

UNIVERSITAT POLITÈCNICA DE CATALUNYA

INTELIGENCIA ARTIFICIAL

BACHELOR DEGREE IN COMPUTER SCIENCE

---

## PROYECTO DE SBC

---

*Authors:*

Daniel Moreno  
Fabio De Angelis  
Solange Palomino

*Professor:*

Javier Vazquez Salceda

Q2 Curso 2019-2020



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# Índice

<b>Índice</b>	<b>1</b>
<b>Introducción</b>	<b>2</b>
<b>Identificación del problema</b>	<b>3</b>
Descripción del problema	3
Viabilidad de construir un Sistema Basado en el Conocimiento	3
Fuentes del conocimiento	4
Objetivos del sistema	4
Resultados esperados por el sistema	4
<b>Conceptualización</b>	<b>5</b>
Elementos principales del dominio	5
Problemas y subproblemas	6
Flujo de razonamiento	7
Ejemplo de conocimiento experto	8
<b>Formalización</b>	<b>10</b>
Desarrollo de la ontología	10
Elementos de la ontología	11
Métodos de resolución	19
Abstracción de datos	19
Asociación heurística	20
Refinamiento	20
<b>Implementación</b>	<b>21</b>
Ontología	21
Módulos	22
Prototipos	26

# Introducción

En esta práctica se presenta un problema para resolver mediante el desarrollo de un sistema basado en el conocimiento.

Para la realización de la práctica hemos seguido las siguientes fases:

- **Identificación del problema:** Determinar si el problema es adecuado para desarrollar un SBC. También consiste en buscar las fuentes de conocimiento, determinar el conocimiento necesario para el desarrollo del sistema y establecer los objetivos del sistema.
- **Conceptualización:** En esta fase debemos detallar los elementos del dominio, descomponer el problema en subproblemas, detallar el flujo de razonamiento y las entradas y salidas de cada subproblema.
- **Formalización:** En esta fase deberemos decidir el formalismo de representación adecuado, identificar el espacio de búsqueda y analizar la tipología de los subproblemas y decidir los métodos de resolución de problemas.
- **Implementación:** En esta fase deberemos construir una ontología del dominio, encajar los problemas identificados en las metodologías de resolución de problemas, construir los módulos que correspondan a cada subproblema y construir un prototipo inicial, que iremos aumentando incrementalmente.
- **Validación y prueba:** En esta fase debemos escoger casos representativos y resolverlos mediante el sistema. Esta fase se irá repitiendo a medida que se desarrolle el prototipo.

# Identificación del problema

## Descripción del problema

Actualmente, la sociedad vive en un contexto en el que la inminente rutina de estudio y/o trabajo ha abarcado casi todas las energías y horas de las personas, por lo que mayoría de las personas tienden hacia lo que se denomina como sedentariedad. Muchos buscan alternativas para mantenerse en movimiento dentro del mismo trabajo tales como las pausas activas que ayudan a la reducción del estrés, sin embargo, lo ideal es que las personas cuenten con una sesión de ejercicios dentro de la misma rutina diaria.

Ante ello, la cadena de gimnasios *Coaching Potato* se plantea ayudar a sus futuros clientes haciéndoles una recomendación de un programa de entrenamiento personalizado. Para ello necesita un programa capaz de recomendar ejercicios a los nuevos clientes dependiendo de sus hábitos, condición física, objetivo y su dieta.

## Viabilidad de construir un Sistema Basado en el Conocimiento

Antes de intentar resolver el problema, debemos saber identificar qué tipo de problema es y cómo puede resolverse, teniendo en cuenta cuestiones como: la información del dominio que se tiene, el tamaño, cómo es la solución que se ha de ofrecer, entre otros.

Se podría intentar abarcarlo como un problema de búsqueda, porque consiste en desplazarse por el espacio de ejercicios y construir un conjunto de ellas respetando las restricciones y los objetivos del cliente, pero el espacio de exploración sería muy grande y las posibles soluciones serían demasiadas.

Tampoco lo podemos resolverlo de forma algorítmica, ya que una función no nos podrá representar todas las decisiones de exploración en este problema.

Por lo tanto, la mejor forma de resolver el problema es mediante la construcción de un Sistema Basado en el Conocimiento (en adelante SBC) porque tiene un tamaño adecuado y necesitamos mostrar una solución justificada.

## Fuentes del conocimiento

Las fuentes de conocimiento de un SBC son muy importantes, ya que, a partir de estas, el sistema coge toda la información para después poder tomar decisiones. Las más importantes son las siguientes:

- **Cliente** : nos indicarán sus condiciones físicas iniciales (peso, altura, edad...), sus hábitos en el trabajo y fuera de él, su salud (lesiones y dieta) y el objetivo del programa de entrenamiento.
- **Internet** : donde hemos recopilado una serie de ejercicios, de los cuales tenemos su dificultad, los músculos implicados, repeticiones...

También tendremos como fuente del conocimiento el sentido común, partiendo de nuestra propia experiencia.

## Objetivos del sistema

Los objetivos que debe cumplir nuestro sistema SBC son los siguientes:

- Obtener toda la información de los nuevos clientes: El sistema debe recolectar la información del usuario a partir de una serie de preguntas que el cliente debe ser capaz de responder.
- Inferir toda la información posible de los clientes: El sistema debe inferir la intensidad inicial de los ejercicios a recomendar que el nuevo cliente puede soportar a partir de la información ingresada.
- Presentar adecuadamente un programa de ejercicios: El sistema debe generar, a partir de una serie de ejercicios almacenados previamente en el sistema, una serie de secuencias diarias que cubrirán una semana de ejercicios enfocados en la intensidad inicial recomendada para el usuario, así como en su objetivo y en su tiempo disponible.

## Resultados esperados por el sistema

El sistema generará un programa de ejercicios de 7 días, donde se mostrará qué ejercicios hacer y cuantas repeticiones de cada uno, indicando los músculos implicados y el motivo por el cual han sido escogidos.

Estos resultados son generados a partir de la evaluación de una serie de instancias almacenadas previamente en el sistema, así como de la información brindada por el usuario relativo a sus hábitos, problemas de salud, alimentación y su objetivo.

# Conceptualización

Esta fase consiste en obtener la información necesaria para decidir el conocimiento que utilizará nuestro sistema para resolver el problema. Para obtener toda la información necesaria descomponemos la conceptualización en 3 fases:

- Detallar los elementos principales del dominio que el sistema ha de conocer y sus características para realizar una descripción informal de una ontología.
- Descomponer el problema en subproblemas mediante refinamientos sucesivos.
- Detallar el flujo de razonamiento y las entradas y salidas de cada subproblema.

## Elementos principales del dominio

En esta fase debemos recopilar los elementos principales del dominio y sus características principales con tal de obtener una ontología inicial. Los elementos principales que hemos obtenido son los siguientes:

- **Persona :**
  - Nombre
  - Género
  - Altura
  - Peso
  - Edad
  - Hábitos
  - Salud
  - Dieta
  - Objetivo
  - PPM1
  - PPM2
  - presión sanguínea mínima
  - presión sanguínea máxima
- **Ejercicio:**
  - Objetivo
  - Tipo de ejercicio
  - Nombre
  - Grupos musculares que ejercita
  - Número de repeticiones
  - Dificultad
  - Aparato
  - Si es recomendado o no

Otro elemento principal del dominio es el **Programa**, que es la salida del proyecto, el cual está formado por una secuencia de ejercicios distribuidos en los días de una semana. También tenemos el concepto de **Hábitos**, los cuales los dividimos en los hábitos en el trabajo y fuera del trabajo. Finalmente, también **Salud**, donde rescatamos principalmente su Dieta y sus Problemas de Salud.

## Problemas y subproblemas

Hemos identificado los siguientes subproblemas con el fin de resolver el problema. El proceso de resolución se basa en la resolución secuencial de estos subproblemas:

### **Conocer la información del cliente nuevo**

El primer subproblema consiste la recolección de datos a partir de una serie de preguntas al nuevo cliente con el fin de ingresar al sistema toda la información posible del nuevo cliente. Estas preguntas estarán relacionadas a la información personal, los hábitos (dentro y fuera del trabajo), los problemas de salud, la dieta y el objetivo del programa de ejercicios a obtener.

### **Análisis sobre la intensidad inicial**

A partir de toda la información recopilada por las preguntas del subproblema anterior, el sistema será capaz de inferir la intensidad inicial de los ejercicios que se recomiendan para el cliente nuevo.

### **Obtención de ejercicios adecuados**

A partir de la intensidad inicial apta para el cliente nuevo, del objetivo del programa y de sus problemas de salud obtendremos el conjunto de ejercicios adecuados para el cliente en cuestión

### **Presentación de la solución al cliente**

Después del filtrado de ejercicios se obtiene una serie de ejercicios recomendados. A partir de este conjunto y del tiempo disponible diario del cliente el sistema los agrupará en secuencias. Seguidamente, se mostrará la rutina de ejercicios a realizarse cada día, además de cuántas repeticiones o tiempo debe dedicarle a cada uno de ellos.

## Flujo de razonamiento

El cliente, a través de las preguntas dirigidas por el sistema, introducirá sus datos personales, información general sobre sus hábitos (en el trabajo y fuera de este), su dieta (alimentación por grupo de nutrientes y frecuencia), sus problemas de salud o lesiones y su objetivo para con el programa de ejercicios personalizado. A partir de toda esta información introducida por el cliente, el sistema analizará e inferirá la intensidad inicial con la que debe iniciar.

Esta intensidad inicial de los ejercicios viene inferida, en primer lugar, de forma independiente, dependiendo de cada uno de los campos que determinan la condición física de la persona como: la dieta, los hábitos y la condición física del usuario. Luego, se determina una intensidad inicial general que evalúa cada una de las intensidades independientes obtenidas anteriormente.

Una vez tenemos la intensidad inicial obtendremos un conjunto de ejercicios con la misma intensidad que puede soportar el cliente y el mismo objetivo que nos ha indicado. También eliminaremos todos los ejercicios que puedan contraindicar el problema de salud que nos ha indicado el cliente, en caso lo tuviera.

Finalmente, se organizan los ejercicios en grupos de secuencias destinados a un día en específico respetando el tiempo disponible que tiene el cliente cada día. Estas secuencias cubrirán un total de siete días.



# Ejemplo de conocimiento experto

Este tipo de sistemas se destacan principalmente por el input de información y conocimiento que recibe como input o que ya se encuentra almacenada, la cual se basa principalmente en la experticia de alguna persona que domina un campo específico.

Un ejemplo del conocimiento experto extraído del dominio en el que el sistema se basa para ser capaz de inferir a partir de los datos ingresados es la comprensión del concepto de la frecuencia cardíaca de una persona para poder evaluar la intensidad de los ejercicios a recomendar a una persona dependiendo de características básicas personales como las siguientes:

- Edad
- Género
- Pulsaciones por minuto en reposo
- Pulsaciones por minuto después de actividad física

En el sistema presentado en el presente informe hemos establecido que el usuario ingrese las pulsaciones por minuto que registra después de un minuto de carrera sostenida.

A partir de los datos ingresados, se procede a calcular la frecuencia cardíaca máxima de la persona de acuerdo a la siguiente fórmula:

K: Constante de evaluación

Si Género=Femenino entonces  $K=226$

Si Género=Masculino entonces  $K=220$

Frecuencia cardíaca máxima =  $K - \text{edad}$

La frecuencia cardíaca máxima es una estimación del valor más elevado de pulsaciones por minuto al que el corazón de una persona es capaz de bombear bajo la máxima presión, causado en este caso por el ejercicio que realiza.

Es importante conocer la frecuencia cardíaca máxima para poder establecer la intensidad a la que una persona debe realizar sus ejercicios para poder cumplir su objetivo dentro del entrenamiento. Por ello, las intensidades se distribuyen en grupos que van desde muy suaves hasta lo máximo. En nuestro caso, sólo consideraremos tres de estas intensidades para estar en el mismo contexto de las otras evaluaciones de intensidad que realiza nuestro SBC: baja, media, alta.

Se sabe que bajo este tipo de entrenamiento una persona realiza ejercicio y se mide las pulsaciones por minuto cada cierto tiempo, si sus pulsaciones están dentro del rango de la intensidad muy alta, significa que está trabajando dentro de un porcentaje de su frecuencia cardíaca máxima, en ese caso, se acerca a la fracción superior.

En este sistema nos basamos en esta evaluación (pero en sentido inverso) para poder obtener una recomendación de intensidad de ejercicio al que la persona debe someterse. Por ejemplo, si una persona después de correr mide sus pulsaciones por minuto (PPM2) y este se encuentra dentro del rango que constituye aproximadamente el 70%-85% de su frecuencia cardíaca máxima entonces la intensidad de ese ejercicio realizado es alta. En este caso, si en un minuto de carrera sostenida (ejercicio poco intenso) se encuentra en ese rango significa que la persona no está acostumbrada a ese ritmo por lo que se recomienda un nivel de intensidad mucho menor.

En ese sentido utilizamos la siguiente fórmula para obtener la correspondiente intensidad recomendada dependiendo de las pulsaciones por minuto de una persona registrados en dos estados:

t: Factor de deducción

PPM1: Pulsaciones por minuto en reposo

PPM2: Pulsaciones por minuto después de ejercicios (un minuto de carrera sostenida)

F\_max: Frecuencia cardíaca máxima

U: Variable de evaluación

$$U(t) = PPM1 + [ ( F\_max - PPM1 ) * t ]$$

Una vez contamos con ese valor evaluamos la intensidad de acuerdo a la siguiente tabla:

Tmin	Tmax	Evaluación	Intensidad recomendada
0%	49%	$U(Tmin) \leq PPM2 \leq U(Tmax)$	alta
50%	69%		media
70%	100%		baja

# Formalización

Esta fase consiste en desarrollar la construcción de la ontología en la que se basará todo el conocimiento del sistema que describe el presente informe. Para llevarlo a cabo, descomponemos esta fase en tres etapas:

- Desarrollo de la ontología
- Elementos de la ontología
- Métodos de resolución
- Justificación

## Desarrollo de la ontología

Como bien sabemos, no existe una ontología universal que se pueda usar en todos los campos de conocimiento existentes, por ende crearemos una que contemple todas las características de este problema en específico.

A continuación detallaremos los primeros pasos para la construcción de la misma:

### Determinar dominio y cobertura

Tal y como se puede deducir del enunciado del problema, este se enfoca en una realidad relativa a la actividad física, salud y calidad de vida de una persona. Así el dominio tanto desde el punto de vista de un experto como de un ingeniero es la misma: Medicina.

Sin embargo la cobertura de la misma es amplia, ya que la decisión de una recomendación de secuencias de ejercicios se basará en varios aspectos de la cotidianidad de una persona, es decir, se toma en cuenta diversos aspectos como la alimentación, hábitos diarios (dentro y fuera del trabajo), problemas de salud, ejercicios que contraindiquen ciertas actividades, músculos relacionados a ciertos problemas, ejercicios relacionados a ejercitar ciertos grupos musculares, entre otros.

En el presente proyecto calificamos todos esos factores como importantes.

## Determinación del uso de la ontología

La ontología es importante, en primer lugar, para poder entender y redefinir (si fuese necesario) los conceptos que abarcan tanto el problema como la solución del mismo. En segundo lugar, es primordial para poder representar y almacenar el conocimiento base que necesita el sistema con el objeto de obtener deducciones y así poder tomar decisiones.

En este caso, lo utilizamos principalmente para establecer las relaciones que tienen cada uno de los conceptos. Asimismo, para almacenar la información personal ingresada por el usuario. Previamente, también está almacenada la información de los ejercicios que se puede recomendar, así como los problemas de salud a los que pueden contraindicar, o los grupos musculares a los que ejercitan.

## Determinación de la ontología como medio de respuesta

Una ontología también nos ayuda a poder responder preguntas específicas relacionadas a sus conceptos. En este caso, nuestra ontología nos brinda respuesta sobre la condición física de una persona, sus hábitos, sus problemas de salud y su dieta. Por otro lado, también nos da información acerca de su objetivo dentro del programa de recomendaciones de ejercicios y sobre los ejercicios que este debería realizar.

## Mantenimiento de la ontología

La presente ontología será únicamente aplicada para el desarrollo de un sistema de recomendación de programas de ejercicios, dentro de la asignatura de Inteligencia Artificial. Su mantenimiento será realizado por el propio equipo de desarrollo del programa durante el tiempo que necesitemos para llevarlo a cabo. No obstante, consentimos el hecho de que pueda ser reutilizado en otro proyecto.

## Elementos de la ontología

La lista de términos importantes que son relevantes a representar en la ontología son los que están relacionados a los datos personales de una persona, así como su salud reflejada en sus hábitos (dentro y fuera del trabajo), su dieta, y problemas de salud. En el mismo sentido también son importantes los conceptos que relacionan a un ejercicio en sí, tales como el tiempo que se necesita realizar un ejercicio, si necesita una herramienta o no, si contraindica a una enfermedad específica o si ejercita un grupo muscular.

## Clases y jerarquía

Tal como se deduce del problema, los términos principales serán los que constituyan una clase principal, por ende una jerarquía mayor, dentro de nuestra ontología.

A continuación presentamos una visión general de la ontología que planteamos para cubrir los conceptos del enunciado del problema y de la solución misma.

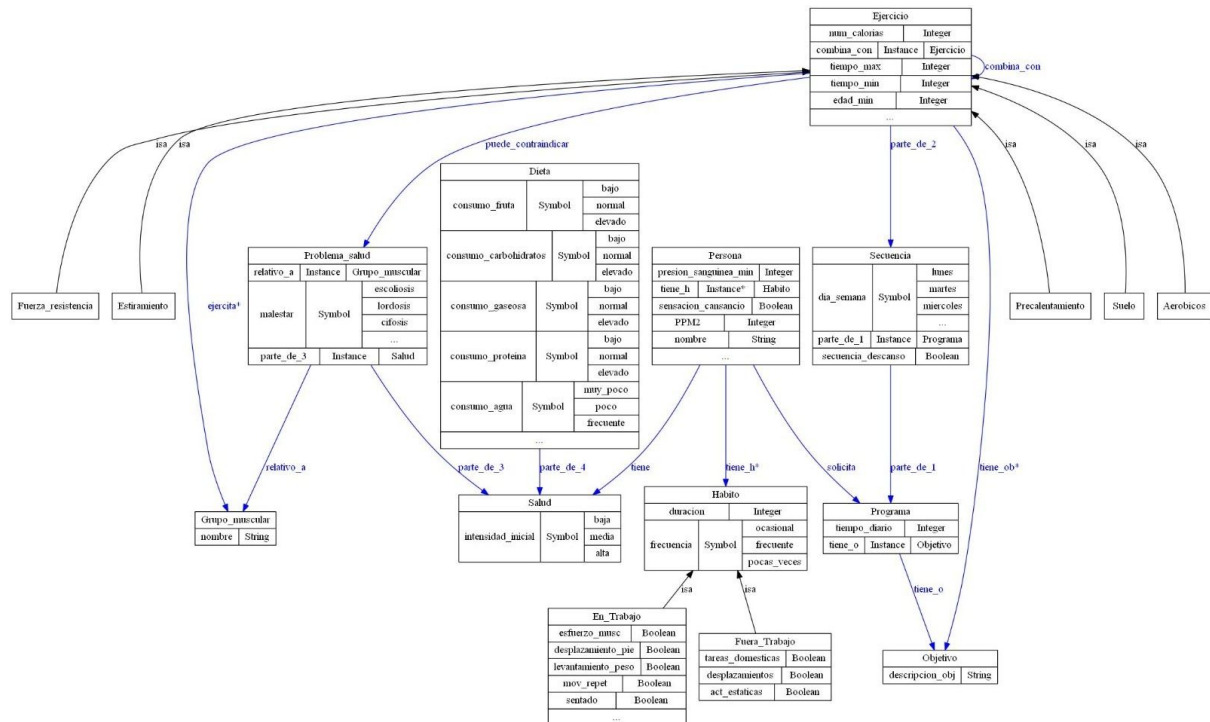


Figura 1.- Ontología para el Sistema Basado en Conocimientos para la recomendación del programa de ejercicios (exportado en Protegé)

A continuación describiremos las entidades de forma independiente en una vista más específica.

## Persona

Persona		
presion_sanguinea_min		Integer
tiene_h	Instance*	Habito
sensacion_cansancio		Boolean
PPM2		Integer
nombre		String
...		

Figura 2.- Clase Persona (exportado en Protegé)

Sus atributos son los siguientes:

- nombre
- genero
- altura
- peso
- edad
- presion\_sanguinea\_min
- presion\_sanguinea\_max
- tiene\_h
- sensacion\_cansancio
- tirantez\_muscular
- PPM1
- PPM2
- tiene
- IMC
- solicita

## Salud

Salud		
intensidad_inicial	Symbol	baja
		media
		alta

Figura 3.- Clase Salud (exportado en Protegé)

Sus atributos son los siguientes:

- intensidad\_inicial: Nivel de Intensidad inicial de ejercicio al que el usuario está acostumbrado. Depende de sus hábitos

## Hábito

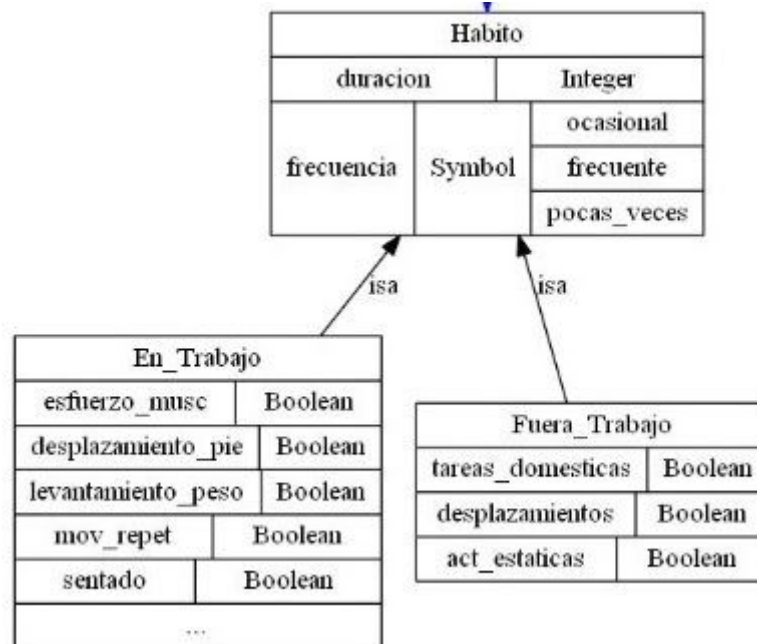


Figura 4.- Clase Hábito (exportado en Protegé)

Sus atributos son los siguientes:

- duracion
- frecuencia
- esfuerzo\_musc
- desplazamiento\_pie
- levantamiento\_peso
- sentado
- mov\_repet
- de\_pie
- tareas\_domesticas
- desplazamientos
- act\_estaticas

## Dieta

Dieta		
consumo_fruta	Symbol	bajo
		normal
		elevado
consumo_carbohidratos	Symbol	bajo
		normal
		elevado
consumo_gaseosa	Symbol	bajo
		normal
		elevado
consumo_proteina	Symbol	bajo
		normal
		elevado
consumo_agua	Symbol	muy_poco
		poco
		frecuente
...		

Figura 5.- Clase Dieta (exportado en Protegé)

Sus atributos son los siguientes:

- consumo\_fruta
- consumo\_carbohidratos
- consumo\_proteina
- consumo\_gaseosa
- consumo\_agua
- consumo\_grasa
- consumo\_verduras
- consumo\_picante
- abuso\_sal
- parte\_de\_4



## Problema de salud

Problema_salud		
relativo_a	Instance	Grupo_muscular
malestar	Symbol	escoliosis
		lordosis
		cifosis
		...
parte_de_3	Instance	Salud

Figura 6.- Clase Problema de salud (exportado en Protegé)

Sus atributos son los siguientes:

- relativo\_a
- malestar
- parte\_de\_3

## Grupo muscular

Grupo_muscular	
nombre	String

Figura 7.- Clase Grupo muscular (exportado en Protegé)

Sus atributos son los siguientes:

- nombre

## Programa

Programa		
tiempo_diario	Integer	
tiene_o	Instance	Objetivo

Figura 8.- Clase Programa (exportado en Protegé)

Sus atributos son los siguientes:

- tiempo\_diario
- tiene\_o

## Secuencia

Secuencia		
dia_semana	Symbol	lunes
		martes
		miercoles
		...
parte_de_1	Instance	Programa
secuencia_descanso		Boolean

Figura 9.- Clase Secuencia (exportado en Protegé)

Sus atributos son los siguientes:

- secuencia\_descanso
- dia\_semana
- parte\_de\_1

## Ejercicio

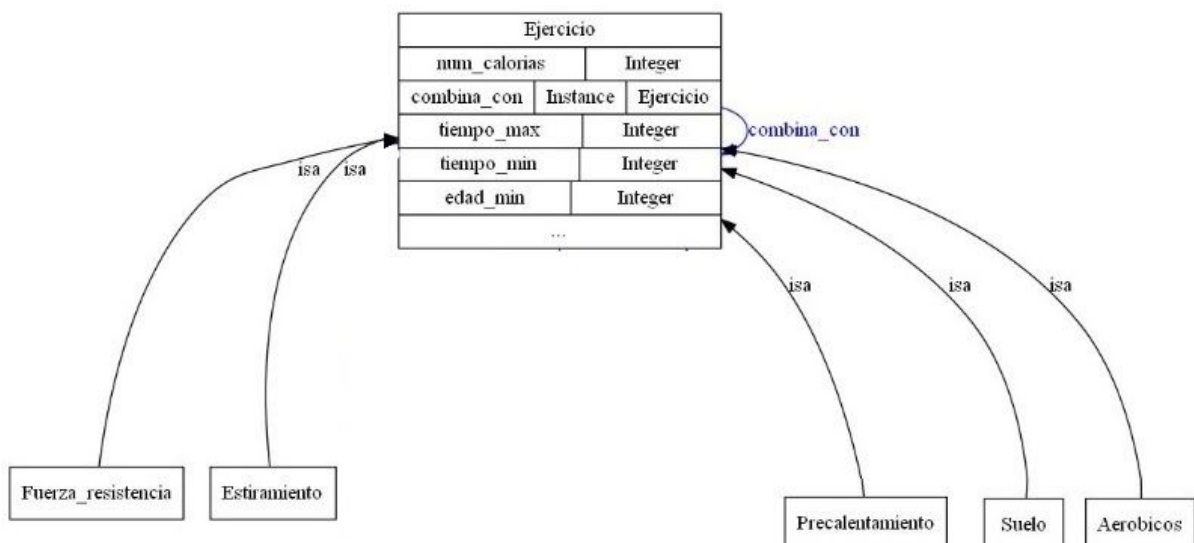


Figura 10.- Clase Ejercicio (exportado en Protegé)

Sus atributos son los siguientes:

- nombre\_ejercicio
- descripcion
- aparato
- dificultad
- edad\_min
- edad\_max
- tiene\_ob
- parte\_de\_2
- repeticiones\_min
- repeticiones\_max
- num\_calorias
- puede\_contraindicar
- ejercita
- es\_recomendado
- tiempo\_min
- tiempo\_max
- combina\_con

## Objetivo

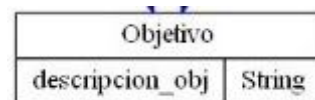


Figura 11.- Clase Ejercicio(exportado en Protegé)

Sus atributos son los siguientes:

- descripcion\_obj

# Métodos de resolución

En el presente proyecto hemos escogido a la clasificación heurística como la metodología de resolución de nuestro Sistema Basados en Conocimientos ya que el problema básicamente es de análisis. Por consiguiente es el experto quien decide qué conjunto de soluciones son aptas para un problema específico, en este caso se recopila el conjunto de instancias que van acorde a las recomendaciones del sistema.

Nuestra problemática constituye uno de análisis ya que su solución se basa en las mini-soluciones que previamente estén registrados en el sistema, es decir, de los ejercicios que estén almacenados en la base de datos del sistema, y de los cuales se tiene que tomar la decisión de si son aptos o no para que sean recomendados hacia el usuario.

Cabe precisar que el equipo que desarrolla este proyecto no está contemplando muchos de los ejercicios existentes y quizá tampoco se hayan tomado en cuenta muchos factores influyentes, tal como el hecho de que algún grupo muscular se vea afectado por algún ejercicio incluido en el sistema pero esta relación no haya sido planteada. Esto básicamente se da por la gran cantidad de información existente y al objetivo educativo que tiene este proyecto.

Por otro lado, siguiendo la línea de la clasificación heurística, es importante mencionar que esta se divide en tres etapas, las cuales describiremos a continuación:

## Abstracción de datos

El sistema de recomendación de programas de ejercicios empieza a recolectar la información del usuario. Estos son los datos que inicialmente utilizamos como el input de esta etapa. Como no queremos basar la recomendación en datos explícitos, se recolectan los datos y se almacenan directamente a la única instancia de la clase Persona. Luego empieza la rutina de inferencia de datos donde a través de una serie de reglas se van abstrayendo datos hacia un concepto más general.

Como se explicó anteriormente, almacenados esa información en una instancia del sistema, sin embargo, no lo volvemos a consultar más (existen excepciones) una vez que ya se ha realizado la abstracción de los datos.

Un ejemplo de la abstracción de datos aplicado en el sistema es el caso del índice de masa corporal (IMC) de una persona, el cual depende de su peso y su altura. Estos dos últimos son ingresados por el usuario a través de las preguntas que se realizan en una especie de formulario. Una vez que se obtienen se almacenan, se calcula el IMC y el resultado se clasifica entre cuatro categorías “obeso”, “sobrepeso” “normopeso” y “bajo peso” dependiendo de los rangos de IMC que abarca cada uno.

En el mismo sentido, así como se han incluido reglas para lograr la abstracción de los datos concretos, también se han utilizado reglas que abstraen aún más la información basándose en datos previamente abstraídos.

Cabe mencionar que el módulo (en el fichero que se ejecuta en CLIPS) que se encarga de este proceso de abstracción es el módulo “inferir\_datos”. Sin embargo, hay abstracciones menos complejas que se realizaron directamente en la consulta de los datos, es decir en el módulo “hacer\_preguntas”.

## Asociación heurística

En este proceso ya tenemos todos los datos en forma abstracta, por lo que nuestro Problema Concreto ya se encuentra en el plano del Problema Abstracto. En ese sentido, debemos proceder a transformarlo en una Solución Abstracta.

Si nos enfocamos en el proyecto, este proceso se lleva a cabo cuando procedemos a inferir la intensidad general de los ejercicios que se va a recomendar al usuario de acuerdo a las demás intensidades previas. Ya que la intensidad de los ejercicios ya es parte de la solución del problema.

Asimismo, una vez que contamos con la intensidad general, se procede a capturar los ejercicios que van acorde a esa recomendación. Como hasta ese momento no se recurre a la información concreta recolectada al inicio del programa, podemos decir que aún nos encontramos en el plano de Solución Abstracta.

Cabe mencionar que el módulo (en el fichero que se ejecuta en CLIPS) que se encarga de este proceso de asociación heurística es el módulo “modulo\_inferencia\_ejercicios”.

## Refinamiento

Esta es la última etapa de la clasificación heurística, la cual no siempre se requiere. Sin embargo, en este proyecto sí lo vemos necesario ya que requerimos de la información concreta ingresada al inicio de la ejecución del programa para poder generar la secuencia de ejercicios por día para completar así el programa completo.

Requerimos por ejemplo de datos concretos como el objetivo del programa y el tiempo diario que la persona tiene disponible para poder seleccionar los ejercicios y para poder organizarlos dentro de las secuencias diarias.

Cabe mencionar que el módulo (en el fichero que se ejecuta en CLIPS) que se encarga de este proceso de asociación heurística es el módulo “modulo\_inferencia\_ejercicios”.

# Implementación

Para cuando ya estamos en esta etapa significa que contamos con la ontología más adecuada para representar el conocimiento. Asimismo, ya hemos definido el método de resolución que se adapta al problema y los procesos de razonamiento que seguirá el sistema. Por ello, ahora nos enfocaremos en la implementación de todo lo anterior, es decir, de la ontología. Además encajaremos la descomposición de subproblemas con la metodología de clasificación heurística. Utilizaremos el lenguaje CLIPS para poder llevarlo a cabo.

## Ontología

En el presente proyecto se han seguido las pautas recibidas tanto en las clases teóricas como en los laboratorios de la asignatura. Una de ellas fue el apoyo para el desarrollo de la ontología en el programa Protege, el cual tiene herramientas para poder exportar las clases e instancias creadas para importarlas directamente en nuestro fichero del sistema.

Una vez que contamos con las clases (en el fichero .Ponts) y las instancias (en el fichero .Pints) procedemos al desarrollo del programa en CLIPS, un lenguaje de programación basado en reglas, aplicado principalmente para la construcción de los SBC.

En este fichero .clp hemos incluido la ontología, las instancias y los diversos módulos que se explicarán más a detalle en las siguientes secciones. Asimismo, se han utilizado una serie de funciones extras y de templates que nos ayudan a tratar mejor la implementación.

Dado que las diversas intensidades deducidas de las características de la persona no están incluidas en la ontología, utilizaremos de templates para almacenar su información en forma local en tiempo de ejecución. Lo importante al final es la intensidad general que se basa en esas otras, es por ello que es la única que se incluye en la ontología.

# Módulos

En esta sección del informe presentaremos los módulos del programa y los defrules que los conforman (en adelante “funciones”) que han sido implementados en el archivo .clp.

## Módulo 1: main MAIN

En este módulo está compuesto sólo por dos funciones básicas: “comienzo” y “existe\_persona”. Estas se definen con el objetivo de presentar el sistema y de dar inicio al mismo, creando un hecho específico que da paso al inicio de muchas otras reglas en los siguientes módulos.

## Módulo 2: hacer\_preguntas

Este módulo es invocado por la última regla del módulo Main. Consta de 9 funciones que nos ayudan a interactuar con el usuario por medio de preguntas sobre sus hábitos, alimentación, problemas de salud, condición física y demás. La información recolectada es almacenada en una única instancia de la clases Persona de la ontología.

Este módulo es muy importante ya que sin este el sistema no puede recolectar conocimiento sobre la persona por lo que no será capaz de inferir algún tipo de ejercicio que conforme una secuencia para el programa.

Las preguntas están planteadas de forma secuencial para que la persona pueda encontrar secuencia y lógica a sus respuestas. Además se le da indicaciones y opciones para que el usuario sepa qué tipo de respuesta es la que se espera en una pregunta específica.

## Módulo 3: inferir\_datos

Este módulo está dedicado a la inferencia de información a partir de los datos recolectados en el módulo anterior. Está conformado por 48 funciones. Tal como hemos descrito en el apartado anterior, este módulo da paso a la abstracción de datos para poder pasar de un plano de problemas concreto al plano de problema abstracto.

Este módulo nos ayuda primordialmente a la creación del perfil general de la persona, así como a plantear un camino hacia la construcción de un programa que vaya acorde a sus necesidades.

En esta sección del programa se evalúan además algunas abstracciones base que sirven para abstracciones de mayor nivel, como es el caso de la Intensidad general que se infiere de las intensidades que son inferidas en este mismo módulo pero que sirven como input para el general.

Es en este módulo donde se usan los deftemplates que almacenan la intensidad de ejercicio recomendado después de analizar cierta característica:

```
..... NIVEL RECOMENDADO DE EJERCICIO
.....

;;; deftemplate para determinar la intensidad de ejercicio dependiendo de su genero, edad
y pulsaciones por minuto

(deftemplate intensidad_Pulsaciones
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))
)

;;;deftemplate para almacenar la intensidad inicial dependiendo de su dieta

(deftemplate intensidad_Dieta
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))
)

;;;deftemplate para almacenar la intensidad inicial dependiendo de sus habitos

(deftemplate intensidad_HabitosTrabajo
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))
)

(deftemplate intensidad_HabitosFuera
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))
)
```



```
;;; deftemplate para almacenar la intensidad inicial dependiendo de su condicion actual  
(imc, sensaciones despues de correr, presion_sanguinea)
```

```
(deftemplate intensidad_Condicion  
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))  
)
```

```
..... GENERAL  
.....
```

```
;;; deftemplate para almacenar la intensidad inicial recomendada resumiendo todos los  
factores anteriores
```

```
(deftemplate intensidad_CondGeneral  
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot media (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))  
)
```

En el caso del deftemplate de la presión sanguínea no se almacena una intensidad recomendada, sino que realmente se almacena la evaluación de la presión de la persona dependiendo de si se encuentra en los rangos de presión determinados para que sea alta, baja o normal.

```
(deftemplate presion_sanguinea  
  (slot alta (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot normal (type SYMBOL)(allowed-values TRUE FALSE)(default FALSE))  
  (slot baja (type SYMBOL)(allowed-values TRUE FALSE) (default FALSE))  
)
```

## Módulo 4: modulo\_inferencia\_ejercicios

Este módulo representa la asociación heurística, ya que a partir de la información abstracta obtenida en el módulo anterior, se da el paso a la búsqueda de los ejercicios que van acorde a las restricciones de la condición de la persona. Es así como se van filtrando poco a poco los que no cumplen con la intensidad que se recomienda al usuario que solicita un programa de ejercicios.

En este módulo también se utilizan `deftemplates` con el objeto de almacenar temporalmente todos los ejercicios que van cumpliendo con los requisitos del usuario.

```

.....
))))))))))))))))))
PARA ALMACENAR LOS EJERCICIOS RECOMENDADOS

(deftemplate instancias_ejercicios
  (multislot inst_ej (type INSTANCE)(allowed-classes Ejercicio)(cardinality 1
?VARIABLE))
  (slot cantidad (type INTEGER))
)

(deftemplate instancias_problema_sal
  (multislot inst_pr (type INSTANCE)(allowed-classes Problema_salud)(cardinality 1
?VARIABLE))
  (slot cantidad (type INTEGER))
)

(deftemplate instancias_lunes
  (multislot inst_lun (type INSTANCE)(allowed-classes Ejercicio)(cardinality 1
?VARIABLE))
  (slot cantidad (type INTEGER))
)

```

En el caso del deftemplate “instancias\_lunes”, su estructura se repite para los demás días de la semana, a excepción del día jueves. Esto debido a que como equipo hemos decidido que el día jueves será considerado como un día de descanso dentro de la rutina de entrenamiento.

## Módulo 5: modulo inferencia programa

Este módulo se utiliza para organizar los ejercicios que serán recomendados al usuario. Sin embargo, se debe tener en cuenta que los ejercicios pertenecen a una secuencia, para que el usuario los practique de forma diaria, es por ello que se deben organizar una serie de 7 secuencias para cubrir una semana completa de entrenamiento. Asimismo se debe considerar el tiempo diario que la persona tiene disponible para sus ejercicios.

Este módulo representa el paso de la solución abstracta a un plano de solución concreta, es decir, representa el Refinamiento de la solución. Por lo que es en este módulo donde se imprime en pantalla la respuesta del SBC para recomendación de programas de ejercicio.

## Prototipos

En esta parte de la documentación nos centraremos en explicar detalladamente el proceso de prototipado rápido y desarrollo incremental que vamos a realizar en este proyecto.

Como es la primera vez que todos los miembros del equipo manejamos un lenguaje de programación basado en reglas orientado a un Sistema Basado en Conocimientos, cabe mencionar que nos ha costado entender su sintaxis y funcionamiento. Así como la manipulación de los datos y entendimiento de errores.

### Inicial

Como equipo nos concentramos en trabajar de forma incremental enfocandonos de que el proceso de decisión del sistema se muestre en pantalla para que se pueda entender el camino que se ha ido tomando, de tal forma que podamos explicar la recomendación final del SBC. Nuestro primer prototipo se basó en la recolección de datos.

A pesar de la limitación mencionada anteriormente, fuimos aprendiendo a medida que avanzábamos con el proyecto mismo, entonces hasta este momento ya sabíamos interactuar con el usuario y podíamos almacenar su información en las clases de la ontología exportada de Protegé.

### Extendido

En esta versión del proyecto fuimos incrementando la funcionalidad del sistema ya que añadimos las reglas de inferencia para comenzar con la abstracción. Como se puede observar en el fichero .clp del proyecto, son estos módulos de inferencia los que tienen mayor extensión, ya que decidimos trabajar por niveles, siendo la primera la inferencia de las intensidades base dependiendo de ciertas características de la persona. Para luego comenzar a inferir un nivel de intensidad de ejercicios más general.

Cabe mencionar que en esta etapa surgieron muchos problemas ya que nuestra comprensión del lenguaje de programación estaba un poco más afianzada. Sin embargo, mientras más entendíamos se nos ocurrían ideas un poco más complejas relacionando el problema con una implementación en otro lenguaje de programación, e intentando buscar la manera de hacerlo en Clips. Por ello nos encontramos con muchos errores que poco a poco fuimos solucionando.

## Final

En esta versión del proyecto añadimos la parte de la organización de los ejercicios en secuencias de tal forma que se pueda abarcar el tiempo del usuario en casi su totalidad. Asimismo añadimos más instancias de Problemas de salud, Ejercicios y sus relaciones. Esto para probar la funcionalidad del programa.

Sin embargo, en este punto de la implementación se presentaron muchos errores de ejecución que por ahora no hemos podido resolver. Hasta esta parte hemos conseguido que los ejercicios sean seleccionados dependiendo del objetivo de la persona, intensidad recomendada, problemas de salud relacionados y grupos musculares. Asimismo hemos procurado que toda la toma de decisiones se refleje en la ejecución y se pueda justificar la recomendación final del SBC.