Proyecto Integración de Redes Neuronales, Lógica Difusa y Sistemas Expertos.

Autor: Nicolás Rodríguez Fernández
Ingeniería de Sistemas, Universidad Tecnológica de Pereira, Pereira, Colombia
Correo-e: nickoro094@utp.edu.co

***Resumen*— Un sistema experto es un programa de computadora que utiliza tecnologías de inteligencia artificial (AI) para simular el juicio y el comportamiento de un ser humano o una organización que tiene conocimiento y experiencia en un campo en particular.**
Palabras clave— **Inteligencia Artificial, lógica, Sistema experto, agente.**

***Abstract*— An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field.**

***Key Word* — Artificial Intelligence, logic, expert system, agent.**

I. INTRODUCCIÓN

Un sistema experto es un programa de computadora que utiliza tecnologías de inteligencia artificial (AI) para simular el juicio y el comportamiento de un ser humano o una organización que tiene conocimiento y experiencia en un campo en particular.

Típicamente, un sistema experto incorpora una base de conocimientos que contiene experiencia acumulada y un motor de inferencia o reglas, un conjunto de reglas para aplicar la base de conocimientos a cada situación particular que se describe en el programa. Las capacidades del sistema se pueden mejorar con adiciones a la base de conocimientos o al conjunto de reglas. Los sistemas actuales pueden incluir capacidades de aprendizaje automático que les permitan mejorar su rendimiento en función de la experiencia, al igual que los humanos.

La lógica difusa es una extensión de la lógica booleana de Lotfi Zadeh en 1965, basada en la teoría matemática de los conjuntos difusos, que es una generalización de la teoría de conjuntos clásica. Al introducir la noción de grado en la verificación de una condición, permite una condición en un estado distinto de verdadero o falso, la lógica difusa proporciona una flexibilidad muy valiosa para el razonamiento, que hace posible tomar en cuenta inexactitudes y incertidumbres. Una ventaja de la lógica difusa para formalizar el razonamiento humano es que las reglas se establecen en lenguaje natural. Por ejemplo, aquí hay algunas reglas de conducta que un conductor sigue, asumiendo que no quiere perder su licencia de conducir:

If the light is red...	if my speed is high...	and if the light is close...	then I brake hard.
If the light is red...	if my speed is low...	and if the light is far...	then I maintain my speed.
If the light is orange...	if my speed is average...	and if the light is far...	then I brake gently.
If the light is green...	if my speed is low...	and if the light is close...	then I accelerate.

Figura 1. Ejemplo de lógica difusa. Conducta de un agente conductor.

II. DESCRIPCIÓN DEL PROBLEMA

El problema objeto del proyecto está referido a movilidad en una ciudad en el que se toma en cuenta las características que afectan la calidad del servicio.
Las variables a tomar en cuenta, entre otras, son las siguientes:

1. Día de la semana
2. Mes
3. Hora
4. Estado de la carretera
5. Lluvia

Si en el desarrollo de la solución se perciben otras variables claves, éstas podrán ser agregadas al sistema. Esta parte que se acaba de describir requiere de un modelo difuso en el cual se establezcan los niveles de verdad de cada variable lingüística, asociado al despliegue de rangos de valores en características específicas (como por ejemplo, Mucho, Poco, Nada, etcétera). Seguidamente se estable un mapa de patrones de entrada-salida, en el cual se establecen, como entradas, las características iniciales: día de la semana, mes, hora, estado de la carretera, lluvia y código de la ubicación en la ciudad.

Fecha de Recepción: (Letra Times New Roman de 8 puntos)
Fecha de Aceptación: Dejar en blanco

La salida es la probabilidad de accidentalidad. Este modelo requiere de una red neuronal backpropagation.

El último modelo es un Sistema Experto. En él se establece, a partir de las entradas y salidas de la red neuronal, una recomendación generada por un sistema de reglas. Las recomendaciones pueden ser: velocidad máxima, encender luces, transitar con cuidado, no adelantar, entre otras.

Cada uno de los sistemas debe ser programado de manera independiente, pero los resultados de un sistema se utilizan como entrada del siguiente (según se requiera)

III. OBJETIVOS

El objetivo del proyecto LD+RN+SE (Lógica Difusa + Red Neuronal + Sistema Experto) es aplicar las tres tecnologías de Computación Blanda en un problema que requiera el uso de métodos heurísticos, incertidumbre y aprendizaje.

IV. MODELOS

La lógica difusa se basa en la teoría de conjuntos difusos, que es una generalización de la teoría de conjuntos clásica. Decir que la teoría de conjuntos borrosos es una generalización de la teoría de conjuntos clásica significa que esta última es un caso especial de la teoría de conjuntos borrosos. Como una metáfora en la teoría de conjuntos, la teoría de conjuntos clásica es un subconjunto de la teoría de conjuntos borrosos, como lo ilustra la figura 2.

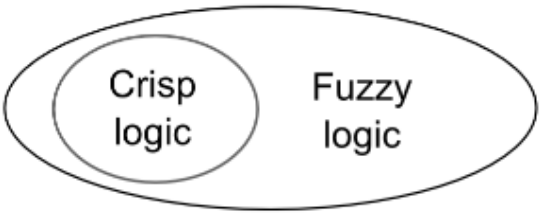


Figura 2. “La teoría de sets clásica es un subset de la teoría de sets difusa.

2.1 Modelo de Lógica Difusa

i. Sea X un conjunto. Un subconjunto difuso A de X se caracteriza por una función de membresía. fª : X -> [0,1]

La forma de la función de membresía se elige arbitrariamente siguiendo el consejo del experto o mediante estudios estadísticos: sigmoide, hiperbólico, tangente, exponencial, gaussiano o cualquier otra forma. La figura 3 muestra la diferencia entre un conjunto convencional y un conjunto difuso,

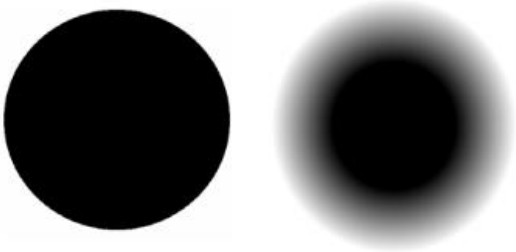


Figura 3. Conjunto convencional (Izquierda) y conjunto difuso (Derecha)

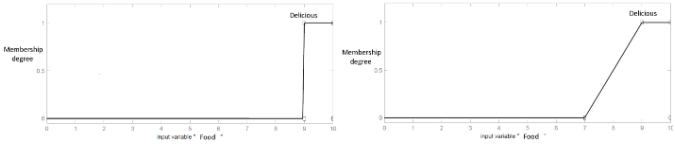


Figura 4. Comparación entre función de identidad de un set convencional y la función de membresía en lógica difusa.

Para definir las características de los conjuntos difusos, estamos redefiniendo y expandiendo las características habituales de los conjuntos clásicos. Los conjuntos difusos tienen una serie de propiedades. Aquí hay definiciones de las propiedades más importantes. Sea X un conjunto y A, un subconjunto difuso de X y μA, la función de pertenencia que lo caracteriza.μA (x) se denomina grado de pertenencia de x en A

ii. La altura de A, denotada h (A), corresponde al límite superior del código de su función de pertenencia: h(A) = sup{μA(x)|x∈X}

iii. A se dice que está normalizada si y sólo si h (A) = 1. En la práctica, es extremadamente raro trabajar en conjuntos difusos no normalizados.

iv. El Soporte de A es el conjunto de elementos de X que pertenecen al menos a algunos A. En otras palabras, el soporte es el conjunto supp (A) = {x∈X | μA (x)> 0}.

v. El Kernel de A es el conjunto de elementos de X que pertenecen totalmente a A. En otras palabras, el kernel noy (A) = {x∈X | μA (x) = 1}. Por construcción, noy (A) ⊆supp (A).

vi. Un corte α de es el subconjunto clásico de elementos con un grado de pertenencia mayor o igual que α: corte α (A) = {x∈X | μA (x)> α}.

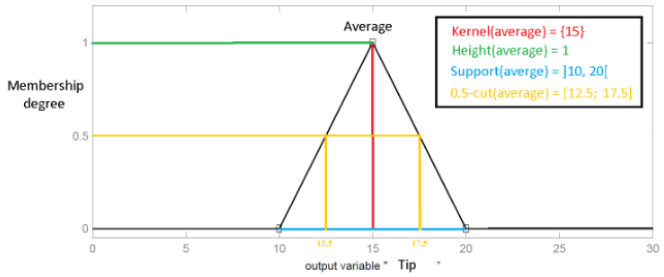


Figura 5. Una función de membresía con propiedades visibles,

Podemos ver que si A era un conjunto convencional, simplemente tendríamos (A) = noy (A) yh (A) = 1 (ouh (A) = 0siA =). Por lo tanto, nuestras definiciones pueden recuperar las propiedades inusuales de los conjuntos clásicos.

El concepto de función de membresía discutido anteriormente nos permite definir sistemas difusos en lenguaje natural, ya que la función de membresía combina la lógica difusa con las variables de lineamientos que definiremos ahora.

vii. Sea V una variable, X el rango de valores de la variable y TVa conjunto finito o infinito de conjuntos difusos. La variable lingüística corresponde al triplete (V, X, TV)

Cuando definimos los conjuntos difusos de variables lingüísticas, el objetivo no es definir exhaustivamente las variables lingüísticas. En su lugar, solo definimos algunos subconjuntos difusos que serán útiles más adelante en la definición de las reglas que aplicamos.

2.2 Defuzzificación

Al igual que con todos los operadores difusos, el diseñador de sistemas difusos debe elegir entre varias definiciones posibles de defuzzificación.

Presentaremos brevemente los dos métodos principales de defuzzificación: el método de la media de los máximos (MeOM) y el método del centro de gravedad (COG).

La defuzzificación de MeOM establece la salida (decisión de la cantidad de la punta) como el promedio de Abscisas de los máximos del conjunto difuso resultante de la agregación de los resultados de la aplicación.

$$Décision = \frac{\int_S y \cdot dy}{\int_S dy}$$

Donde, $S = \{y_m \in R, \mu(y_m) = SUP_{y \in R}(\mu(y))\}$

, y R es el conjunto difuso que resulta de la agregación de los resultados de la implicación.

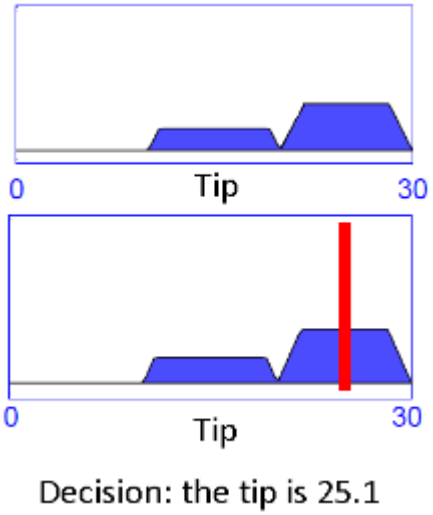


Figura 6. Defuzzificación por método media de los máximos (MeOM)

La defuzzificación COG es más comúnmente utilizada. Define la salida como correspondiente a la abscisa del centro de gravedad de la superficie de la función de pertenencia que caracteriza el conjunto difuso resultante de la agregación de los resultados implícitos.

$$Décision = \frac{\int_S y \cdot \mu(u) \cdot dy}{\int_S \mu(u) \cdot dy}$$

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

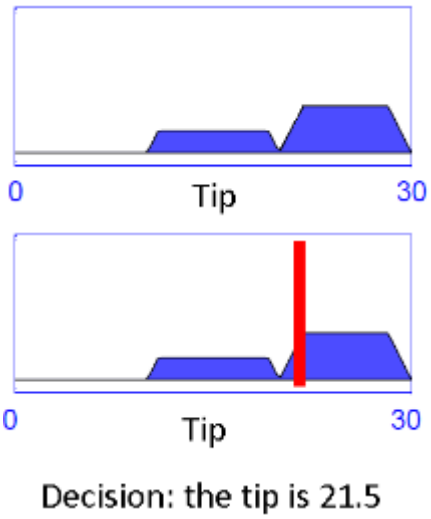


Figura 7. Defuzzificación por método centro de gravedad (COG)

Esta definición evita que las discontinuidades puedan aparecer en la defuzzificación de MeOM, pero es más compleja y tiene un mayor costo computacional.

Como vemos en las figuras 6 y 7 que muestran las defuzzificaciones de MeOM y COG aplicadas a nuestro ejemplo, la elección de este método puede tener un efecto significativo en la decisión final.

Estado Carretera	Presencia Lluvia	Hora
Mala[0-35]	Poca[0-35]	Madrugada[0,7]
Regular[35-65]	Moderada[35-65]	Dia[7,18]
Buena[65-100]	Mucha[65-100]	Noche[18,0]

2.3. Modelo Red Neuronal (Backpropagation)

$$f(x) = \begin{cases} 1 & si\ w \cdot x + b > 0, \\ 0 & en\ otro\ caso \end{cases}$$

Donde *w* es un vector de pesos con valores reales, $w \cdot x$ es el producto punto $\sum_{i=1}^{m} w_i x_i$, donde *m* es el número de entradas de la red neuronal, y *b* es el *bias*. El *bias* cambia el límite de la decisión moviéndolo lejos del origen y no depende de ningún valor de entrada.

El valor de $f(x)$ (0 o 1) se usa para clasificar *x* como una instancia positiva o negativa, en el caso de un problema de clasificación binaria. Si *b* es negativo, entonces la combinación ponderada de entradas debe producir un valor positivo mayor que | b | con el fin de empujar la neurona clasificadora sobre el umbral de 0. Espacialmente, el sesgo altera la posición (aunque no la orientación) del límite de decisión.

El algoritmo de aprendizaje de la red neuronal no termina si el conjunto de aprendizaje no es linealmente separable. Si los vectores no son linealmente separables, el aprendizaje nunca alcanzará un punto donde todos los vectores se clasifiquen correctamente.

2.3. Funcionamiento

La red neuronal consta de 4 partes.

- Valores de entrada o una capa de entrada
- Pesos y Bias
- Función de activación
- Suma neta

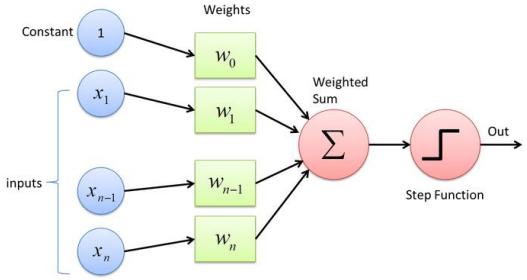


Figura 8. Red neuronal

La red neuronal funciona siguiendo los pasos:

a) Todas las entradas *x* se multiplican con sus pesos *w*. Llamémoslo *k*.

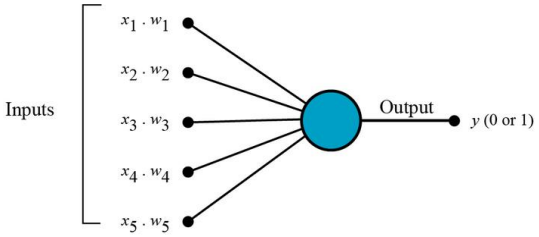


Figura 9. Multiplicando entradas por pesos (5 entradas).

b) Sumar todos los valores obtenidos, a esto lo llamamos Sumatoria de Pesos.

$$\sum_{k=1}^{5} k$$

c) Aplicamos la sumatoria de pesos a la función de activación correcta

$$f(x) = \begin{cases} 0 & si\ 0 > x \\ 1 & si\ x \geq 0 \end{cases}$$

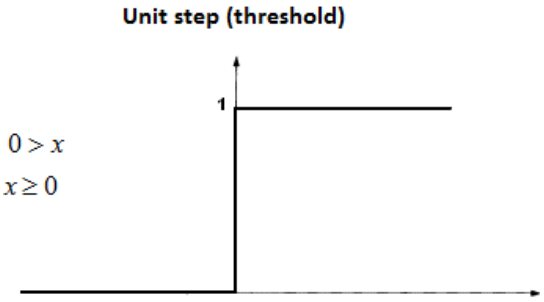


Figura 10. Función de Activación

2.4. Pesos y bias

Los pesos muestran la fuerza del nodo particular.

Un valor de bias le permite cambiar la curva de la función de activación hacia arriba o hacia abajo.

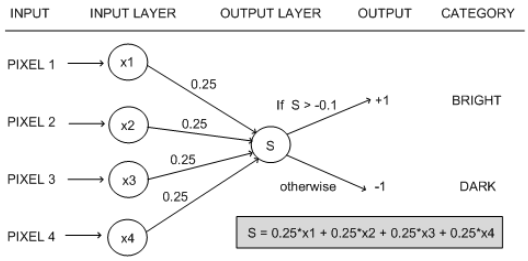


Figura 11. Ejemplo de red neuronal mostrando pesos y bias.

2.5. Función de Activación

La función de activación es usualmente una abstracción representando una tasa de potencial de activación gatillándose en la celda. En su forma simplificada, esta función es binaria, esto es, se activa la neurona o no.

En resumen, las funciones de activación se utilizan para asignar la entrada entre los valores requeridos como (0, 1) o (-1, 1).

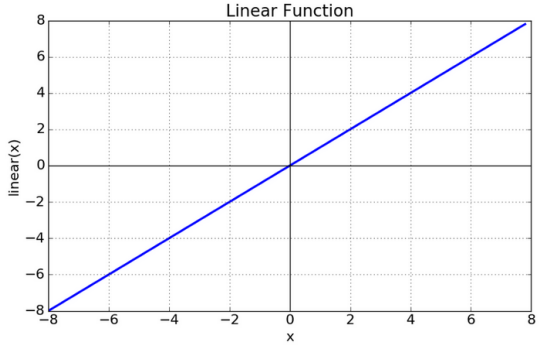


Figura 12. Función de activación lineal

2.6. Convergencia

El Backpropagation es un clasificador lineal, por lo tanto, nunca llegará al estado con todos los vectores de entrada clasificados correctamente si el conjunto de entrenamiento *D* no es linealmente separable, es decir, si los ejemplos positivos no pueden separarse de los ejemplos negativos por un hiperplano. En este caso, ninguna solución "aproximada" se abordará gradualmente bajo el algoritmo de aprendizaje estándar, sino que el aprendizaje fallará por completo. Por lo tanto, si la separabilidad lineal del conjunto de entrenamiento no se conoce a priori, se debe usar una de las variantes de entrenamiento a continuación.

Si el conjunto de entrenamiento es linealmente separable, entonces se garantiza que la red neuronal converge. Además, hay un límite superior en el número de veces que el Backpropagation ajustará sus pesos durante el entrenamiento.

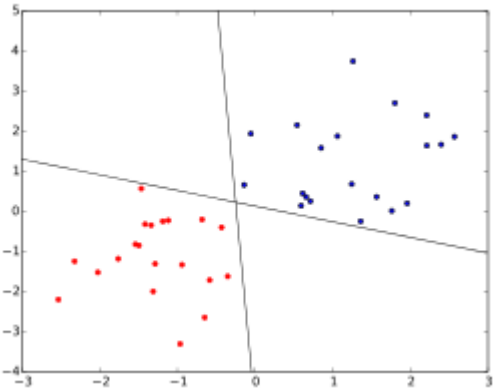
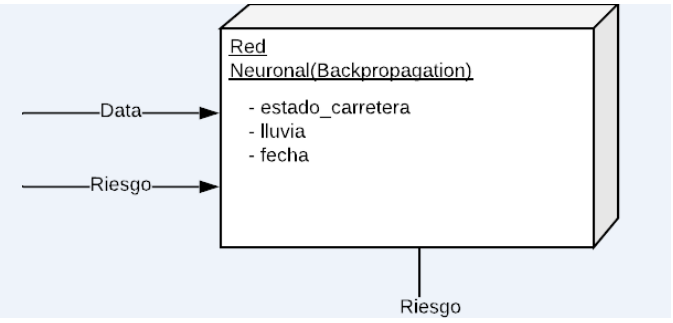


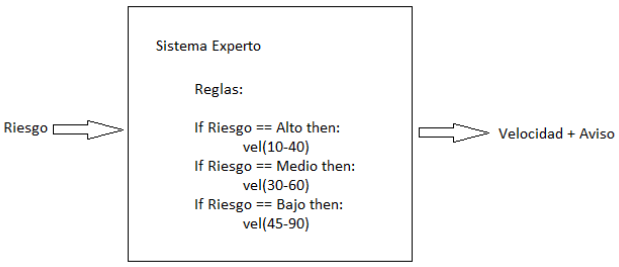
Figura 13. Dos clases de puntos y dos de los infinitos límites lineales que los separan. A pesar de que los límites están casi en ángulo recto entre sí, el algoritmo de percepción no tiene forma de elegir entre ellos.

Si bien se garantiza que el algoritmo de Backpropagation convergerá en alguna solución en el caso de un conjunto de entrenamiento separable linealmente, todavía puede elegir cualquier solución y los problemas pueden admitir muchas soluciones de calidad variable.

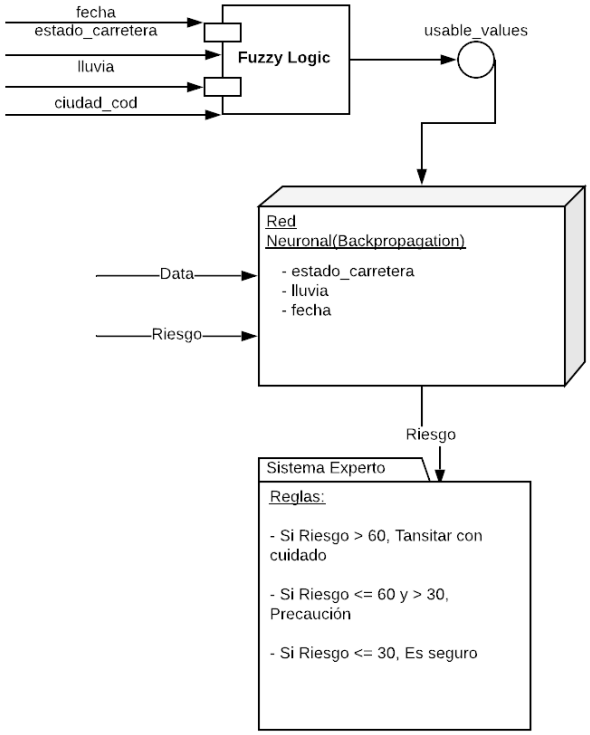


1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

2.5 Modelo Sistema Experto



V. ARQUITECTURA INTEGRADA



VI. HERRAMIENTAS Y MÉTODOS

Se utilizará para la implementará la solución con las herramientas:

- Python 3.7
  - Librerias
  - Fuzzylite
  - Pyknow

VII. CONCLUSIONES

Al utilizar las técnicas algorítmicas de la lógica difusa que nos permite lidiar con incertidumbres en la evaluación de los sistemas expertos hemos logrado obtener una mejor efectividad a la hora de entrenar y aplicar redes neuronales con Backpropagation.

En los resultados del proyecto se logró obtener una probabilidad de accidentalidad en carretera basado en entradas inciertas transformadas con lógica difusa (mucho, poco, nada) a enteros que pudiesen ser trabajados con los métodos de Backpropagation y sistemas expertos.

REFERENCIAS

[1] Jangi, 1992] Jangi, R. (1992). Neuro-Fuzzy modeling: Architecture, Analysis and Application. PhD thesis, University of California, Berkeley.

[2] Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. Information and Control, 8(3):338 – 353

[3] J.P. Aurrand-Lions, L. Fournier, P. Jarri, et al. Application of fuzzy control for ISIS vehicule braking. In Proceedings of Fuzzy and Neu-ronal Systems, and Vehicule applications'91, 1991

[4] Wang, Li-Xin "Adaptative Fuzzy Systems and Control. Design and Stability Analysis" Prentice Hall, NewJersey, 1994

[5] Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory.

[6] Minsky M. L. and Papert S. A. 1969. Perceptrons. Cambridge, MA: MIT Press.

[7] Widrow, B., Lehr, M.A., "30 years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," Proc. IEEE, vol 78, no 9 (1990).

[8] https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos