correcciones que nos lleven a resultados muy similares con los obtenidos por el usuario.

Las pruebas usadas en este punto se detallan en la Tabla 4.8:

Código	Descripción de prueba
PE001	Verificar que los resultados de la obtención de índices sean los mismos que obtiene el usuario realizándolo tal cual lo hace en su labor diaria.
PE002	Verificar que los resultados del diagnóstico sean los mismos a los que llegaría el usuario experto teniendo en cuenta los hechos ingresados al sistema.
PE003	Verificar que los valores de distribución de los alimentos sean coherentes y similares con los que sugeriría el usuario experto en su labor diaria.

Tabla 4.8 Pruebas con usuario experto

4.2.2. Resultado de las pruebas.

Las pruebas fueron realizadas por el ingeniero de conocimiento. El resultado de las pruebas que se realizaron al sistema se muestra a continuación:

- PU001: Verificar que se puede crear y modificar usuarios en el sistema.
 - a) Campo de entrada: Ap. Paterno

Clases de equivalencia	Valores	Identificador
CV	Caracteres alfanuméricos	1
CNV	Caracteres alfanuméricos no	2
CNV	Vacío	3

- b) Campo de entrada: Ap. Materno

Clases de equivalencia	Valores	Identificador
CV	Caracteres alfanuméricos	4
CNV	Caracteres alfanuméricos no	5
CNV	Vacío	6

- c) Campo de entrada: Nombres

Clases de equivalencia	Valores	Identificador
CV	Caracteres alfanuméricos	7
CNV	Caracteres alfanuméricos no	8
CNV	Vacío	9

- d) Campo de entrada: DNI

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	10
CNV	Vacío	11

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1,4,7,10	Paterno,	Se muestra	OK

		Materno, Nombre, 12345678	mensaje de confirmación.	
2	2, 4, 7, 10	%#\$, Materno, Nombre, 12345678	Se muestra el mensaje: Carácter no válido, corrija	NO OK
3	3, 4, 7, 10	“, Materno, Nombre, 12345678	Se muestra el mensaje: todos los campos son obligatorios	NO OK
4	1,5,7,10	Paterno, %#\$, Nombre, 12345678	Se muestra el mensaje: Carácter no válido, corrija.	NO OK
5	1, 6, 7, 10	Paterno, “”, Nombre, 12345678	Se muestra el mensaje: todos los campos son obligatorios	NO OK
6	1, 4, 8, 10	Paterno, Materno, %#\$, 12345678	Se muestra el mensaje: Carácter no válido, corrija.	NO OK
7	1, 4, 9, 10	Paterno, Materno, “”, 12345678	Se muestra el mensaje: todos los campos son obligatorios	NO OK
8	1, 4, 7, 11	Paterno, Materno, Nombre, “”	Se muestra el mensaje: todos los campos son obligatorios	NO OK

Tabla 4.9 Casos de prueba PU001

- PU002: Verificar que se puede crear y modificar pacientes en el sistema.
 - a) Campo de entrada: Ap. Paterno

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	1
CNV	Valores no alfanuméricos	2
CNV	Vacío	3

- b) Campo de entrada: Ap. Materno

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	4
CNV	Valores no alfanuméricos	5
CNV	Vacío	6

c) Campo de entrada: Nombres

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	7
CNV	Valores no alfanuméricos	8
CNV	Vacío	9

d) Campo de entrada: DNI

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	10
CNV	Vacío	11

e) Campo de entrada: Peso

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	12
CNV	Vacío	13

f) Campo de entrada: Talla

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	14
CNV	Vacío	15

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1, 4, 7, 10, 12, 14	Paterno, Materno, Nombre, 12345678, 50.2, 1.70	Se muestra mensaje de confirmación	OK
2	2, 4, 7, 10, 12, 14	%&#, Materno, Nombre, 12345678,	Se muestra el mensaje: Carácter no válido, corrija.	NO OK

		50.2, 1.70		
3	3, 4, 7, 10, 12, 14	“, Materno, 12345678, 50.2, 1.70	Se muestra el mensaje: todos los campos son obligatorios	NO OK
4	1, 5, 7, 10, 12, 14	Paterno, %&#, Nombre, 12345678, 50.2, 1.70	Se muestra el mensaje: Carácter no válido, corrija.	NO OK
5	1, 6, 7, 10, 12, 14	Paterno, “”, Nombre, 12345678, 50.2, 1.70	Se muestra el mensaje: todos los campos son obligatorios	NO OK
6	1, 4, 8, 10, 12, 14	Paterno, Materno, %&#, 12345678, 50.2, 1.70	Se muestra el mensaje: Carácter no válido, corrija.	NO OK
7	1, 4, 9, 10, 12, 14	Paterno, Materno, “”, 12345678, 50.2, 1.70	Se muestra el mensaje: todos los campos son obligatorios	NO OK
8	1, 4, 7, 11, 12, 14	Paterno, Materno, Nombre, “”, 50.2, 1.70	Se muestra el mensaje: todos los campos son obligatorios	NO OK
9	1, 4, 7, 10, 13, 14	Paterno, Materno, Nombre, 12345678, “”, 1.70	Se muestra el mensaje: todos los campos son obligatorios	NO OK
10	1, 4, 7, 10, 12, 15	Paterno, Materno, Nombre, 12345678, 50.2, “”	Se muestra el mensaje: todos los campos son obligatorios	NO OK

Tabla 4.10 Casos de prueba PU002

- PU003: Verificar que se puede crear y modificar alimentos en la ontología usada en el sistema.
 - a) Campo de entrada: Instancia

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	1

CNV	Valores alfanuméricos	no	2
CNV	Vacío		3

b) Campo de entrada: Valor energético

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	4
CNV	Vacío	5

c) Campo de entrada: Proteínas

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	6
CNV	Vacío	7

d) Campo de entrada: Carbohidratos

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	8
CNV	Vacío	9

e) Campo de entrada: Grasas

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	10
CNV	Vacío	11

f) Campo de entrada: Azúcar

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	12
CNV	Vacío	13

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1, 4, 6, 8, 10, 12	Instancia, 1.0, 2.0, 3.0, 4.0, 5.0	Se muestra mensaje de confirmación	OK
2	2, 4, 6, 8, 10, 12	%&#, 1.0, 2.0, 3.0, 4.0, 5.0	Se muestra el mensaje: Carácter no válido, corrija.	NO OK

3	3, 4, 6, 8, 10, 12	“, 1.0, 2.0, 3.0, 4.0, 5.0	Se muestra el mensaje: todos los campos son obligatorios	NO OK
4	1, 5, 6, 8, 10, 12	Instancia, “”, 2.0, 3.0, 4.0, 5.0	Se muestra el mensaje: todos los campos son obligatorios	NO OK
5	1, 4, 7, 8, 10, 12	Instancia, 1.0, “”, 3.0, 4.0, 5.0	Se muestra el mensaje: todos los campos son obligatorios	NO OK
6	1, 4, 6, 9, 10, 12	Instancia, 1.0, 2.0, “”, 4.0, 5.0	Se muestra el mensaje: todos los campos son obligatorios	NO OK
7	1, 4, 6, 8, 11, 12	Instancia, 1.0, 2.0, 3.0, “”, 5.0	Se muestra el mensaje: todos los campos son obligatorios	NO OK
8	1, 4, 6, 8, 10, 13	Instancia, 1.0, 2.0, 3.0, 4.0, “”	Se muestra el mensaje: todos los campos son obligatorios	NO OK

Tabla 4.11 Casos de prueba PU003

- PU004: Verificar que se puede iniciar sesión en el sistema.
 - a) Campo de entrada: Usuario

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	1
CNV	Valores no alfanuméricos	2
CNV	Vacío	3

- b) Campo de entrada: Contraseña

Clases de equivalencia	Valores	Identificador
CV	Valores alfanuméricos	4
CV	Valores no alfanuméricos	5
CNV	Vacío	6

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1, 4	Usuario, pass	Se cierra la ventana de logueo y se abre la ventana principal	OK
2	1, 5	Usuario, %&#	Se cierra la ventana de logueo y se abre la ventana principal	OK
3	2, 4	%&#, pass	Se muestra el mensaje: carácter incorrecto, corrija	NO OK
4	3, 4	“”, pass	Se muestra el mensaje: todos los campos son obligatorios	NO OK
5	1, 6	Usuario, “”	Se muestra el mensaje: todos los campos son obligatorios	NO OK

Tabla 4.12 Casos de prueba PU004

- PU005: Verificar que se puede generar un diagnóstico para el paciente.
 - Campo de entrada: Sexo

Clases de equivalencia	Valores	Identificador
CV	Valor M o F	1
CNV	Otros valores	2
CNV	Vacío	3

- Campo de entrada: Edad

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	4
CNV	Vacío	5

c) Campo de entrada: Talla

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	6
CNV	Vacio	7

d) Campo de entrada: Peso actual

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	8
CNV	Vacio	9

e) Campo de entrada: Azúcar en sangre

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	10
CNV	Vacio	11

f) Campo de entrada: Colesterol

Clases de equivalencia	Valores	Identificador
CV	Valores numéricos	12
CNV	Vacio	13

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1, 4, 6, 8, 10, 12	M, 20, 1.1, 1.2, 1.3, 1.4	Se autocompleta la zona de diagnóstico	OK
2	2, 4, 6, 8, 10, 12	P, 20, 1.1, 1.2, 1.3, 1.4	Se muestra mensaje de error: Genero incorrecto	NO OK
3	3, 4, 6, 8, 10, 12	“, 20, 1.1, 1.2, 1.3, 1.4	Se muestra el mensaje: todos los campos son obligatorios	NO OK
4	1, 5, 6, 8, 10, 12	M, “”, 1.1, 1.2, 1.3, 1.4	Se muestra el mensaje: todos los campos son obligatorios	NO OK

5	1, 4, 7, 8, 10, 12	M, 20, "", 1.2, 1.3, 1.4	Se muestra el mensaje: todos los campos son obligatorios	NO OK
6	1, 4, 6, 9, 10, 12	M, 20, 1.1, "", 1.3, 1.4	Se muestra el mensaje: todos los campos son obligatorios	NO OK
7	1, 4, 6, 8, 11, 12	M, 20, 1.1, 1.2, "", 1.4	Se muestra el mensaje: todos los campos son obligatorios	NO OK
8	1, 4, 6, 8, 10, 13	M, 20, 1.1, 1.2, 1.3, ""	Se muestra el mensaje: todos los campos son obligatorios	NO OK

Tabla 4.13 Casos de prueba PU005

- PU006: Verificar que se puede generar y modificar la sugerencia de dieta obtenida para un paciente.
En esta prueba, se deben elegir que alimentos formarán parte de la dieta semanal, se debe elegir por lo menos un alimento de los distintos tipos.
- PU008: Verificar que el sistema calcule los índices que servirán para el diagnóstico.
En esta prueba, se hace uso de los mismos casos de prueba que en la PU005 ya que es requisito que se completen esos datos para que se generen los índices y estos a su vez son requisitos para el diagnóstico.
- PU009: Verificar que se puede realizar la carga inicial de la ontología.
a) Campo de entrada: Ruta

Clases de equivalencia	Valores	Identificador
CV	Ruta existente	1
CNV	Ruta inexistente	2
CNV	Vacío	3

- Casos de prueba

CP	Clase equivalente	Valores	Resultado esperado	Observación
1	1	C:\Users\hcano\TESIS2\Smart_nutrition\Estructura_alimentos.xls	Se muestra mensaje: Ontología	OK

			generada	
2	2	C:\\\\Users\\hcano\\TESIS2\\Smart_nutrition\\Estructura_alimentos.xls	Se muestra mensaje de error: Ruta incorrecta	NO OK
3	3	“”	Se muestra mensaje de error: campo obligatorio	NO OK

Tabla 4.14 Casos de prueba PU009



5. Observaciones, conclusiones y recomendaciones.

En este capítulo se abordan las observaciones, conclusiones y recomendaciones a las que se llegan luego de la implementación del sistema.

5.1. Observaciones.

Los problemas que son tratados en este proyecto son los siguientes:

1. Identificar los distintos alimentos y sus aportes nutricionales y el hecho de tenerlos plenamente identificados de manera casi instantánea.
2. Obtención de los distintos índices que son utilizados para el proceso de diagnóstico como para la elaboración de la sugerencia de dieta.
3. Automatización del proceso de diagnóstico y generación de dietas balanceadas, así como la simplificación de consultas sobre los distintos alimentos del medio.

El sistema será capaz de solucionar dichos problemas, ya que su concepción está enfocada a la solución de todas estas problemáticas.

Los pacientes podrán ser evaluados de manera exacta y rápida, podrán conocer cuál es su peso ideal y cuanta es su necesidad calórica diaria, además, los especialistas podrán acceder de manera rápida y sencilla a la información de los alimentos, calcular los índices y necesidades calóricas de los pacientes muy fácilmente y consultar los distintos reportes que el sistema les ofrece.

5.2. Conclusiones.

Las conclusiones a las que se llega, luego de implementar el sistema, son las siguientes:

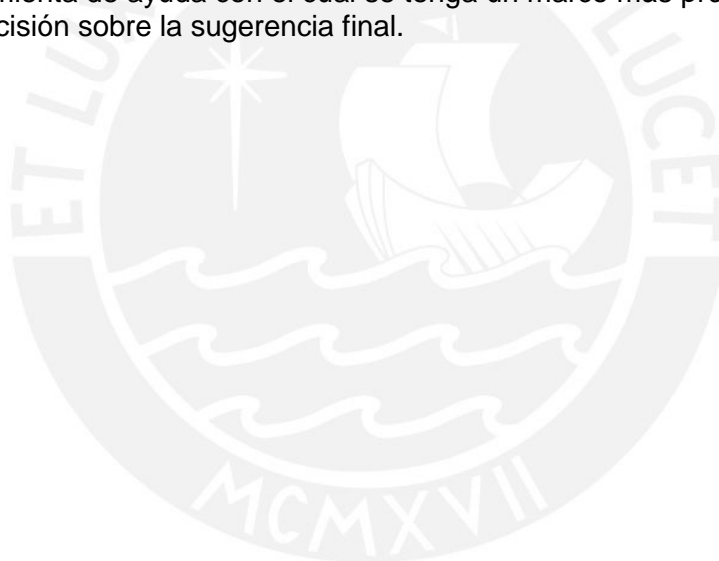
1. El sistema ayudará a aquellas personas con problemas de peso a tratarse de una manera adecuada.
2. Debido a la automatización del proceso de obtención de índices, diagnóstico, consulta de alimentos y generación de dietas, el especialista podrá realizar sus consultas de manera rápida.
3. Se logró desarrollar el algoritmo para que el motor de inferencia obtenga un diagnóstico basado en la información ingresada por el usuario sobre el paciente a tratar.
4. El usuario podrá realizar tanto la inscripción del paciente, atenderlo y consultar reportes en base a los resultados anteriormente registrados en el sistema.
5. El usuario experto podrá modificar los datos de todas las fórmulas usadas para los cálculos internos, modificando los coeficientes que forman parte de estas fórmulas en caso estudios posteriores indiquen que estos coeficientes deben modificarse así como la información de los distintos alimentos a nivel de valor energético, carbohidratos, grasas, proteínas y azúcar, así como también agregar nuevos alimentos a la estructura de alimentos, con ello conseguimos un alto grado de configuración del sistema.

6. La categorización del estatus socioeconómico de los pacientes es un aspecto bastante subjetivo, por ello, el sistema brinda la facilidad de que sean ellos quienes indiquen que alimentos son los que van a formar parte de la sugerencia de dieta, ocultándose de dicha posibilidad de selección, aquellas que pueden afectar su salud.

5.3. Recomendaciones.

Para hacer uso de este sistema de manera correcta, se deben seguir las siguientes recomendaciones:

1. El usuario experto deberá mantener actualizada la base de conocimiento y la de hechos, ya que estas son las fuentes para que el motor de inferencia realice su trabajo.
2. Los datos que ingrese el usuario sobre el paciente a tratar deben ser verdaderos, ya que solo de esta forma se garantiza que tanto el diagnóstico como la sugerencia sean lo más próximos a la realidad.
3. El sistema no pretende suplir en ningún momento al experto, solo ser una herramienta de ayuda con el cual se tenga un marco más próximo para la toma de decisión sobre la sugerencia final.



6. Referencias.

- [UOMMC11] UNIVERSITY OF MARYLAND – MEDICAL CENTER.
2011 Publicación en la web. Consulta: 15 de Abril de 2011
 < http://www.umm.edu/esp_ency/article/002468.htm>
- [VCA64] VILLEE, Claude A.
1964 Biología. Séptima edición.
- [PAA06] PEÑA AYALA, Alejandro.
2006 Sistemas Basados en Conocimientos: Una base para su concepción y desarrollo. México: Instituto Politécnico Nacional. Consulta:
 <http://www.wolnm.org/apa/articulos/Sistemas_Basados_Conocimiento.pdf?target=>>
- [GJM02] GUTIERREZ, José Manuel
2002 Sistemas Expertos basados en reglas
 Universidad de Cantabria: Dpto. de Matemática Aplicada.
- [NNFMD05] NOY, Natalya F. and MCGUINNESS, Deborah L.
2005 Desarrollo de ontologías: Guía para crear tu primera ontología
 Stanford, CA: Stanford University
- [BDM11] BWM 2.0 Diet Manager
 Consulta: 19 de Abril del 2011
 <<http://www.bioweblogic.com/products/bwm20/>>
- [ESN11] Equilibra Software Nutricional
 Consulta: 19 de Abril del 2011
 < <http://www.generalsoftec.com/equilibra.html>>
- [PDT11] Perfect Diet Tracker
 Consulta: 15 de Abril del 2011
 <<http://www.perfect-diet-tracker.com>>
- [LDE11] DE TORRES AURED, Lourdes
 La Dieta Equilibrada – Guía para enfermeras de atención primaria
 publicada en la web)
 Consulta: 11 de Abril del 2011
 <<http://www.nutricion.org/publicaciones/pdf/Gu%C3%ADa%20AP-Diet%C3%A9ticaWeb.pdf>>
- [OMS11] ORGANIZACIÓN MUNDIAL DE LA SALUD
 Obesidad y Sobrepeso – Nota Descriptiva N°311
 Consulta: 14 de Abril del 2011
 <<http://www.who.int/es/> >
- [IFBC12] Inferencing Forward and Backward Chaining
 Consulta: 21 de Septiembre del 2012

<<http://www.docstoc.com/docs/41606902/Inferencing-Forward-and-Backward-Chaining>>

- [RGM92] ROMERO GARCÍA, Mariluz
1992 Sistema experto para la formulación de dietas alimenticias por via enteral en el ser humano (SEFDA)
- [RAE12] REAL ACADEMIA ESPAÑOLA
Consulta: 21 de Septiembre del 2012
<<http://lema.rae.es/drae/>>
- [J12] JAVA
Consulta: 21 de Septiembre del 2012
<http://www.java.com/es/download/whatis_java.jsp>
- [DN12] Microsoft .Net
Consulta: 20 de Septiembre del 2012
<<http://www.microsoft.com/net>>
- [MSQL12] MySQL
Consulta: 21 de Septiembre del 2012
<<http://www.mysql.com/>>
- [MSS12] Microsoft SQL Server
Consulta: 21 de Septiembre del 2012
<<http://www.microsoft.com/sqlserver/en/us/product-info.aspx>>
- [ORA12] ORACLE
Consulta: 21 de Septiembre del 2012
<<http://www.oracle.com/pls/db102/homepage>>
- [DS12] DOCTISSIMO
Consulta: 15 de Octubre del 2012
<<http://salud.doctissimo.es>>
- [NETB12] NETBEANS
Consulta: 04 de Noviembre del 2012
<www.netbeans.org>
- [EE12] CANO NUÑEZ, Jesús
Entrevista del 30 de Noviembre del 2012