**D01:**

**INFORME DEL PROYECTO**

**C3L®**

**PABLO AMOR MOLINA 100495855**

**SERGIO VERDE LLORENTE 100495899**

**HUGO CUEVAS ROMERA 100495962**

ÍNDICE

[Resumen ejecutivo 3](#_Toc166706464)

[Descripción del sistema de inferencia 4](#_Toc166706465)

[Borrosificación 4](#_Toc166706466)

[Evaluación de reglas 4](#_Toc166706467)

[Composición 5](#_Toc166706468)

[Deborrosificación 5](#_Toc166706469)

[Metodología del sistema 5](#_Toc166706470)

[Inicialización 5](#_Toc166706471)

[Bucle general 5](#_Toc166706472)

[Función auxiliar ‘procesapp’ (sistema de inferencia) 6](#_Toc166706473)

[Función auxiliar ‘fuzzy’ 7](#_Toc166706474)

[Cierre final 9](#_Toc166706475)

[Presupuesto 9](#_Toc166706476)

[Anexo 10](#_Toc166706477)

# Resumen ejecutivo

Hemos creado un sistema que facilita a cualquier banco la decisión a la hora de conceder préstamos personales a sus usuarios. Para ello hemos tomado el modelo del sistema de inferencia borrosa de Mamdani, modelo más común e idóneo para resolver situaciones tales como la que se nos presenta en este caso.

Una vez ya hemos establecido la situación inicial y nuestro objetivo, nuestro equipo ha definido unas variables (como el empleo, edad, ingresos, etc.) que, mediante el uso de los conjuntos borrosos, servirán para determinar el nivel de riesgo de una solicitud de préstamo. Estas variables se utilizarán en las reglas de inferencia también definidas por nuestro equipo. Estas reglas determinarán el nivel de riesgo de un préstamo según las distintas características que presente el cliente solicitante en las variables definidas inicialmente.

Finalmente, hemos creado el sistema de inferencia (descrito posteriormente) a partir de las variables y reglas anteriormente mencionadas. Dicho sistema recogerá los datos de las solicitudes de préstamo recibidas por el Banco Pichin y proporcionará los valores de riesgo de cada solicitud, facilitando la decisión en la concesión de los préstamos correspondientes.

Aquí se muestra un diagrama del sistema creado:

Diagrama

Descripción generada automáticamente

# Descripción del sistema de inferencia

Nuestro sistema de inferencia sigue el modelo de inferencia borrosa de Mamdani, el cual se divide en 4 pasos:

1. Borrosificación de las variables de entrada
2. Evaluación de reglas
3. Composición
4. Deborrosificación de las variables de salida



Estos pasos se ejecutarán en cada una de las aplicaciones (solicitudes de préstamo) que reciba el programa para evaluar el nivel de riesgo correspondiente.

## Borrosificación

En este paso recogemos los valores de entrada de la aplicación (localizada en *Applications*) y los transformamos a grados de pertenencia de los conjuntos borrosos establecidos en el archivo *InputVarSets* mostrados anteriormente.

## Evaluación de reglas

Este paso consiste en el cálculo del antecedente y consecuente de cada una de las reglas que se reciben en el archivo *Rules*.

En primer lugar, calcularemos la similitud entre las entradas de la aplicación y los antecedentes de las reglas borrosas. El resultado a este cálculo será el valor mínimo de entre todos los grados de pertenencia de los antecedentes de la regla. Por ejemplo, si tenemos tres antecedentes y sus grados son 0, 1 & 0.5, entonces la similitud de la regla será igual a 0.

Posteriormente, calcularemos el consecuente aplicando el resultado del antecedente a la función de pertenencia del consecuente mediante el método de recorte. Para ello comprobaremos la regla consecuente (en nuestro caso representa los 3 niveles de riesgo posibles), obtenemos su función e iremos recortando dicha función con la similitud del antecedente. Es decir, en aquellos puntos del eje Y donde la consecuente sea mayor que la similitud, la consecuente tomará el valor calculado inicialmente (para todos los puntos del eje X en la gráfica).

Ejemplo visual de una función de pertenencia (similitud = 0.5):

Forma

Descripción generada automáticamente

Función previa Función posterior al recorte

## Composición

En este paso uniremos las funciones de pertenencia resultantes del paso anterior de todas las reglas impuestas, introduciendo como entrada el conjunto de dichas funciones y obteniendo como resultado un conjunto borroso de cada variable de salida.

## Deborrosificación

En este paso final cogeremos el conjunto borroso calculado en el paso de composición y, mediante el método del cálculo del centroide del área del conjunto borroso, obtendremos un valor numérico único que representa el riesgo de una solicitud de préstamo. Todos los valores serán devueltos en un archivo único *Results*.

# Metodología del sistema

Hemos dividido la metodología del sistema en diferentes fases de ejecución que nos permitirán comprender fácilmente sus mecanismos y funcionalidades.

## Inicialización

Para comenzar la ejecución debemos inicializar todas las listas y diccionarios con los que vamos a trabajar. Estos diccionarios recogerán las variables de entrada, las funciones de riesgo, los datos de las aplicaciones o solicitudes de préstamo, y las reglas de inferencia. Para poder declarar e inicializar estos elementos hemos utilizado funciones ya diseñadas previamente y archivos de texto que nos proporcionan los datos requeridos.

Tras crear completamente todos los elementos, abrimos el archivo de escritura que guardará los resultados de nuestro sistema de inferencia.

## Bucle general

Entramos en el bucle fundamental de la ejecución de nuestro sistema. Este bucle se repetirá en cada una de las aplicaciones que aparezcan en la lista *Apps*, la cual guarda los datos de todas las solicitudes de préstamo recibidas diariamente por el banco. El bucle ejecuta dos funciones:

* Llama a la función auxiliar *procesapp*, introduciendo como parámetros la aplicación concreta, junto con las listas y diccionarios iniciales. Esta función devolverá como resultado el valor del riesgo de la solicitud, valor que se guardará en una variable local.
* Escribe el resultado, guardado en la variable, en el archivo de escritura *resultados*.

## Función auxiliar ‘procesapp’ (sistema de inferencia)

Esta función realiza todos los pasos del sistema de inferencia borrosa de Mamdani descritos en apartados anteriores. Al comienzo de esta función inicializamos las variables de salida, que representarán los valores en los ejes X e Y de nuestra gráfica de salida.

Posteriormente, ejecutaremos otra función auxiliar llamada *fuzzy* que explicaremos posteriormente. Esta función recibe como parámetros la solicitud en cuestión y la lista de variables borrosas de entrada, y realiza el proceso completo de borrosificación.

Tras el proceso de borrosificación, se ejecuta un bucle que realiza los procesos de evaluación de reglas y composición en cada una de las reglas localizadas en la lista *rule* pasada como parámetro.

En la evaluación de reglas, arrancaremos con el cálculo del antecedente. Para realizar este cálculo debemos recoger en una lista el valor de similitud (o *memDegree*) de cada uno de los antecedentes de la regla y, una vez estén todos los valores recogidos, escogeremos el menor y lo guardamos como el valor de pertenencia o *strenght* de la regla. A continuación, realizamos el cálculo del consecuente, el cual arrancamos copiando la gráfica correspondiente a uno de los valores (*Low*, *Medium* o *High*) de la variable borrosa *Risk*. Ahora nuestro sistema hará una agregación parcial aplicando el método de recorte. En este proceso comprobamos que todos los valores de la gráfica consecuente en el eje Y sean menores o iguales que el valor de pertenencia. En los puntos de X en los que no se cumpla este caso, el valor de Y de la gráfica tomará el valor de pertenencia, haciendo que la gráfica quede “recortada”.

El proceso de composición se realiza en el mismo bucle en el que realizamos la agregación parcial. Dentro del bucle iremos comprobando que el valor de la variable de salida Y sea mayor o igual que el valor Y de la gráfica consecuente en ese punto X. En caso contrario, el valor de salida Y tomará el valor de la gráfica. Este proceso se realiza con todas las reglas existentes, dando como resultado la unión de las gráficas consecuentes de todas ellas.

Finalmente ejecutamos el proceso de deborrosificación, usando las variables salidaX y salidaY, y calculando el centroide de la gráfica resultante de la aplicación insertada en la función *procesapp*. El valor del centroide será devuelto como salida de la función.

Para la deborrosificación utilizamos este conjunto borroso:

**Risk:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## Función auxiliar ‘fuzzy’

Esta función recibe una solicitud de préstamo o aplicación y la lista de las variables borrosas de entrada. El objetivo de esta función es realizar el proceso de borrosificación, calculando y guardando el valor de similitud para cada una de las posibilidades de cada variable borrosa para la aplicación pasada como parámetro.

Los conjuntos borrosos utilizados para el calculo de la similitud en esta función son:

**Edad:**

Gráfico, Gráfico de líneas, Histograma

Descripción generada automáticamente

**Income:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Assets:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Amount:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Job:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**History:**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## Cierre final

Para finalizar el programa de ejecución cerramos el archivo de escritura de los resultados que previamente hemos modificado con los valores de riesgo de cada una de las solicitudes.

# Presupuesto

Para analizar el presupuesto de este proyecto podemos partirlo en sus fases de realización, la primera fase seria la fase de análisis en la que se estudia el objetivo y se realiza una primera idea de cómo se podría llegar a él, este paso sería un 15% del proyecto lo que serían 5h de trabajo. La segunda fase del proyecto o diseño del mismo en este caso ha sido muy rápida ya que solo se necesita un software interno y por lo tanto ocupa un 10% del proyecto, por lo que el diseñador usara 3h de su trabajo. La fase de desarrollo es la fase de más trabajo ya que se centra en toda la realización del código fuente desde la borrosificación hasta el cálculo del centroide, por ello podemos estimar que el desarrollo del proyecto ocupara un 65% del mismo lo que estimamos que podrían llegar a ser 16h de nuestro programador. Por último, la fase de pruebas e implementación de este que podría tardar de 1-2h de trabajo.

El valor de las horas será:

Analista: 25 €/h

Diseñador: 21 €/h

Programador: 18€/h

El precio total del proyecto rondaría los 430€.

# Anexo

Hemos entregado todos los archivos del código fuente ya que hemos encontrado un error en la función “readFuzzySetsFile” del archivo MFIS\_Read\_Functions.py por el que al crear la listaX se creaba con un valor menos del necesario y, por lo tanto, la lista se salía del rango al acceder a un índice no existente.