Problema clásico de la ruina del jugador y con un oponente infinitamente rico a través de caminatas aleatorias y Python

Classic Gambler Ruin problem and with an infinitely rich opponent through random walks and Python

Maria Cristina Medel Lopez[1], Gladys Denisse Salgado Suárez[2], Francisco Solano Tajonar Sanabria[1] y Fernando Velasco Luna[1]

[1]Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla,
[2]Universidad de las Américas Puebla.
*Correo-e: maria.medell@alumno.buap.mx

PALABRAS CLAVE:
Proceso estocástico, Caminata aleatoria, Python, Número pseudoaleatorio.

RESUMEN

La teoría de los procesos estocásticos permite estudiar sistemas o fenómenos que evolucionan en el tiempo de forma aleatoria y que varían en un conjunto bien definido de estados, son muy diversas las áreas en las que sistemas de este tipo están presentes, por ejemplo, en economía, meteorología y en el desarrollo de múltiples procesos cotidianos y no tan cotidianos. De ahí la relevancia de estudiar este tipo de procesos y la teoría alrededor de ellos, para que, así como con el estudio de la probabilidad se pueda generar herramientas útiles en la toma de decisiones. En el presente trabajo se estudian el problema de la ruina del jugador en su versión clásica y otra modificada, como caminatas aleatorias, la cuales son un caso particular de las Cadenas de Márkov, con el propósito de explorar las propiedades de las caminatas que modelan a ambas versiones e interpretarlas en su respectiva simulación en el lenguaje de programación Python. Al mismo tiempo abordar el uso de generadores de números pseudoaleatorios, conceptos de recursividad y listas dinámicas definidas como una clase con sus respectivos métodos, objetivos que fueron alcanzados. Como principales resultados están la reproducción de las trayectorias que describen eventos simples de la caminata aleatoria y la estimación de las probabilidades de ruina y duración esperada propias de cada juego. Este es un problema clásico del que es posible partir para estudiar conceptos y propiedades de los procesos estocásticos y de programación.

KEYWORDS:
Stochastic process, Random walk, Python Pseudorandom Number.

ABSTRACT

The theory of stochastic processes makes it possible to study systems or phenomena that evolve in time randomly and that vary in a well-defined set of states, the areas in which systems of this type are present are very diverse, for example, in economics, meteorology and in the development of multiple everyday and not so everyday processes. Hence the relevance of studying this type of process and the theory around them, so that, as well as with the study of probability, useful tools can be generated in decision-making. In the present work, the problem of the player's ruin is studied in its classic and modified versions, such as random walks, which are a particular case of the Markov Chains, with the purpose of exploring the properties of the walks that model both versions and interpret them in their respective simulation in the Python programming language. At the same time addressing the use of pseudo-random number generators, recursion concepts and dynamic lists defined as a class with their respective methods, objectives that were achieved. The main results are the reproduction of the trajectories that describe simple events of the random walk and the estimation of the probabilities of ruin and expected duration of each game. This is a classic problem from which it is possible to start to study concepts and properties of stochastic processes and programming.

• **Recibido:** 20 de agosto 2021 • **Aceptado:** 9 de abril 2022 • **Publicado en línea:** 30 de noviembre 2022

1 INTRODUCCIÓN

PROCESOS ESTOCÁSTICOS.

Las caminatas aleatorias son un caso particular de las Cadenas de Márkov, un proceso estocástico a tiempo discreto con múltiples aplicaciones. Un proceso estocástico se define como una colección de variables aleatorias $\{Xt,\ t \geq 0\}$ definidas en un mismo espacio de probabilidad e indexadas por un conjunto T que representa al tiempo (Paul G. Hoel, 1972), si dicho conjunto es un intervalo entonces el proceso es a tiempo continuo, si, por otro lado, es un subconjunto de los números enteros, entonces, se dice a tiempo discreto.

El conjunto de valores que pueden tomar las variables aleatorias, el cual se puede denotar como S, se denomina espacio de estados y este también puede ser finito (de dos elementos por ejemplo 0 y 1), infinito contable, o infinito incontable. Los procesos estocásticos se pueden clasificar de acuerdo con el tipo de conjuntos que son S y T, si estos son discretos o continuos o de acuerdo con las relaciones de dependencia entre las variables aleatorias que lo conforman (Rincón, 2012). Las cadenas de Márkov son un tipo de proceso estocástico $\{Xn,\ n \geq 0\}$ a tiempo discreto que se caracterizan por cumplir la propiedad de Márkov, la cual se expresa en la ecuación (1).

$$\boldsymbol{P(X_{n+1} = x_{n+1} | X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n)} \quad (1)$$

Para interpretar esta propiedad primero se aclara que, X_n=x_n es encontrarse en el estado x_n ∈ S, al momento $n \geq 0$, por lo tanto, esta propiedad quiere decir que la probabilidad de encontrarse en un cierto estado x_(n+1) al momento $n + 1$, dado que se conoce el estado en el que se encontró el proceso en las etapas $n, n-1, \dots, 0$, es la misma que, la probabilidad de encontrarse en dicho estado x_(n+1) dado que solo se conoce el estado del proceso al momento n. Es decir que, al conocer el estado en la etapa previa, el conocimiento del resto de los estados previos por los que pasó el proceso no aporta más información, de manera que puede disponerse de ella. Las caminatas aleatorias simples y unidimensionales son un caso particular de las cadenas de Márkov, se parte de un estado inicial, uno de los posibles valores de, el estado siguiente será el resultado de sumar o restar una unidad al estado inicial, con probabilidades p y q respectivamente. Pasar de un estado a otro se denomina una transición entre estados, la dinámica de transición descrita se puede apreciar en la Figura 1, en la que sin pérdida de generalidad se considera que se parte del valor cero.

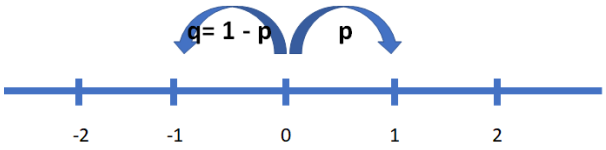


Figura 1: Desplazamiento de una caminata aleatoria simple unidimensional.

El estado de una caminata aleatoria en cualquier etapa n se puede entender entonces como el resultado de la suma de los primeros n elementos de una sucesión de variables aleatorias independientes e idénticamente distribuidas {ξ_i,i≥1}, las cuales pueden tomar únicamente los valores 1 y −1 (Gallager, 2013). Lo anterior se expresa en la siguiente ecuación.

$$\boldsymbol{X_n = X_0 + \xi_1 + \xi_2 + \cdots + \xi_n} \quad (2)$$

La caminata aleatoria cumple la propiedad de Márkov ya que el encontrarse en cierto estado en una etapa específica, depende únicamente del estado en el que estuvo previamente. Por ello se dice que una caminata aleatoria es un caso particular de las cadenas de Márkov. Las probabilidades de transición en un solo paso para una caminata aleatoria simple se definen a continuación.

$$\boldsymbol{P(X_n = j | X_{n-1} = i) = \begin{cases} p & si\ j = i+1 \\ 1-p & si\ j = i-1 \end{cases}} \quad (3)$$

En donde p es un número en (0,1) y la probabilidad para cualquier otro valor de j es igual a cero. Los elementos de un proceso estocástico son así: su espacio de estados S, el conjunto de parámetros T, su función de transición y su distribución inicial (Paul G. Hoel, 1972). Sin embargo, en el estudio de los procesos estocásticos hay más características y propiedades que nos permiten estudiarlos, para el modelo que se usó en este trabajo fue necesario además de los elementos básicos, la función o probabilidades de transición en n pasos. Las probabilidades de transición en uno o más pasos sirven para clasificar los estados, en el caso de las cadenas también es posible a partir de ellos clasificar a las cadenas mismas. Si partiendo de un estado i la probabilidad de que el proceso se vuelva a encontrar en dicho estado es 1, se dice entonces que i es un estado recurrente, un caso particular de los estados recurrentes es lo que se denomina estado absorbente, este es un estado que cumple que una vez que el proceso se encuentra en él, en la etapa siguiente con probabilidad uno se encontrará en el mismo estado. Cuando es el caso que partiendo de un estado *i*, hay una probabilidad positiva de jamás regresar a él, dicho estado es llamado transitorio.

PROBLEMA DE LA RUINA DEL JUGADOR.

Este problema se publicó en latín por primera vez en el tratado de Huygens en el año de 1657, como el último de una lista de cinco problemas (Jesús Basulto, 2009). En él se plantea un escenario con dos jugadores, que se pueden identificar como jugador A y jugador B respectivamente, cada uno inicia con 12 fichas, partida a partida se lanzan tres dados, si el resultado del lanzamiento suma 11 entonces el jugador A cede una ficha a su oponente, por otro lado, si el resultado es 14, el jugador B cede una unidad al jugador A, el juego se termina cuando alguno de los dos se ha quedado sin más fichas que apostar. La pregunta en dicho documento fue ¿Cuál es la probabilidad que cada jugador tiene de ganar? Las respuestas dadas en ese tiempo utilizaron principios combinatorios y herramientas como los árboles de probabilidad, así como el planteamiento de múltiples ecuaciones que representasen los posibles escenarios partida a partida. La teoría de los procesos estocásticos permite emplear una nueva perspectiva para este problema, al tomar como punto de referencia la experiencia del jugador A, se tiene que su capital al inicio tiene un valor especifico, y este cambiará en el tiempo de forma aleatoria.

Es por ello por lo que este problema se encuentra presente en distintos libros introductorios a la teoría de procesos estocásticos. La manera en la que se enuncia el problema de la ruina del jugador es de forma más general, se plantea a dos jugadores A y B, cada uno inicia con un capital k y $a - k$ respectivamente, es decir que hay en juego un número finito de a unidades, ambos participarán en partidas de un juego que no permite empates, en el que la probabilidad de ganar del jugador A siempre es $p \in (0,1)$ y la del jugador B es $1 - p$, partida tras partida, el jugador que resulte perdedor cede una unidad de su capital a su oponente, se pregunta ahora ¿cuál es la probabilidad de quedar arruinado? ¿Cuál es la duración esperada del juego? La versión que se pondrá en comparación es una en la que se tienen las mismas condiciones, salvo que ahora el capital del jugador B es muy grande, tanto que se considera infinito, piense en por ejemplo un escenario en el que se está jugando contra la casa en un casino, cuyos recursos son inmensos, esta versión recibe el nombre de, la ruina del jugador con un oponente infinitamente rico (Peter W. Jones, 1957), se hacen las mismas preguntas sobre la probabilidad de ruina y la duración esperada. Se dijo anteriormente que el capital del jugador A, evoluciona en el tiempo de forma aleatoria, pero además lo hace en etapas del tiempo discretas y el estado del proceso en cada nueva etapa es resultado de la suma o resta de una unidad al paso inmediatamente previo, de ahí que este sistema sea modelado como una caminata aleatoria simple y unidimensional para ambas versiones.

ELEMENTOS DE LA SIMULACIÓN.

Para simular el problema clásico de la ruina del jugador y la versión con un oponente infinitamente rico, es necesario contar con un generador de números pseudoaleatorios. Python cuenta con un módulo de generadores de números de este tipo para varias distribuciones, uniformes, normales, entre otras herramientas con las que es posible seleccionar elementos enteros o de tipo flotante dentro de un rango y varias instrucciones más. Casi todas las funciones de dicho módulo dependen de una función básica random(

), esta genera un número flotante aleatorio en el rango semiabierto [0.0, 1.0), el generador principal que utiliza Python es el Mersenne Twister (Lib/random.py, 2021), por las características de este generador no se recomienda para tareas de criptografía, sin embargo, para los propósitos de simular los escenarios de juego descritos en la sección anterior, es de bastante utilidad. La forma en que nos interesa que se distribuyan los números aleatorios es la uniforme, de manera que, al asignar un cierto valor a la probabilidad de ganar en cada ronda, el resultado del experimento aleatorio se pueda determinar con la generación de un número en el intervalo semiabierto mencionado anteriormente y revisando si este número es menor o igual que el valor definido, o mayor. Una rutina que ayuda a apreciar este comportamiento es el definir una región y dibujar puntos cuyas coordenadas sean generadas por las funciones que producen los números pseudoaleatorios verificando que no haya algún acumulamiento o tendencia marcada en la posición de los puntos, tal como se ve en la Figura 2, la cual representa la generación de 500 puntos de forma aleatoria.

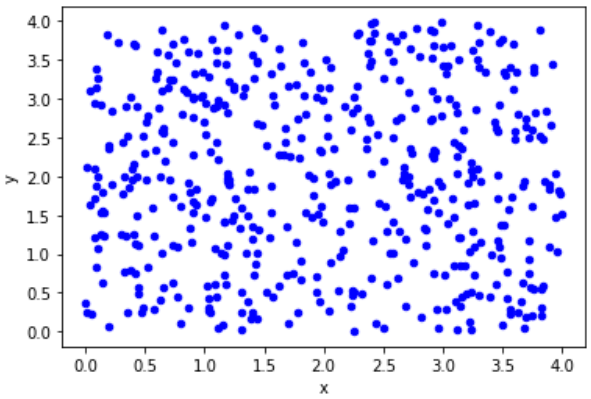


Figura 2. Distribución de puntos con coordenadas generadas por la función random, en una región cuadrada.

Para cada simulación se espera ver la trayectoria que describa su desarrollo, por lo que es necesario guardar el historial del recorrido hecho por la caminata aleatoria hasta llegar a la ruina de alguno de los dos jugadores, en el caso de la versión clásica, o sólo a la ruina del jugador A, para la versión con un oponente infinitamente rico, en caso de que realmente se alcance ese estado. La cantidad de elementos en este historial, lo cual es interpretado como la cantidad de partidas antes de llegar a alguno de los escenarios descritos anteriormente va a variar entre simulaciones, por lo que se vuelve necesario definir y posteriormente utilizar listas dinámicas. En su forma simple una lista se conforma por nodos, estos constan al menos dos campos, uno el cual alberga un dato del tipo que el programador defina y otro que corresponde a un apuntador, en la Figura 3 se ve un esquema de ello.

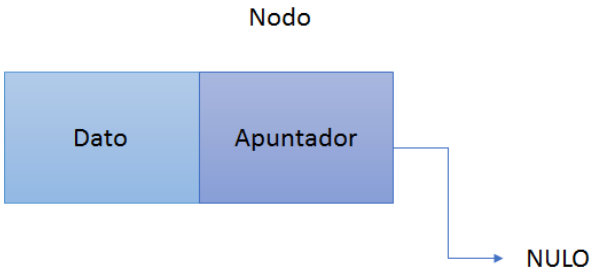


Figura 2: Nodo simple.

En la Figura 3 se observa a un nodo cuyo apuntador se dirige a NULO, esto es porque no está asignado a ningún otro objeto o variable. Agregar elementos a la lista significa que estos apuntadores estén relacionados por medio de los apuntadores, tal como se presenta en la Figura 4.

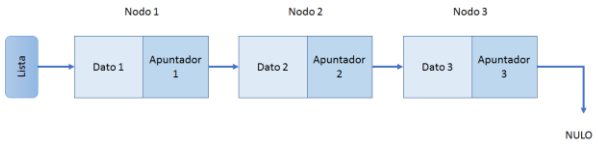


Figura 3: Lista con tres nodos.

Por lo que tener el registro de la trayectoria se haría agregando tantos nodos como la simulación lo vaya requiriendo hasta llegar a alguna de las condiciones en las que el juego termina. Agregar nuevos elementos a la lista se puede definir como un atributo o método del objeto nodo. Una forma en que esto se puede realizar es por medio de un método recursivo, al que se le da el nodo raíz de la lista y este la recorre hasta encontrar el nodo con apuntador nulo, en ese nodo es en el que se inserta el nuevo elemento. Los métodos recursivos son sumamente útiles, sin embargo, al igual que otros recursos al momento de programar, hay un límite. El lenguaje de programación Python tiene un límite de recursión que sirve para evitar caer en ciclos infinitos, con una instrucción de la biblioteca *sys* el sistema nos devuelve dicho límite, para el caso de Python 3.8, este es de 3000, hay instrucciones que sirven para

modificarlo, pero, la sugerencia suele ser optar por reescribir dichas rutinas recursivas de forma iterativa.

2 METODOLOGÍA

Se revisaron ambas versiones del problema de la ruina del jugador identificando las diferencias que hay entre ellos. Con la teoría de procesos estocásticos y particularmente de cadenas de Márkov, se estudiaron los elementos que definen a cada caminata aleatoria, para ello se revisaron las secciones correspondientes en (Paul G. Hoel, 1972; Rincón, 2012; Gallager, 2013).

Se estudiaron las soluciones desarrolladas y sugeridas a las preguntas ¿Cuál es la probabilidad que tiene el jugador A de quedar arruinado? ¿Cuál es la duración esperada del juego? En ellas se vio que, los valores correspondientes a la probabilidad de ganar en cada ronda, capital inicial y en el caso de la versión clásica también el capital total, son los parámetros que condicionan las respuestas a dichas preguntas. Estas soluciones se encuentran en (Peter W. Jones, 1957; Rincón, 2012), en donde se tiene el uso de resultados de la probabilidad condicional, pero también se propone la opción de responder a dichas preguntas con el planteamiento y resolución de ecuaciones en diferencia.

El haber definido cada etapa del proceso estocástico como está en (2), es decir, como la suma de los incrementos y decrementos de una unidad al capital del jugador A, acumulados partida tras partida, permitió que la programación de las rutinas de simulación resultaran más accesibles, ya que se plantearon ciclos en los que se parte de una variable que contenga el valor del capital inicial, posteriormente, dentro del ciclo se incrementa o disminuye en una unidad el valor del capital inicial, luego se pasa a verificar si se ha llegado a alguna condición de paro, estas condiciones son: el jugador A ha perdido todo, el jugador B ha perdido todo o que se haya llegado al límite de iteraciones fijado. La respuesta a la probabilidad de ruina y la duración esperada del juego, quedó determinada por ciertas expresiones que dependen de los parámetros que definen a la caminata aleatoria, sin embargo se planteó hacer la estimación de estos valores con el uso de las simulaciones, para esto último no es necesario conocer todo el desarrollo del proceso, solo se requiere conocer el número de partidas requeridas antes de alcanzar una condición de paro y saber si dicha condición fue la ruina del jugador A, la del jugador B, o la de ninguno. De lo anterior se tuvo que se requieren dos rutinas de simulación, una en la que se tenga el registro de todos los estados por los que pasa el proceso (para reproducir la trayectoria) y otra en la que solo se devuelva el número de partidas que tomó la simulación en acabar y el motivo por el cual terminó, por lo que para la primera clase de simulación si se requirió del uso de listas dinámicas para guardar el historial y con ello hubo que respetar el límite de recursividad, mientras que, para la segunda clase de simulación este límite se pudo rebasar sin problema alguno, pero ciertamente es necesario fijar un límite, ya que, como se vio para el problema de la ruina del jugador con un oponente infinitamente rico, en distintos escenarios, el juego tiene probabilidades positivas de no terminar jamás. Finalmente se hizo una comparación de las características identificadas por medio de la literatura y de las observadas en las simulaciones.

3 RESULTADOS

Los parámetros que definen a la caminata aleatoria para el problema clásico de la ruina del jugador son: probabilidad de perder en cada partida (p), capital inicial (k) y capital total (a), la definen en el sentido de que con dichos se establecen las probabilidades de transición, por otro lado, el capital inicial es el estado inicial, mientras que el capital total representa un estado absorbente para el proceso estocástico que modela este problema. Los dos estados absorbentes en la versión clásica son el estado cero y el estado a, ya que una vez alcanzados el juego termina, esto es que, para el resto de las etapas a partir de haber llegado a uno de dichos estados, la caminata permanecerá en ese valor. Se hacen observaciones similares para el problema de la ruina del jugador con un oponente infinitamente rico y se resumen sus principales diferencias en la Tabla 1.

Tabla 1: Características de las caminatas aleatorias.

Problema Clásico	Problema con un Oponente Infinitamente Rico
Caminata con espacio de estados finito $\{0,1,\dots,k,\dots,a\}$	Caminata con espacio de estados infinito $\{0,1,\dots,k,\dots\}$
Estados absorbentes: $\{0,a\}$	Estado absorbente: $\{0\}$

Al estudiar la probabilidad de ruina y la duración esperada del juego para la versión clásica, se encontró que era necesario hacer diferencia entre el caso simétrico y el asimétrico. Por ejemplo, si se seguía la resolución con ecuaciones en diferencia, suponer igualdad de probabilidades para ambos jugadores generaba una ecuación distinta y con ello una expresión menos complicada que aquella en la que se emplea la variable p. Se tuvo así que, para un jugador que inicia con un capital de k unidades, que tiene una probabilidad de ganar en cada ronda p y hay un total de a fichas en juego, la probabilidad de quedar arruinado (la cual se denotará como u_k) y la duración esperada del juego (la cual se denotará como d_k) son las siguientes,

con $s = \frac{1-p}{p}$

$$u_k = \begin{cases} \dfrac{s^k - s^a}{1 - s^a} & si \quad p \neq 0.5 \\ \dfrac{a-k}{a} & si \quad p = 0.5 \end{cases} \quad (4)$$

$$d_k = \begin{cases} \dfrac{1}{1-2p}\left(k - \dfrac{a(1-s^k)}{1-s^a}\right) & si \quad p \neq 0.5 \\ a(a-k) & si \quad p = 0.5 \end{cases} \quad (5)$$

De (4) y (5) es posible revisar como varían las probabilidades de ruina y la duración esperada para un juego con cierto capital total en juego, variando las opciones de capital inicial y de probabilidad de ganar en cada ronda, tal como se ve en las Figuras 5 y 6.

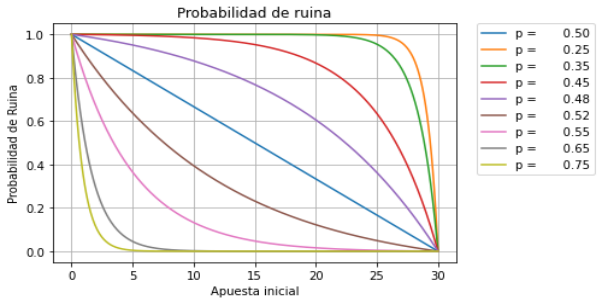


Figura 4: Probabilidad de ruina para distintos valores de p, cuando el capital total en juego es de 30 unidades.

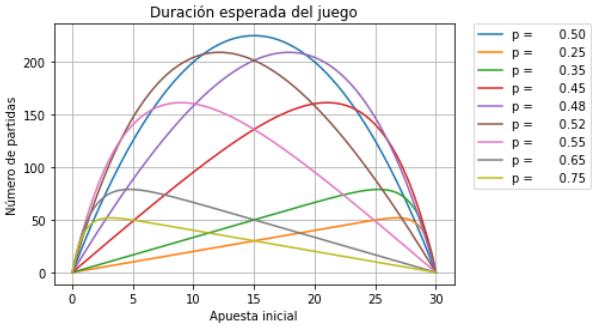


Figura 5: Duración esperada para distintos valores de p, cuando el capital total en juego es de 30 unidades.

Las Figuras 5 y 6 corresponden a un caso del problema de la ruina del jugador clásico, en el que se tienen 30 fichas en juego. Para el caso simétrico, la probabilidad de ruina se comporta de forma lineal, en cuanto a la duración esperada, esta alcanza su máximo valor cuando las condiciones de juego son más justas, es decir, cuando ambos jugadores tienen igualdad de probabilidades de ganar partida a partida y además inician con el mismo capital.

Para el problema de la ruina del jugador con un oponente infinitamente rico se parte de las expresiones (4) y (5), ya que la probabilidad de ruina y duración esperada serían el valor que pudiesen u_k y d_k en el límite cuando a tiende a infinito. Calcular dicho límite requiere que se haga una diferencia no solo entre el caso simétrico y asimétrico, ya que se tiene que el término s se eleva a la potencia a, y dicho término es en realidad el cociente de $1-p$ sobre p, por lo que se diferencia el caso en el que p es mayor o menor que 0.5, con lo que se tiene la diferencia entre tres casos posibles, los resultados obtenidos se presentan en la Tabla 2. Puede verse que para los casos en que el valor de p, es mayor o igual a 0.5, la duración esperada es infinita, por lo que se considera el escenario en que, en el proceso de generar las trayectorias, la simulación del problema alcance el límite de recursividad propio de Python. También resulta interesante observar que la ruina es segura para los casos en que el valor de p es menor o igual que 0.5, de otra forma, la ruina no es segura y eso quiere decir que hay una probabilidad positiva de no alcanzar el único estado absorbente.

Tabla 2: Probabilidad de ruina y duración esperada para el problema con un oponente infinitamente rico.

Valor de p	u_k	d_k
$p < 0.5$	1	$\frac{k}{1-2p}$
$p = 0.5$	1	∞
$p > 0.5$	s^k	∞

En cuanto a las simulaciones se definen algoritmos para realizar la simulación en el lenguaje de programación Python, pero de manera general tienen una estructura similar a la que se ve en la Figura 7, primero se ingresa el valor de los parámetros que definen a la caminata, estos son: probabilidad de ganar en cada ronda (p), capital inicial (k) y capital total (a), para el problema con un oponente infinitamente rico se omite este último parámetro, posteriormente se define el límite de iteración, como se dijo, en el presente trabajo se consideraron dos rutinas de simulación, una para reproducir las trayectorias y otra para guardar únicamente la información de número de rondas jugadas y condición de paro alcanzada, también es necesario inicializar una variable para ir contando el número de rondas jugadas, la parte esencial de la simulación es el resultado de la ronda, para ello se definió una parte del algoritmo en la que se genera un número aleatorio en el intervalo [0.0, 1.0), si el número es menor o igual que p, se interpreta entonces como que el jugador A ganó, con ello se registra un incremento o decremento del capital en una unidad, posteriormente se actualiza el número de iteraciones y se procede a evaluar alguna de las condiciones de paro, estas son haber alcanzado la ruina, para la versión clásica, otra condición de paro es haber ganado todo el capital en juego y por último verificar si no se ha llegado al límite de iteraciones, por último, su implementación requirió del uso de las librerías *random* y *matplotlib*.

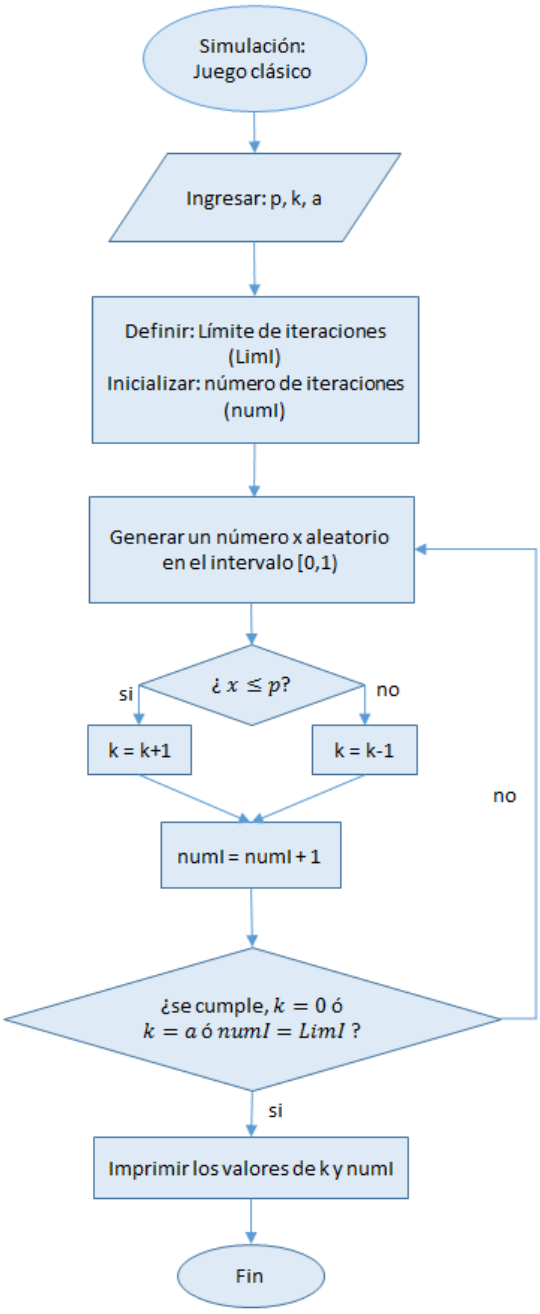


Figura 6: Diagrama de flujo del algoritmo empleado para realizar las simulaciones del problema de la ruina del jugador.

Para que el algoritmo regrese la trayectoria, es decir, el historial del desarrollo que tuvo el capital del jugador A, es necesario agregar entre las variables a la lista dinámica y en la parte en que se actualiza el número de

iteraciones, insertar un nuevo nodo con el nuevo capital, en las Figuras 8 y 9 se tiene ejemplos de algunas trayectorias obtenidas para el problema clásico y modificado respectivamente. En la Figura 8 se tiene el resultado de simular el problema clásico con los parámetros $p = 0.52$, $k = 10$ y $a = 30$, donde se observa que después de 128 partidas el jugador ganó todo el capital en juego, la Figura 9 muestra el resultado de 25 simulaciones con los mismos parámetros. En cuanto a la Figura 10, se muestran varias trayectorias que corresponden a la simulación del problema de la ruina del jugador con un oponente infinitamente rico, en el caso simétrico y cuando el jugador A, inicia con un capital de 25 unidades, recordar que para dicho caso, la probabilidad de ruina es 1 y la duración esperada es infinita, se observa que algunas trayectorias ya terminaron en la inevitable ruina del jugador A, sin embargo otras continúan, pero dentro del programa se ha alcanzado el límite de recursión (3000), que se traduce en el número de elementos o nodos que conforma la lista o bien, el número de partidas, por lo que se concluye que para esas trayectorias el juego no ha terminado.

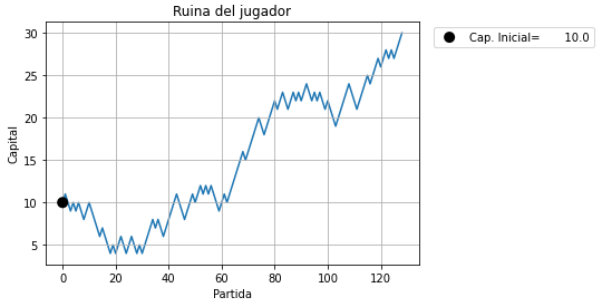


Figura 7: Simulación de una trayectoria para el problema cásico de la ruina del jugador con p=0.52, k=10, a=30.

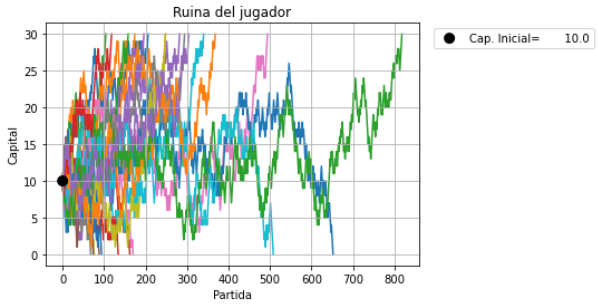


Figura 8: 25 simulaciones del problema clásico de la ruina del jugador con p=0.52, k=10, a=30.

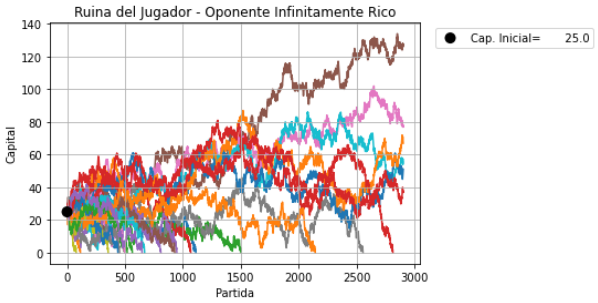


Figura 9: Simulación de varias trayectorias para el problema de la ruina del jugador con un oponente infinitamente rico, con p= 0.5, k=25.

Para la versión con un oponente infinitamente rico basta con hacer un cambio en las condiciones de paro, esta es que, ya no se contempla la posibilidad de que el capital del jugador A, representado por la variable k, alcance el capital total en juego, ya que este es infinito. Como se dijo anteriormente, la probabilidad de ruina y la duración esperada quedan determinadas por las expresiones obtenidas en (4) y (5) para el problema clásico, mientras que para el problema modificado, se tienen los resultados de la Tabla 2, sin embargo al tener acceso a un lenguaje de programación, en este caso Python, es posible ejecutar múltiples simulaciones con el objetivo de generar datos que sean útiles para estimar la probabilidad de ruina como una proporción entre juegos perdidos y el total de juegos en los que se participó, mientras que la duración esperada queda determinada por el promedio de duraciones. Para dichas estimaciones no es necesario reproducir las trayectorias y el límite de iteraciones puede ser bastante más grande. Resultado de ello se puede ver en la Figura 11.

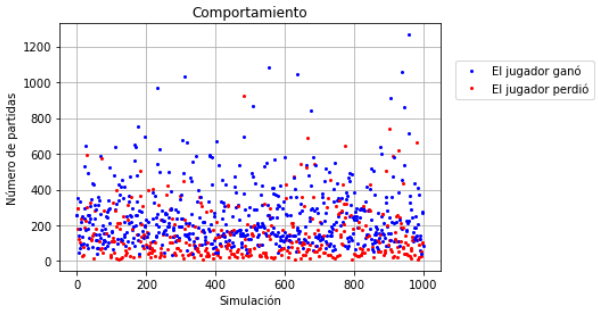


Figura 10: Resultados de 1000 simulaciones del problema clásico de la ruina del jugador con p=0.52, k=10, a=30.

En la Figura 11 se muestra el resultado de 1000 simulaciones del problema de la ruina del jugador en su versión clásica con los parámetros, p = 0.52, k = 10 y a = 30. Si bien hacer las estimaciones permiten comparar su resultado con el valor devuelto por las expresiones correspondientes, otro elemento interesante es que con esta herramienta se puede apreciar otro aspecto de la duración del juego que es la dispersión alrededor del promedio o bien, del valor esperado. Esto mismo se puede realizar para el problema con un oponente infinitamente rico, ya que en esta versión del problema se encuentra que para los casos en que el valor del parámetro p es mayor o igual que 0.5, la duración esperada del juego es infinita, por lo que, se vuelve interesante explorar el desarrollo de las trayectorias, aumentando el número de iteraciones posibles, es decir el límite de rondas que se pueden jugar, el resultado de ello se encuentra en la Figura 12, en la que se ve el resultado de 1000 simulaciones, donde se consideró una probabilidad de ganar en cada ronda igual a 0.5 y un capital inicial igual a 20 unidades, los puntos de color rojo representan juegos perdidos y el resto son juegos que después de 500000 no se habían terminado aún, sin embargo de la Tabla 2, se sabe que la ruina es segura.

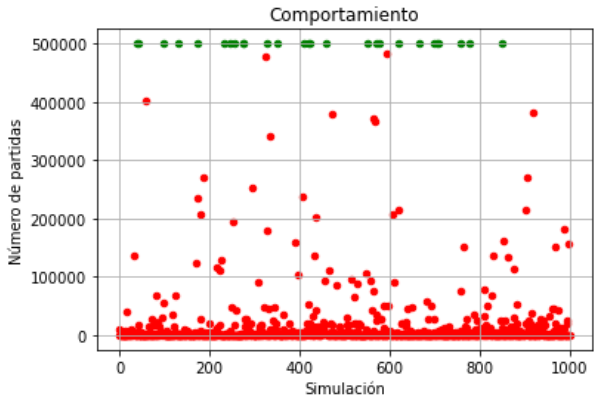


Figura 11: Resultado de 1000 simulaciones del problema de la ruina del jugador con un oponente infinitamente rico, con los parámetros p=0, k=20.

4. CONCLUSIONES

Para el problema clásico de la ruina del jugador, la sensibilidad de la probabilidad de ruina a las distintas opciones de capital inicial disminuye para cada vez más valores posibles de p, conforme el capital total aumenta. Esto es que, la probabilidad de ruina es más próxima a cero o a uno para la mayoría de los valores posibles con los que puede iniciar el jugador A, para valores de p fuera de intervalos cada vez más pequeños alrededor de 0.5.

Las simulaciones permiten explorar más características de los modelos, como lo fue en este caso la dispersión del número de partidas en que terminó el juego, considerando el valor esperado obtenido teóricamente. Para el problema de la ruina del jugador con un oponente infinitamente rico se tiene que la ruina es segura para valores de p menores o iguales a 0.5, sin embargo, aunque para el caso simétrico hay seguridad de que el jugador A quedará arruinado, el número de partidas para que esto ocurra puede ser demasiado grande, en las simulaciones que no mostraban la trayectoria y que por lo tanto el límite de iteraciones era mayor, se llegó a plantear un límite de 100000 partidas y en 500 simulaciones se encontraban juegos que aún no terminaban, por lo que se considera un caso interesante.

La teoría de procesos estocásticos, particularmente de las cadenas de Márkov, permite estudiar una gran variedad de situaciones y fenómenos. El problema de la ruina del jugador fue estudiado primeramente con resultados combinatorios, sin embargo, en este trabajo se vio que el problema clásico así como la versión de un oponente infinitamente rico y otras más que pudiesen formularse, se pueden modelar como caminatas aleatorias y con ello utilizar los conceptos y herramientas más básicos de esta teoría, de manera similar ocurre con las herramientas computacionales, las cuales deben irse integrando cada vez más en las tareas de aprendizaje. Este juego permite explorar ideas y aspectos importantes, así como interesantes de la simulación de fenómenos aleatorios.

REFERENCIAS

Gallager, R. G. (2013). *Stochastic Processes: Theory for Applications.* Cambridge, Inglaterra: Cambridge University Press.

Higgins, W. F. (1957). *An Introduction to Probability Theory and Its Applications.* Nueva York, EUA: John Wiley and Sons.

Jesús Basulto, M. D. (2009). La resolución de Montmort (1708, 1713) de los cinco problemas propuestos por Huygens en su tratado (1657). *IV Congreso Internacional de Historia de la Estadístico y de la Probabilidad*, 407-420.

Lib/random.py. (2021, septiembre 14). *random - Generar números pseudoaleatorios*. Retrieved from random - Generar números pseudoaleatorios: https://docs.python.org/es/3/library/random.html

Paul G. Hoel, S. C. (1972). *Introduction to Stochastic Processes.* Los Ángeles, California, EUA: Houghton Mifflin.

Peter W. Jones, P. S. (1957). *Stochastic Processes An Introduction.* Florida, EUA: CRC Press.

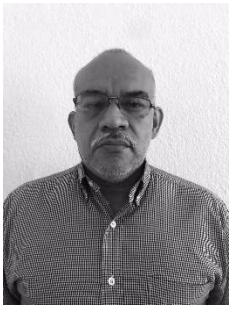
Rincón, L. (2012). *Introducción a los Procesos Estocásticos.* CDMX, México: Facultad de Ciencias UNAM.

Acerca de los autores

María Cristina Medel López. Obtuvo el grado de licenciada en Matemáticas Aplicadas por la Benemérita Universidad Autónoma de Puebla. En su paso por la licenciatura fue colaboradora y ponente en el evento Encuentro Internacional en la Enseñanza de la Probabilidad y la Estadística en 2018, 2019 y 2021, también participó en el Congreso Nacional de la Sociedad Matemática Mexicana y en el Seminario Nacional de Tecnología Computacional en la Enseñanza y el Aprendizaje de las Matemáticas en 2019, así como, en la XIV Semana Internacional de la Estadística 2021, con trabajos relacionados a la enseñanza y el aprendizaje de la probabilidad empleando recursos como juegos o problemas clásicos.

Dra. Gladys Denisse Salgado Suarez. Obtuvo el grado de Doctora en Ciencias (Matemáticas) por la Benemérita Universidad Autónoma de Puebla (BUAP) docente de tiempo parcial en el departamento de Actuaría, Física y Matemáticas de la Universidad de la Américas Puebla, en Puebla, Puebla, actualmente también es árbitro y miembro del comité editorial, así como organizadora y colaboradora del Encuentro Internacional de la Enseñanza en la Probabilidad y la Estadística (EIEPE). Obtuvo el reconocimiento Sofía Kovalevskaia, por parte de la Sociedad Matemática Mexicana y la fundación Sofía Kovalevskaia en 2018. Como investigadora cuenta con múltiples artículos indizados, arbitrados y capítulos de libro publicados en diversas revistas especializadas. Sus áreas de interés en investigación son Probabilidad (procesos de decisión de Márkov), Estadística de Procesos Estocásticos y Educación Matemática.

Dr. Francisco Solano Tajonar Sanabria. Profesor adscrito a la Facultad de Ciencias Físico Matemáticas de la Benemérita Universidad Autónoma de Puebla, desde marzo de 1984. Doctorado, Maestría y Licenciatura en Ciencias Matemáticas otorgados por la Benemérita Universidad Autónoma de Puebla. Integrante del Cuerpo Académico de Probabilidad y Estadística. Las líneas de investigación de interés son Probabilidad y Estadística, en particular Análisis de Supervivencia, Teoría de Riesgo, Matemáticas Financieras, Teoría de Colas.



Dr. Fernando Velasco Luna. Cursó la Licenciatura en Matemáticas en la Universidad Veracruzana, realizó los estudios de maestría en el área de Probabilidad y Estadística, recibió el grado de Dr. en Matemáticas con énfasis en Probabilidad y Estadística por la Universidad Veracruzana en el año 2011. Es profesor Investigador de la Facultad de Ciencias Físico Matemáticas de la Benemérita Universidad Autónoma de Puebla desde el año 2012. Imparte los cursos del área de Probabilidad y Estadística en las Licenciaturas de matemáticas, Actuaría, y Matemáticas Aplicadas, ha dirigido tesis de licenciatura, escrito artículos de investigación. Sus intereses de investigación son en el área de Probabilidad y Estadística.