



## Taller 2: Lógica Difusa aplicada a un sistema de apoyo en diagnóstico de hardware.

En experiencias previas se ha logrado aplicar conceptos de LPO para desarrollar una suerte de herramienta para el diagnóstico en desperfectos en componentes de un Computador. Para esta segunda experiencia se utilizará la misma temática, pero con la diferencia que en esta ocasión esta última abordada desde otra perspectiva, en específico, mediante el uso de **lógica difusa**, también conocida como **fuzzy logic**.

Uno de los aprendizajes que deja (y se esperaba que notarán) la implementación de la primera experiencia, es que la lógica de primer orden es limitada respecto a lo que se puede llegar a modelar, por ejemplo en un diagnóstico de componentes, *¿Cuántos grados por encima de lo permitido se encontraba el sistema con falla de refrigeración?*, ¿A qué nos referimos con “grados permitido” ?.

En general depende del tipo de gabinete y las propias características del resto de componentes, por lo que si un computador presenta una temperatura de 50° C, puede que se interprete esta temperatura normal, así como puede ser una temperatura alta. Por ende en este tipo de situaciones se presenta un grado de **ambigüedad**, el cual es aprovechado y empleado en lógica difusa para representar esta determinada situación desde un enfoque más cercano a la realidad.

En Lógica difusa existe una función de pertenencia ( $\mu$ ) que toma cualquier elemento  $R^n$  de un **conjunto de discurso** ( $X$ ) y obtiene infinitos valores posibles de pertenencia, dentro del intervalo  $[0, 1]$ . Es decir por cada elemento del conjunto discurso, este tiene un valor de pertenencia, el cual se interpreta como su grado de pertenencia al conjunto difuso.

$$\mu_A : U \rightarrow [0, 1]$$

Así, se define el conjunto difuso ( $A$ ) como el conjunto de todos los elementos del conjunto de discurso y sus respectivos valores de pertenencia,

$$X = \{x_1, x_2, x_3, x_4, \dots, x_n\}, x \in U, \forall i = 1, 2, 3, \dots, n$$

$$A = \{\mu_A(X) / X\}$$

$$A = \{\mu_A(x_1) / x_1, \mu_A(x_2) / x_2, \mu_A(x_3) / x_3, \dots, \mu_A(x_n) / x_n\}$$

A partir de los elementos anteriores, la lógica difusa permite que **agentes inteligentes** pueden tomar decisiones o acciones en función de las entradas recibidas, basándose en los siguientes pasos:



1. **Fusificación:** Se reciben las entradas desde el entorno y se fusifican. Consiste en llevar el valor numérico o **nítido** de la entrada y proyectarlo a un valor lingüístico, con su respectivo valor de pertenencia de algún conjunto difuso.
2. **Sistema de Inferencia:** Los valores fusificados se aplican a un conjunto de reglas predefinidas para el sistema, aplicando **algún método de inferencia** y se obtiene la **consecuencia lógica** de la entrada, por ende la acción del agente.
3. **Desfusificación:** Al resultado de la inferencia realizada en el punto anterior, se le aplica **algún método para calcular los resultados nítidos** que toma el agente y realizar las acciones pertinentes e interactuar con el entorno. Algunos de los métodos más empleados son el de Centro de Área (COA) y Bisector de Área(BOA).

En el taller 1, donde se puso en práctica la lógica de primer orden (LPO), para la implementación de la base de conocimiento y entrada a las consultas usted, consideró como referencia la información de posibles errores y defectos de los componentes, disponible en [ScienceDirect Hardware errors](#), [ticinformatica](#) y [prodatarecover](#). Lo que principalmente se apunta con esta información es **rendimiento**, con los componentes:

- Sistema de refrigeración
- Memoria RAM
- Procesador
- Placa madre
- Memoria física (Disco Duro)
- Tarjeta gráfica
- Fuente de poder

***Puede agregar más componentes si así lo estima necesario.***

Mientras, se cuenta con las siguientes reglas obtenidas de acuerdo a la opinión de algunos usuarios que han acudido a las páginas anteriormente mencionadas:

1. **Si** el sistema de refrigeración **y** la memoria RAM **y** el procesador **y** la tarjeta de gráfica tienen un desempeño deficiente **y** la fuente de poder tiene un desempeño moderado **entonces** el rendimiento debe ser deficiente.
2. **Si** el procesador tiene un rendimiento moderado **y** el sistema de refrigeración es malo **y** la memoria RAM es moderada **y** la placa madre es buena **y** la tarjeta gráfica es mala **y** la fuente de poder es buena **entonces** el rendimiento debiese ser bueno.
3. **Si** la placa madre es óptima **y** el procesador es óptimo **y** la memoria RAM es moderada **y** la fuente de poder es moderada **entonces** el rendimiento debe ser bueno.
4. **Si** la fuente de poder es moderada **o** la memoria física es buena **o** la placa madre es moderada **o** el sistema de refrigeración es óptimo **entonces** el rendimiento debe ser moderado.



5. **Si** la memoria RAM es óptima **y** la placa madre es óptima **entonces** el rendimiento debe ser óptimo.
6. **Si** la placa madre es deficiente **o** el sistema de refrigeración es moderado, **y**, el procesador es bueno **o** la memoria RAM es buena, **entonces** el rendimiento debe ser malo.
7. **Si** la tarjeta gráfica es moderada **y** el procesador es moderado **y** la placa madre es buena **y** el sistema de refrigeración es óptimo **y** la memoria ram es moderada **entonces** el rendimiento debe ser moderado.

En este taller se debe modelar el diagnóstico de **rendimiento esperado**, aplicando lógica difusa para determinar el grado de rendimiento que se tiene en el computador con los componentes (deficiente, malo, moderado, bueno y óptimo, en orden de rendimientos, de peor a mejor), con el fin de ver el rendimiento de la configuración de componentes ingresada. Para el desarrollo de la aplicación informática debe utilizar el lenguaje de programación **Python** y la librería [scikit-fuzzy](#) para generar representaciones gráficas que expliquen el trabajo realizado con los conjuntos difusos, las reglas de inferencias y los cálculos para la obtención de los resultados.

### Consideraciones del sistema

- Debe considerar la interacción en tiempo real con el usuario (pregunta/respuesta).
- Utilice la implicación o **inferencia de Mamdani**.
- Implemente los conjuntos difusos necesarios para cumplir con las reglas definidas previamente (para definir las funciones de pertenencia, puede hacer todos los supuestos que estime necesarios, siempre siendo estos justificados en detalle).
- Aplique la **desfusificación** con ambos de los siguientes métodos:
  - Centro de Área (COA).
  - Bisector de Área (BOA).Identifique para qué casos funciona mejor uno u otro método, tanto en calidad de resultados como en factor de tiempo.
- Considere escalas de parametrización.
- Cree tres nuevas reglas en base al conocimiento adquirido por la combinación de componentes estudiada en el taller 1. Al menos una de estas reglas debe tener el operador “**o**”; la estructuración de estas reglas debe ser respaldada bibliográficamente y debe tener como mínimo dos variables lingüísticas por regla.
- **Extra:** Las entradas **pueden** ser recibidas por consola o intérprete de python, pero en caso de implementar una interfaz gráfica para estas, se tendrá en consideración al momento de revisión con puntaje adicional.



## Entrega

Debe ser realizada mediante la subida de un archivo en formato ZIP por uno de los integrantes del grupo al curso en la plataforma <https://uvirtual.usach.cl>, el que debe contener:

- Un archivo README, con las instrucciones para ejecutar el programa.
- Debe considerar la entrega de un **informe**, el cual debe contener:
  - Portada
  - Introducción
  - Desarrollo
  - Análisis de resultados, donde dada una entrada establecer si el modelo es adecuado (describe la realidad). Lo anterior para al menos dos casos de prueba.
  - Conclusiones
  - Bibliografía
- Una carpeta con el nombre **src** que contenga el código fuente y los archivos necesarios para ejecutar el programa.

Fecha de entrega: Hasta el 1 de Marzo 2021 hasta las 12:00 (medio día).

Entregas posteriores a la fecha de entrega serán evaluadas con la nota mínima.