

**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e Ingenierías**

**Licenciatura en ingeniería informática**



**Proyecto Final: Sistema Experto**

**Seminario de Sistemas Basados en Conocimiento**

**Profesor: Julio Esteban Valdes Lopez**

**Alumnos:**

**Briceño Caguado, Luis Gerardo - 219473333**

**Miranda Mercado, Valeria - 219416801**

**08/11/2025**

# Índice

<b>Índice.....</b>	<b>2</b>
Introducción.....	3
Explicación.....	3
Base de Conocimiento.....	4
Grupos musculares.....	4
Ejercicios.....	4
Definición de splits (full-body, push-pull-legs, upper-lower).....	6
Motor de Inferencia.....	7
Selección del ejercicio.....	7
Generación de lista de ejercicios por grupo muscular.....	8
Generación de rutinas según el tipo de split.....	8
Generación de rutina Full Body.....	9
Generación de rutina Upper-Lower.....	9
Generación de rutina Push-Pull-Legs.....	10
Inicio del Sistema.....	11
Predicados auxiliares.....	13
Nota para split Upper-Lower.....	13
Traducción de opciones a rutinas y objetivos:.....	13
Visualización de resultados:.....	13
<b>Conclusiones.....</b>	<b>15</b>

## **Introducción**

En el presente documento se describe el desarrollo de un sistema experto como parte del proyecto de la materia Seminario de Solución de Problemas de Sistemas Basados en Conocimiento. El tema seleccionado por los integrantes del equipo consiste en la generación de rutinas de entrenamiento, diseñadas en función del objetivo del usuario (fuerza o hipertrofia) y del tipo de split elegido (full-body, push-pull-legs o upper-lower).

El propósito de este proyecto es aplicar los principios fundamentales de los sistemas basados en conocimiento, empleando un motor de inferencia, una base de hechos y una base de conocimiento que permitan ofrecer recomendaciones adaptadas a las necesidades de cada usuario. Con ello, se busca demostrar cómo la inteligencia artificial simbólica puede ser utilizada para resolver problemas prácticos dentro del ámbito del acondicionamiento físico y el entrenamiento personalizado.

Asimismo, el desarrollo de este sistema experto representa una oportunidad para comprender el funcionamiento interno de los mecanismos de razonamiento automatizado, desde la representación del conocimiento hasta la toma de decisiones mediante reglas lógicas. Además, contribuye al fortalecimiento de las habilidades de análisis, diseño e implementación de soluciones inteligentes, mediante el uso de PROLOG, un lenguaje de programación lógico y declarativo ampliamente utilizado en el desarrollo de sistemas basados en conocimiento e inteligencia artificial simbólica.

Este documento detalla las etapas del desarrollo del sistema, que incluyen la definición del tema elegido, el diseño de la base de conocimiento, la implementación del motor de inferencia, la realización de pruebas funcionales y, finalmente, las conclusiones y reflexiones derivadas del proyecto. Asimismo, se abordan los retos asociados al aprendizaje y aplicación de un lenguaje de programación declarativo, como PROLOG, utilizado para la construcción del sistema experto.

## **Explicación**

Se consideró que la generación de rutinas de ejercicio basadas en el objetivo y el tipo de split mediante un sistema experto sería un tema adecuado para aplicar los conocimientos adquiridos en la materia, ya que estos factores permiten estructurar rutinas de entrenamiento a partir de criterios definidos y lógicos.

Por ejemplo, según el objetivo del usuario y el split de rutina elegido se determinan los ejercicios adecuados.

Posteriormente, con base en el tipo de split elegido, se organiza la distribución de los ejercicios y los días de entrenamiento, dado que el tipo de split define la manera en que se dividen los grupos musculares a trabajar por sesión.

## Base de Conocimiento

Tomando en cuenta estos factores, se procedió a construir el sistema experto basado en conocimiento. En primer lugar, se definió la base de conocimiento, elaborada a partir de la recopilación de información de la encuesta simulada con un experto en entrenamiento físico y del conocimiento previo del equipo. Con esta información, se identificaron los principales grupos musculares que se suelen considerar en una rutina de entrenamiento, los cuales se representaron en la base de conocimiento mediante hechos en PROLOG de la siguiente forma:

### *Grupos musculares*

- **grupo\_muscular**(pecho).
- **grupo\_muscular**(espalda).
- **grupo\_muscular**(hombros).
- **grupo\_muscular**(biceps).
- **grupo\_muscular**(triceps).
- **grupo\_muscular**(cuadriceps).
- **grupo\_muscular**(femoral).
- **grupo\_muscular**(gluteos).
- **grupo\_muscular**(pantorrillas).
- **grupo\_muscular**(abdomen).

### *Ejercicios*

Posteriormente, se definieron al menos cinco ejercicios por grupo muscular, con el objetivo de dotar al sistema experto de una base de conocimiento amplia y variada. Se estableció este número mínimo para mantener la practicidad en la elaboración del sistema experto basado en conocimiento, sin comprometer la diversidad de rutinas que el sistema puede generar.

La forma en que se decidió declarar los hechos en **PROLOG** fue la siguiente:

- **ejercicio** (Grupo, Nombre, Tipo)

#### **Donde:**

- **Grupo** representa el grupo muscular trabajado en el ejercicio.
- **Nombre** indica el nombre del ejercicio.
- **Tipo** hace referencia a si el ejercicio es compuesto o de accesorio, lo cual permite dar variedad a las rutinas generadas por el motor de inferencia.

Ya que planteamos los aspectos que creíamos necesarios para los ejercicios nos dimos a la tarea de definir los hechos por grupo muscular: pecho, espalda, hombros, biceps, triceps, cuadriceps, femoral, glúteos, pantorrillas, y abdomen.

A continuación, se muestran los hechos definidos en la base de conocimiento:

#### **PECHO**

**ejercicio**(pecho, press\_banca, compuesto).

**ejercicio**(pecho, press\_inclinado, compuesto).  
**ejercicio**(pecho, fondos, compuesto).  
**ejercicio**(pecho, aperturas, accesorio).  
**ejercicio**(pecho, cruces\_polea, accesorio).

### **ESPALDA**

**ejercicio**(espalda, dominadas, compuesto).  
**ejercicio**(espalda, remo\_barra, compuesto).  
**ejercicio**(espalda, jalon\_pecho, accesorio).  
**ejercicio**(espalda, peso\_muerto, compuesto).  
**ejercicio**(espalda, remo\_mancuerna, accesorio).

### **HOMBROS**

**ejercicio**(hombros, press\_militar, compuesto).  
**ejercicio**(hombros, press\_arnold, compuesto).  
**ejercicio**(hombros, elevaciones\_laterales, accesorio).  
**ejercicio**(hombros, remo\_menton, compuesto).  
**ejercicio**(hombros, elevaciones\_posteriores, accesorio).

### **BÍCEPS**

**ejercicio**(biceps, curl\_barra, accesorio).  
**ejercicio**(biceps, dominadas\_supinas, compuesto).  
**ejercicio**(biceps, curl\_martillo, accesorio).  
**ejercicio**(biceps, curl\_concentrado, accesorio).  
**ejercicio**(biceps, curl\_predicador, accesorio).

### **TRICEPS**

**ejercicio**(triceps, press\_cerrado, compuesto).  
**ejercicio**(triceps, fondos\_triceps, compuesto).  
**ejercicio**(triceps, extension\_polea, accesorio).  
**ejercicio**(triceps, patada\_mancuerna, accesorio).  
**ejercicio**(triceps, rompecraneos, accesorio).

### **CUADRICEPS**

**ejercicio**(cuadriceps, sentadilla, compuesto).  
**ejercicio**(cuadriceps, prensa, compuesto).  
**ejercicio**(cuadriceps, sentadilla\_bulgara, accesorio).  
**ejercicio**(cuadriceps, sentadilla\_frontal, compuesto).  
**ejercicio**(cuadriceps, extension\_pierna, accesorio).

### **FEMORAL**

**ejercicio**(femoral, peso\_muerto\_rumano, compuesto).  
**ejercicio**(femoral, curl\_femoral, accesorio).  
**ejercicio**(femoral, buenos\_dias, compuesto).  
**ejercicio**(femoral, curl\_nordico, accesorio).  
**ejercicio**(femoral, curl\_femoral\_sentado, accesorio).

### **GLÚTEOS**

**ejercicio**(gluteos, hip\_thrust, compuesto).  
**ejercicio**(gluteos, puente\_gluteo, accesorio).

**ejercicio**(gluteos, sentadilla\_profunda, compuesto).

**ejercicio**(gluteos, peso\_muerto\_sumo, compuesto).

**ejercicio**(gluteos, patada\_gluteo, accesorio).

#### **PANTORRILLAS**

**ejercicio**(pantorrillas, elevaciones\_talones, accesorio).

**ejercicio**(pantorrillas, donkey\_raises, accesorio).

**ejercicio**(pantorrillas, elevaciones\_prensa, accesorio).

**ejercicio**(pantorrillas, saltos\_cuerda, accesorio).

**ejercicio**(pantorrillas, elevaciones\_unilateral, accesorio).

#### **ABDOMEN**

**ejercicio**(abdomen, crunch, accesorio).

**ejercicio**(abdomen, plancha, accesorio).

**ejercicio**(abdomen, elevaciones\_piernas, accesorio).

**ejercicio**(abdomen, rueda\_abdominal, accesorio).

**ejercicio**(abdomen, crunch\_maquina, accesorio).

#### **Definición de splits (full-body, push-pull-legs, upper-lower)**

Una vez definidos los ejercicios por grupo muscular, se procedió con la definición de los tipos de split o rutinas. En esta sección se establecieron los hechos correspondientes a cada tipo de rutina y los grupos musculares trabajados en cada una de ellas.

Por ejemplo, en el caso de la rutina **full-body**, se sabe (según la información obtenida en la simulación de encuesta con el experto) que se debe trabajar un ejercicio de piernas, uno de pecho, uno de espalda, uno de hombros o brazos, y uno de abdomen. Esto se debe a que este tipo de rutina busca entrenar todo el cuerpo en una sola sesión, evitando el sobreentrenamiento de un mismo grupo muscular.

En la rutina **push-pull-legs**, el entrenamiento se divide en tres secciones principales.

- En **push**, se trabajan pecho, hombros y tríceps.
- En **pull**, se enfocan los ejercicios en espalda, bíceps y abdomen.
- Finalmente, en **legs**, se incluyen los grupos musculares de piernas, tales como cuádriceps, femorales, glúteos y pantorrillas.

Por último, en la rutina **upper-lower**, la división se realiza entre tren superior e inferior:

- En **upper**, se trabaja pecho, espalda, hombros, bíceps y tríceps.
- En **lower**, se incluyen cuádriceps, femorales, glúteos y pantorrillas.

A partir de esta información, los hechos definidos en la base de conocimiento quedaron estructurados de la siguiente manera:

**rutina**(GrupoMuscular,[Ejercicios], [Tipo\_por\_cada\_Ejercicio]).

**rutina**(full\_body, [pecho, espalda, cuadriceps, hombros, biceps, triceps, abdomen],[compuesto, accesorio, compuesto,\_,\_,\_]).

**rutina**(upper, [pecho, pecho, espalda, espalda, hombros, biceps, triceps], [compuesto, accesorio, compuesto, accesorio,\_,\_,\_]).

**rutina**(lower, [cuadriceps, cuadriceps, femoral, gluteos, pantorrillas], [compuesto, accesorio,\_, compuesto, \_]).

**rutina**(push, [pecho, pecho, hombros, hombros, triceps],[compuesto, accesorio, compuesto, accesorio, \_]).

**rutina**(pull, [espalda, espalda, biceps, abdomen],[compuesto, accesorio, accesorio, \_]).

**rutina**(legs, [cuadriceps, cuadriceps, femoral, gluteos, pantorrillas],[compuesto, accesorio, accesorio, compuesto, \_]).

## **Motor de Inferencia**

Luego, se pasó a una etapa más compleja: el diseño e implementación del motor de inferencia. En esta parte, se definieron las reglas lógicas que permiten al sistema realizar un proceso de razonamiento basado en los datos proporcionados por el usuario (contenidos en la base de hechos) y el conocimiento estructurado previamente (definido en la base de conocimiento).

El motor de inferencia actúa como el “cerebro” del sistema experto, ya que se encarga de evaluar las condiciones establecidas en las reglas, comparar los hechos disponibles y deducir conclusiones que permiten generar la rutina de entrenamiento más adecuada.

En términos generales, este componente aplica un mecanismo de inferencia hacia adelante (forward chaining), en el cual se parte de los hechos conocidos (por ejemplo, el tipo de rutina o el objetivo del usuario) y se buscan las reglas que puedan aplicarse para derivar nuevas conclusiones, como los grupos musculares o ejercicios específicos que deben incluirse.

De esta forma, el sistema experto no se limita a mostrar información estática, sino que razona de manera automatizada para ofrecer una respuesta personalizada en función de las condiciones establecidas, demostrando la utilidad práctica del razonamiento lógico en la generación de recomendaciones inteligentes.

### ***Selección del ejercicio***

La selección de ejercicios se implementó mediante el predicado `selecciona_ejercicio/3`. Este proceso consiste en generar una lista de ejercicios que correspondan a un grupo muscular y tipo específicos (compuesto o accesorio), utilizando el predicado `findall/3`.

Posteriormente, a partir de esa lista, se selecciona un ejercicio de forma aleatoria mediante `random_member/2`, lo que permite introducir variabilidad en los resultados y evitar repeticiones constantes.

**selecciona\_ejercicio**(Grupo, Ejercicio, Tipo) :-  
findall((X,Tipo), ejercicio(Grupo,X, Tipo, \_), Lista),

random\_member((Ejercicio,Tipo), Lista).

```
?- selecciona_ejercicio(pantorrillas, Ejercicio, Tipo).  
Ejercicio = elevaciones_prensa,  
Tipo = accesorio.  
  
?- selecciona_ejercicio(pecho, Ejercicio, compuesto).  
Ejercicio = press_banca_inclinado.  
  
?- selecciona_ejercicio(espalda, Ejercicio, accesorio).  
Ejercicio = pullover.
```

### ***Generación de lista de ejercicios por grupo muscular***

Posteriormente, se diseñó una regla que permitiera generar una lista de ejercicios correspondiente a cada división dentro de una rutina. Para ello, se implementó el predicado `ejercicios_por_grupos/3`, el cual recibe tres listas como parámetros: una con los grupos musculares, otra con los ejercicios seleccionados y una tercera con los tipos de ejercicio (compuesto o accesorio).

Dentro de este predicado, se invoca a `selecciona_ejercicio/3`, pasando como argumentos el primer elemento de la lista de grupos musculares (`Grupo`), la variable `Ejercicio` y la cabeza de la lista de tipos (`Tipo`). De esta manera, se obtiene un ejercicio correspondiente a cada combinación de grupo muscular y tipo de ejercicio.

El proceso se repite de forma recursiva, llamando nuevamente a `ejercicios_por_grupos/3` con las colas de las listas, hasta que estas quedan vacías. Así, se construye una lista completa de ejercicios que cubre todos los grupos musculares definidos en la rutina.

**`ejercicios_por_grupos([], [], []).`**

**`ejercicios_por_grupos([Grupo|Gs],[Ejercicio|Es],[Tipo|Ts]) :-`**

`selecciona_ejercicio(Grupo, Ejercicio, Tipo),  
ejercicios_por_grupos(Gs, Es, Ts) .`

```
?- ejercicios_por_grupos([pecho, espalda, cuadriceps],E, T).  
E = [press_pecho_máquina, jalon_al_pecho, press_piernas],  
T = [compuesto, accesorio, compuesto].  
  
?- ejercicios_por_grupos([pecho, espalda, cuadriceps],E, [accesorio,accesorio,accesorio]).  
E = [cables_cruzados, remo_sentado, extension_pierna].
```

### ***Generación de rutinas según el tipo de split***

Para generar las rutinas correspondientes a cada tipo de split, se implementó el predicado `genera_rutina/2`, el cual se adapta según la estructura del entrenamiento que se desea construir.



### ***Generación de rutina Full Body***

En el caso del split Full Body, el predicado recibe como primer parámetro el nombre del tipo de rutina y, como segundo parámetro, una lista que representa los días en los que se divide dicho entrenamiento.

Dentro de su definición, se invoca al predicado rutina/3, al cual se le pasa el nombre del split junto con dos variables: Grupos y Tipo. Esto permite obtener los grupos musculares que se trabajan en esa rutina y el tipo de ejercicio correspondiente a cada uno (por ejemplo, compuesto o accesorio).

A continuación, se llama al predicado ejercicios\_por\_grupos/3, enviando como argumentos las listas Grupos y Tipo devueltas por rutina/3, además de la variable Ejercicios, que almacenará los ejercicios seleccionados para cada grupo muscular.

Finalmente, se construye la variable Dia, que asocia el nombre del split con la lista de ejercicios generada. De este modo, se obtiene una estructura organizada que representa la rutina completa del tipo Full Body.

**genera\_rutina**(full\_body, [Dia]) :-  
rutina(full\_body, Grupos, Tipo),  
ejercicios\_por\_grupos(Grupos, Ejercicios, Tipo),  
Dia = dia(full\_body, Ejercicios, Tipo).

```
?- genera_rutina(full_body, [D]).  
D = dia(full_body, [press_banca_inclinado, hiperextension, press_piernas, p  
ress_arnold, curl_martillo, maquina_para_fondos, elevaciones_de_piernas], [  
compuesto, accesorio, compuesto, compuesto, accesorio, compuesto, accesorio  
]).
```

### ***Generación de rutina Upper-Lower***

Para el split Upper-Lower, se definió el predicado genera\_rutina/2, al cual se le pasa como primer argumento el nombre del split y, como segundo, una lista que contiene las variables Dia1 y Dia2, correspondientes a los dos días en los que se divide este tipo de rutina: uno enfocado en el tren superior (Upper) y otro en el tren inferior (Lower).

En este caso, se utiliza el predicado rutina/3 dos veces: una para obtener los grupos musculares y tipos de ejercicios asociados al día Upper, y otra para los correspondientes al día Lower. Cada llamada devuelve las listas GruposUpper y GruposUpper\_Tipos, así como GruposLower y GruposLower\_Tipos.

Posteriormente, estas listas se envían como argumentos al predicado ejercicios\_por\_grupos/3, el cual genera la lista de ejercicios seleccionados para cada grupo muscular, tanto del tren superior como del inferior.

Finalmente, los resultados se asignan a las variables Dia1 y Dia2, creando así una estructura organizada que asocia cada día con sus ejercicios respectivos.

**genera\_rutina**(upper\_lower, [Dia1, Dia2]) :-  
rutina(upper, GruposUpper, GruposUpper\_Tipos),  
rutina(lower, GruposLower, GruposLower\_Tipos),  
ejercicios\_por\_grupos(GruposUpper, EjerciciosUpper, GruposUpper\_Tipos),  
ejercicios\_por\_grupos(GruposLower, EjerciciosLower, GruposLower\_Tipos),  
Dia1 = dia(upper, EjerciciosUpper, GruposUpper\_Tipos),  
Dia2 = dia(lower, EjerciciosLower, GruposLower\_Tipos).

```
?- genera_rutina(upper_lower, [Dia1, Dia2]).  
Dia1 = dia(upper, [press_banca_inclinado, cables_cruzados, dominadas, remo_sentado, press_arnold, curl_concentrado, extension_tricep_un_brazo], [compuesto, accesorio, compuesto, accesorio, compuesto, accesorio, accesorio]),  
Dia2 = dia(lower, [sentadilla_bulgara, extension_pierna, curl_femoral, peso_muerto_sumo, elevaciones_talones], [compuesto, accesorio, accesorio, compuesto, accesorio]).
```

### ***Generación de rutina Push-Pull-Legs***

Para el split Push-Pull-Legs, el proceso de generación de la rutina sigue la misma lógica que en los casos anteriores, con la diferencia de que este tipo de división contempla tres días de entrenamiento, correspondientes a los movimientos de empuje (Push), tracción (Pull) y piernas (Legs).

En este caso, el predicado genera\_rutina/2 recibe como primer argumento el nombre del split y, como segundo, una lista compuesta por las variables DiaPush, DiaPull y DiaLegs, que representan los tres días de entrenamiento.

El predicado rutina/3 se invoca tres veces, una por cada tipo de división, para obtener los grupos musculares y tipos de ejercicios asociados a cada día. Posteriormente, se llama al predicado ejercicios\_por\_grupos/3 con los valores obtenidos, con el fin de generar la lista de ejercicios seleccionados para cada grupo muscular en los tres tipos de entrenamiento.

Finalmente, los resultados se asignan a las variables DiaPush, DiaPull y DiaLegs, formando la estructura final de la rutina.

**genera\_rutina**(push\_pull\_legs, [DiaPush, DiaPull, DiaLegs]) :-  
rutina(push, GruposPush, GruposPush\_Tipo),  
rutina(pull, GruposPull, GruposPull\_Tipo),  
rutina(legs, GruposLegs, GruposLegs\_Tipo),  
ejercicios\_por\_grupos(GruposPush, EjerciciosPush, GruposPush\_Tipo),  
ejercicios\_por\_grupos(GruposPull, EjerciciosPull, GruposPull\_Tipo),  
ejercicios\_por\_grupos(GruposLegs, EjerciciosLegs, GruposLegs\_Tipo),  
DiaPush = dia(push, EjerciciosPush, GruposPush\_Tipo),  
DiaPull = dia(pull, EjerciciosPull, GruposPull\_Tipo),  
DiaLegs = dia(legs, EjerciciosLegs, GruposLegs\_Tipo)

```
?- genera_rutina(push_pull_legs, [DiaPush, DiaPull, DiaLegs]).
DiaPush = dia(push, [press_pecho_máquina, mariposa_pec_deck, press_arnold,
elevaciones_frontales, patada_mancuerna], [compuesto, accesorio, compuesto,
accesorio, accesorio]),
DiaPull = dia(pull, [dominadas, remo_sentado, curl_predicador, plancha], [c
ompuesto, accesorio, accesorio, accesorio]),
DiaLegs = dia(legs, [press_piernas, extension_pierna, curl_femoral, sentadi
lla_profunda, elevaciones_talones], [compuesto, accesorio, accesorio, compu
esto, accesorio]).
```

## Inicio del Sistema

Para comenzar la ejecución del sistema experto, se solicita al usuario seleccionar el tipo de rutina que desea generar.

Esta selección se realiza mediante una entrada numérica donde:

- 1 corresponde a Full Body
- 2 a Push–Pull–Legs
- 3 a Upper–Lower

El valor introducido por el usuario se guarda en la variable Opc y se pasa al predicado tipo\_rutina/2, el cual traduce el número ingresado al nombre del split correspondiente.

Posteriormente, se solicita al usuario seleccionar el objetivo del entrenamiento, eligiendo entre:

- 1 para Fuerza
- 2 para Hipertrofia

De manera similar, el valor leído se almacena en OpcObjetivo y se asocia a su correspondiente concepto mediante el predicado objetivo\_entrenamiento/2.

Una vez definidos ambos parámetros, el sistema llama al predicado genera\_rutina/2, que se encarga de construir la rutina de acuerdo con el tipo de split seleccionado, generando una lista de días (Dias) con los ejercicios asignados.

Luego, se ejecuta el predicado mostrar\_rutina/2, encargado de desplegar la rutina en pantalla junto con las recomendaciones de entrenamiento según el objetivo elegido.

Como extra tenemos el predicado si\_upplow/1 con el cual mostramos una nota específica para ese Split y también el volver a preguntar si quiere generar otra rutina, con el fin de crear otra sin necesidad de volver a escribir inicio.

### inicio :-

```
writeln('SISTEMA EXPERTO GENERADOR DE RUTINAS'),
writeln('Seleccione el tipo de rutina:'),
writeln('1. Full Body'),
writeln('2. Push–Pull–Legs'),
writeln('3. Upper–Lower'),
read(Opc),
tipo_rutina(Opc, TipoRutina),
```

```

writeln('Seleccione el objetivo:'),
writeln('1. Fuerza'),
writeln('2. Hipertrofia'),
read(OpcObjetivo),
objetivo_entrenamiento(OpcObjetivo, Objetivo),

```

```

genera_rutina(TipoRutina, Dias),
mostrar_rutina(Dias, Objetivo).
si_upplow(TipoRutina);
(writeln("\nGenerar otra rutina(s/n): '),
read(OpcRutina), OpcRutina==s,inicio, nl;!)).

```

```

?- inicio.
SISTEMA EXPERTO GENERADOR DE RUTINAS
Seleccione el tipo de rutina:
1. Full Body
2. Push|Full|Legs
3. Upper|Lower
|: 1.
Seleccione el objetivo:
1. Fuerza
2. Hipertrofia
|: 1.

Día: full_body

press_banca_inclinado(compuesto): 4|6 repeticiones, 80|90% del peso máximo
remo_sentado(accesorio): 4|6 repeticiones, 80|90% del peso máximo
press_piernas(compuesto): 4|6 repeticiones, 80|90% del peso máximo
press_arnold(compuesto): 4|6 repeticiones, 80|90% del peso máximo
curl_concentrado(accesorio): 4|6 repeticiones, 80|90% del peso máximo
press_frances(accesorio): 4|6 repeticiones, 80|90% del peso máximo
crunch_maquina(accesorio): 4|6 repeticiones, 80|90% del peso máximo

Generar otra rutina(s/n):
|: s.

```

```

Generar otra rutina(s/n):
|: s.
SISTEMA EXPERTO GENERADOR DE RUTINAS
Seleccione el tipo de rutina:
1. Full Body
2. Push|Full|Legs
3. Upper|Lower
|: 2.
Seleccione el objetivo:
1. Fuerza
2. Hipertrofia
|: 2.

Día: push

press_pecho_máquina(compuesto): 8|12 repeticiones, 65|75% del peso máximo
cables_cruzados(accesorio): 8|12 repeticiones, 65|75% del peso máximo
press_arnold(compuesto): 8|12 repeticiones, 65|75% del peso máximo
tiron_a_la_carra(accesorio): 8|12 repeticiones, 65|75% del peso máximo
patada_mancuerna(accesorio): 8|12 repeticiones, 65|75% del peso máximo

Día: pull

dominadas(compuesto): 8|12 repeticiones, 65|75% del peso máximo
hiperextension(accesorio): 8|12 repeticiones, 65|75% del peso máximo
curl_barra(accesorio): 8|12 repeticiones, 65|75% del peso máximo
crunch(accesorio): 8|12 repeticiones, 65|75% del peso máximo

Día: legs

press_piernas(compuesto): 8|12 repeticiones, 65|75% del peso máximo
extension_pierna(accesorio): 8|12 repeticiones, 65|75% del peso máximo
curl_femoral_sentado(accesorio): 8|12 repeticiones, 65|75% del peso máximo
hip_thrust(compuesto): 8|12 repeticiones, 65|75% del peso máximo
elevaciones_prensa(accesorio): 8|12 repeticiones, 65|75% del peso máximo

Generar otra rutina(s/n):
|: n.

```

## Predicados auxiliares

### *Nota para split Upper-Lower*

**si\_upplow**(upper\_lower):-

```
format('\nNota: Puedes repetir este split dos veces por semana, alternando el enfoque entre
grupos musculares\n'),
(writeln('\nGenerar otra rutina(s/n): '),
read(OpcRutina), OpcRutina==s,inicio, nl;!).
```

```
?- si_upplow(upper_lower).

Nota: Puedes repetir este split dos veces por semana, alternando el enfoque
entre grupos musculares

Generar otra rutina(s/n):
|: n.

true.
```

### *Traducción de opciones a rutinas y objetivos:*

**tipo\_rutina**(1, full\_body).

**tipo\_rutina**(2, push\_pull\_legs).

**tipo\_rutina**(3, upper\_lower).

```
?- tipo_rutina(1, Rutina).
Rutina = full_body.

?- tipo_rutina(2, Rutina ).
Rutina = push_pull_legs.

?- tipo_rutina(3, Rutina ).
Rutina = upper_lower.
```

**objetivo\_entrenamiento**(1, fuerza).

**objetivo\_entrenamiento**(2, hipertrofia).

```
?- objetivo_entrenamiento(1, Objetivo).
Objetivo = fuerza.

?- objetivo_entrenamiento(2, Objetivo).
Objetivo = hipertrofia.
```

### *Visualización de resultados:*

El conjunto de predicados `mostrar_rutina/3`, `mostrar_ejercicios/3` y `recomendaciones/3` tiene como función mostrar en pantalla la rutina generada por el sistema experto. El proceso inicia con `mostrar_rutina/3`, que recibe una lista de días y el objetivo del entrenamiento (fuerza o hipertrofia). Este predicado recorre la lista día por día, mostrando el nombre de cada uno y llamando a `mostrar_ejercicios/3` para desplegar los ejercicios correspondientes. Por su parte,

mostrar\_ejercicios/3 recorre la lista de ejercicios de cada día y, para cada uno, invoca el predicado recomendaciones/3, el cual imprime las indicaciones específicas de repeticiones y porcentaje de peso dependiendo del objetivo elegido. Si el objetivo es fuerza, se recomiendan pocas repeticiones con un alto porcentaje del peso máximo; mientras que si el objetivo es hipertrofia, se sugiere un número mayor de repeticiones con un peso moderado. De esta manera, mediante el uso de recursividad, el sistema muestra toda la rutina día por día y ejercicio por ejercicio junto con las recomendaciones adecuadas.

**mostrar\_rutina**([],\_).

**mostrar\_rutina**([dia(Nombre\_Split, Ejercicios,Tipo)|Resto], Objetivo) :-

```
format('\nDía: ~w \n', [Nombre_Split]), nl,
mostrar_ejercicios(Ejercicios, Tipo, Objetivo),
mostrar_rutina(Resto, Objetivo).
```

```
?- mostrar_rutina([dia(upper, [press_banca_inclinado, cables_cruzados, dominadas, remo_sentado, press_arnold, curl_concentrado, extension_tricep_un_brazo], [compuesto, accesorio, compuesto, accesorio, compuesto, accesorio, accesorio]), fuerza).

Día: upper

press_banca_inclinado(compuesto): 4||6 repeticiones, 80||90% del peso máximo
cables_cruzados(accesorio): 4||6 repeticiones, 80||90% del peso máximo
dominadas(compuesto): 4||6 repeticiones, 80||90% del peso máximo
remo_sentado(accesorio): 4||6 repeticiones, 80||90% del peso máximo
press_arnold(compuesto): 4||6 repeticiones, 80||90% del peso máximo
curl_concentrado(accesorio): 4||6 repeticiones, 80||90% del peso máximo
extension_tricep_un_brazo(accesorio): 4||6 repeticiones, 80||90% del peso máximo
true
```

**mostrar\_ejercicios**([], [],\_).

**mostrar\_ejercicios**([E|Es], [T|Ts], Objetivo) :-

```
recomendaciones(E, T, Objetivo),
mostrar_ejercicios(Es, Ts, Objetivo).
```

```
?- mostrar_ejercicios([press_banca_inclinado, cables_cruzados, dominadas, remo_sentado, press_arnold, curl_concentrado, extension_tricep_un_brazo], [compuesto, accesorio, compuesto, accesorio, compuesto, accesorio, accesorio], hipertrofia).

press_banca_inclinado(compuesto): 8||12 repeticiones, 65||75% del peso máximo

cables_cruzados(accesorio): 8||12 repeticiones, 65||75% del peso máximo
dominadas(compuesto): 8||12 repeticiones, 65||75% del peso máximo
remo_sentado(accesorio): 8||12 repeticiones, 65||75% del peso máximo
press_arnold(compuesto): 8||12 repeticiones, 65||75% del peso máximo
curl_concentrado(accesorio): 8||12 repeticiones, 65||75% del peso máximo
extension_tricep_un_brazo(accesorio): 8||12 repeticiones, 65||75% del peso máximo
true.
```

**recomendaciones**(Ejercicio,Tipo, fuerza) :-

```
format('~w(~w): 4-6 repeticiones, 80-90% del peso máximo\n', [Ejercicio, Tipo]), !.
```

**recomendaciones**(Ejercicio, Tipo, hipertrofia) :-

```
format('~w(~w): 8-12 repeticiones, 65-75% del peso máximo\n', [Ejercicio, Tipo]).
```

```
?- recomendaciones(press_banca_inclinado, compuesto, fuerza).  
press_banca_inclinado(compuesto): 4|6 repeticiones, 80|90% del peso máximo  
true .  
  
?- recomendaciones(press_banca_inclinado, compuesto, hipertrofia).  
press_banca_inclinado(compuesto): 8|12 repeticiones, 65|75% del peso máximo  
true .
```

## Conclusiones

Realizar este proyecto supuso un reto divertido. No pensábamos que haber hecho la encuesta simulada con el experto serviría tanto, y la verdad es que fue la base fundamental para poder desarrollar el Sistema Experto Basado en Conocimiento (SEBC). Si bien nos esforzamos por abarcar lo que creíamos necesario, hubo algunos conocimientos que se nos escaparon, aunque nada que no pudiera resolverse investigando.

Pensar en un sistema basado en conocimiento al principio parecía sencillo, pero hay muchas variables que considerar respecto a lo que se desea lograr. Nos ocurrió que la parte de la base de conocimiento fue relativamente sencilla, ya que solo consistía en definir los hechos que utilizaríamos; sin embargo, el motor de inferencia resultó ser una experiencia más complicada. Sabíamos lo que queríamos hacer, pero como Prolog es un lenguaje de programación lógico declarativo, no estábamos del todo familiarizados con su lógica. A pesar de haber realizado trabajos en clase con problemas por resolver, integrar toda esa lógica aprendida para que las partes funcionaran en conjunto fue difícil. Tropezamos una y otra vez con la lógica, pero lo bueno fue que, al repetir tanto el proceso, comprendimos más con cada problema y el resto del trabajo no resultó tan complicado una vez que empezamos.

Como conclusión final, los Sistemas Expertos Basados en Conocimiento son sistemas diseñados para resolver problemas muy específicos, en los cuales se construyen para obtener resultados a partir de reglas definidas y hechos concretos. Creemos que los SEBC son muy útiles en este tipo de temas, ya que su conocimiento es concreto y sus decisiones se forman a partir de una base establecida.