

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

SISTEMAS INFORMÁTICOS
2009-2010

INGENIERO EN INFORMÁTICA



**Búsqueda de comportamiento
en una máquina simple
aleatoria**

Autores:

Alberto Edroso Jarne

Marcos Miguel García Toledo

Tutor:

Francisco Javier Crespo Yáñez

TÍTULO:	Búsqueda de comportamiento en una máquina simple aleatoria
AUTORES:	Alberto Edroso Jarne Marcos Miguel García Toledo
TUTOR:	Francisco Javier Crespo Yáñez
DEPARTAMENTO:	Sistemas Informáticos y Computación

Abstract

Nuestro trabajo ha consistido en la búsqueda de un posible comportamiento en una máquina simple aleatoria, como es el lanzamiento de una moneda. Hemos simulado dichos lanzamientos mediante la generación de números pseudoaleatorios, especificando el sesgo de la moneda y recogiendo las variables de cada experimento.

Hemos realizado una amplia experimentación, de la cual hemos extraído y analizando diferentes variables en relación a las distancias[1] para comprobar si aportan nueva información útil, con la que aumentar la credibilidad del sesgo estimado por las máquinas fuzzy implementadas que utilizan únicamente el porcentaje de apariciones.

Palabras clave

Aleatoriedad, moneda, lanzamiento, fuzzy, análisis, sesgo, correlación.

Abstract

Our project is about the seeking of a behavior on a simple random machine as is the toss of a coin. We have simulated those tossing through the generation of pseudo-random numbers specifying the coin bias and collecting variables for each experiment.

We have accomplished a huge experimentation, from which we have inferred and analyzed different kinds of variables such as distances[1] to probe if they can generate new information to increase the credibility of the estimated bias by the fuzzy machines implemented, which only use the percentage of heads.

Keywords

Randomness, coin, tossing, fuzzy, analysis, bias, correlation.

A nuestras familias y amigos.

Agradecimientos

Al profesor D. Francisco Javier Crespo, por su investigación previa y software facilitado para el desarrollo de este proyecto.

Índice general

1. Introducción	15
2. Estado del arte	21
2.1. Historia del arte	21
2.2. Estudios sobre la percepción de la aleatoriedad	32
2.2.1. Prejuicios cognitivos	34
2.3. Lógica clásica	42
2.4. Lógica <i>fuzzy</i>	45
2.5. Coeficiente de correlación de Pearson	54
2.6. Números pseudoaleatorios	55
3. Planteamiento y solución	59
3.1. ¿Qué implica el lanzamiento de una moneda?	59
3.1.1. ¿Cómo se agrupan los lanzamientos de una moneda?	61
3.2. ¿Cómo vamos a lanzar nuestra moneda?	62
3.3. Parámetros empleados en la experimentación	65
3.4. Objetivos de la investigación	66
3.5. Desarrollo del proyecto	68

3.5.1.	Diagrama de clases	68
3.5.2.	Estructura: Clases y paquetes	70
3.6.	Máquinas desarrolladas	71
3.6.1.	Máquinas <i>fuzzy</i>	71
4.	Experimentación	75
4.1.	Desarrollo de la aplicación	76
4.2.	Exposición de los resultados obtenidos	78
4.2.1.	Análisis de las apariciones de caras	78
4.2.2.	Análisis de las distintas distancias	80
5.	Discusión	87
5.0.3.	Distancias máximas	88
5.0.4.	Distancias 0, 1 y 2	88
6.	Conclusión y líneas futuras	91
7.	Apéndices y Anexos	95

Índice de figuras

2.1. Figura de Keller sobre probabilidades de una moneda	26
2.2. Maquina desarrollada para el lanzamiento de monedas. Vista 1	31
2.3. Maquina desarrollada para el lanzamiento de monedas. Vista 2	32
2.4. Conjuntos fuzzy o difusos para la altura de una persona	46
2.5. Conjuntos crisp-clasicos para la altura de una persona	47
2.6. Conjuntos difusos entre los se definirán las operaciones	47
2.7. Intersección entre dos conjuntos difusos	48
2.8. Cual es el valor de pertenencia de $x=4.5$ a la interseccion de los conjuntos difusos A y B	49
2.9. Union entre dos conjuntos difusos	50
2.10. Cual es el valor de pertencia de $x=4.5$ a la union de los con- juntos difusos A y B	50
2.11. Complemento de un conjunto difuso	51
2.12. Cual es el valor de pertenencia de $x=6$ al complemento del conjunto difusos A	52
2.13. Bitmap de valores pseudoaleatorios generados por la funcion Rand de PHP	57

2.14. Bitmap de valores pseudoaleatorios generados por la pagina web Random.org	58
2.15. Bitmap de valores pseudoaleatorios generados por la funcion Random de la libreria Math en Java	58
3.1. Árbol de probabilidades de un ensayo de Bernoulli de tres experimentos	60
3.2. Representacion de la distribucion binomial	61
3.3. Histograma del lanzamiento de una moneda al aire 100 veces .	63
3.4. Histograma de una moneda girando sobre su eje 100 veces . .	64
3.5. Diagrama de clases parte 1	68
3.6. Diagrama de clases parte 2	69
3.7. Valores de la funcion triangular de pertenencia fuzzy al con- junto fuzzy de sesgo 50	72
3.8. Valores de la funcion trapezoidal de pertenencia fuzzy al con- junto fuzzy de sesgo 50	73
4.1. Grafica de las apariciones de caras por sesgo	78
4.2. Grafica de las distancias acumuladas de caras por sesgo	80
4.3. Apariciones de la distancia 0 por experimento	82
4.4. Apariciones de la distancia 1 por experimento	83
4.5. Apariciones de la distancia 2 por experimento	84
4.6. Grafica de las distancias maximas de caras por sesgo	86
7.1. Interfaz principal de la aplicacion	96
7.2. Menu de lanzamiento	97

7.3. Resultado de apariciones	98
7.4. Distancias maxima y media	98
7.5. Distancia individual 0	99
7.6. Distancia individual 3	99
7.7. Vista de menu de maquinas	100
7.8. Opciones menu	101
7.9. Ventana de experimentacion	102
7.10. Ventana de graficas	103
7.11. Seleccion de graficas	104
7.12. Ventana de obtencion de resultados	105

Capítulo 1

Introducción

*”Ten paciencia con todas las cosas,
pero sobre todo contigo mismo”*

San Francisco de Sales

Nuestro trabajo consistirá en la observación y estudio de un sistema aleatorio simple, siendo el sistema de estudio el lanzamiento de una moneda, instrumento el cual nos da únicamente dos valores mutuamente excluyentes, como son cara y cruz. Obtendremos los lanzamientos mediante la generación de números pseudoaleatorios, por medio de computador.

Existen estudios previos que han tratado de demostrar que el lanzamiento de una moneda no se trata de un suceso aleatorio[8][14]. Vamos a centrarnos en las dos investigaciones más destacadas:

- En 1986, Keller determinó que cuando se lanza una moneda, ésta tiene una cierta velocidad inicial y un cierto espín, que de acuerdo con la mecánica clásica determina el cómo va a caer.

De esta forma Keller llegó a la conclusión de que el plano en que se representan las dos variables, la velocidad inicial y el espín de la moneda queda particionado en regiones, de modo que si los valores de las variables están en una región la moneda cae tal como se lanzó y si están en la región contigua cae al contrario y así, de modo análogo y alternativamente, ocurre con todas las regiones [8].

Finalmente Keller concluyó que la parte de la moneda que está inicialmente arriba es más probable que salga al final con 0.51 de probabilidades mientras que la parte inferior con 0.49 (Véase figura 2.1) [8].

- En 2004 los investigadores estadounidenses de la Universidad de Stanford, Persi Diaconis, Susan Holmes y Richard Montgomery demostraron que el resultado de lanzar una moneda es perfectamente determinista y en absoluto aleatorio bajo unas mismas condiciones iniciales controladas[14].

El resultado del lanzamiento viene determinado por las condiciones iniciales (posición, orientación y velocidad inicial y condiciones ambientales). Conocidas estas variables, es posible predecir el resultado "tan fielmente como podemos predecir el movimiento de la Luna o el de una manzana que cae"[14].

Podríamos resumir diciendo que si la fuerza con la que se hace el lanzamiento es siempre la misma, el resultado es siempre el mismo, si y solo si las condiciones de lanzamiento no varían.

Para probar el determinismo del lanzamiento de una moneda, constru-

yeron una máquina que lanzaba una moneda al aire mediante un muelle y la recogían en un recipiente, siempre bajo las mismas condiciones. El resultado era que la posición final era la misma que la posición inicial

Nuestro trabajo se basará en la observación y estudio de un sistema aleatorio simple, como es el lanzamiento de una moneda. Se analizarán los lanzamientos simulados mediante números pseudoaleatorios generados por computador.

Para ello vamos a implementar un simulador que nos permita realizar una experimentación, basada en múltiples lanzamientos de monedas, y así después analizar los datos resultantes para construir máquinas basadas en lógica fuzzy, para luego tratar de estimar con suficiente credibilidad el sesgo de la moneda en base a su comportamiento.

Basándonos en los resultados de forma empírica y a partir de esa práctica, intentaremos obtener una teoría que sea aplicada eficientemente.

La lógica *fuzzy* o lógica difusa es una metodología que nos proporciona de una manera simple y elegante la forma de obtener una conclusión a partir de una información de entrada vaga, ambigua, imprecisa, con ruido o incompleta, como serán nuestras variables de estudio. En general, la lógica difusa imita cómo una persona percibe la información y toma decisiones basada en dicha información.

Las variables que vamos a analizar de la forma anteriormente expuesta son:

- Porcentaje de apariciones

Se refiere al número de veces en % que ha aparecido cara con respecto

a la longitud total del experimento sin tener en cuenta el orden de sus apariciones.

Es la variable más representativa por que cuando el número de tiradas tiende a infinito el valor teórico de la variable coincide con el sesgo.

- Distancia acumulada de apariciones[1]

El concepto de distancia se refiere al número de apariciones consecutivas de un valor (en el caso de la moneda, cara o cruz).

$$distancia = numero_apariciones_consecutivas_valor - 1$$

- Distancia máxima de apariciones[1]

Contabiliza cual ha sido la distancia máxima que haya aparecido durante un experimento.

- Distancias individuales[1]

Esta última variable engloba una subvariable por cada distancia observada, siendo en realidad 100 variables. Indica para una determinada distancia y un determinado sesgo cuántas veces ha aparecido y en cuántos experimentos.

La motivación del uso de distancias viene dada por la necesidad de añadir credibilidad al sesgo estimado teniendo en cuenta únicamente la variable del porcentaje de apariciones, debido a que en la experimentación se da que hay una zona de intersección que genera cierta indeterminación al poder pertenecer ésta a varios sesgos (Véase figura 4.1). Las distancias serán usadas como fuentes de información a la hora de decantarnos por un comportamiento.

Comprobaremos si existe una correlación lineal entre las nuevas variables propuestas y las apariciones de caras, obteniendo los resultados de dichas variables a partir de una amplia experimentación. Con dicho coeficiente podremos saber que variables aportan nueva información útil respecto a las apariciones. Para obtener dicho coeficiente, emplearemos la fórmula de la correlación de Pearson:

$$r = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y}$$

Aún así, el hecho de que no exista una correlación lineal entre dos variables no implica que no exista ninguna relación entre ambas, pudiendo existir una correlación no lineal entre ambas.

El valor del índice de correlación varía en el intervalo $[-1, +1]$:

- Si $r = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada relación directa: cuando una de ellas aumenta, la otra también lo hace en proporción constante.
- Si $0 < r < 1$, existe una correlación positiva.
- Si $r = 0$, no existe relación lineal. Pero esto no necesariamente implica que las variables son independientes: pueden existir todavía relaciones no lineales entre las dos variables.
- Si $-1 < r < 0$, existe una correlación negativa.
- Si $r = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada relación inversa:

cuando una de ellas aumenta, la otra disminuye en proporción constante.

Obtenidas los diferentes coeficientes de correlación, entre el porcentaje de caras y las distintas distancias, trataremos de determinar cuales de ellas aportan más credibilidad a las máquinas ya existentes.

Capítulo 2

Estado del arte

*”La más larga caminata
comienza con un paso”
~ Proverbio hindú ~*

2.1. Historia del arte

Las civilizaciones más antiguas emplearon a utilizar monedas, evidentemente aparte de como medio de comercio, como forma para interpretar la voluntad divina. La referencia más antigua que se posee de este hecho se remonta a la antigua Grecia, donde se utilizaba el resultado de lanzar cuatro dados para predecir el futuro y conocer así la voluntad favorable o desfavorables de los dioses [5]

Incluso se sometía al azar la elección de los cargos, ya que era vista como lo más democrático y justo: Las elecciones favorecerían a los más ricos,

elocuentes y famosos, mientras que el sorteo repartía el trabajo de la administración entre toda la ciudadanía, integrándolos dentro de la experiencia democrática que, en palabras de Aristóteles, suponía "gobernar y ser gobernado en turnos"[23].

La asignación por sorteo, (proceso denominado *Dikasteria*[5]) de un cargo a un individuo estaba basada simplemente en su condición de ciudadano, y no en su mérito o cualquier forma de popularidad que pudiera ser comprada.

Este método fue considerado un medio para prevenir la compra corrupta de votos y dar a los ciudadanos una igualdad política total, ya que todos tenían la misma probabilidad de obtener un cargo gubernamental confiando en la aleatoriedad y justicia que el sorteo otorgaba.

Aparte de como medio justo y equilibrado de resolución de conflictos, la creencia de que el azar era la voluntad de los dioses estaba presente en casi todas las civilizaciones de la Edad Antigua. Pero esta no es una práctica que quedase desechada por la razón humana evolucionada, como apunta el autor de *Mito y sociedad en la Grecia Antigua*. Muchas otras culturas a lo largo de la historia han seguido relacionando los conceptos religión y azar.

Un ejemplo de la unión de voluntad divina y azar en una civilización posterior, como en la cultura sintoísta de mitades del s. XVI, es cuando en la batalla de Okehazama, en el 1560, el gran general japonés Nobunaga quería atacar a pesar de que su ejército era muy inferior al del enemigo, que le aventajaba en una proporción de diez a uno. Nobunaga estaba seguro de la victoria, pero sus hombres se mostraban temerosos a entrar en un combate tan desigual [9]. Cuando marchaban al combate, se detuvieron a orar en un templo y al salir, Nobunaga reunió a sus hombres y les dijo:

-Mientras oraba, he recibido una revelación divina: Voy a tirar esta moneda al aire, y si sale cara, debemos pelear con toda confianza y seguridad, pues la victoria estará de nuestro lado. Si sale cruz, no debemos ir a la batalla, pues seguro que seremos derrotados.

Así pues, lanzó la moneda y salió cara. Los soldados estaban tan deseosos de luchar y tan seguros de la victoria, que pelearon con inusitado valor y derrotaron sin problemas al enemigo.

Al día siguiente, le dijo un oficial a Nobunaga: *-Ciertamente, es imposible cambiar la voluntad de los dioses. -Por supuesto -le replicó Nobunaga mientras le mostraba una moneda que tenía cara por ambos lados.*

Es ya durante el Renacimiento (siglos XV y XVI) cuando el desarrollo del análisis matemático en los juegos de azar basados en maquinas simples aleatorias surge. Es entonces cuando se produce un abandono progresivo de las explicaciones teológicas para dar paso a un enfoque mas científico, reconsiderando los experimentos simples aleatorios.

A pesar de la prevalencia de los juegos en todas las épocas y culturas, por un largo periodo de tiempo hubo una cierta investigación occidental sobre el tema, posiblemente debido a la desaprobación que la Iglesia Católica mantenía en cuestiones de juegos y la adivinación, los cuales consideraba una práctica pecaminosa, pudiendo acarrear la práctica y estudio de los mismos graves sanciones y castigos.

Si bien Gerolamo Cardano y Galileo escribieron sobre los juegos de azar (recordemos que este ultimo fue condenado a morir en la hoguera por la Inquisición), fueron los franceses Blaise Pascal (1623-1662), Pierre de Fermat (1601-1665) y el holandés Christiaan Huygens (1629-1695) quienes nos

condujeron a lo que hoy se conoce como teoría de probabilidad. El cálculo de probabilidades se consolida como disciplina desde la segunda mitad del siglo XVII, siendo su primer impulsor Isaac Newton (1642-1727). Se crea la primera definición de probabilidad por Pierre Simon, marqués de Laplace (1749-1827).

El primer análisis sobre un sistema aleatorio como lo es la ruleta, lo realizó otro francés, Henri Poincarè (1854-1912). En su estudio demostraba que la mecánica de Newton, con toda su perfección y precisión, era incapaz de predecir que número va a salir en una tirada de dados o donde va a caer la bola en una ruleta [2].

Lo que Poincarè se planteaba era cómo la bola de la ruleta al ser lanzada y rebotar, obedeciendo unas reglas perfectamente conocidas, demostraba un comportamiento aleatorio. Ya que si uno supone que al analizar los rebotes que hace con física mecánica, finalmente se debería saber exactamente dónde va a caer la bola.

Henri Poincarè apuntaba en 1896:

”Una causa pequeñísima, que se nos escapa, determina un efecto considerable que no podemos dejar de ver, y entonces afirmamos que este efecto es debido al azar. Si conociésemos exactamente las leyes de la naturaleza y la situación del universo en el instante inicial, podríamos predecir la situación de este mismo universo en un instante posterior” [4].

Lo que quiere decir es que en un mundo lleno de factores incalculables e indetectables, habrá siempre cosas que la mecánica de Newton no podrá pre-

decir, ya que con el tiempo estas causas cambiarán de estado y echaran a perder las predicciones previamente calculadas. En el caso de la ruleta de Poincarè, por citar algunos de esos factores: Fuerza, ángulo de lanzamiento, posición inicial. . .

Estos estudios fueron continuados por Hopf (1934-1937) sobre el clásico problema de probabilidad de la aguja de Buffon [10], donde se plantea la cuestión de al lanzar una aguja sobre un suelo de tablas de madera paralelas entre ellas y distanciadas entre si de manera uniforme, cual seria la probabilidad de que la aguja se quede cruzada entre dos maderas. Hopf concluyó que se puede demostrar que si la distancia entre las maderas es igual a la longitud de la aguja, la probabilidad de que la aguja se quede cruzada entre alguna de las maderas es $2 = \pi$.

Posteriormente, el estudio detallado ya centrado sobre el lanzamiento de monedas empezó por Joseph B. Keller en 1986 con su revolucionario artículo *The Probability of Heads* donde determinó que las monedas no son justas[8].

Cuando se lanza una moneda, ésta tiene una cierta velocidad inicial y un cierto espín que es lo que, de acuerdo con la mecánica clásica, determina el cómo va a caer.

Keller llegó a la conclusión de que lo que realmente ocurre es que el plano en que se representan las dos variables, la velocidad inicial y el espín de la moneda queda particionado en regiones, de modo que si los valores de las variables están en una región la moneda cae tal como se lanzó y si están en la región contigua cae al contrario y así, de modo análogo y alternativamente, ocurre con todas las regiones. [8]

Las regiones resultantes, tal como muestra la figura 2.1 tienen contornos

con forma de hipérbolas y, a medida que aumentan la velocidad y el espín, las regiones se aproximan más y más, de modo que pequeños cambios en las condiciones iniciales determinan que el resultado del lanzamiento sea cara o cruz. Este análisis explica el porqué una moneda es aleatoria y el porqué, aunque se conozcan con una altísima precisión las condiciones iniciales, el resultado final va a ser aleatorio, es decir, no se puede predecir.

Otro aspecto muy importante que revela el análisis precedente es que si el número de revoluciones por segundo es pequeño, aproximadamente 15, como cuando se lanza una moneda a una altura de unos 30 cm, entonces los resultados sucesivos del lanzamiento de la moneda están correlados. Esta correlación es del orden de 0,1, lo que supone una correlación nada despreciable, que podría usarse para jugar con ventaja. Como cabe esperar, la correlación disminuye a medida que aumentan los valores de ambas variables, de modo que los resultados de sucesivos lanzamientos tienden a ser independientes [8].

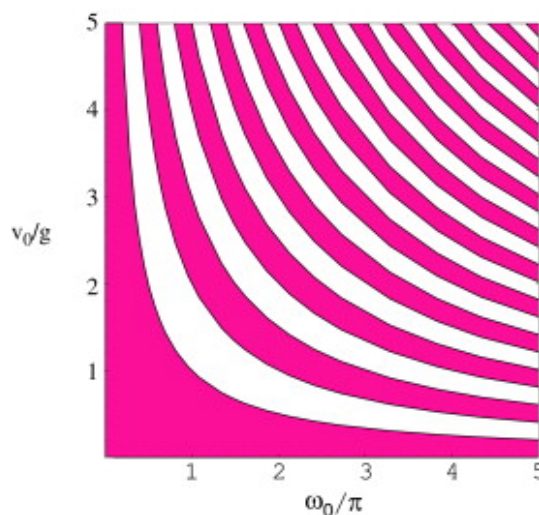


Figura 2.1: Figura de Keller sobre probabilidades de una moneda

Finalmente Keller concluyó que la cara de la moneda que está inicialmente

arriba es más probable que salga al final con 0.51 de probabilidades, rosa en la figura, que la otra, blanco en la figura, con 0.49[8].

Esta sorprendente conclusión de Joseph B. Keller, que hasta entonces había pasado desapercibida en estudios y práctica. ¿Cómo nunca se había percibido una desviación en el sesgo del lanzamiento de una moneda?

Por ejemplo, en la Segunda Guerra Mundial el matemático sudafricano John Kerrich [6], prisionero en un campo de concentración alemán en Dinamarca, lanzó 10000 veces una moneda, anotando únicamente los resultados para mas tarde publicarlos en su libro *An Experimental Introduction to the Theory of Probability* [7].

El matemático obvió tomar nota de qué lado de la moneda colocaba hacia arriba en cada lanzamiento, así que no pudo descubrir esta desviación con sus resultados, los cuales determinaban que la moneda que usaba para hacer los lanzamientos era aproximada a una moneda equilibrada, ya que evidentemente y sin control, fueron aproximadamente igual el número de veces las que puso la moneda mostrando la cara o la cruz, contando con que es una desviación insignificante para tal masa de experimentos.

En la práctica, el valor de esta desviación puede parecer mínimo e irrelevante, pero por ejemplo los Casinos, en la actualidad, en su oferta de juegos no incluyen nunca el lanzamiento de monedas, mientras que sí incluyen por ejemplo el lanzamiento de dados o ruleta, los cuales pueden presentar una desviación parecida dada una posición inicial en el lanzamiento. Pero como apuntaba Poincarè en su estudio [4] sus resultados van a seguir siendo impredecibles, entre otras cosas por tener un mayor numero de posibles variables que una moneda.

Como ejemplo de la desviación que pueden presentar estos sistemas probabilísticos en los Casinos, fue el conocido caso de la familia Pelayo en el casino de Torrelodones[12].

Basándose en la premisa de que algunas ruletas concretas deben tener alguna imperfección física y que no existe la ruleta perfectamente aleatoria, ya que todas pueden sufrir abombamientos, diferencias milimétricas en el tamaño de los casilleros de los números, flexibilidad de las placas separadoras ...

Basta con examinar los números ganadores durante varios miles de lanzamientos buscando un sesgo hacia los que más frecuentemente aparecen. Si la ruleta tiene una pequeña deformación, digamos, el 21 está en un ligero abombamiento, tal vez salga con más frecuencia de lo que cabría esperar y superados ciertos valores es favorable apostararlo (puede que ese sesgo supere la ventaja teórica del 2,7 % del casino por la inclusión del valor 0).

En su libro, García-Pelayo cuenta que tras examinar al menos 5.000 lanzamientos sobre una ruleta real, se analizan los números que han salido más de lo normal. (Salir más de lo normal significa que ese número aparezca más de $1/36$ de las veces, que sería lo habitual para obtener un premio).

Para saber si esa desviación es debida a un sesgo real del mecanismo de la ruleta o al puro azar, se comparan esos valores con dos límites:

- El primer límite es aquel que en una simulación realmente aleatoria por ordenador abarca al 95 % de los casos (sólo un 5 % de los casos se pasan del límite).
- El segundo límite es el que engloba al 99,95 % de las simulaciones (sólo

un 0,05 % de los casos pasan ese límite). Si tras esas 5.000 tiradas comprobadas algún número supera el primer límite significa que casi con toda probabilidad habrá un sesgo real sobre ese número en esa ruleta debido a algún defecto indeterminado de la ruleta.

Si se supera el segundo límite, el más estricto, el sesgo será según García-Pelayo absolutamente seguro y cierto mientras esa ruleta no se modifique o manipule. Por ejemplo: si tras 20.000 pruebas se espera el valor de +278 como límite al 99,95 % y se observa que el 36 ha salido +633 veces de lo normal es que tiene un defecto en ese número. Concluye en que algo realmente extraño le pasa al 36 y hay que jugarlo porque es un número ganador. Si ese sesgo supera el 2,7 % (ruleta europea) ó 5,4 % (en ruletas americanas, por inclusión del doble 0) de margen que tiene el casino, que es lo que sucede al pasar esos dos límites, la ruleta puede considerarse, en palabras de García-Pelayo, una caja de ahorros más que un juego de azar” [12].

Los casinos, obviamente tomaron sus contramedidas cuando descubrieron a los sistemistas, intercambiando las ruletas entre si, cambiando de modelos, intercambiando piezas de unas y otras ... A parte de prohibirles la entrada a muchos casinos.

Finalmente tras un largo contencioso de sentencias y recursos relativas a la prohibición que pesaba sobre ellos a entrar en el Casino Gran Madrid después de que ganaran astronómicas sumas de dinero gracias a este método, El Tribunal Supremo ha terminado por dar la razón a los Pelayo. De hecho, la sentencia definitiva viene a decir que ser ingenioso, calcular estadísticas o analizar los datos en un ordenador de casa no es hacer trampas («irregularidades

en la práctica de los juegos», según denunciaba el casino) y que por tanto los Pelayos tienen derecho a entrar en el casino cuando quieran[11].

Otra historia similar es la de los Eudaemons, un grupo de estudiantes del M.I.T. (Massachusetts Institute of Technology) que en entre en los 70 atacó el juego de la ruleta de varios casinos de Las Vegas, con un primitivo ordenador escondido en un zapato [13].

Volviendo al lanzamiento de monedas y centrándonos en la no aleatoriedad demostrada que poseen éstas una vez están en el aire y bajo condiciones conocidas, los canadienses Matthew Clark y Brian Westerberg, realizaron un estudio publicado en el *Journal de la Association Médicale Canadienne* (CMAJ)[15] en el que pidieron a 13 internados de la clínica St. Paul de la universidad de Columbia Británica en Vancouver lanzar una moneda al aire 300 veces con la intención de que cayera con la cara hacia arriba.

Todos los participantes recibieron una breve formación sobre cómo mejorar y perfeccionar la manera de lanzar una moneda para que el resultado sea el deseado. De esta forma se lograron más veces que el resultado fuera cara.

Siete de ellos lograron una preponderancia significativa y el mejor logró un 68% de resultados positivos.

Esta investigación muestra que cuando los participantes reciben simples instrucciones sobre el lanzamiento de la moneda y tienen algunos minutos para entrenarse, más de la mitad de ellos pueden orientar el resultado en proporciones significativas.

Los investigadores estadounidense de la Universidad de Standord, Persi Diaconis, Susan Holmes y Richard Montgomery publicaron *Dynamics of coin tossing is predictable*[14], donde muestran que el resultado de lanzar una

moneda es perfectamente determinista y en absoluto aleatorio bajo unas mismas condiciones iniciales controladas[14].

El resultado del lanzamiento viene determinado por las condiciones iniciales (posición, orientación y velocidad inicial y condiciones ambientales). Conocidas estas variables, es posible predecir el resultado "tan fielmente como podemos predecir el movimiento de la Luna o el de una manzana que cae" [14].

Podríamos resumir diciendo que si la fuerza con la que se hace el lanzamiento es siempre la misma, el resultado es siempre el mismo, si y solo si las condiciones de lanzamiento no varían.

Para probar el determinismo del lanzamiento de una moneda, construyeron una máquina que lanzaba una moneda al aire mediante un muelle y la recogían en un recipiente, siempre bajo las mismas condiciones. El resultado era que la posición final era la misma que la posición inicial.

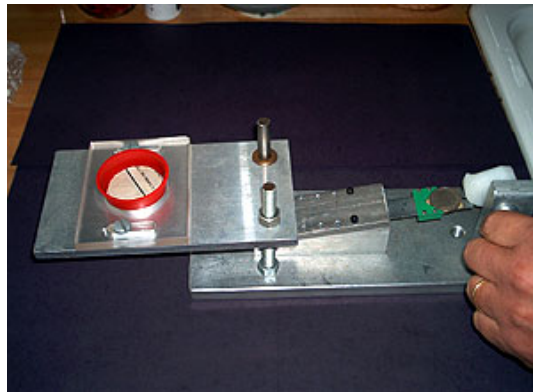


Figura 2.2: Máquina desarrollada para el lanzamiento de monedas. Vista 1

Hay que remarcar que las condiciones externas al lanzamiento eran siempre las mismas, sin ningún factor externo que pudiera distorsionar o modificar el vuelo de la moneda.

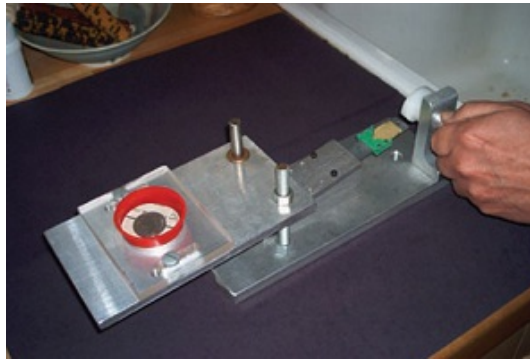


Figura 2.3: Máquina desarrollada para el lanzamiento de monedas. Vista 2

2.2. Estudios sobre la percepción de la aleatoriedad

En esta sección vamos a centrarnos en las percepciones populares sobre la aleatoriedad analizadas en el campo de la psicología, que determina que son frecuentemente equivocadas y erróneas, siendo basadas en falacias lógicas. A continuación numeraremos y comentaremos las más comunes.

Un número está "debido"

Este argumento dice que "ya que todos los números eventualmente saldrán en una selección aleatoria, aquellos que todavía no hayan salido están 'debidos' y es más probable que salgan pronto"[16].

Esta lógica es solo correcta si es aplicada a un sistema donde los números que salen son removidos del sistema. Como por ejemplo sobre una baraja, una vez que una carta es extraída del mazo, la próxima carta es menos probable que sea una de este tipo y más probable que sea otra cualquiera.

Sin embargo, si esa carta es regresada al mazo y este se vuelve a barajar,

hay una oportunidad igual de sacar esa misma carta que cualquier otra carta la próxima vez.

La misma verdad se aplica a cualquier otro caso, donde los objetos son seleccionados independientemente y nada es removido del sistema después de cada acción, así como una tirada de dado, de moneda o la mayoría de las loterías.

Un número está "maldito"

Este argumento es casi el opuesto del de anteriormente citado, ya que dice que los números que han salido de forma menos seguida en el pasado continuarán viniendo con menor frecuencia en el futuro.

Un argumento similar, "número bendito", puede estar diciendo que los números que han salido con mayor frecuencia en el pasado lo harán en el futuro. Esta lógica será únicamente válida si la tirada es parcial y los resultados no tienen iguales posibilidades de salir, por ejemplo, con una moneda con un sesgo 80 %.

Si sabemos con seguridad que la tirada es justa (50 %-50 %), entonces podemos garantizar que eventos previos no tienen influencia en eventos futuros.

En la naturaleza, eventos inesperados o inciertos, (tales como terremotos, huracanes...) raramente ocurren con frecuencias perfectamente iguales. De esta manera el aprender qué eventos es posible que tengan probabilidades más altas observando resultados tiene sentido. Lo que es falaz es aplicar esta lógica a sistemas que son especialmente diseñados para que todos los resultados sean igualmente probables, tales como dados, ruletas, etc [17].

2.2.1. Prejuicios cognitivos

Un prejuicio cognitivo es una distorsión cognitiva que afecta al modo en el que los humanos perciben la realidad. Algunos de estas distorsiones han recibido apoyo empírico en el campo de la psicología, otros están siendo considerados como categorías generales de prejuicios [16] [17].

Falacia del jugador

La falacia del jugador es una falacia lógica por la que se cree erróneamente que los sucesos pasados afectan a los futuros en lo relativo a actividades aleatorias, como en muchos juegos de azar. Puede comprender las siguientes ideas equivocadas:

- Un suceso aleatorio tiene más posibilidades de ocurrir porque no ha ocurrido durante cierto período.
- Un suceso aleatorio tiene menos posibilidades de ocurrir porque no ha ocurrido durante cierto período.
- Un suceso aleatorio tiene más posibilidades de ocurrir si ocurrió recientemente.
- Un suceso aleatorio tiene menos posibilidades de ocurrir si ocurrió recientemente.

Las anteriores son ideas equivocadas que surgen cotidianamente en razonamientos sobre probabilidades. Mucha gente pierde dinero apostando debido a una creencia errónea en esta falacia.

Las probabilidades de que algo suceda la próxima vez no están necesariamente relacionadas con lo que ya sucedió, especialmente en muchos juegos de azar.

La falacia del jugador puede ilustrarse considerando el lanzamiento repetido de una moneda. Si ésta está equilibrada (Sesgo=0.50), las opciones de que salga cara son exactamente 0.5 (una de cada dos). Las opciones de que salgan dos caras seguidas es $0,5 * 0,5 = 0,25$ (una de cada cuatro), las de obtener tres caras seguidas son $0.5*0.5*0.5=0.125$ (una de cada ocho), y así sucesivamente.

Supongamos que se han sacado cuatro caras seguidas. Un creyente en la falacia del jugador diría: «Si en el siguiente lanzamiento saliese cara, habrían salido cinco consecutivas. La probabilidad de que esto suceda es $0,55 = 0,03125$, así que por tanto en el siguiente lanzamiento la probabilidad de que salga cara es sólo 1 entre 32.»

Éste es el paso falaz en el razonamiento. Si la moneda está equilibrada, entonces por definición la probabilidad debe ser siempre 0,5 (obviando la desviación de Keller [8]) tanto para cara como para cruz. Aunque la posibilidad de lograr una serie de cinco caras consecutivas es de sólo 1 cada 32 (0,03125), lo es antes de que la moneda se tire por primera vez. Después de los primeros cuatro lanzamientos los resultados ya no son desconocidos, y por tanto no cuentan.

La probabilidad de lograr cinco caras consecutivas es la misma que la de cuatro caras seguidas de una cruz. Las cruces no son más probables. Cada uno de los dos posibles resultados tiene la misma probabilidad independientemente del número de veces que la moneda se haya lanzado antes y de los

resultados obtenidos. Razonar que es más probable que el próximo lanzamiento será cruz en vez de cara debido a los anteriores lanzamientos es la falacia: la idea de que una racha de suerte pasada influye de alguna forma en las posibilidades futuras.

A veces los jugadores argumentan: *«Acabo de perder cuatro veces seguidas. Como la moneda está equilibrada y por tanto a la larga los resultados lo estarán también, si me limito a seguir jugando terminaré por recuperar mi dinero»*.

Sin embargo, es irracional considerar las cosas a la larga comenzando desde antes de empezar a jugar: Debe considerarse siempre a la larga desde la posición actual, y no puede esperarse que el juego se equilibre desde la posición inicial por que ya se acumulen cuatro juegos perdidos.

Como ejemplo, la estrategia popular conocida como la Martingala, técnica consistente en doblar la apuesta (comenzar con 1, si se pierde apostar 2, luego 4... hasta que se gane) no funciona. Originariamente la Martingala se refería a un tipo de estrategia de apuesta muy popular en Francia en el siglo XVIII. La más simple de estas estrategias fue diseñada para un juego en el que el apostante gana la apuesta en caso de que al lanzar una moneda caiga de cara y pierde en caso de que salga cruz. Actualmente se ha extrapolado a la ruleta jugando a colores, ya que como comentábamos anteriormente en los casinos actuales no están contemplados juegos de lanzamiento de monedas.

Esta y otras estrategias parecidas canjean muchas pequeñas ganancias por unas pocas pérdidas enormes, sin contar con tres factores determinantes:

- Las mesas tienen un límite de apuesta: En España por ejemplo, el límite de los casinos 777.es es 900 euros[19]. Esto quiere decir que tras una secuencia de once resultados negativos al jugador no se le permite seguir apostando y pierde el dinero.
- Que el dinero del jugador es limitado y no finito: Aunque el casino permitiera apuestas máximas enormes, el hecho de ir doblando la apuesta hace crecer la apuesta de forma exponencial. Las cantidades crecen de forma desorbitada al poco tiempo.
- Las secuencias largas de varios resultados negativos acaban apareciendo a largo plazo. Sucesos improbables, aunque sean realmente muy improbables, acaban produciéndose si uno espera suficiente tiempo a que sucedan [3].

Con una cantidad infinita de capital disponible y sin límite de apuesta impuesto por el casino, podría adoptarse esta estrategia.

Adviértase que la falacia del jugador es bastante diferente del siguiente hilo de razonamiento (que lleva a la conclusión opuesta): «*la moneda da cara más veces que cruz, por lo que no está equilibrada, así que apostaré que en el siguiente lanzamiento también saldrá cara*». Esto no es una falacia, si bien el primer paso (del argumento a partir de un número finito de observaciones a la afirmación de sesgo de la moneda) es muy delicado y en sí mismo proclive a falacias de su propio tipo peculiar.

Algunos afirman que la falacia del jugador es un sesgo cognitivo provocado por una heurística psicológica llamada heurística de representatividad [17].

El problema de Monty Hall

El problema de Monty Hall es un problema matemático de probabilidad que está inspirado por el concurso televisivo estadounidense *Let's Make a Deal* (Hagamos un trato).[18]

Se ofrece un concurso cuya mecánica es la siguiente:

- Al concursante se le ofrece la posibilidad de escoger entre tres puertas. Tras una de ellas se encuentra un coche, y tras las otras dos hay una cabra. El concursante gana el premio que se oculta detrás de la puerta que escoja.
- Después de que el concursante escoja una puerta, el presentador abre una de las otras dos puertas, mostrando una cabra. Siempre puede hacerlo ya que incluso si el concursante ha escogido una cabra, queda otra entre las puertas que ha descartado y el presentador conoce lo que hay detrás de cada puerta.
- Entonces, ofrece al concursante la posibilidad de cambiar su elección inicial y escoger la otra puerta que descartó originalmente, que continúa cerrada.

El problema es: ¿Debe hacerlo o no? La solución se basa en tres suposiciones básicas:

- Que el presentador siempre abre una puerta.
- Que la escoge entre las restantes después de que el concursante escoja la suya

- Que tras ella siempre hay una cabra.

Estas suposiciones no se encuentran explícitamente en el enunciado.

La probabilidad de que el concursante escoja en su primera oportunidad la puerta que oculta el coche es de $1/3$, por lo que la probabilidad de que el coche se encuentre en una de las puertas que no ha escogido es de $2/3$. ¿Qué cambia cuando el presentador muestra una cabra tras una de las otras dos puertas?

Una suposición errónea es que, una vez sólo queden dos puertas, ambas tienen la misma probabilidad (un 50%) de contener el coche. Es errónea ya que el presentador abre la puerta después de la elección de jugador. Esto es, la elección del jugador afecta a la puerta que abre el presentador. No es un suceso aleatorio ni inconexo.

Si el jugador escoge en su primera opción la puerta que contiene el coche (con una probabilidad de $1/3$), entonces el presentador puede abrir cualquiera de las dos puertas. Además, el jugador pierde el coche si cambia cuando se le ofrece la oportunidad.

Pero, si el jugador escoge una cabra en su primera opción (con una probabilidad de $2/3$), el presentador sólo tiene la opción de abrir una puerta, y esta es la única puerta restante que contiene una cabra. En ese caso, la puerta restante tiene que contener el coche, por lo que cambiando lo gana.

En resumen, si mantiene su elección original gana si escogió originalmente el coche (con probabilidad de $1/3$), mientras que si cambia, gana si escogió originalmente una de las dos cabras (con probabilidad de $2/3$). Por lo tanto, el concursante debe cambiar su elección si quiere maximizar la probabilidad de

ganar el coche.

La explicación matemática[18] es la siguiente:

Sea $X : (\omega, P) \rightarrow 1, 2, 3$ la puerta aleatoria detrás de la cual se encuentra el coche.

Sea $Y : (\omega, P) \rightarrow 1, 2, 3$ la puerta que escoge aleatoriamente el candidato.

Las variables aleatorias X e Y son estocásticamente independientes.

Sea $M : (\omega, P) \rightarrow \{cabra, coche\}$ lo que se encuentra detrás de la puerta que el moderador, de manera aleatoria, escoge (entre las que aún no se han abierto). Se cumple entonces $[M=cabra]$ con probabilidad=1 (esto es, siempre).

La probabilidad que el candidato se lleve el coche bajo el supuesto que él no cambia de puerta es entonces

$$P[X = Y | M = cabra] = P[X = Y] = 1/3$$

.

La probabilidad que el candidato se lleve el coche bajo el supuesto que él cambia de puerta es entonces

$$P[X \neq Y | M = cabra] = 1 - P[X = Y] = 2/3$$

.(Esta es la solución correcta.)

Una solución incorrecta se obtiene de la siguiente interpretación: Si, por otro lado, el presentador escoge de manera aleatoria y uniforme entre las

puertas que aún no se han abierto, entonces la probabilidad que el candidato se lleve el coche (dado que él no cambia de puerta) es

$$\begin{aligned}
 P[X = Y|M = \text{cabra}] &= P[X = Y]/P[M = \text{cabra}] = \\
 P[X = Y]/(P[M = \text{cabra}|X = Y]P[X = Y] + \\
 P[M = \text{cabra}|X \neq Y]P[X \neq Y]) &= (1/3)/(1/3 + (1/2) * (2/3)) = 1/2.
 \end{aligned}$$

Por lo tanto, 0,5 es la probabilidad que el candidato se lleve el coche (dado que él cambia de puerta), pero esta respuesta no es aplicable a nuestro problema.

Otra forma para ver el planteamiento es la siguiente: Definimos los eventos A: El concursante elige la puerta con el premio antes de cambiar de opción y; B: El concursante elige la puerta con el premio después de cambiar de opción. Entonces aplicando el teorema de Probabilidad Total, tenemos:

$$\begin{aligned}
 P[B] &= P[BA] + P[B \neg A] = P[B|A]P[A] + \\
 P[B|\neg A]P[\neg A] &= \\
 (0)(1/3) + (1)(2/3) &= 2/3.
 \end{aligned}$$

$P[B|A] = 0$, puesto que son eventos mutuamente excluyentes. $P[A] = 1/3$, debido a que desde el inicio elige una puerta de tres y todas son equiprobables. $P[B|\text{not}A] = 1$, es porque si eligió la puerta incorrecta desde el principio y posteriormente realizar el cambio, siempre ganará. $P[\neg A] = 2/3$, porque $P[\neg A] = 1 - P[A] = 1 - 1/3 = 2/3$.

Esto ocurre ya que lo que muestra el presentador no afecta a tu elección original, sino sólo a la otra puerta no escogida. Una vez que se abre una puerta y se muestra la cabra, esa puerta tiene una probabilidad 0 de contener un coche, por lo que deja de tenerse en cuenta. Si el conjunto de dos puertas tenía una probabilidad de contener el coche de $2/3$, entonces, si una tiene una probabilidad de 0, la otra debe tener una probabilidad de $2/3$. La elección, básicamente, consiste en preguntarte si prefieres seguir con tu puerta original o escoger las otras dos puertas. La probabilidad de $2/3$ se traspa a la otra puerta no escogida (en lugar de dividirse entre las dos puertas restantes de modo que ambas tengan una probabilidad de $1/2$) porque en ningún caso puede el presentador abrir la puerta escogida inicialmente. Si el presentador escogiese al azar entre las dos puertas con cabras (incluyendo la del concursante), abriese una de ellas y luego diese de nuevo a elegir, entonces las dos puertas restantes sí tendrían la misma probabilidad de contener el coche.

2.3. Lógica clásica

La lógica crisp o lógica clásica es un sistema formal diseñado para analizar ciertos tipos de argumentos. En la lógica proposicional, las fórmulas representan proposiciones y las constantes lógicas son operaciones sobre las fórmulas que producen otras fórmulas de mayor complejidad.

Como otros sistemas lógicos, la lógica proposicional intenta esclarecer nuestra comprensión de la noción de consecuencia lógica para el rango de argumentos que analiza.

En la lógica clásica, las constantes lógicas son tratadas como funciones de

verdad. Es decir, como funciones que toman conjuntos de valores de verdad y devuelven valores.

Por ejemplo, la constante lógica "no" es una función que si toma el valor de verdad 1, devuelve 0, y si toma el valor de verdad 0, devuelve 1. Por lo tanto, si se aplica la función "no" a una letra que represente una proposición falsa, el resultado será algo verdadero. Si es falso que "está lloviendo", entonces será verdadero que "no está lloviendo".

El significado de las constantes lógicas no es nada más que su comportamiento como funciones de verdad. Cada constante lógica se distingue de las otras por los valores de verdad que devuelve frente a las distintas combinaciones de valores de verdad que puede recibir. Esto quiere decir que el significado de cada constante lógica puede ilustrarse mediante una tabla que despliegue los valores de verdad que la función devuelve frente a todas las combinaciones posibles de valores de verdad que puede recibir.

A continuación se incluye una tabla con los operadores lógicos, tablas de verdad, símbolos y equivalente lingüístico de los mismos.

Conectiva	Tabla	Lenguaje natural	Notación															
Negación	<table border="1"> <thead> <tr> <th>ϕ</th> <th>$\neg\phi$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table>	ϕ	$\neg\phi$	1	0	0	1	no	\neg									
ϕ	$\neg\phi$																	
1	0																	
0	1																	
Conjunción	<table border="1"> <thead> <tr> <th>ϕ</th> <th>ψ</th> <th>$\phi \wedge \psi$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	ϕ	ψ	$\phi \wedge \psi$	1	1	1	1	0	0	0	1	0	0	0	0	y	\cap
ϕ	ψ	$\phi \wedge \psi$																
1	1	1																
1	0	0																
0	1	0																
0	0	0																
Disyunción	<table border="1"> <thead> <tr> <th>ϕ</th> <th>ψ</th> <th>$\phi \vee \psi$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	ϕ	ψ	$\phi \vee \psi$	1	1	1	1	0	1	0	1	1	0	0	0	o	\cup
ϕ	ψ	$\phi \vee \psi$																
1	1	1																
1	0	1																
0	1	1																
0	0	0																
Condición	<table border="1"> <thead> <tr> <th>ϕ</th> <th>ψ</th> <th>$\phi \rightarrow \psi$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	ϕ	ψ	$\phi \rightarrow \psi$	1	1	1	1	0	0	0	1	1	0	0	1	si	\rightarrow
ϕ	ψ	$\phi \rightarrow \psi$																
1	1	1																
1	0	0																
0	1	1																
0	0	1																
Bicondicional	<table border="1"> <thead> <tr> <th>ϕ</th> <th>ψ</th> <th>$\phi \leftrightarrow \psi$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	ϕ	ψ	$\phi \leftrightarrow \psi$	1	1	1	1	0	0	0	1	0	0	0	1	si y solo si	\leftrightarrow
ϕ	ψ	$\phi \leftrightarrow \psi$																
1	1	1																
1	0	0																
0	1	0																
0	0	1																

Cuadro 1: Tabla con operadores lógicos clásicos

2.4. Lógica *fuzzy*

La lógica *fuzzy* o lógica difusa es una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de una información de entrada vaga, ambigua, imprecisa, con ruido o incompleta. En general la lógica difusa imita como una persona toma decisiones basada en información con las características mencionadas. Una de las ventajas de la lógica difusa es la posibilidad de implementar sistemas basados en ella tanto en hardware como en software, o en combinación de ambos [26].

El concepto clave para entender como trabaja la lógica difusa es el de **conjunto difuso**. Se puede definir un conjunto difuso de la siguiente manera:

Teniendo un posible rango de valores al cual llamaremos U , por ejemplo $U = R_n$, donde R_n es un espacio de n dimensiones, a U se le denominara Universo de Discurso. En U se tendrá un conjunto difuso de valores llamado F el cual es caracterizado por de una función de pertenencia uf tal que $uf : U \rightarrow [0, 1]$, donde $uf(u)$ representa el grado de pertenencia de un u que pertenece a U en el conjunto difuso F .

Conjuntos difusos

Supongamos que se desea representar con conjuntos difusos la variable altura de una persona, en este caso el universo de discurso será el rango de posibles valores de la altura que tenga una persona adulta, se escogerá un rango entre 140 cm y 200 cm, valores por fuera de este rango son posibles pero son muy escasos. El universo de discurso $U = [140, 200]$, para denominar los conjuntos difusos se suelen trabajar con etiquetas lingüísticas similares a

las que se usan de manera coloquial por ejemplo, en la vida diaria decimos que una persona es Muy Baja (MB), Baja (B), Mediana (M), Alta (Alta) y Muy Alta (MA).

Etiqueta	Rango $[min, max]$
MB	[140, 160]
B	[160, 170]
M	[170, 180]
A	[180, 190]
MA	[190, 200]

Cuadro 2.1: Listado de alumnos

Empleando lógica *fuzzy* para determinar estos conjuntos de alturas, la representación sería de la forma:

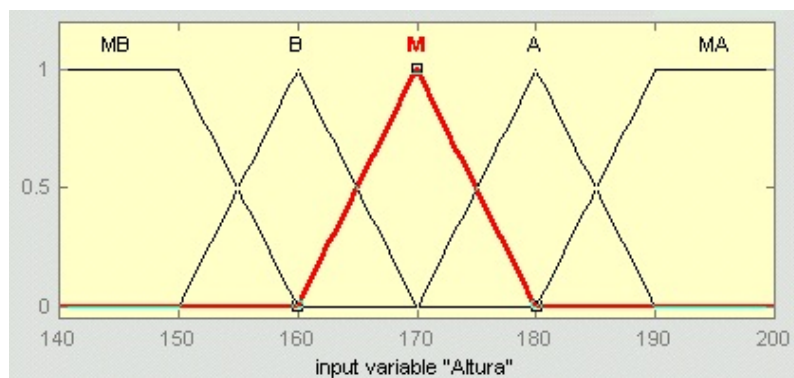


Figura 2.4: Conjuntos fuzzy o difusos para la altura de una persona

Mientras que si sobre el ejemplo anterior se desea trabajar con conjuntos clásicos (lógica *crisp*) o lógica clásica se tendrán únicamente dos opciones, o alguien Alto (A) o Bajo (B). Se supondrá que alguien Alto si mide mas de 170cm es caso contrario es bajo:

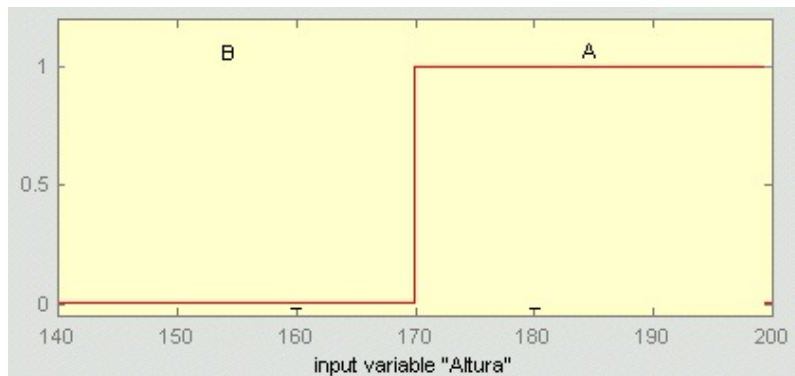


Figura 2.5: Conjuntos crisp-clásicos para la altura de una persona

Operaciones entre conjuntos difusos

De manera similar a la que entre los conjuntos *crisp* se realizan operaciones entre ellos, en conjuntos *fuzzy* se puede hacer lo mismo, pero debido a la naturaleza diferente de ellos la formulación de estas operaciones es algo especial.

En la figura 2.6, se muestran dos conjuntos difusos los cuales nos servirán para definir las operaciones fundamentales que entre ellos se pueden realizar:

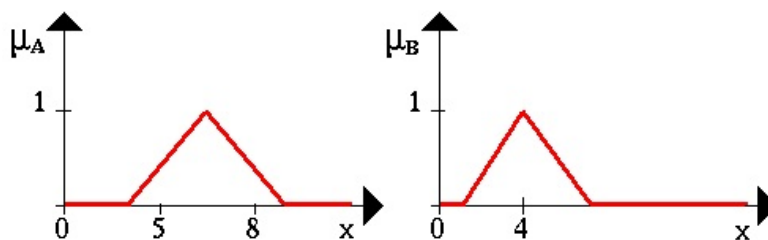


Figura 2.6: Conjuntos difusos entre los se definirán las operaciones

Intersección

La idea intuitiva de intersección heredada de los conjuntos crisp expresa que el conjunto intersección de dos conjuntos A y B , se define como los elementos que están en el conjunto A y (AND) en el conjunto B ; de esta manera la intersección entre conjuntos se puede entender como el una operación tipo AND entre los mismos.

Siguiendo esta idea, se podría representar de forma gráfica la intersección de los conjuntos difusos mostrados en la figura 2.6.

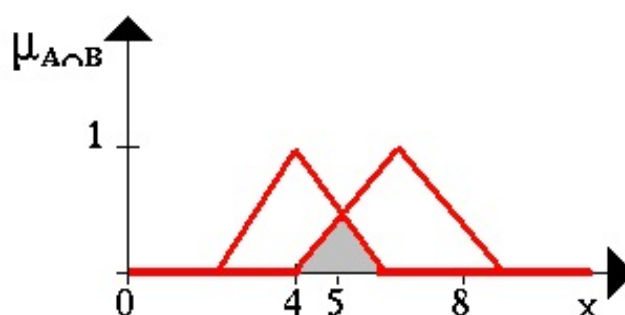


Figura 2.7: Intersección entre dos conjuntos difusos

De manera similar a como se define el nivel de pertenencia a un conjuntos difuso, vamos a encontrar el nivel de pertenencia de valor $x = 4,5$ a la intersección de los dos conjuntos difusos mostrados.

Gráficamente sabemos que el valor $x = 4,5$ tiene un nivel de pertenencia de 0,8 al conjunto A y de 0,2 al conjunto B , y el valor de pertenencia de $x = 4,5$ a la intersección (zona sombreada) se desea expresar como una operación entre estos valores se observa que de estos dos valores, el que "toca" la zona sombreada es el de 0,2 por lo que de manera intuitiva se puede afirmar que el valor de pertenencia del valor dado a la intersección de los

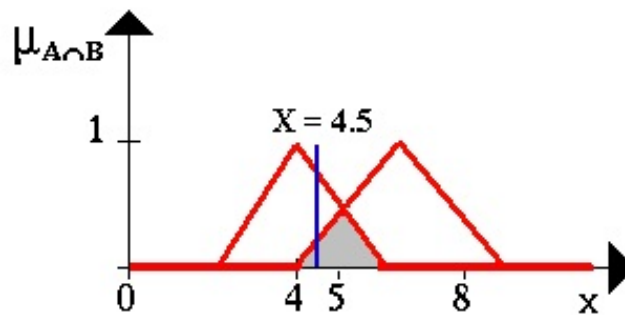


Figura 2.8: Cual es el valor de pertenencia de $x=4.5$ a la intersección de los conjuntos difusos A y B

conjuntos A y B es el valor mínimo de los valores de pertenencia del dicho valor a los conjuntos de manera individual, de manera matemática lo anterior se puede expresar así:

$$C = A \cap B, \forall x \in U$$

$$\mu_{A \cap B} = \min\{\mu_A(x), \mu_B(x)\}$$

Unión

La idea intuitiva de unión heredada de los conjuntos crisp expresa que el conjunto unión de dos conjuntos A y B , se define como los elementos que están en el conjunto A OR están en el conjunto B . de esta manera la intersección entre conjuntos se puede entender como el una operación tipo OR entre los mismos.

Siguiendo esta idea, se podría expresar la unión de los conjuntos difusos mostrados en la figura 2.6 de esta forma:

De manera similar a como se define el nivel de pertenencia a un conjuntos

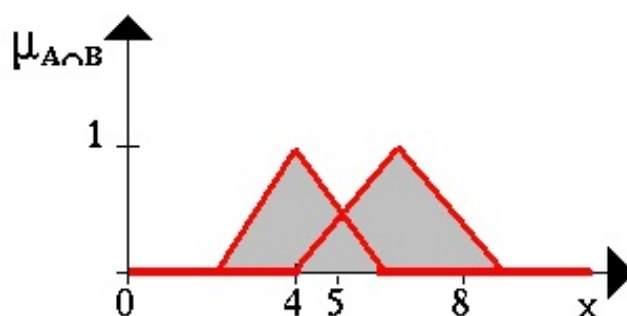


Figura 2.9: Unión entre dos conjuntos difusos

difuso, vamos a encontrar el nivel de pertenencia de valor $x = 4.5$ a la unión de los dos conjuntos difusos mostrados.

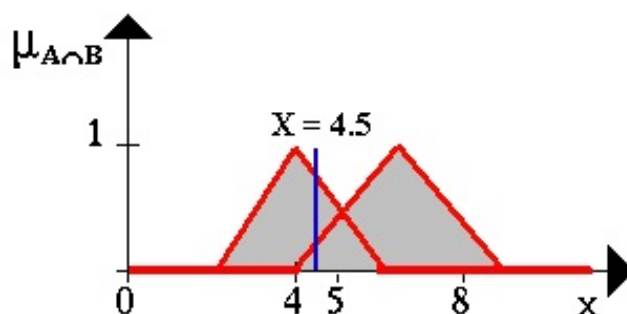


Figura 2.10: Cual es el valor de pertenencia de $x=4.5$ a la unión de los conjuntos difusos A y B

Gráficamente sabemos que el valor $x = 4,5$ tiene un nivel de pertenencia de 0,8 al conjunto A y de 0,2 al conjunto B , y el valor de pertenencia de $x = 4,5$ a la unión (zona sombreada) se desea expresar como una operación entre estos valores se observa que de estos dos valores, el que "toca" la zona sombreada es el de 0,8 por lo que de manera intuitiva se puede afirmar que el valor de pertenencia del valor dado a la unión de los conjuntos A y B es el valor máximo de los valores de pertenencia de dicho valor a los conjuntos de

manera individual, de manera matemática lo anterior se puede expresar así:

$$C = A \cup B, \forall x \in U$$

$$\mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$$

Complemento

En conjuntos crisp se define el complemento como el conjunto de los elementos que le faltan a un conjunto para ser igual al conjunto universo.

De la misma manera en conjuntos difusos se habla del complemento como el conjunto formado por los valores de pertenencias que le permitirían al conjunto obtener el valor máximo de pertenencia posible, siendo 1 el valor máximo de pertenencia que un conjunto difuso puede suministrar, este conjunto se podría formar prestándole a 1 los valores de pertenencia del conjunto difuso al que se desea encontrar el complemento.

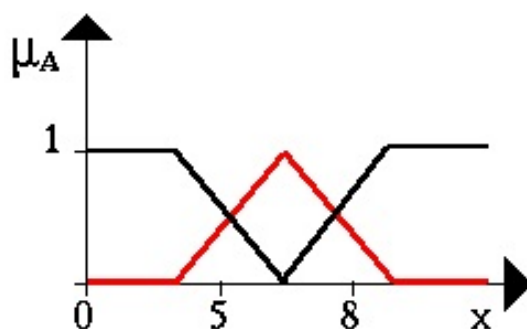


Figura 2.11: Complemento de un conjunto difuso

En la imagen anterior, el conjunto complemento se ha dibujado con trazo negro. De manera similar a como se define el nivel de pertenencia a un

conjuntos difusos, vamos a encontrar el nivel de pertenencia de valor $x = 6$ al complemento del conjunto difuso A .

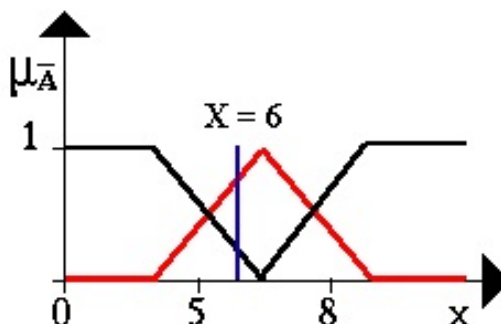


Figura 2.12: Cual es el valor de pertenencia de $x=6$ al complemento del conjunto difusos A

En $x = 6$ se observa que el valor de pertenencia al conjunto A es de 0,8, si pensamos en el complemento como lo que le falta a esta valor para alcanzar el máximo valor posible (1), sabemos que se tendría el nivel del pertenencia de $x = 6$ al complemento es de 0,2, en la gráfica se puede verificar esta conclusión. Matemáticamente esta operación se expresa así:

$$\mu_{\bar{A}} = 1 - \mu_A$$

La lógica difusa se adapta mejor al mundo real en el que vivimos. La clave de esta adaptación al lenguaje se basa en comprender los cuantificadores de nuestro lenguaje.

Esta lógica se basa en reglas heurísticas de la forma

SI (antecedente) **ENTONCES** (consecuente)

Donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Las reglas de las que dispone el motor de inferencia de un sistema difuso pueden ser formuladas por expertos, o bien aprendidas por el propio sistema

Sin embargo, la utilización de la lógica borrosa para el control de sistemas tiene sus ventajas y desventajas y por lo tanto hay que conocerlas y analizarlas, entre otras plantearemos las siguientes:

- Con los sistemas basados en la lógica borrosa se pueden evaluar mayor cantidad de variables, entre otras, variables lingüísticas, no numéricas, simulando el conocimiento humano.
- Se relacionan entradas y salidas, sin tener que entender todas las variables, permitiendo que el sistema pueda ser más confiable y estable que uno con un sistema de control convencional.
- Se puede simplificar la asignación de soluciones previas a problemas sin resolver.
- Ante un problema que tiene solución mediante un modelo matemático, obtenemos peores resultados usando Lógica Difusa.

Es posible obtener prototipos, rápidamente, ya que no requiere conocer todas las variables acerca del sistema antes de empezar a trabajar, siendo su desarrollo más económico que el de sistemas convencionales, porque son más fáciles de designar.

También se simplifica la adquisición y representación del conocimiento y unas pocas reglas abarcan gran cantidad de complejidades.

2.5. Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson es un índice que mide la relación lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida de las variables. La definición empleada es:

$$r = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y}$$

El coeficiente de correlación entre dos variables aleatorias X e Y es el cociente donde XY es la covarianza de (X,Y) y X y Y las desviaciones típicas de las distribuciones marginales.

Siendo la covarianza:

$$S_{XY} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n} \sum_{i,j} x_i \cdot y_j n_{ij} - \bar{x}\bar{y}$$

Y la varianza:

$$\sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^N (X_i - \mu)^2}{N}}$$

El valor del índice de correlación varía en el intervalo [-1, +1]:

- Si $r = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada relación directa: cuando una de ellas aumenta, la otra también lo hace en proporción constante.

- Si $0 < r < 1$, existe una correlación positiva.
- Si $r = 0$, no existe relación lineal. Pero esto no necesariamente implica que las variables son independientes: pueden existir todavía relaciones no lineales entre las dos variables.
- Si $-1 < r < 0$, existe una correlación negativa.
- Si $r = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada relación inversa: cuando una de ellas aumenta, la otra disminuye en proporción constante.

2.6. Números pseudoaleatorios

La librería Math (`java.util.Random`) de Java, de la cual usamos la función `Random` para generar los lanzamientos, nos proporciona valores pseudoaleatorios para la generación de los lanzamientos de las monedas.

Estos valores serán considerados siempre como pseudoaleatorios y nunca como puramente aleatorios, debido a que la función empleada trata de no mostrar un patrón de creación no evidente, y que así parezcan ser números aleatorios.

El método base para este tipo de generación es el método congruencial de generación de números en el conjunto $\{1, 2, \dots, (N-1)\}$. Para ello se necesita:

- Los valores A, B como enteros.
- Un valor inicial, llamado semilla, X_0 en el conjunto $\{1, 2, \dots, (N-1)\}$

Se aplica el esquema recursivo:

$$X_1 = (A \cdot X_0 + B) \bmod N$$

$$X_2 = (A \cdot X_1 + B) \bmod N$$

...

$$X_{k+1} = (A \cdot X_k + B) \bmod N$$

Donde por $X \bmod N$ se indica el resto de la división entera de X entre N.

Por ejemplo, $12 \bmod 7 = 5$

Por lo tanto, si se conoce cómo generar valores en $\{1, 2, \dots, (N - 1)\}$ es posible generar valores en el intervalo $[0,1)$, como sigue:

- Se toma N que será un entero grande (long int).
- Se toma una semilla $X_0 \in \{1, 2, \dots, N - 1\}$.
- Se construye la secuencia $Y_0, Y_1, Y_2 \dots$, obtenida aplicando la expresión $Y_k = X_k/N$.

Así que por lo tanto, siempre que se parta de la misma semilla, se obtendrá la misma secuencia de valores (de diferente longitud, evidentemente). Para lo cual es necesario cambiar la semilla en cada lanzamiento.

Así que de acuerdo con esta demostración y a fin de proporcionar la

mayor aleatoriedad posible a la función, por cada lanzamiento de moneda cambiaremos la semilla de la función Random de Java.

Una demostración gráfica de esta pseudoaleatoriedad[21] la podemos ver en los siguientes bitmaps generados con numeros aleatorios de la función Rand() de PHP y por la página web Random.org:

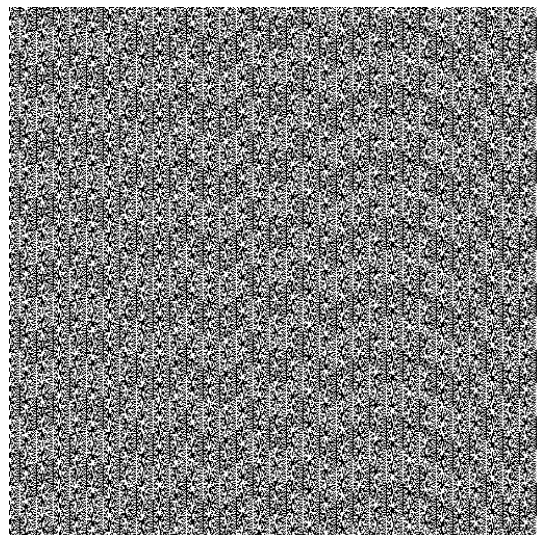


Figura 2.13: Bitmap de valores pseudoaleatorios generados por la función Rand de PHP

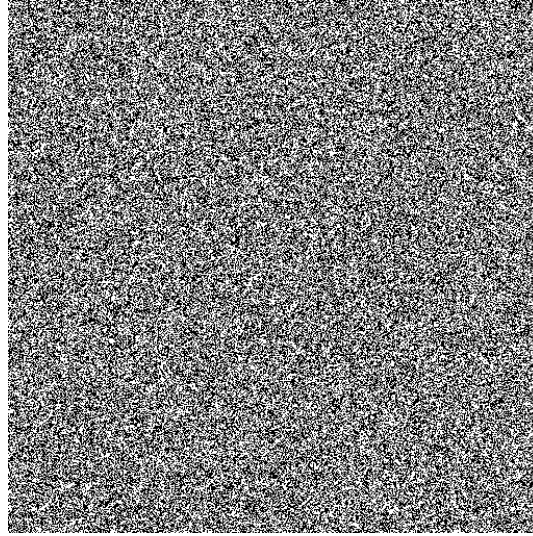


Figura 2.14: Bitmap de valores pseudoaleatorios generados por la pagina web Random.org

La secuencia generada por la función `Random` (`java.util.Random`) de la librería `Math` empleado por Java, genera el patrón[22] mostrado en la figura 2.15.

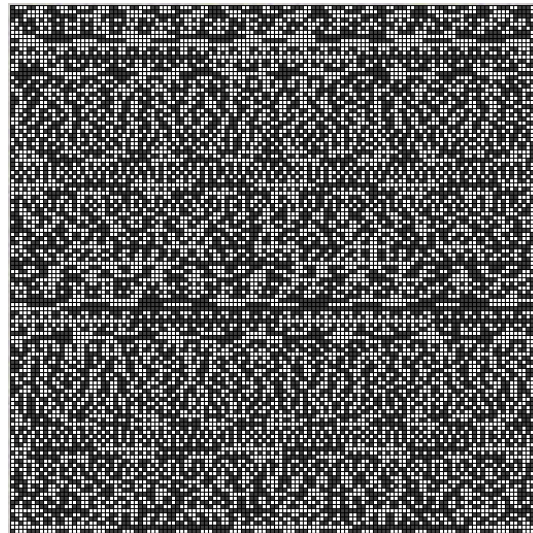


Figura 2.15: Bitmap de valores pseudoaleatorios generados por la funcion `Random` de la libreria `Math` en Java

Capítulo 3

Planteamiento y solución

*”La formulación de un problema,
es incluso más importante que su solución.”*

Albert Einstein

En esta sección vamos a tratar la naturaleza del problema propuesto así como las soluciones planteadas para la resolución del mismo.

3.1. ¿Qué implica el lanzamiento de una moneda?

La acción de lanzar una moneda al aire una vez, es un experimento con dos eventos mutuamente excluyentes. Esto quiere decir que en un lanzamiento sólo podrá aparecer un valor: cara o cruz. Nunca los dos valores a la vez.

Podremos decir que el experimento de lanzar una moneda es equivalente

a la generación al azar de un bit, pudiendo ser éste 0 o 1.

Este tipo de experimento está considerado un **ensayo de Bernoulli**. Un experimento aleatorio en el que sólo se pueden obtener dos resultados, habitualmente etiquetados como éxito (E) y fracaso (F). Desde el punto de vista de la teoría de la probabilidad, estos ensayos están modelados por una variable aleatoria que puede tomar sólo dos valores, 0 y 1. Habitualmente, se utiliza el 1 para representar el éxito. [2]

El espacio muestral del proceso está formado por cada uno de los caminos del árbol de probabilidades. La probabilidad de un camino, por ejemplo, del camino EFE es:

$$p(EFE) = p * q * p$$

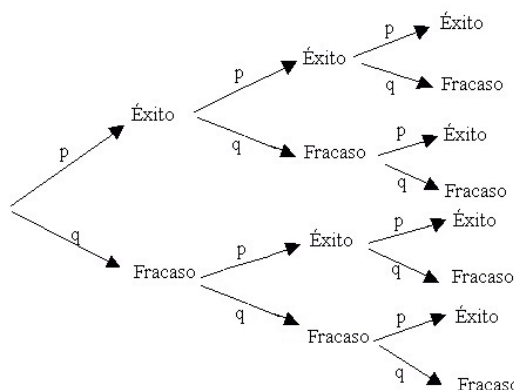


Figura 3.1: Árbol de probabilidades de un ensayo de Bernoulli de tres experimentos

En nuestro caso, los valores no estarán ligados a un éxito o a un fracaso, sino a una cara o a una cruz.

3.1.1. ¿Cómo se agrupan los lanzamientos de una moneda?

Por lo tanto y según lo expuesto en el apartado anterior, cada lanzamiento de moneda va a constituir un ensayo de Bernoulli.

En teoría de la probabilidad y estadística, la distribución de probabilidad de una variable aleatoria es una función que asigna a cada suceso definido sobre la variable aleatoria (cara o cruz) la probabilidad de que dicho suceso ocurra. La distribución de probabilidad está definida sobre el conjunto de todos los eventos rango de valores de la variable aleatoria, en nuestro caso tendremos dos valores, cara o cruz.

La forma que tendrá, por ejemplo el lanzamiento de una moneda justa y equilibrada (Sesgo=50) tendrá la forma:

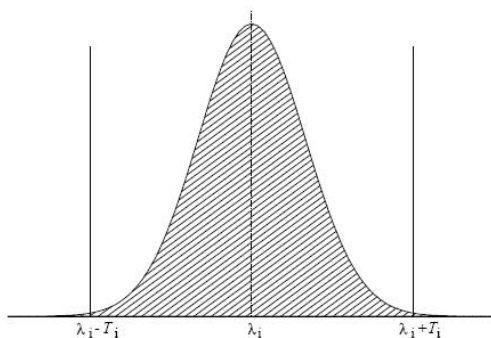


Figura 3.2: Representación de la distribución binomial

Esta distribución es de la forma binomial, la cual es una distribución de probabilidad de una variable aleatoria discreta ampliamente utilizada, así como de las más importantes.

En una distribución de probabilidad discreta, la variable aleatoria asigna un valor numérico a cada resultado en el espacio muestral del experimento.

La distribución binomial tiene que ver con una clase especial de experimento llamado experimento binomial [3].

Para calcular cuál es la probabilidad de obtener un determinado número de caras o cruces al tirar una moneda, está la formula:

$$b(x; n, \theta) = \binom{n}{x} \theta^x (1 - \theta)^{(n-x)}$$

Donde n es el número de veces que hacemos el experimento (lanzar una moneda en este caso), θ es la probabilidad de que salga lo que queremos (en nuestro será caso el sesgo elegido) y x es el número de caras que buscamos.

3.2. ¿Cómo vamos a lanzar nuestra moneda?

Evidentemente la forma como la que lancemos la moneda estará directamente relacionada con el sesgo que presente la misma[14], en este punto vamos a plantear dos formas posibles lanzando 100 veces una moneda para observar la dependencia del sesgo:

- Lanzar la moneda al aire haciendo que gire sobre ella misma

Colocando la moneda en la parte superior del dedo pulgar e impulsando la moneda al aire, haciendo que gire ganando altura para luego caer en la palma de la mano.

De los 100 lanzamientos podemos ver que se aproxima a lo que sería una moneda justa concentrándose el mayor número de caras en el sesgo 50, teniendo así 50 % de posibilidades cara y 50 % de posibilidades de cruz.

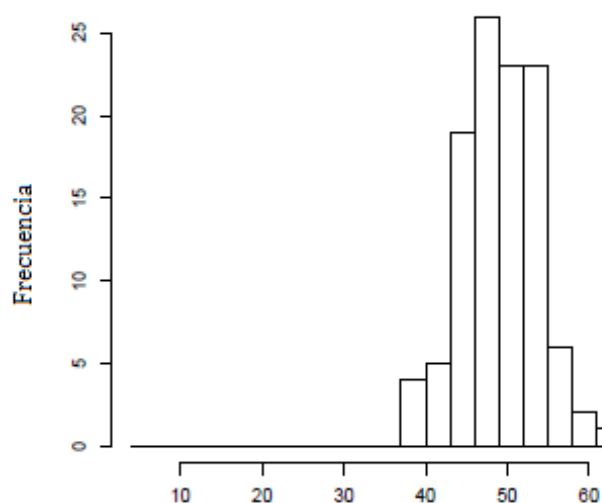


Figura 3.3: Histograma del lanzamiento de una moneda al aire 100 veces

- Hacerla girar sobre su canto en una mesa

Colocando la moneda en una superficie lisa, apoyada sobre su borde mientras es sostenida con la mano derecha, para con la mano izquierda impulsarla para que gire sobre su propio borde para finalmente caer sobre la mesa.

De los 100 lanzamientos podemos ver una enorme desviación del sesgo hacia las cruces.

Si una moneda es lanzada al aire y en vez de recogerla en el aire con la mano, como fue expuesto en el punto 1 de este apartado, la mayoría de las veces caerá con su canto sobre el suelo, para girar y finalmente caer hacia un lado. De esta forma, estaríamos consiguiendo una enorme desviación.

Aún así, la mayoría de la gente prefiere un lanzamiento de esta última forma ya que desconoce los factores anteriormente expuestos, y pese a ser

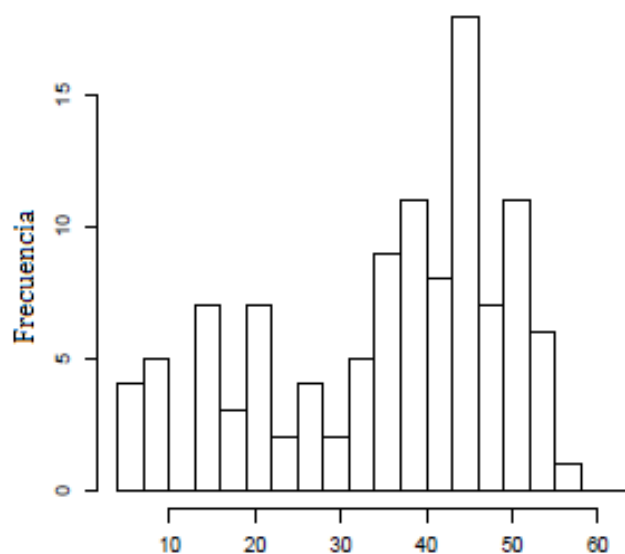


Figura 3.4: Histograma de una moneda girando sobre su eje 100 veces

menos equilibrado y justo que uno que recoge la moneda en el aire, tal vez se decanten por esta opción por que el factor humano no intervine en la intercepción de la moneda, pudiendo así por ejemplo manipular la moneda una vez en la palma de su mano.

Otro factor importante sobre el lanzamiento de una moneda, demostrado por Murray y Teare es que un penique americano, sobre 6000 lanzamientos al menos una vez acabará sobre su borde, sin decantarse por cara o cruz.[20]

Para nuestro experimento, cuando hablemos de una moneda justa y equilibrada, estaremos hablando exactamente de una proporción 50%-50% generada por números pseudoaleatorios, sin tener en cuenta la desviación de Keller[8] ni la causalidad de Murray y Teare [20].

3.3. Parámetros empleados en la experimentación

En esta sección expondremos las variables planteadas para su análisis y estudio. Dichas variables serán observadas en baterías de experimentos de 99 lanzamientos, puesto que la aplicación CoinToss contabiliza distancias hasta 99, haremos experimentos de dicha longitud para que sea no pueda darse una distancia mayor a la que podamos contabilizar.

- Porcentaje de caras
- Distancia máxima de caras
- Distancias individuales

La motivación del uso de distancias viene dada por la necesidad de añadir credibilidad al sesgo estimado teniendo en cuenta únicamente la variable del porcentaje de apariciones, debido a que en los resultados obtenidos a partir de la experimentación se observa que hay una zona de intersección que genera cierta indeterminación al poder pertenecer ésta a varios sesgos (Véase figura 4.1). Las distancias serán usadas como fuentes de información a la hora de decantarnos por un comportamiento.

Porcentaje de apariciones de caras

Se refiere al número de veces en % que ha aparecido cara con respecto a la longitud total del experimento sin tener en cuenta el orden de sus apariciones.

Es la variable más representativa por que cuando el número de tiradas tiende a infinito el valor teórico de la variable coincide con el sesgo.

Distancias[1]

El concepto de distancia se refiere al número de apariciones consecutivas de un valor (en el caso de la moneda, cara o cruz).

$$distancia = numero_apariciones_consecutivas_valor - 1$$

Distancia máxima de caras[1]

Esta variable contabiliza cual ha sido la distancia máxima que haya aparecido durante un experimento.

Distancias individuales[1]

Esta última variable engloba una subvariable por cada distancia observada, siendo en realidad 100 variables. Indica para una determinada distancia y un determinado sesgo cuántas veces ha aparecido y en cuántos experimentos.

3.4. Objetivos de la investigación

Los objetivos principales que se esperan cumplir como resultado del proceso de experimentación y observación son los siguientes:

- Elaborar una amplia experimentación en la que podamos observar el comportamiento de las variables de estudio anteriormente expuestas.
- Determinar si las distancias[1] aportan credibilidad al sesgo estimado a partir del porcentaje de apariciones.
- Comparar las distintas subvariables extraídas a partir de las distancias, y determinar cual de ellas es la que más información útil aporta.

- Determinar el sesgo de la moneda a partir de un experimento de 99 tiradas con suficiente credibilidad.

Otros objetivos relacionados con el proceso de investigación son los siguientes:

- Desarrollar el software necesario para generar y almacenar la experimentación.
- Desarrollar el software que permita observar gráficamente las variables de estudio en dicha experimentación.

3.5. Desarrollo del proyecto

3.5.1. Diagrama de clases

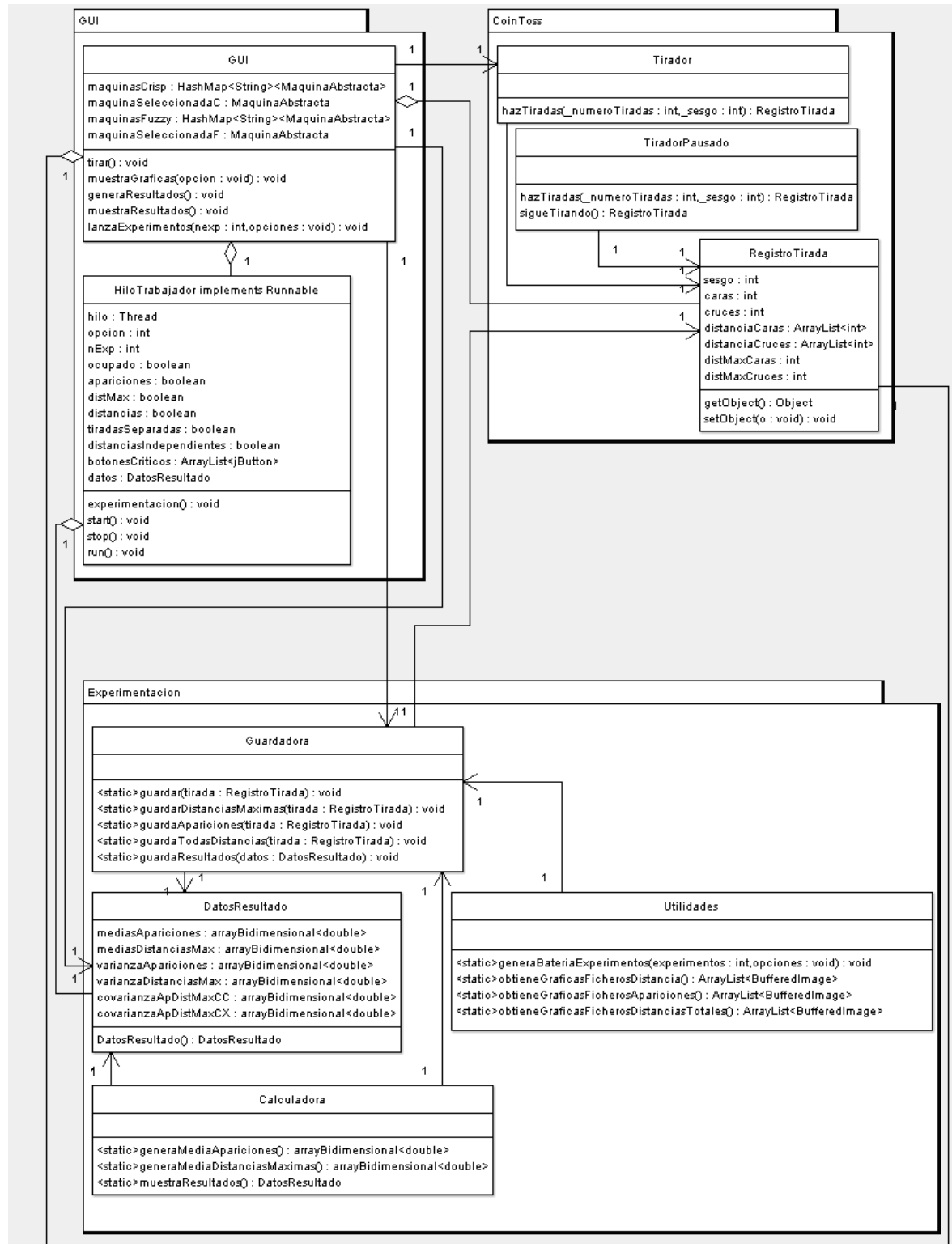


Figura 3.5: Diagrama de clases parte 1

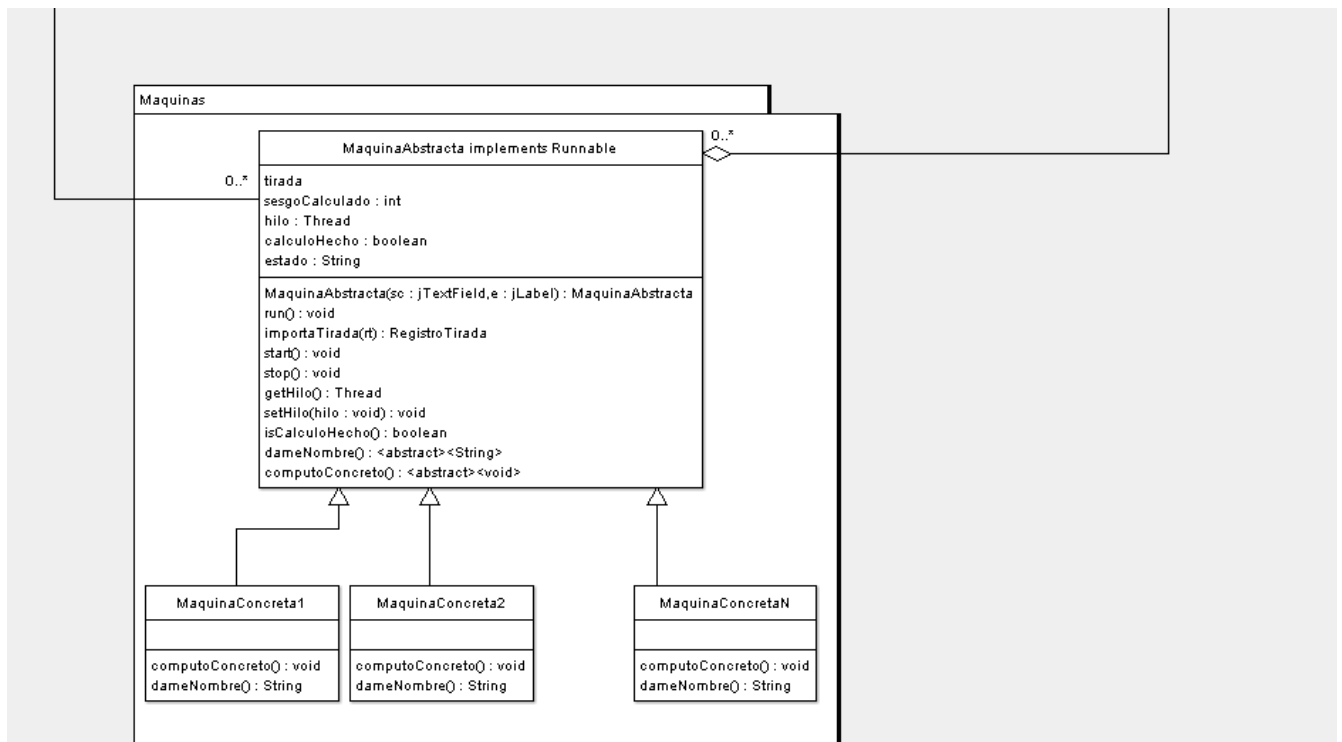


Figura 3.6: Diagrama de clases parte 2

3.5.2. Estructura: Clases y paquetes

Paquete GUI

Es el paquete donde se encuentra la GUI principal, con opción a realizar experimentos individuales escogiendo los parámetros, y lanzando por cada experimento todas las maquinas, fuzzy y crisp, cada una de ellas en un hilo independiente. Permite la opción de lanzar un hilo que realice baterías de experimentos, con diferentes opciones, guardándolos en ficheros. También tiene la opción de obtener gráficas mediante los ficheros de resultados. Además de la opción de lanzar un hilo que calcule las varianzas y covarianzas de las apariciones de caras y cruces y las distancias máximas de caras y cruces, con el fin de obtener su índice de correlación, medido por la formula de la correlación de Pearson (Véase apartado 2.5).

Paquete CoinToss

Este paquete contiene las clases encargadas de realizar y almacenar los experimentos de forma individual, con el número de tiradas y el sesgo que se desee. Éste paquete es una adaptación a nuestro proyecto de la aplicación CoinToss facilitada por el profesor D. Francisco Javier Crespo Yáñez (Ver Javadoc).

Paquete Maquinas

Este paquete contiene la clase MaquinaAbstracta, de la que todas las maquinas que vayamos a construir deben heredar. Implementa la interfaz *Runnable*, haciendo concretos todos los métodos de la interfaz. Se encarga

además de actualizar las partes de la interfaz concerniente al estado de la maquina y a su resultado. Se encarga también de importar la tirada actual. El único método abstracto de `MaquinaAbstracta` es `void computoConcreto()`

Paquete Maquinas.Fuzzy

Contiene todo lo necesario (motor de inferencia y tipos de datos especiales) para la implementación de las maquinas fuzzy utilizadas.

Paquete Experimentacion

Este paquete se encarga de el tratamiento masivo de experimentos usando ficheros. Genera baterías de experimentos de distintos sesgos y longitud de las tiradas de 99. Lee experimentos previos de ficheros y añade nuevos experimentos, genera gráficas de ficheros de experimentación y obtiene el índice de correlación de Pearson a partir de ficheros de experimentación.

3.6. Máquinas desarrolladas

En base al comportamiento de los resultados de las variables anteriormente expuestas, se van a hacer máquinas que intenten estimar y prever ese comportamiento en base a unas reglas previamente definidas en función de las lógicas *crisp* y *fuzzy*.

3.6.1. Máquinas *fuzzy*

A continuación vamos a exponer las máquinas Fuzzy construidas a partir de la observación sobre la experimentación de la variable de las apariciones

de caras y cruces.

Ambas maquinas utilizan el mismo motor de inferencia de código abierto proporcionado por www.mygnet.net[27]. Se diferencian en su función de "fuzzyficación".

Máquina triangular

La función de *fuzzyficación* para los conjuntos Fuzzy tiene forma triangular. Tiene once conjuntos Fuzzy, correspondientes a la división de los sesgos $[0, 100]$ en conjuntos de 10. Esta función está centrada en el porcentaje de caras igual al sesgo de ese conjunto. Cada conjunto fuzzy tiene la siguiente forma:

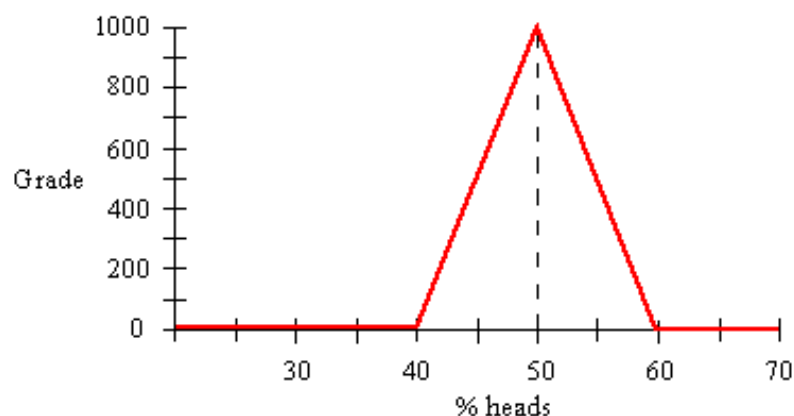


Figura 3.7: Valores de la funcion triangular de pertenencia fuzzy al conjunto fuzzy de sesgo 50

Máquina trapezoidal

La función de *fuzzyficación* para los conjuntos Fuzzy tiene forma trapezoidal. Tiene once conjuntos Fuzzy, correspondientes a la división de los sesgos $[0, 100]$ en conjuntos de 10. Esta función está centrada en el porcentaje de

caras igual al sesgo de ese conjunto. Cada conjunto fuzzy tiene la siguiente forma:

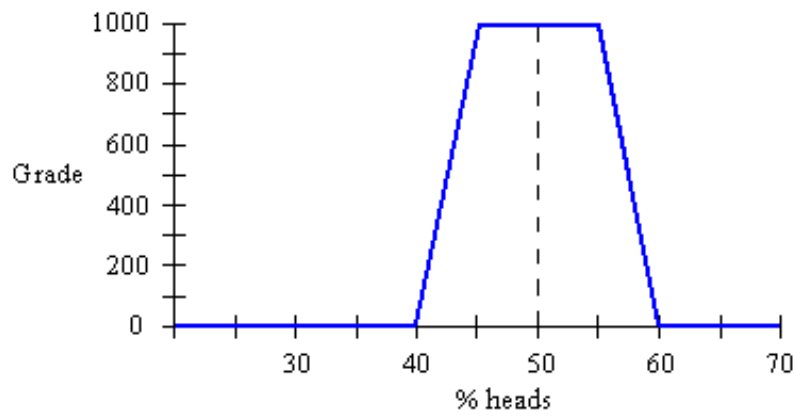


Figura 3.8: Valores de la función trapezoidal de pertenencia fuzzy al conjunto fuzzy de sesgo 50

Capítulo 4

Experimentación

*” A veces al final de un largo camino
puedes encontrarte otro camino más largo.”*

En esta sección vamos a hablar del desarrollo de nuestra aplicación, así como de la experimentación que hemos realizado. Indicar que en números totales, nuestra experimentación consta de un un total de 500.000 experimentos por cada sesgo, recordemos que el espectro de desviaciones lo hemos dividido en once sesgos, lo que lleva a un total de 16.500.000 experimentos analizados.

A parte, cada uno de estos experimentos constará de 99 lanzamientos de moneda con su respectivo sesgo.

La sección de exposición de resultados estará dividida en los análisis de caras, para posteriormente centrarnos en los diferentes tipos de distancias analizadas.

4.1. Desarrollo de la aplicación

La herramienta con la que iniciamos nuestra investigación, fue la aplicación CoinToss proporcionada por el profesor D. Francisco Javier Crespo Yáñez, que nos permite simular el lanzamiento de una moneda con el sesgo escogido y el número de tiradas que se desea lanzar.

Para nuestra investigación hemos desarrollado una aplicación multi-hilo que nos permite realizar un experimento con sesgo y número de tiradas deseadas, para que luego, cada una de esas máquinas que hayamos implementado, dado esa tirada estimen un sesgo, pudiendo ver qué sesgo ha calculado cada máquina, para así poder comprobar su fiabilidad.

Hemos decidido que el lenguaje de programación sea Java, por ser el lenguaje con el que más familiarizados estamos, y además, tiene hilos ya implementados en la clase *Thread*, que usamos en nuestra aplicación. El entorno utilizado es NetBeans 6.5 o superior, para versión de Java 1.6 o superior.

La decisión de que cada máquina se ejecutase en un hilo propio fue tomada en base a que de otra manera, la aplicación se bloquearía hasta que cada una de las máquinas estimase el sesgo. Además podemos consultar el resultado de las máquinas que primero hayan acabado, y su estado, o solo su estado en el caso que la máquina estuviese todavía estimando el sesgo. Añadimos máquinas de prueba para comprobar la escalabilidad del sistema. Las nuevas máquinas añadidas son *MaquinaRandom*, *MaquinaReglasSimples* y *MaquinaBinomialSencilla*.

Llegados a este punto ya tenemos una aplicación que nos permite hacer tiradas con datos a elegir, y diferentes máquinas que se lanzarán al hacer la

tirada y estimar el sesgo. Pero necesitamos experimentación para observar los datos obtenidos y creamos el paquete Experimentacion.

Decidimos que los experimentos sean de 99 tiradas, y tantas repeticiones del experimento por sesgo (sesgos de 0 a 100 de 10 en 10). En un principio guardamos sólo los ficheros de información de las variables a observar, y no cada experimento por separado, puesto que para 500.000 experimentos, el fichero tiradasS.txt, siendo S el sesgo, ocupa casi 700 Mb.

Luego nos dimos cuenta que aunque para generar las gráficas baste con los otros archivos, para calcular la correlación entre las diferentes variables, necesitamos todos los experimentos por separado, lo que nos obliga a repetir el experimento, esta vez generando los ficheros tiradasS.txt.

Comprobamos que generar experimentos es una operación costosa que bloquea la aplicación, así que le asignamos un hilo que use la clase Utilidades para generar baterías de experimentos, además, el hilo deshabilitará todos los botones que le requieran para trabajo, a fin de no lanzar varios hilos de escritura y lectura sobre los mismos ficheros.

Para mayor información consultar Javadoc y manual de la aplicación (Véase Anexo 1).

4.2. Exposición de los resultados obtenidos

4.2.1. Análisis de las apariciones de caras

Evolución del número de apariciones de caras

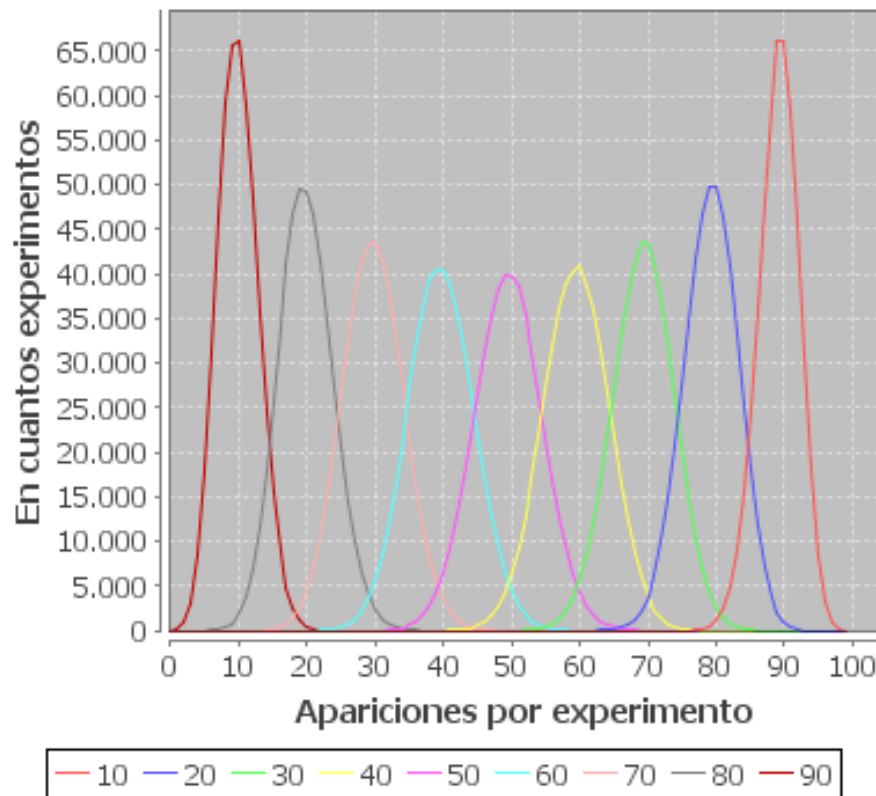


Figura 4.1: Gráfica de las apariciones de caras por sesgo

En la gráfica podemos observar un comportamiento en las apariciones de las caras que permite diferenciar la variación de la aparición de las caras dependiendo del sesgo. Podemos ver que las apariciones por cada sesgo, siguen una distribución binomial (Véase Sección 3.2.1).

Se han realizado 500000 experimentos por cada sesgo y en base a estos

resultados hemos implementado nuestras máquinas *fuzzy* (Véase Sección 3.6).

El problema se encuentra en la zona de intersección de las curvas, donde es tan posible el hecho de que la moneda se comporte siguiendo el patrón de un sesgo u otro. Para aumentar la credibilidad de la estimación en estas zonas de intersección, a la hora de tratar de obtener el sesgo, vamos a tratar de encontrar otras variables a analizar, comprobando previamente que la variable proporciona nueva información útil respecto a esta variable.

4.2.2. Análisis de las distintas distancias

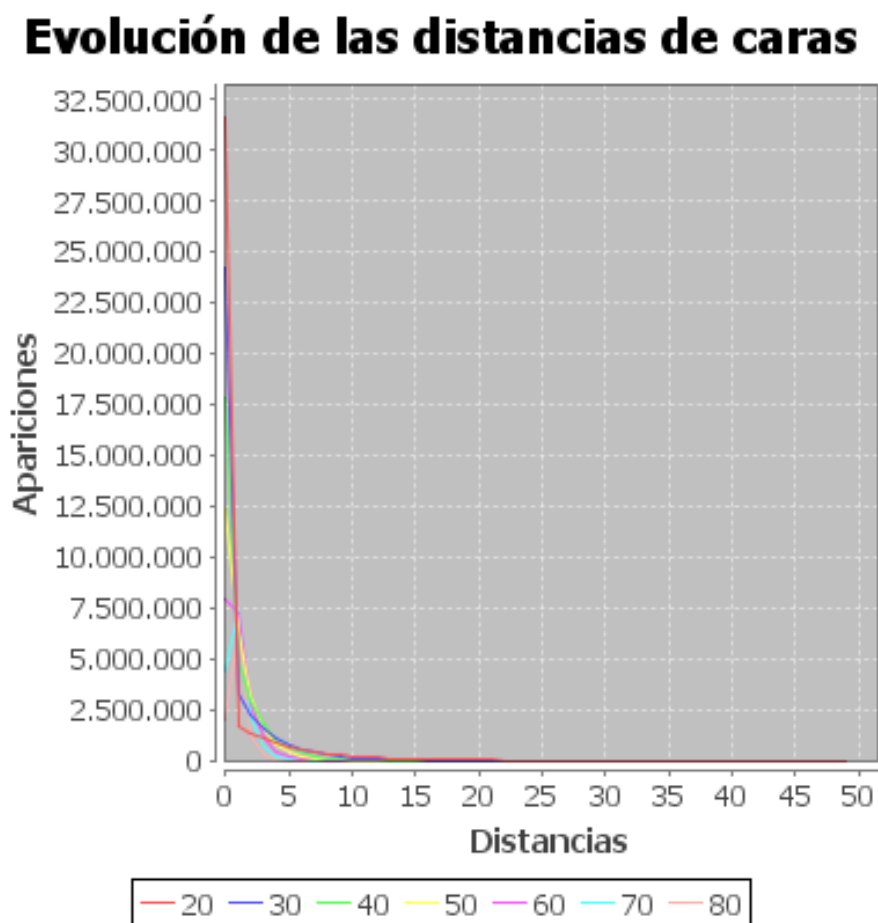


Figura 4.2: Grafica de las distancias acumuladas de caras por sesgo

En la gráfica obtenida a partir de los ficheros Sdist.txt donde S es el sesgo (ver Anexo 2), muestra la información acerca de cuántas veces ha aparecido cada distancia en todos los experimentos.

Al ser las distancias acumuladas esta gráfica no está pensada para el análisis de nuevas variables, sino para ver un comportamiento global de las distancias en todos los experimentos.

La primera variable de las distancias a analizar, son las distintas individuales 0, 1 y 2 por separado viendo el número de veces que aparece esa distancia por experimento. A continuación mostramos las gráficas de los resultados obtenidos de la experimentación realizando 500.000 experimentos por cada sesgo y su tabla de correlaciones respecto a la variable apariciones.

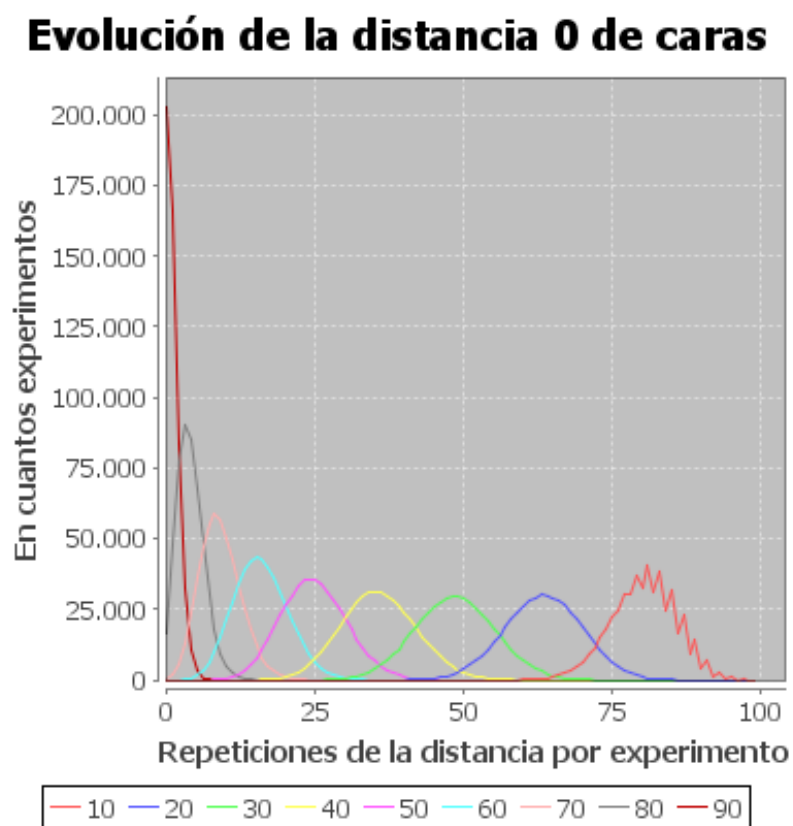


Figura 4.3: Apariciones de la distancia 0 por experimento

Sesgo	Correlación
0	1'0000
10	0'9837
20	0'9673
30	0'9472
40	0'9223
50	0'8904
60	0'8488
70	0'7897
80	0'7037
90	0'5493
100	1'0000

Cuadro 4.1: Tabla de correlaciones para las apariciones y la distancia 0 por sesgo obtenidas

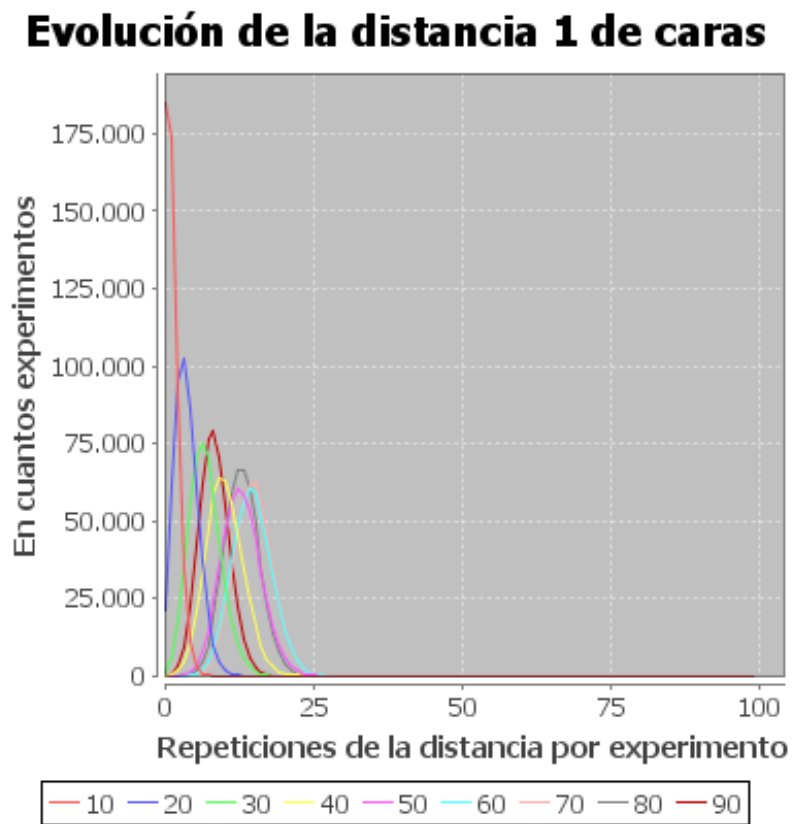


Figura 4.4: Apariciones de la distancia 1 por experimento

Sesgo	Correlación
0	1'0000
10	-0'5016
20	-0'5761
30	-0'5663
40	-0'4949
50	-0'3668
60	-0'1665
70	0'1101
80	0'4398
90	0'7565
100	1'0000

Cuadro 4.2: Tabla de correlaciones para las apariciones y la distancia 1 por sesgo obtenidas

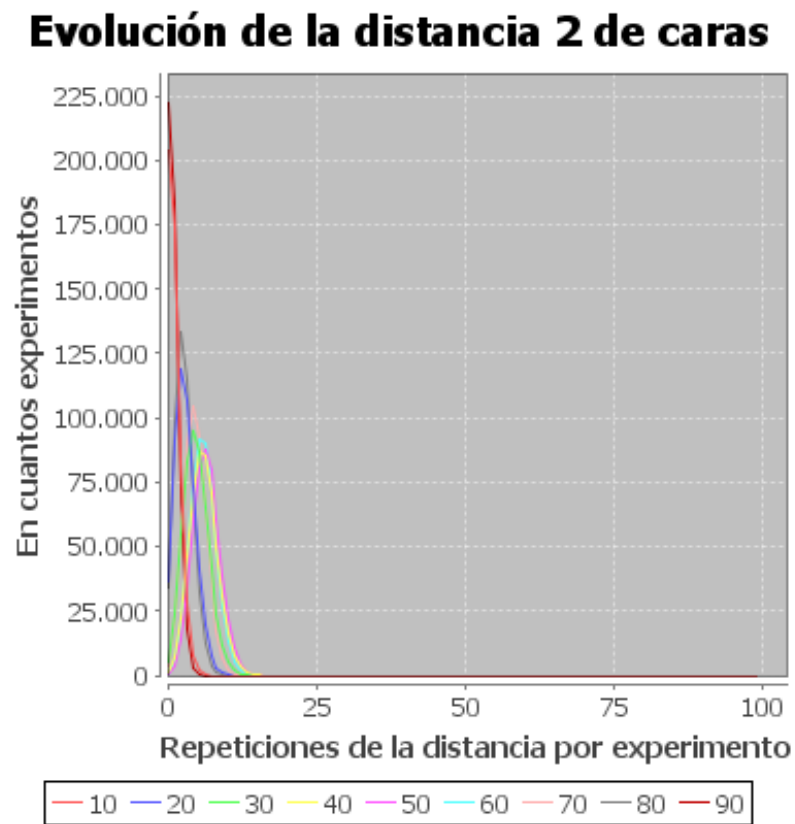


Figura 4.5: Apariciones de la distancia 2 por experimento

Sesgo	Correlación
0	1'0000
10	-0'4512
20	-0'4533
30	-0'3563
40	-0'1968
50	0'0120
60	0'2300
70	0'4072
80	0'5116
90	0'4873
100	1'0000

Cuadro 4.3: Tabla de correlaciones para las apariciones y la distancia 2 por sesgo obtenidas

Las gráficas generadas de esta manera son 100, una para cada distancia observada, aunque únicamente hemos incluido las tres primeras porque tal como se puede observar, a medida que aumenta la distancia que estamos observando, los valores convergen a 0 o 1. Esto es debido a que cuanto mayor es la distancia observada, menos veces puede aparecer por tirada, porque como es evidente, en un experimento de 9 tiradas es posible encontrar, como máximo cinco distancias 0 de caras, pero una sola distancia 5.

En búsqueda de una variable para el análisis basada en las distancias, hemos tomado las distancias máximas obtenidas por cada experimento. Las gráficas muestran el resultado de las distancias máximas por experimento, para 500.000 experimentos, almacenados en los ficheros S.txt (Véase Anexo 2) donde S es el sesgo.

Evolución de las distancias máximas de caras

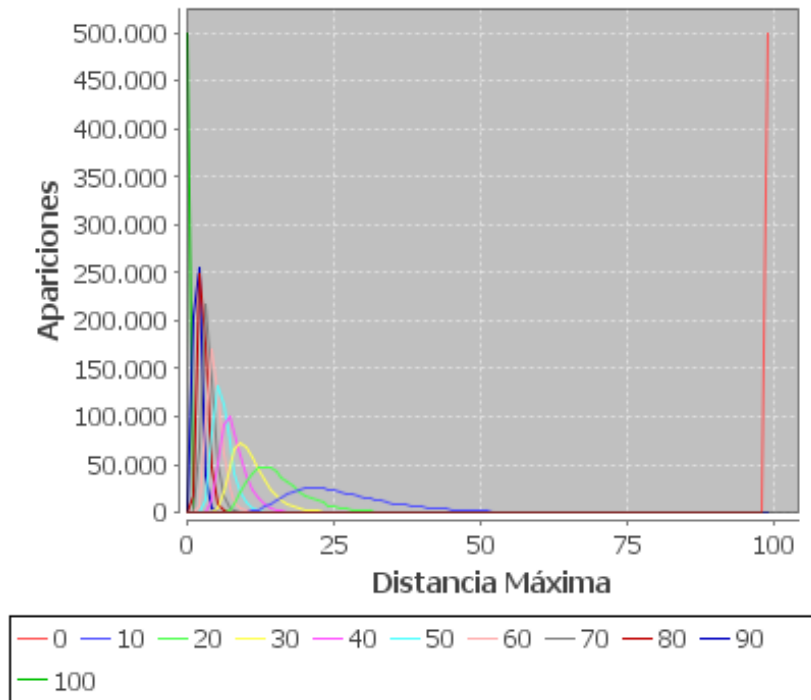


Figura 4.6: Grafica de las distancias maximas de caras por sesgo

Sesgo	Correlación
0	1'0000
10	0'5989
20	0'5286
30	0'4868
40	0'4627
50	0'4398
60	0'4287
70	0'4251
80	0'4223
90	0'4523
100	1'0000

Cuadro 4.4: Tabla de correlaciones para las apariciones y las distancias máximas por sesgo obtenidas

Capítulo 5

Discusión

"Si bien buscas, encontrarás"

Platón

A partir de la experimentación obtenida y almacenada en ficheros, hemos obtenido los diferentes índices de correlación de Pearson (Véase Sección 2.5) entre las variables estadísticas:

- Las apariciones de caras y la distancia máxima de caras para cada sesgo (Véase Cuadro 4.4).
- Las apariciones de caras y las distancias 0 (Véase Cuadro 4.1).
- Las apariciones de caras y las distancias 1 (Véase Cuadro 4.2).
- Las apariciones de caras y las distancias 2 (Véase Cuadro 4.3).

5.0.3. Distancias máximas

Como vemos, para los sesgos 0 y 100 obtenemos un índice de correlación 1. Esto quiere decir que proporcionan la misma información. Ya que cuando en una tirada de longitud 99, tenemos un sesgo 0, las apariciones y la distancia máxima de caras será la misma, 99, al igual que con sesgo 100, la distancia máxima y las apariciones de caras serán 0.

Viendo la gráfica de las distancias máximas, observamos que para cada sesgo el comportamiento de las distancias máximas siguen distintas distribuciones, al contrario de la gráfica de las apariciones que sigue una distribución binomial. Esto conlleva a que la correlación entre las apariciones de caras y las distancias máximas de caras varíe según el sesgo.

Aún así, podemos observar que hay una correlación entre ambas variables, que oscila dependiendo del sesgo, en el intervalo $[0'40 - 0'60]$ por lo que deducimos que existe una relación entre las variables, y por lo tanto podrá ser usada para aportar nueva información a la hora de estimar la pertenencia a un sesgo.

5.0.4. Distancias 0, 1 y 2

Al igual que con las distancias máximas, para los sesgos 0 y 100 las distancias 0, 1 y 2 tienen un índice de correlación de 1, puesto que sus valores no varían a lo largo de los experimentos.

Distancia 0

Viendo la gráfica de la distancia 0 podemos observar un comportamiento diferenciado por cada sesgo que parece seguir una distribución binomial.

Como podemos encontrar un comportamiento diferenciado por cada sesgo, podría emplearse una función de *fuzzyficación* que tuviese en cuenta el número de veces que aparece la distancia 0 por experimento a la hora de determinar el comportamiento de la moneda.

Al obtener el coeficiente de correlación entre el porcentaje de caras y la distancia 0, podemos observar que los coeficientes obtenidos oscilan dependiendo del sesgo, en el intervalo $[0'54 - 0'99]$, con la mayoría de los datos cercanos a 0'90. Esto nos indica que existe una alta correlación entre ambas variables, por lo que aporta información redundante.

Distancia 1 y 2

Viendo ambas gráficas podemos observar que su comportamiento es el que sigue una distribución binomial, salvo que a diferencia de la anteriormente expuesta distancia 0, existe una mayor superposición. Además, a medida que vaya aumentando la distancia que tomemos como variable, la superposición será mayor, por lo que no se podría estimar un comportamiento diferenciado para cada sesgo.

A medida que vamos tomando como variables distancias mayores, se puede observar en las tablas de correlación cómo los coeficientes son cada vez más cercanos a 0, por lo que entre ambas variables no existe relación lineal. Pero esto no necesariamente implica que las variables sean independientes:

pueden existir todavía relaciones no lineales entre las dos variables.

Pero dado que aumentando la distancia observada la superposición es mayor, aunque la información no sea redundante, para su uso habría que tener en cuenta otros factores o combinarlo con el uso de otras distancias.

Capítulo 6

Conclusión y líneas futuras

*”El futuro pertenece a quienes
creen en la belleza de sus sueños.”*

Eleanor Roosevelt

La motivación del estudio realizado a la par de su correspondiente experimentación, ha sido hecha en base a la búsqueda de un posible comportamiento en el lanzamiento simulado de una moneda a través de la generación pseudoaleatoria de numeros, y dependiente de las variaciones en el sesgo.

Para ello, se ha determinado a través de las variables y subvariables empleadas, que para determinar la pertenencia a un determinado sesgo, a parte de la estimación ofrecida por el porcentaje de apariciones de caras, que sería la forma más precisa, como forma de acotación a estos conjuntos de sesgos, la variable distancias máximas puede ofrecer una información menos redundante que la que nos pudieran ofrecer el resto de subvariables de distancias, que

como vimos en el apartado Experimentación (Véase sección 4) tienen una gran superposición.

Sobre las variables de apariciones y distancias máximas, se ha trazado una correlación, mediante la fórmula de la correlación de Pearson (Véase Sección 2.3), que ha determinado que efectivamente hay una correlación entre ambas variables, que oscila dependiendo del sesgo, más concretamente entre el rango de valores $[0'40 - 0'60]$ por lo que deducimos que existe una relación entre las variables, y por lo tanto podrá ser usada para aportar nueva información útil a la hora de estimar la pertenencia a un sesgo.

Como líneas de desarrollo futuras, sería posible a partir de nuestra experimentación hacer una suma de credibilidades de forma nítida y borrosa para tratar de determinar si esa suma permite hallar el sesgo inicial.

Además, indicar que nuestro sistema permite añadir nuevas máquinas a nuestro sistema de forma sencilla. Primero hemos de crear una máquina que herede de `MaquinaAbstracta` en el paquete `Maquinas`. La máquina que creamos puede usar todos los paquetes que sean necesario, siempre y cuando sean declarados como subpaquetes de máquinas. Hemos de declarar su constructora, que recibe un `jLabel` y un `jTextField`, que le pasa a la constructora de la superclase.

La nueva máquina que se implemente, ha de implementar los métodos `dameNombre()` y `computoConcreto()`. `computoConcreto()` es el cuerpo del hilo, y será donde se estime el sesgo, según el tipo de máquina que hallamos decidido implementar. Al final de `computoConcreto()`, el sesgo estimado, llamémosle "e", tendrá que guardarse con el método `setSesgoCalculado(e)`. El formato del método sería el siguiente:

```
public void computoConcreto(){
int sesg; //Variable donde guardaremos el sesgo
... //Calculamos el sesgo y lo guardamos en sesg
setSesgoCalculado(sesg); //Actualizamos la variable
//de la superclase para que la aplicación pueda leerla
}
```

Una vez implementada nuestra nueva máquinas, iremos a la clase GUI.java, e incluiremos nuestra nueva máquina en el método incluyeMaquinas(), supongamos que queremos añadir la máquina fuzzy llamada mf, dentro de ese método añadiríamos la siguiente sentencia:

```
private void incluyeMaquinas(){
//CRISP
maquinasCrisp = new HashMap<String, MaquinaAbstracta>();
...//incluimos las máquinas crisp
maquinaSeleccionadaC = maux;
//FUZZY
maquinasFuzzy = new HashMap<String, MaquinaAbstracta>();
...//incluimos las máquinas
//al final de todas las inclusiones añadimos la nueva máquina
maux = new mf(this.jTextSesgoC1, this.jLabelEstado1);
maquinasFuzzy.put("mf", maux);
...
}
```

```
maquinaSeleccionadaF = maux;  
}
```

Además, en el `jList` de máquinas fuzzy habrá que añadir "mf". Es importante que el nombre que aparezca en el `jList` sea el mismo que la clave utilizada en el `HashMap`, porque ese nombre será el que se use para acceder al `HashMap`. El procedimiento para añadir una máquina crisp es análogo al anterior.

En el caso de encontrar una nueva variable de investigación, habría primero que crear en guardadora el método que dado un `RegistroTirada` leyese del fichero, añadiese la nueva información y sobrescribiese dicho fichero. Una vez tenemos el método estático para actualizar ficheros con una tirada, en el método `generaBateriaExperimentos(...)` de la clase `Utilidades`, añadimos un nuevo parámetro `bool` con el que decidimos si deseamos guardar.

```
if (guardarNuevaVariable)  
Guardadora.nuevoMetodo(rt);
```

Por último añadiríamos en `utilidades` los métodos de generación de gráficas para esos ficheros, y en `Calculadora` los métodos de cálculo de varianza y covarianza para las nuevas variables observadas.

Una vez hecho esto, solo sería necesario añadir un nuevo `checkbox` en `jExperimentar` para guardar esos datos, En `jGraficas` añadir una línea con la nueva opción, que llame al nuevo `generarGraficas` de `Utilidades`, y crear una nueva ventana de resultados para las nuevas variables observadas.

Capítulo 7

Apéndices y Anexos

Anexo 1: Guía de uso de la aplicación

CoinPlus es una aplicación que principalmente fue diseñada para simular lanzamientos de monedas mediante números pseudoaleatorios, y para que permitiese observar los datos de la tirada. Una vez con esos datos, diversas máquinas implementadas estimaran el sesgo.

Junto con la memoria se adjunta una carpeta de proyecto NetBeans y una carpeta con el archivo ejecutar (.jar)

La interfaz principal es:

CoinPlus 2010 v0.3.1
 Archivo Experimentación

Sesgo real Número de tiradas Tirada Paso
 Una en una De n en n Sin Parada

Guardar Tirada

Español English

Resultado Caras Cruces %
 Diferencia caras-cruces: %

Distancia máxima: Caras Cruces %
 Cantidades: Caras Cruces
 Diferencia: %

Distancia media: Caras Cruces %
 Cantidades: Caras Cruces
 Diferencia: %

Distancia 0: Caras Cruces %
 Diferencia: %

CRISP FUZZY

Random
 Reglas simples
 Binomial sencilla

Maquina: Nombre Máquina
 Sesgo calculado:
 Estado: Ocupado

Figura 7.1: Interfaz principal de la aplicación

Desde esta ventana podemos realizar experimentos individuales, y guardarlos en ficheros, rellenar los campos sesgo, con el sesgo que se desea lanzar la moneda, y en Número de tiradas, las veces que se desea tirar la moneda, va en el siguiente campo:



The screenshot shows a software interface for coin toss experiments. It includes two input fields: 'Sesgo real' with the value '50' and 'Número de tiradas' with the value '99'. There are three buttons: 'Tirada' (highlighted in blue), 'Paso', and 'Guardar'. To the right, there are three radio buttons: 'Una en una', 'De n en n', and 'Sin Parada' (which is selected). There are also two empty input fields labeled 'n' and 'Tirada'. At the bottom, there is a large text area containing a long sequence of 'C' and 'X' characters, representing the results of the tosses.

Figura 7.2: Menu de lanzamiento

El checkbox habilita el guardado, pero se recomienda guardar solo experimentos de tamaño 99, puesto que son las tiradas con las que se generan las baterías de experimentos. Los radioButton permiten escoger si queremos hacer tiradas paso a paso. Las tiradas paso a paso pueden servir para ver la evolución de las distancias y el porcentaje de caras, pero hay que tener en cuenta que las tiradas paso a paso no se guardan, ni son evaluadas por las máquinas.

Aquí es donde se muestra la información relativa al número de caras:

Resultado			
Caras	<input type="text" value="48"/>	Cruces	<input type="text" value="51"/>
	<input type="text" value="48.484848"/>	%	<input type="text" value="51.515152"/>
		%	
Diferencia caras-cruces:			
	<input type="text" value="3"/>		<input type="text" value="3.030303"/>
		%	

Figura 7.3: Resultado de apariciones

Distancias máximas y distancias medias obtenidas, junto cuantas veces se han dado esas distancias

Distancia máxima:				Distancia media:			
Caras:	<input type="text" value="5"/>	Cruces:	<input type="text" value="5"/>	Caras:	<input type="text" value="2"/>	Cruces:	<input type="text" value="1"/>
	<input type="text" value="050505"/>	%	<input type="text" value="5.050505"/>	%	<input type="text" value="020202"/>	%	<input type="text" value="1.010101"/>
		%				%	
Cantidades:				Cantidades:			
Caras:	<input type="text" value="2"/>	Cruces:	<input type="text" value="1"/>	Caras:	<input type="text"/>	Cruces:	<input type="text"/>
Diferencia:	<input type="text"/>		<input type="text"/>	%	<input type="text"/>		<input type="text"/>
		%				%	

Figura 7.4: Distancias maxima y media

Además, puedes ver cada una de las distancias por separado, para ver cuantas veces se repite esa distancia en la tirada. Puedes cambiar de una distancia a otra pulsando los botones con flechas. Hw aquí un ejemplo para la tirada anterior, de las distancias 0 y 3:

Distancia 0:

Caras: Cruces:

% %

< **Diferencia:** >

%

Figura 7.5: Distancia individual 0

Distancia 3:

Caras: Cruces:

% %

< **Diferencia:** >

%

Figura 7.6: Distancia individual 3

El último apartado de la pantalla principal, es el de las máquinas. Tiene la siguiente forma:

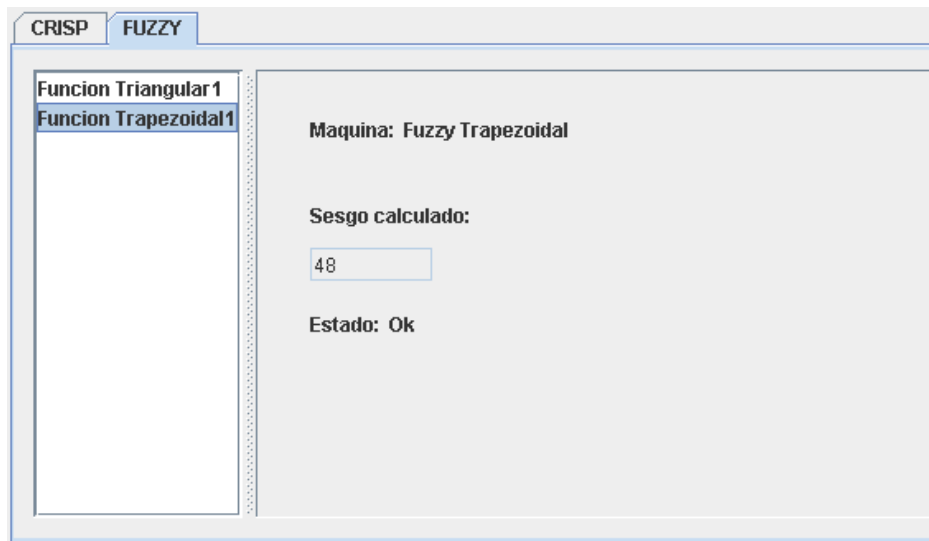


Figura 7.7: Vista de menu de maquinas

La pestaña superior, nos permite alternar entre las máquinas crisp y las máquinas fuzzy, cada cual con su propia lista. Seleccionando una máquina de la lista, a su derecha podrás ver su nombre, el sesgo que ha tirado para la tirada anterior (si la hubiese) y el estado del hilo. Todas las máquinas estimarán un sesgo cuando se realice una tirada sin parada.

Ahora entra en juego el apartado de experimentación, imprescindible para ver si las nuevas variables aportan información útil sobre el sesgo. En la barra de menú podemos encontrar la sección de experimentación.

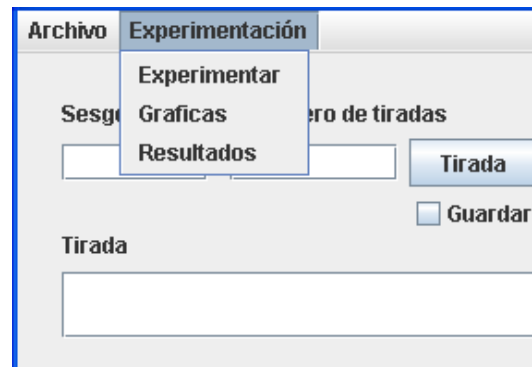


Figura 7.8: Opciones menu

Se divide en tres apartados, que utilizan los ficheros de experimentación. Como desde los tres apartados se accede a esos ficheros, y las operaciones pueden ser muy largas, dependiendo del número de experimentos, habrá un hilo que se encargue de esas tres aplicaciones, no pudiéndose usar una, mientras no se haya terminado la operación en curso.

El primer apartado que hay que ver es el de "Experimentar", puesto que genera los ficheros de experimentación que se usarán en los otros dos apartados. El menú de experimentación tiene la siguiente forma.

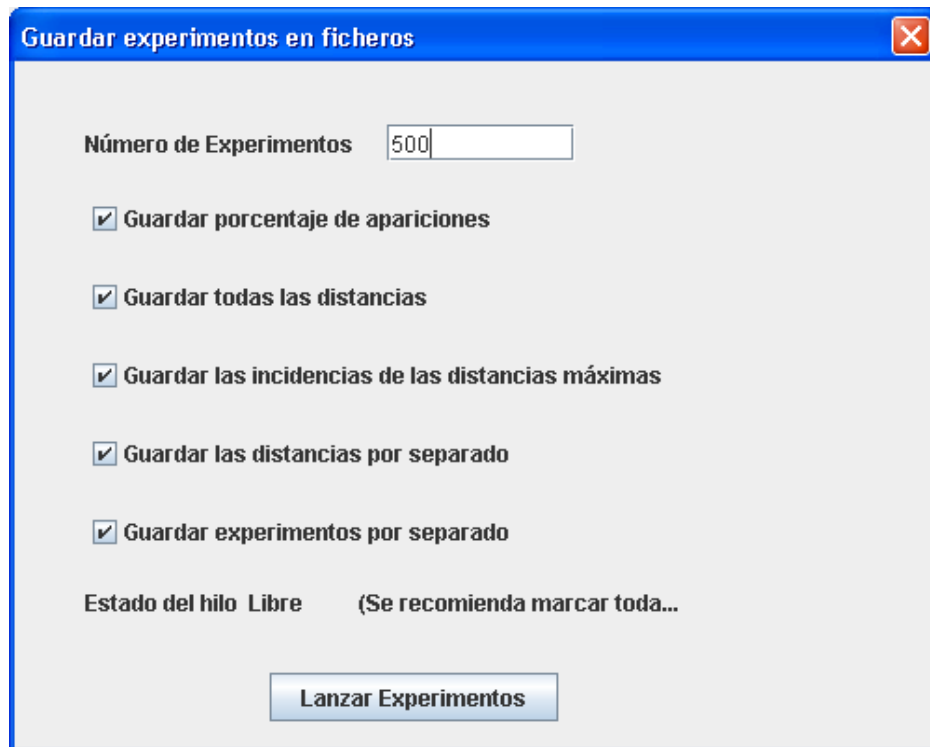


Figura 7.9: Ventana de experimentacion

Realiza n experimentos por cada sesgo de 0 a 100 de 10 en 10 (500 experimentos por cada sesgo) y cada experimento consiste en tirar la moneda 99 veces. Al final se guardará la información en los ficheros que se hayan guardado, las 4 primeras opciones guardan en fichero con lectura y sobreescritura, y se usan para generar las gráficas y calcular las medias sin necesidad de una lectura completa de los experimentos. Guardan en ficheros Sap.txt, S.txt, Sdist.txt y distancias/dNsS.txt, respectivamente, donde S es el sesgo, y N la distancia. (Ver anexo ficheros)

La última opción es para guardar cada experimento por separado, concatenando cada nuevo experimento de sesgo S al final del archivo tiradasS.txt. (Ver

anexo ficheros)

Cada nueva batería de experimentos es añadida a lo que ya había, lo cual permite dividir la experimentación en periodos de tiempo viables para el usuario.

Una vez generados los ficheros de experimentación, podemos obtener las gráficas de ellos, seleccionando la sección Gráficas del menú Experimentación. Nos muestran la siguiente ventana:

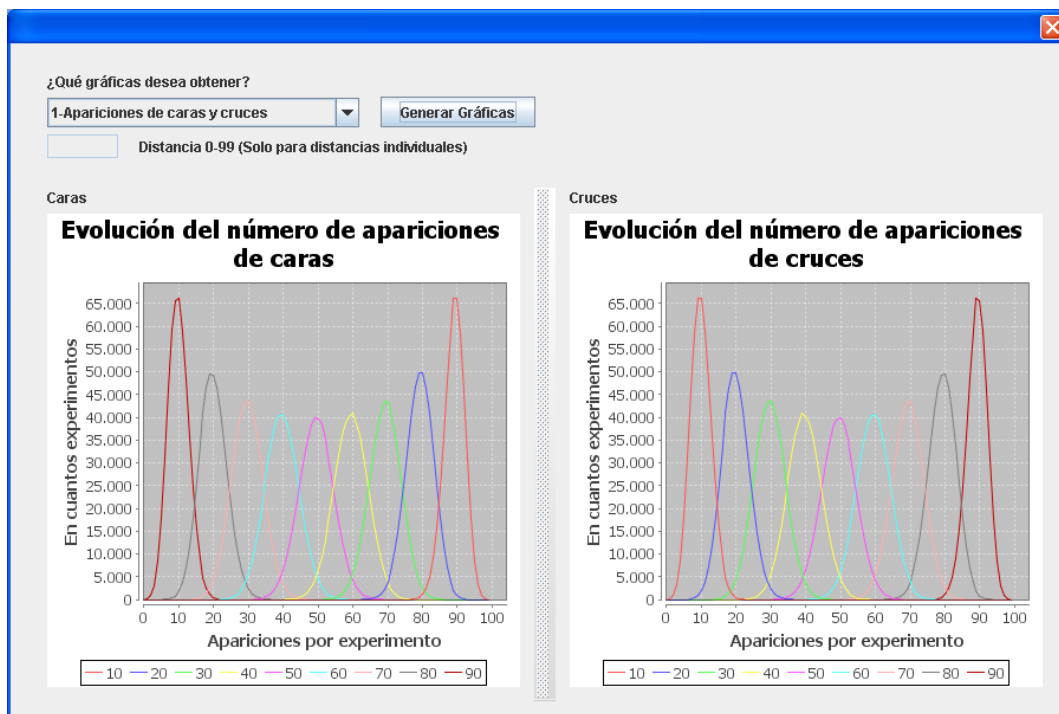


Figura 7.10: Ventana de graficas

Se muestra la ventana de Gráficas para las gráficas de apariciones de caras y cruces. Además, tenemos la opción de seleccionar otras gráficas, y darle luego al botón de generar, que a partir de los ficheros de información respectivos, obtiene las gráficas. En el caso de las distancias individuales, hay

que indicar sobre que distancia se quiere generar la gráfica.

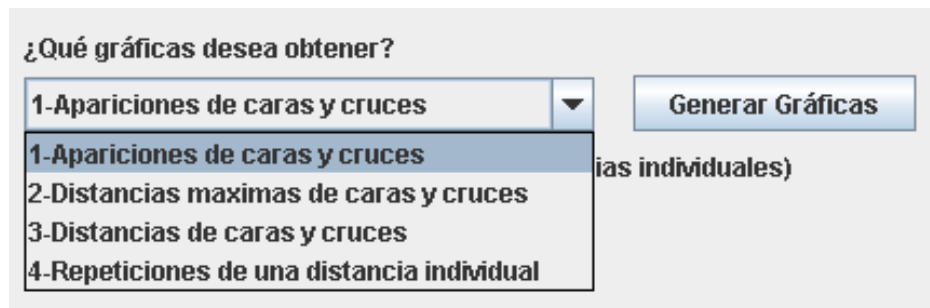


Figura 7.11: Selección de graficas

El último apartado de la experimentación, es Resultados”, o análisis de experimentos, y en esta ventana podremos ver la correlación entre las apariciones y las distancias máximas de caras y cruces, para observar si aportan nueva información.

The screenshot shows a software window titled "Ventana de obtencion de resultados". It features a blue title bar with a close button. The main area is divided into several sections:

- Buttons:** "Obtener resultados" and "Mostrar Resultados" are located at the top left.
- Sesgo:** A central control with a "<" button, a "0" value field, and a ">" button.
- Caras (Left Column):**
 - Apariciones:** Input fields for "Media" and "Varianza".
 - Distancia Máxima:** Input fields for "Media" and "Varianza".
 - Covarianza:** An input field.
 - Coefficiente de correlación:** An input field.
- Cruces (Middle Column):**
 - Apariciones:** Input fields for "Media" and "Varianza".
 - Distancia Máxima:** Input fields for "Media" and "Varianza".
 - Covarianza:** An input field.
 - Coefficiente de correlación:** An input field.
- Distancia 0 de caras (Right Column):** Input fields for "Media", "Varianza", "Covarianza con apariciones", and "Correlacion".
- Distancia 1 de caras (Right Column):** Input fields for "Media", "Varianza", "Covarianza con apariciones", and "Correlacion".
- Distancia 2 de caras (Right Column):** Input fields for "Media", "Varianza", "Covarianza con apariciones", and "Correlacion".

Figura 7.12: Ventana de obtencion de resultados

El botón "Obtener Resultados" lanza el hilo de experimentación si está libre, para que cargue los datos guardados en el fichero Resultados.txt" (Ver anexo formato fichero resultados) y los deja listos para mostrar. En el caso de que el fichero Resultados.txt" no se hubiese creado todavía, a partir de los ficheros de experimentación (todos son necesarios, por eso es recomendable dejarlos siempre todos los campos de experimentar marcados) hará los cálculos de varianzas y covarianzas de apariciones y distancias máximas, los cargará (habilitando el botón "Mostrar Resultados") y guardará los datos obtenidos en ficheros.

En el caso de realizar más experimentación y se quieran actualizar los

resultados, habrá que eliminar el fichero Resultados.txt”, para que se vuelva a generar con los nuevos ficheros de experimentación actualizados.

Junto con el CD de la aplicación, que contiene el proyecto NetBeans, la aplicación proyecto.jar y el javadoc, se entregan dos DVDs que contienen los ficheros de resultados de experimentación tiradasS.txt (Véase Anexo 2)

Anexo 2: Formatos de ficheros de experimentación

Nuestra aplicación guarda varios tipos de ficheros con distinto tipo de información de los experimentos, a través de los cuales genera las gráficas, y calcula varianzas y covarianzas.

El tipo de fichero principal es "tiradasS.txt" donde S es el sesgo de los experimentos incluidos en ese fichero, guarda toda la información de cada uno de los experimentos, añadiendo cada nuevo experimento al final del fichero. Cada experimento tiene el siguiente formato en el fichero.

```
-----Experimento del día (fecha del experimento) -----
tiradas:(numero de tiradas)
Caras:(número de apariciones de caras de ese experimento)
Cruces:(número de apariciones de cruces de ese experimento)
Caras(%):(Porcentaje de caras respecto al número de tiradas)
Cruces(%):(Porcentaje de cruces respecto al número de tiradas)
Dist max c:(Distancia máxima de caras en éste experimento)
Dist max x:(Distancia máxima de cruces en éste experimento)
Distancias:
0:(Número de veces que ha aparecido la distancia 0 para caras)
0:(Número de veces que ha aparecido la distancia 0 para cruces)
...
i:(Número de veces que ha aparecido la distancia i para caras)
i:(Número de veces que ha aparecido la distancia i para cruces)
```

```
...
99:(Número de veces que ha aparecido la distancia 99 para caras)
99:(Número de veces que ha aparecido la distancia 99 para cruces)
-----Experimento del día (fecha del experimento siguiente) -----
...
```

Para estos ficheros hay que tener en cuenta que al aumentar el número de experimentos, el tamaño de los ficheros "tiradasS.txt" aumenta considerablemente, la experimentación aportada por nosotros son 11 ficheros, con sesgos de 0 a 100, agrupados de 10 en 10, con 500000 experimentos cada uno, y cada fichero ocupa casi 700 Mb.

Para facilitar la extracción de información para obtener las gráficas, guardamos distintos ficheros con información parcial, que se puede obtener de los ficheros "tiradasS.txt", pero leer ficheros tan grandes resulta costoso.

Un ejemplo de este tipo de ficheros es "Sap.txt", que guarda las apariciones de caras y cruces para experimentos con un sesgo S. El fichero tiene el siguiente formato:

```
tiradas:(número de experimentos de 99 tiradas)
(C:)
(X:)
0:(indica en cuantos experimentos de sesgo S han habido 0 apariciones
de caras)
0:(indica en cuantos experimentos de sesgo S han habido 0 apariciones
de cruces)
...
```

i:(indica en cuantos experimentos de sesgo S han habido i apariciones de caras)

i:(indica en cuantos experimentos de sesgo S han habido i apariciones de cruces)

...

99:(indica en cuantos experimentos de sesgo S han habido 99 apariciones de caras)

99:(indica en cuantos experimentos de sesgo S han habido 99 apariciones de cruces) <end>

El siguiente fichero es "Sdist.txt", que guarda todas las repeticiones de las distancias acumuladas, su formato es el siguiente:

tiradas:(número de experimentos de 99 tiradas)

(C:)

(X:)

0:(es la suma de las repeticiones de la distancia 0 de caras, para todos los experimentos de sesgo S)

0:(es la suma de las repeticiones de la distancia 0 de cruces, para todos los experimentos de sesgo S)

...

i:(es la suma de las repeticiones de la distancia i de caras, para todos los experimentos de sesgo S)

i:(es la suma de las repeticiones de la distancia i de cruces, para todos los experimentos de sesgo S)

...

99:(es la suma de las repeticiones de la distancia 99 de caras,para todos los experimentos de sesgo S)

99:(es la suma de las repeticiones de la distancia 99 de cruces,para todos los experimentos de sesgo S)

<end>

Para entender como funciona este fichero supongamos un estado inicial del fichero en que la distancia i tiene la siguiente forma:

```
Tiradas:n
...
i:n1
i:n2
...
<end>
```

Leemos el fichero y lo guardamos en un array bidimensional de la forma $\text{int}[2][99]$, siendo 2, caras y cruces, y 99 las distancias, y al añadir una nueva tirada, donde la distancia i de caras aparece m1 veces,y la distancia i de cruces m2 veces, el resultado sería el siguiente:

```
tiradas:n+1
...
i:n1+m1
i:n2+m2
```

...
<end>

También guardamos un fichero con las distancias máximas, en el fichero "S.txt", donde S es el sesgo. El fichero tiene el siguiente formato:

```
tiradas:(número de experimentos de 99 tiradas)
(C:)
(X:)
0:(número de experimentos de sesgo S donde la distancia máxima de
caras es 0)
0:(número de experimentos de sesgo S donde la distancia máxima de
cruces es 0)
...
i:(número de experimentos de sesgo S donde la distancia máxima de
caras es i)
i:(número de experimentos de sesgo S donde la distancia máxima de
cruces es i)
...
99:(número de experimentos de sesgo S donde la distancia máxima de
caras es 99)
99:(número de experimentos de sesgo S donde la distancia máxima de
cruces es 99)
<end>
```

Todos los ficheros anteriores se crean y se leen del directorio principal

del proyecto, en el caso de utilizar el ejecutable, han de estar en el mismo directorio que el ".jar". Pero hay un último tipo de fichero, que es el número de repeticiones de una distancia determinada por un sesgo, los ficheros son de la forma "dNsS.txt" donde N es la distancia observada, y S es el sesgo para el que se observa esa distancia. Son generados un total de 1100 ficheros, por ello se guardan en el subdirectorio distancias. El formato de uno de los ficheros es el siguiente:

```
tiradas:(número de experimentos de 99 tiradas)
(C:)
(X:)
0:(número de experimentos de sesgo S, donde la distancia N de caras
se repite 0 veces)
0:(número de experimentos de sesgo S, donde la distancia N de cruces
se repite 0 veces)
...
i:(número de experimentos de sesgo S, donde la distancia N de caras
se repite i veces)
i:(número de experimentos de sesgo S, donde la distancia N de cruces
se repite i veces)
...
99:(número de experimentos de sesgo S, donde la distancia N de caras
se repite 99 veces)
99:(número de experimentos de sesgo S, donde la distancia N de cruces
se repite 99 veces)
```

<end>

Anexo 3: Formato de fichero de resultados

Los resultados del experimento, tales como calcular las medias y las varianzas, se guardan en el fichero Resultados.txt”, y las variables de nuestro estudio son las apariciones de caras y cruces, las distancias máximas de caras y cruces y las distancias 0, 1 y 2. El fichero tiene el siguiente formato:

MEDIAS DE APARICIONES

0:(Es la media de las apariciones de caras obtenida a partir de todos los experimentos de sesgo 0)

0:(Es la media de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo 0)

...

i:(Es la media de las apariciones de caras obtenida a partir de todos los experimentos de sesgo $i*10$)

i:(Es la media de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo $i*10$)

...

10:(Es la media de las apariciones de caras obtenida a partir de todos los experimentos de sesgo 100)

10:(Es la media de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo 100)

MEDIAS DE DISTANCIAS MAXIMAS

...

VARIANZA DE LAS APARICIONES

0:(Es la varianza de las apariciones de caras obtenida a partir de todos los experimentos de sesgo 0)

0:(Es la varianza de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo 0)

...

i:(Es la varianza de las apariciones de caras obtenida a partir de todos los experimentos de sesgo $i*10$)

i:(Es la varianza de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo $i*10$)

...

10:(Es la varianza de las apariciones de caras obtenida a partir de todos los experimentos de sesgo 100)

10:(Es la varianza de las apariciones de cruces obtenida a partir de todos los experimentos de sesgo 100)

VARIANZA DE LAS DISTANCIAS MAXIMAS

...

COVARIANZA ENTRE APARICIONES Y DIST. MAX. CC

0:(Covarianza entre apariciones de caras y distancia máxima de caras para los experimentos de sesgo 0)

0:(Covarianza entre apariciones de cruces y distancia máxima de cruces para los experimentos de sesgo 0)

...

i:(Covarianza entre apariciones de caras y distancia máxima de caras para los experimentos de sesgo i*10)

i:(Covarianza entre apariciones de cruces y distancia máxima de cruces para los experimentos de sesgo i*10)

...

10:(Covarianza entre apariciones de caras y distancia máxima de caras para los experimentos de sesgo 100)

10:(Covarianza entre apariciones de cruces y distancia máxima de cruces para los experimentos de sesgo 100)

Covarianza entre apariciones y dist max CX

0:(Covarianza entre apariciones de caras y distancia máxima de cruces para los experimentos de sesgo 0)

0:(Covarianza entre apariciones de cruces y distancia máxima de caras para los experimentos de sesgo 0)

...

10:(Covarianza entre apariciones de caras y distancia máxima de cruces para los experimentos de sesgo 100)

10:(Covarianza entre apariciones de cruces y distancia máxima de caras para los experimentos de sesgo 100)

MEDIAS DE LA DISTANCIA 0

0:(Es la media de las repeticiones de la distancia 0 de caras obtenidas a partir de todos los experimentos de sesgo 0)

0:(Es la media de las repeticiones de la distancia 0 de cruces obtenidas

a partir de todos los experimentos de sesgo 0)

...

i:(Es la media de las repeticiones de la distancia 0 de caras obtenidas a partir de todos los experimentos de sesgo $i \cdot 10$)

i:(Es la media de las repeticiones de la distancia 0 de cruces obtenidas a partir de todos los experimentos de sesgo $i \cdot 10$)

...

10:(Es la media de las repeticiones de la distancia 0 de caras obtenidas a partir de todos los experimentos de sesgo 100)

10:(Es la media de las repeticiones de la distancia 0 de cruces obtenidas a partir de todos los experimentos de sesgo 100)

MEDIAS DE LA DISTANCIA 1

...

MEDIAS DE LA DISTANCIA 2

...

VARIANZA DE LA DISTANCIA 0

...

VARIANZA DE LA DISTANCIA 1

...

VARIANZA DE LA DISTANCIA 2

...

COVARIANZA DE LA DISTANCIA 0 Y LAS APARICIONES

...

COVARIANZA DE LA DISTANCIA 1 Y LAS APARICIONES

...

COVARIANZA DE LA DISTANCIA 2 Y LAS APARICIONES

...

Anexo 4: Javadoc

Véase Javadoc en el subdirectorio `/dist` de la carpeta del proyecto Net-Beans.

Bibliografía

- [1] Francisco Javier Crespo Yáñez. Concepto distancias.
- [2] Juan Jose et al. Basulto Santos, Jesús y García del Hoyo. *Historia de la probabilidad y la estadística*. Universidad de Huelva, 1999.
- [3] Mónica Martínez Gómez. *Introducción a la estadística*. Universidad Politécnica de Valencia. Servicio de publicación, 2010.
- [4] H. Poincaré. *El Azar, en Sigma, el Mundo de las Matemáticas*. J.R. Newman, Ediciones Grijalbo, S.A., Barcelona, Vol. 3, pp. 68-82., 1976.
- [5] Jean-Pierre Vernant. *Mito y sociedad en la Grecia Antigua*. Siglo XXI, 2009.
- [6] David S. Moore. *Estadística aplicada básica, p.339*. Antoni Bosch, 1995.
- [7] J. E. Kerrich. *An experimental introduction to the theory of probability*. Antoni Bosch, 1964.
- [8] Joseph B. Keller. The probability of heads, the american mathematical monthly, vol. 93, pp. 191-197. (3), 1986.

-
- [9] Han of harmony. <http://hanofharmony.com/how-to-handle-adversity/>. (2), 2010.
- [10] Ciencianet. <http://ciencianet.com/buffon.html>.
- [11] elmundo.es Entrevista a los Pelayo. <http://www.elmundo.es/encuentros/invitados/2009/11/3927/>.
- [12] Gonzalo García Pelayo. *La fabulosa historia de los Pelayo*. Plaza y Janes, 2003.
- [13] Thomas Bas. *Eudaemonic Pie (Paperback)*. Eudaemonic Pie, 1986.
- [14] Susan Holmes y Richard Montgomery Persi Diaconis. *Dynamical bias in the coin toss*. Stanford University, 2004.
- [15] Matthew Clark y Brian Westerberg. <http://www.cmaj.ca/>.
- [16] Pinker Steven. *Cómo funciona la mente*. Destino.
- [17] T. Gilovich. *How We Know What Isn't So: The Fallibility of Human Reason in Everyday Life*. New York: The Free Press.
- [18] V. V. Bapeswara Rao and M. Bhaskara Rao. *A three-door game show and some of its variants*. The Mathematical Scientist 17, no. 2, pp. 89 a 94.
- [19] Casinos Online 777. <http://es.777.com/juegos-de-casino/ruleta/reglas-de-ruleta>.

-
- [20] Scott W. Teare Daniel B. Murray.
http://link.aps.org/doi/10.1103/PhysRevE.48.2547. The American Physical Society.
- [21] Bo Allen. *http://www.boallen.com/random-numbers.html.*
- [22] A life. *http://alife.co.uk/nonrandom/.*
- [23] Unkwnon. *Tratado sobre Política.* 1317b28-30.
- [24] Javier Sanguino Botella. *Iniciación a L^AT_EX. Un sistema para preparar documentos.* Madrid, Addison-Wesley, ISBN 84-7829-013-3.
- [25] Rodrigo De Castro Korgi. *El universo L^AT_EX. Imprenta en casa.* Universidad Nacional de Colombia, Bogotá, ISBN 958-701-060-4.
- [26] Chantal Maillard. *Logica borrosa.* Ediciones Miguel Gomez, Malaga.
- [27] Nippurl. *http://www.mygnet.net.*