Lógica Difusa. Introducción a los conjuntos difusos y modelado de sistemas difusos.

nombre:Edison Huinaizaca

Plateamienro del problema

En un galpón se tiene una temperatura de 18 grados centígrados, y una humedad de aproximadamente 22 grados centígrados. Según estos valores determinar cual es la velocidad que debería estar funcionando el motor. Para revisar las reglas, función de pertinencia y el proceso revisar el siguiente link: https://medium.com/@javierdiazarca/l%C3%B3gica-difusa-ejercicio-2-bases-de-la-ia-1a8ae594cc15 En base a ello, desarrollar e implementar el sistema dentro de Python o Java en donde me permita modificar los valores de la temperatura y humedad, generando así un sistema experto basado en lógica difusa para obtener la velocidad del motor de aire acondicionado.

Este sistema deberá tener la opción de poder modificar los valores de la temperatura y humedad con un scroll bar y obtener la velocidad de giro. Ademas, deberá presentarme las graficas de pertenencia de INPUT/OUPUT del sistema difuso y como estas varían de acuerdo al cambio de las variables.

Nota: Esta practica remplazara la segunda prueba con los siguientes criterios de evaluación:

- GUI: 40%
- FUZZY LOGIC: 40%informe y pruebas: 20%

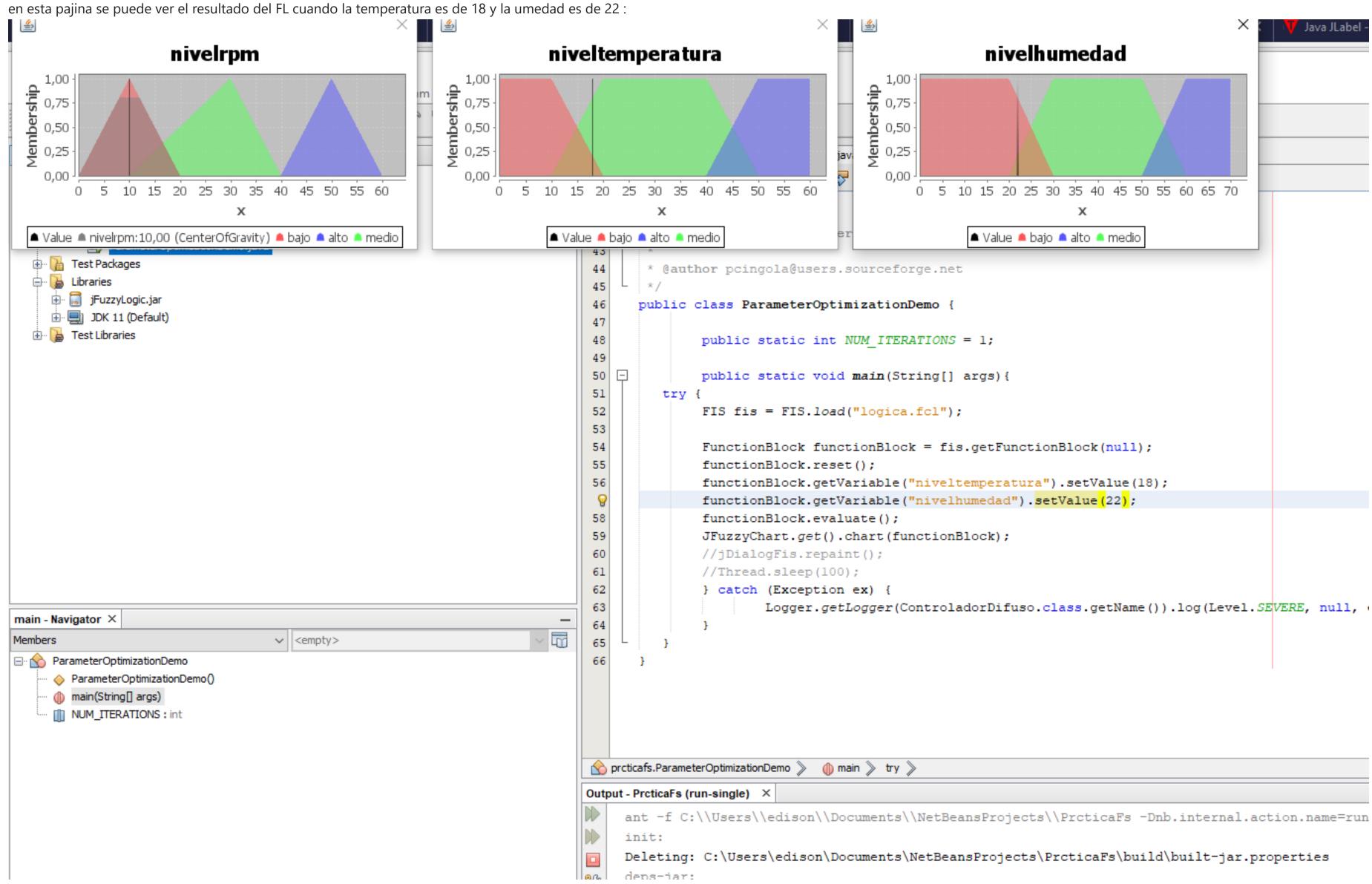
Subir el informe en formato PDF y los códigos fuentes al repositorio Git personal.

Desarrollo

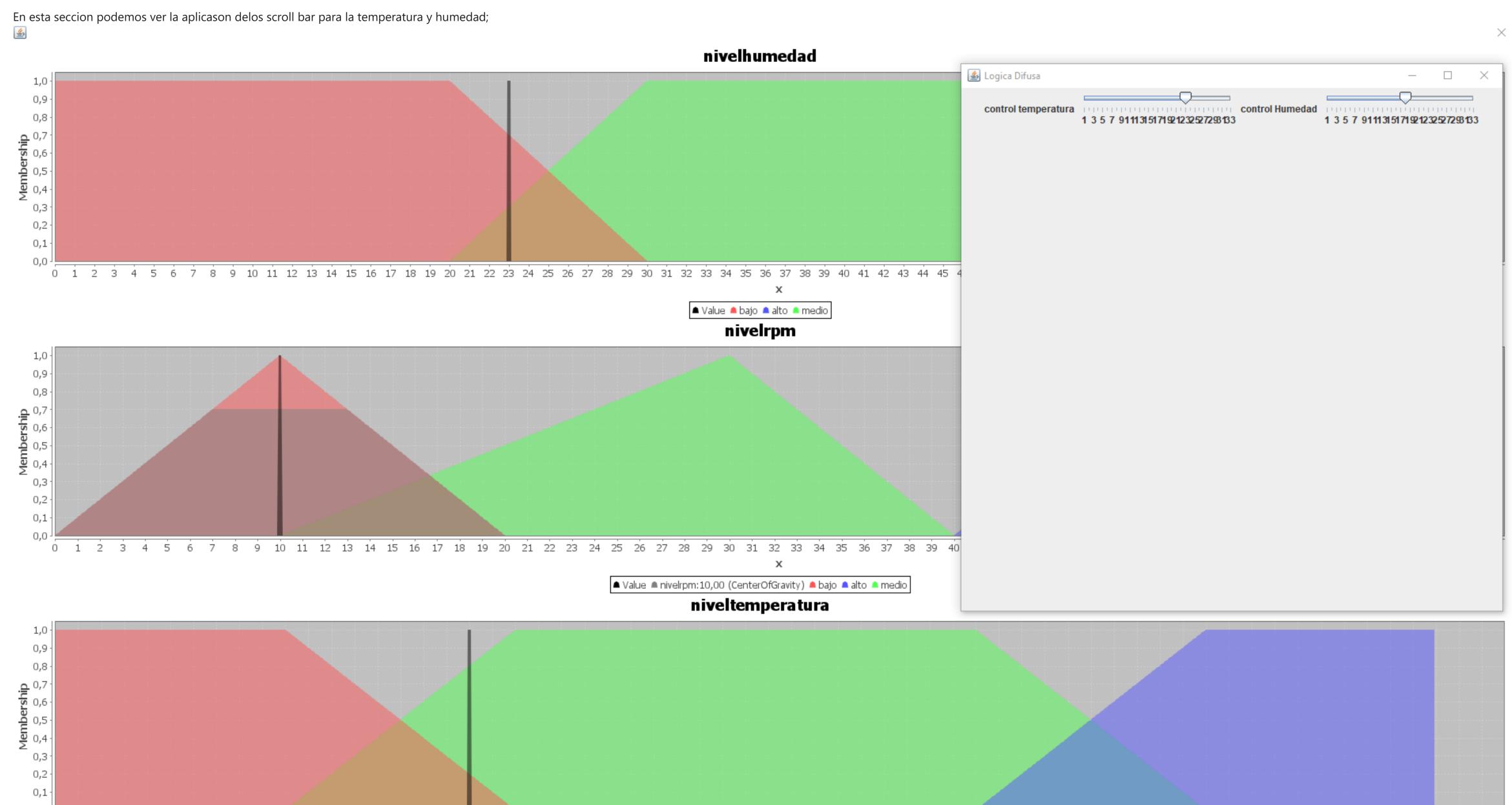
Para el desarrolo de esta aplicasion se uso FL como base y se desarrollo en java en los sigientes parafos se detallara los resultados y su corespondiente explicasion.

Puesta en marcha de el FL en java

es esta seccion se puede ver los resultados cuando se pone en marcha el FL ≗ nivelhumedad nivelrpm niveltemperatura Mempership 0,75 -Membership 0,75 0,25 0,25 Q → Search (Ctrl+I) Membership 0,50 0,25 < → ▼ □ 0 5 10 15 20 25 30 35 40 45 50 55 60 0 5 10 15 20 25 30 35 40 45 50 55 60 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 📤 Value 📤 bajo 🔔 alto 🔔 medio ▲ Value ▲ nivelrpm:26,67 (CenterOfGravity) ▲ bajo ▲ alto ▲ medio 🛕 Value 🔔 bajo 🔔 alto 🔔 medio ⊕ ∏ Test Packages 91 🖨 🍃 Libraries jFuzzyLogic.jar 92 93 public boolean cargar() { ⊞ JDK 11 (Default) 94 // try { 95 functionBlock.getVariable("niveltemperatura").setValue(slhumedad.getValue()); 96 functionBlock.getVariable("nivelhumedad").setValue(slTemperatura.getValue()); 97 functionBlock.evaluate(); 98 jDialogFis = new net.sourceforge.jFuzzyLogic.plot.JDialogFis(FIS); JFuzzyChart.get().chart(FIS.getFunctionBlock(null).getVariable("nivelrpm"),FIS.getFunctionBlock(null).getVariable("nivelrpm").getDefuzzifier(),fals 100 //jDialogFis.repaint(); 101 //Thread.sleep(100); 102 //} catch (NullPointerException ex) { // Logger.getLogger(ControladorDifuso.class.getName()).log(Level.SEVERE, null, ex); 104 105 return true; 106 107 public static void main(String[] args) { 108 new ControladorDifuso().setVisible(true); 109 110 avigator × lembers ✓ | <empty> ⊩ 🕎 ParameterOptimizationDemo ParameterOptimizationDemo() main(String[] args) NUM_ITERATIONS : int prcticafs.ControladorDifuso > ocargar > Output - PrcticaFs (run-single) X ant -f C:\\Users\\edison\\Documents\\NetBeansProjects\\PrcticaFs -Dnb.internal.action.name=run.single -Djavac.includes=prcticafs/ParameterOptimizationDemo.java ^



init:



Estructura del documento logica.fcl

FUNCTION_BLOCK contromotor //declarasion de variables VAR_INPUT nivelhumedad : REAL; END_VAR VAR_INPUT niveltemperatura : REAL; END_VAR VAR_OUTPUT nivelrpm : REAL; END_VAR //Declaracion dela variables que formaran nuestras tres grafica FUZZIFY nivelhumedad TERM alto := (50.0, 0.0) (60.0, 1.0) (70.0, 1.0) ; TERM bajo := (0.0, 1.0) (20.0, 1.0) (30.0, 0.0); TERM medio := (20.0, 0.0) (30.0, 1.0) (50.0, 1.0) (60.0, 0.0); END_FUZZIFY FUZZIFY niveltemperatura TERM bajo := (0.0, 1.0) (10.0, 1.0) (20.0, 0.0); TERM alto := (40.0, 0.0) (50.0, 1.0) (60.0, 1.0); TERM medio := (10.0, 0.0) (20.0, 1.0) (40.0, 1.0) (50.0, 0.0); END_FUZZIFY DEFUZZIFY nivelrpm TERM bajo := TRIAN 0.0 10.0 20.0; TERM alto := TRIAN 40.0 50.0 60.0; TERM medio:= TRIAN 10.0 30.0 40.0; METHOD : COG; DEFAULT := 0.0; END_DEFUZZIFY //creasion de las reglas segun la pagina que fue dada de referencia RULEBLOCK Regla1 ACT : MIN; ACCU : MAX; AND : MIN; RULE 1 : IF nivelhumedad IS alto AND niveltemperatura IS bajo THEN nivelrpm IS bajo; RULE 2 : IF nivelhumedad IS alto AND niveltemperatura IS medio THEN nivelrpm IS medio; RULE 3 : IF nivelhumedad IS alto AND niveltemperatura IS alto THEN nivelrpm IS medio; RULE 4 : IF nivelhumedad IS medio AND niveltemperatura IS bajo THEN nivelrpm IS bajo; RULE 5 : IF nivelhumedad IS medio AND niveltemperatura IS medio THEN nivelrpm IS bajo; RULE 6 : IF nivelhumedad IS medio AND niveltemperatura IS alto THEN nivelrpm IS medio; RULE 7 : IF nivelhumedad IS bajo AND niveltemperatura IS bajo THEN nivelrpm IS bajo; RULE 8 : IF nivelhumedad IS bajo AND niveltemperatura IS medio THEN nivelrpm IS bajo; RULE 9 : IF nivelhumedad IS bajo AND niveltemperatura IS alto THEN nivelrpm IS alto; END_RULEBLOCK END_FUNCTION_BLOCK

es la sigiente seccion podremos ver como es la estroctura del FL y su posterior documentacion

Conclusiones

Conclusiones

Se puede clocluir que el uso del FL es de facil compresion y es de mucha ayuda en el desarrollo de muchas aplicasiones que usen logica difusa como base

Bibliografia y link delcodigo

- http://jfuzzylogic.sourceforge.net/html/manual.html
- https://github.com/edison123344/sistemasExpertos.git
 In []: