TP 1.2 - ESTRATEGIAS EN UNA RULETA

Antonelli, Nicolás
Departamento de Ingeniería en Sistemas
Universidad Tecnológica Nacional - FR Rosario
Rosario, Zeballos 1341
niconelli2@gmail.com

Acciarri, Joshua
Departamento de Ingeniería en Sistemas
Universidad Tecnológica Nacional - FR Rosario
Rosario, Zeballos 1341
acciarrijoshua@gmail.com

1 de mayo de 2020

ABSTRACT

Pequeño estudio sobre las estrategias posibles a aplicar en un juego de ruleta europea simulado; los beneficios y contras de estas estrategias, y análisis económico de su utilización. El objetivo es contemplar si es rentable aplicar en la realidad alguna de nuestras estrategias presentadas.

1. Introducción

Este proyecto es el trabajo práctico 1.2 de la cátedra "Simulación"(UTN FRRO, Departamento ISI). En el mismo, se modelizan ciertas estrategias para aplicar en la ruleta 'europea'[1], estudiando los resultados de dos componentes claves: la frecuencia relativa de las victorias según el tipo de apuesta elegida y el flujo de la 'caja' o del capital disponible. Para ello trabajamos con números pseudoaleatorios para representar los números obtenidos al 'girar' la ruleta, los cuales se almacenan en listas (arrays), para luego graficar los resultados de N 'tiradas' de esta ruleta, lo que representa nuestra población de estudio. Además analizamos cómo se comportan las distintas gráficas si cambiamos el tipo de estrategia o ciertos factores como el capital inicial, valor de la apuesta, tamaño de tiradas (población) y para todos los casos, un caso particular en el que contemplamos el capital como hipotéticamente ilimitado. A su vez, hicimos un estudio de varios *resultados de juego* a la vez, es decir, seis juegos para la misma estrategia corriendo en simultáneo, y graficando el promedio obtenido entre estos juegos.

2. Marco Teórico

2.1. Definiciones

La cantidad de casos de victoria, es decir, que se cumpla el suceso de haber apostado una vez y no perdido el dinero, en N tiradas de ruleta, es una *Variable Aleatoria Discreta*, por la siguiente definición[2]:

> "Una *variable aleatoria* es una función que asigna a cada elemento del espacio muestral un número real. Una variable aleatoria es *discreta* si su recorrido es un conjunto finito o infinito numerable (susceptible de ser contado)". Lo anterior sucede en este caso con un rango de [0; N] donde N es la cantidad de tiradas o la cantidad de veces que hacemos 'girar' la ruleta.

Se tiene entonces definida una población de tamaño N representado por una lista, que se utiliza para obtener la frecuencia de victorias, así como el flujo del capital inicial, el cual se ingresó como parámetro de la simulación. Cada valor(x_i) en cada posición de esta lista representa número aleatorio obtenido al 'girar' la ruleta, repitiendo este proceso para N veces. Donde cada elemento x_i cumple:
$0 \leq x_i \leq 36$ $\quad$ $x \in \mathbb{Z}$ $\quad$ i=1,2,...,N

Ejemplo de una población de números aleatorios de tamaño N=10: {24, 1, 10, 0, 0, 6, 12, 36, 21, 35}

2.2. Fórmulas Empleadas

Cálculo de la Frecuencia de victorias con respecto a los N números de la tirada. Donde:

- n es la frecuencia absoluta de victorias, es decir, la cantidad de veces que ocurrió el suceso "no perder lo apostado en una tirada"
- N es el tamaño de la población, es decir los N números obtenidos en la tirada
- x representa el suceso de victoria

$$f_x = \frac{n_x}{\sum_{i=0}^{N} n_i} = \frac{n_x}{N} \tag{1}$$

Cálculo de la reducción o aumento del capital (Cantidad monetaria) en cada resultado de la ruleta. Donde:

- bet es el tamaño de apuesta
- w es el factor por el que se multiplica el tamaño de apuesta total según sea el tipo de apuesta (por ejemplo: en apuesta a color w=2, en apuesta a un número w=36)
- λ es 0 en caso de derrota, o 1 en caso de victoria

$$F(bet) = -bet + w\lambda bet \tag{2}$$

El valor de F(bet) se agrega al valor del capital para así actualizarlo luego de cada tirada

2.3. Tipos de Apuestas

Cada tipo de apuesta modificará el valor de w antes mencionado. Se incluye en la proporción de paga el valor de la apuesta (bet), puesto que esta no se pierde, y se dan ejemplos de estrategias que tengan este tipo de apuesta en algún momento de la misma. Entre los tipos de apuesta que hemos utilizado, se encuentran:

- **Número simple** (pleno): la paga es *36 a 1*, por lo que w = 36. Entre ellas:
  - Apuesta a un solo número (entre el 0 y el 36)
  - Gerardo Sofovich (utiliza el pago de un pleno)
- **Apuestas sencillas**: la paga es *2 a 1*, por lo que w = 2. Entre ellas:
  - Apuesta a color (rojo o negro)
  - Martingala clásica (utiliza la elección de un color)
  - Martingala modificada (utiliza la elección de un color)
  - Fibonacci (utiliza la elección de un color)
  - D'Alembert (utiliza la elección de un color)
- **Columna y Docena**: la paga es *3 a 1*, por lo que w = 3. Entre ellas:
  - SantE [Original] (utiliza el pago de una columna, apostando a dos de las tres posibles)

3. Estrategias Empleadas

Explicaremos brevemente en qué consisten las estrategias no arbitrarias, es decir, no explicaremos por ejemplo la estrategia de simplemente elegir un color, o la de elegir solo un pleno.

3.1. Martingala

Esta estrategia es muy conocida y tuvo cierta fama en el siglo XVIII. Consiste en una decisión simple: si perdemos en la predicción de nuestra apuesta, duplicaremos el valor neto de la apuesta en la próxima jugada, y si ganamos simplemente reiniciaremos nuestro valor neto de apuesta al inicial[4]. Es decir que para cada jugada N nuestro valor de apuesta V_N será igual a (V_{N-1} * 2) en caso de haber perdido en la jugada (N-1), y será V_0 en caso de haber ganado en la jugada (N-1).

El fuerte de esta estrategia radica en que con una proporción elevada entre el capital total que tenemos y el valor inicial de la apuesta, las probabilidades de perder todo nuestro capital disminuyen considerablemente.

Con una proporción apuesta:capital de 1:100, la probabilidad de perder decrece de la siguiente forma; donde:

- P_0 es la probabilidad pérdida en 1 tirada, apostando a un color (expresado también en forma porcentual).
- V_0 es el valor de la apuesta inicial.
- C_0 es el capital máximo que disponemos en el arranque del juego.
- k es la constante por la cual multiplicamos V_0 y C_0 para mantener la proporcion 1:100
- m es la cantidad máxima de veces seguidas que podemos doblar la apuesta sin quedarnos con capital para seguir duplicando la apuesta. $m \in \mathbb{Z}$.
- P_{m+1} es la probabilidad de quedarnos sin capital alguno, es decir, tener una racha de derrotas 1 vez mayor al máximo posible que podemos soportar (expresado también en forma porcentual).

$$P_0 = (0,5 + \frac{1}{37}) = 0,527 = 52,7\,\% \tag{3}$$

$$V_0\ 2^m \leq C_0 \implies 1k\ 2^m \leq 100k \implies m \leq log_2(100) \implies m \leq 6,74 \implies max(m) = 6 \tag{4}$$

$$P_{m+1} = (P_0)^{m+1} = 0,527^7 = 0,0113 = 1,13\,\% \tag{5}$$

Es decir, con una proporción 1:100, tendríamos que perder 7 veces seguidas ya que $k2^7 > k100$ y quedarnos sin capital, con una probabilidad de solo 1,13 % de que esto suceda. Más de 98,87 % de que ninguna racha de derrotas logre dejarnos sin capital antes de recuperar.

Y con una proporción 1:1000; donde cada constante y variable representa lo mismo que las ecuaciones anteriores, solo que es para una proporción 1:1000 y ya sabemos que $p_0 = 52,7\,\%$:

$$m \leq log_2(1000) \implies m \leq 9,96 \implies max(m) = 9 \tag{6}$$

$$P_{m+1} = (P_0)^{m+1} = 0,527^{10} = 0,0017 = 0,17\,\% \tag{7}$$

Es decir, con una proporción 1:1000, tendríamos que perder 10 veces seguidas ya que $k2^{10} > k1000$ y quedarnos sin capital, con una probabilidad de solo 0,17 % de que esto suceda. Más de 99,83 % de que ninguna racha de derrotas logre dejarnos sin capital antes de recuperar.

3.2. Martingala Modificada

Se basa en la martingala original, con la única modificación de que en caso de perder en la jugada (N-1), nuestro nuevo valor de apuesta V_N será igual a (V_{N-1} * 2 + 1), agregando una unidad monetaria extra. El objetivo de esta decisión es tener 1 unidad monetaria de ganancia 'virtual' en una racha perdedora. Es decir si perdemos 5 veces seguidas y en la sexta ganamos, en esa victoria recuperamos lo perdido +6 unidades monetarias; a diferencia de la martingala clásica en la que la ganancia neta sería 'solo' de +1 unidad monetaria. Las probabilidades de perder con proporciones 1:100 y 1:1000 son aproximadamente las mismas que con la martingala clásica.

3.3. D'Alembert

Otra estrategia que consta de una decisión simple: si ha ocurrido una victoria, disminuye en 1 unidad del valor de apuesta con respecto a la apuesta anterior. Si ha ocurrido una derrota, se aumenta en 1 el valor de la apuesta. Sin embargo, el valor de la apuesta será como mínimo del valor de la apuesta inicial, y si se gana apostando este valor entonces no restaremos del valor de la apuesta ninguna unidad monetaria[5].

Esta estrategia tiene como fuerte que no haremos apuestas tan fuertes en una racha de derrotas.

3.4. Fibonacci

Por comodidad, la apuesta inicial es 1. Teniendo en cuenta la definición de la sucesión de Fibonacci[3]:

$$\begin{cases} fib(0) = fib(1) = 1; \\ fib(n) = fib(n-1) + fib(n-2); \\ n \geq 3; \end{cases} \tag{8}$$

Obtenemos la sucesión de números siguiente:

$$fib(n) = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55...], \forall n \tag{9}$$

El tamaño de apuesta varía según la siguiente decisión: si en la apuesta anterior hubo una derrota, en la actual apostamos el siguiente número de la secuencia de Fibonacci (sin repetir el primer 1); y si en la apuesta anterior hubo una victora, retrocedemos 2 posiciones de la secuencia, llegando como apuesta mínima a la apuesta inicial de 1[6].

3.5. Estrategia de Gerardo Sofovich

Consistía en apostar un pleno simultáneo a 35 de los 37 números disponibles y en una cantidad elevada de apuesta. Con esto conseguimos una muy alta probabilidad de ganar: $(1 - \frac{2}{37})100 = 94,6\%$, aunque también un arriesgado capital a perder. Se recomienda solo jugar con esta estrategia un máximo de 3 tiradas y luego retirarse. Como dato interesante, Gerardo fue betado de muchos casinos reconocidos y de hecho ha tenido serios problemas con algunos otros.

3.6. Estrategia de santE (Original)

Hablamos con un jugador real de ruleta y le pedimos que describa su estrategia original en este juego. Básicamente es apostar a 2 de 3 columnas posibles, y las decisiones de aumentar o disminuir apuesta son similares a la Martingala, con 2 excepciones: tener en cuenta que acá estamos haciendo 2 apuestas por turno en vez de 1, y que si todavía queda un capital considerable, en vez de doblar apuesta, la triplicamos. El jugador menciona también que espera (sin jugar) a que la ruleta tire 2 números consecutivos pertenecientes a una **misma** columna, y luego hace sus apuestas a **las dos restantes distintas** a la que recién ha salido.

Sin embargo esa pausa no ha sido simulada y se simplificó a apostar siempre a la misma columna; así como también se ignoró la posible decisión de que de vez en cuando en vez de doblar la apuesta esta se triplique.

El fuerte de esta estrategia radica en que, a costa de menos ganancia, las probabilidades de perder son aún menores que en la Martingala.

Con una proporción apuesta:capital de 1:1000, podemos estudiar el decrecimiento de las probabilidades de perder, como hicimos con Martingala; donde cada constante y variable representa lo mismo que las ecuaciones de esa estrategia:

$$P_0 = (0,3... + \frac{1}{37}) \approx 0,36036 \approx 36,036\% \tag{10}$$

$$m \leq log_2(1000) \implies m \leq 9,96 \implies max(m) = 9 \tag{11}$$

$$P_{m+1} = (P_0)^{m+1} = 0,36036^{10} = 0,000037 = 0,0037\% \tag{12}$$

Es decir, con una proporción 1:1000, tendríamos que perder 10 veces seguidas ya que $k2^{10} > k1000$ y quedarnos sin capital, con una probabilidad de solo 0,0037% de que esto suceda. Más de 99,9963% de que ninguna racha de derrotas logre dejarnos sin capital antes de recuperar.

4. Estrucura de la Simulación

4.1. Metodología: Librerías y Módulos Utilizados

Numpy [11] Esta librería es muy completa. Primero, la utilización de *Arrays y Matrices* en vez de *Listas*, pues estas últimas son objetos, y con los arrays de Numpy se consigue una mayor eficiencia (es recomendable utilizar arrays y matrices de tamaño fijo, pues como no se almacena de forma contigua espacio extra con Numpy, tamaños variables destruyen esta eficiencia mayor pues aumentar el tamaño sería volver a construirlo).

Numpy Tiene un módulo llamado random (Numpy.Random) que se utilizó para conseguir los números pseudoaleatorios de una forma rápida, y si se necesita, se puede generar un Array de N longitud con todos valores aleatorios.

Por último, también tiene un método llamado *arange* que es como el *for* de siempre, pero con algunos beneficios como que su *Step* puede ser un float, o que se puede generar un array con él.

Pyplot [12] Este módulo de la librería **matplotlib** es ideal para graficar toda la información calculada de una forma simple: unas pocas configuraciones y un array o lista a graficar. Su sintaxis es similar a usar **Matlab**. Los gráficos que genera son muy personalizables y rápidos de generarse. También posee la posibilidad de hacer *Subplots*, es decir más de una gráfica en simultáneo, y plottearlas a todas juntas en una misma ventana. Se utilizó para todas las gráficas de la simulación.

4.2. Método de Resolución Aplicado

El funcionamiento del programa sigue un flujo simple pero efectivo: le damos al usuario la posibilidad de elegir entre las siete estrategias propuestas, con la posibilidad de retirarse en cualquier momento, estando ésto dentro de un bucle que sólo finalizará si el usuario así lo desea. Como pequeño detalle, también agregamos la opción de "no jugar", entre las estrategias. Una vez es elegida la opción, a partir de un array con los nombres de las estrategias, tomamos aquella que el usuario deseaba simular, y procederemos a configurar todo lo que esta requiere.

Inicialmente le pedimos determinados parámetros al usuario que condicionarán los resultados de la simulación, tales como la cantidad de 'tiradas' de la ruleta, el capital inicial con el que cuenta y el valor de su apuesta, la cual se mantendrá por el resto del juego. Luego llamaremos al método correspondiente a la estrategia que eligió el jugador, el cual nos devolverá dos listas que contienen los resultados obtenidos al aplicar la misma en la cantidad N de juegos que él mismo decidió. Hemos dicho que realizaríamos la simulación para dos casos: capital limitado, y capital hipotéticamente infinito. Estas listas contendrán dichos resultados. Cabe resaltar que tanto la ruleta con los números como la estrategia a utilizar, son objetos con su correspondiente clase; pues utilizamos un enfoque en el marco de la programación orientada a objetos.

Finalmente llamamos a nuestro módulo graphing que se encargará de definir, preparar y luego graficar las listas que le hemos pasado. Estas gráficas son almacenadas luego de graficarse, a menos que el usuario indique en el programa inicial que no desea que se guarden, poniendo el *modo save* en "false".

El resultado final serán dos gráficas que a su vez contendrán cuatro subgráficas (dando un total de 8 gráficas). Para entender mejor, concretamente estaríamos hablando de:

- **1. Un juego simple** de N 'tiradas' para un capital inicial y con un valor de apuesta igual a betValue
  - Capital limitado o finito
    - Flujo de la caja (1.1)
    - Frecuencia Relativa (1.2)
  - Capital ilimitado o infinito
    - Flujo de la caja (1.3)
    - Frecuencia Relativa (1.4)
- **2. El promedio de realizar múltiples juegos simultáneos** de N 'tiradas' cada uno para un capital inicial y con un valor de apuesta igual a betValue
  - Capital limitado o finito
    - Flujo de la caja (2.1)
    - Frecuencia Relativa (2.2)
  - Capital ilimitado o infinito
    - Flujo de la caja (2.3)
    - Frecuencia Relativa (2.4)

5. Gráficas Obtenidas

5.1. Resultados Estrategia D'Alembert

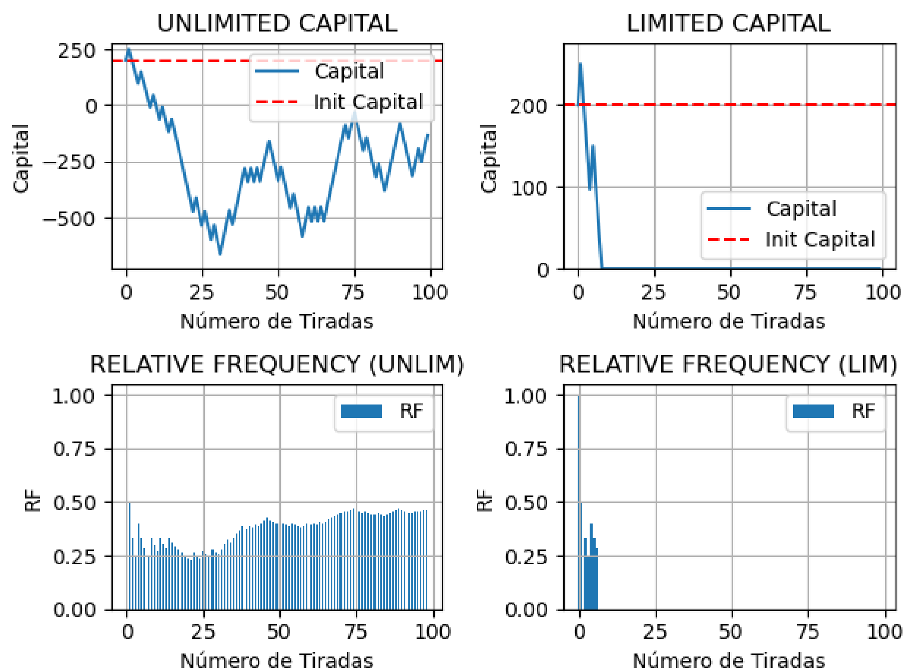


Figura 1: D'Alembert con N=100, capital inicial=200, tamaño de apuesta inicial=20 — Un solo resultado de juego.

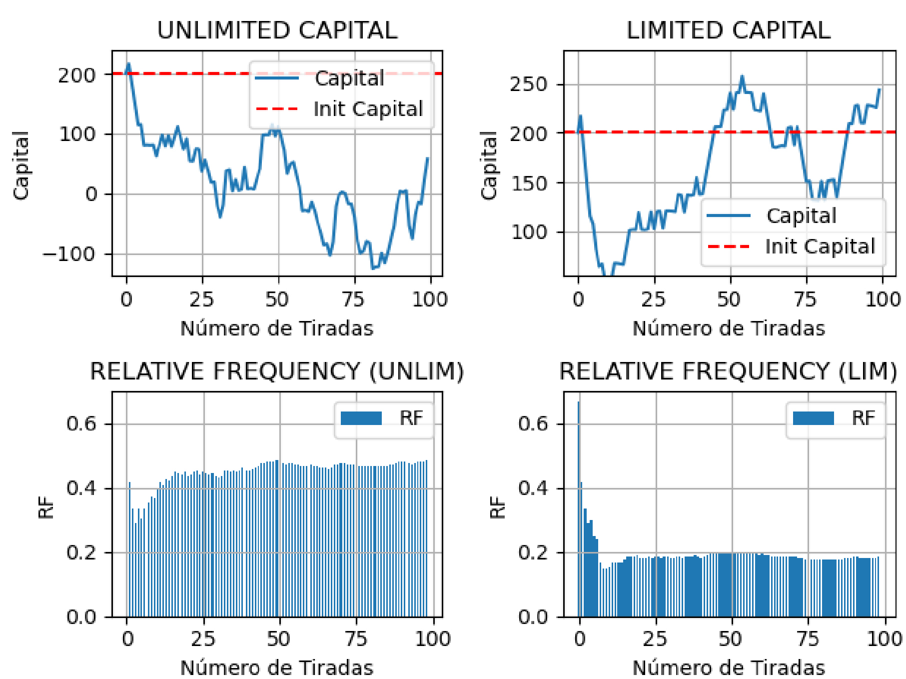


Figura 2: D'Alembert con N=100, capital inicial=200, tamaño de apuesta inicial=20 — Promedio entre 6 resultados de juego.

5.2. Resultados Estrategia Fibonacci

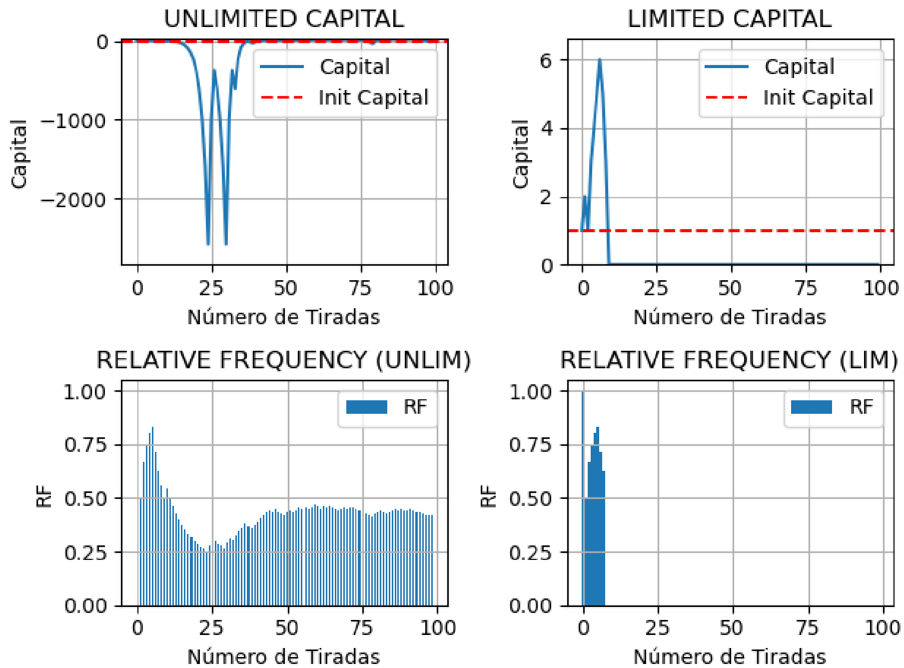


Figura 3: Fibonacci con N=100, capital inicial=1, tamaño de apuesta inicial=1 — Un solo resultado de juego.

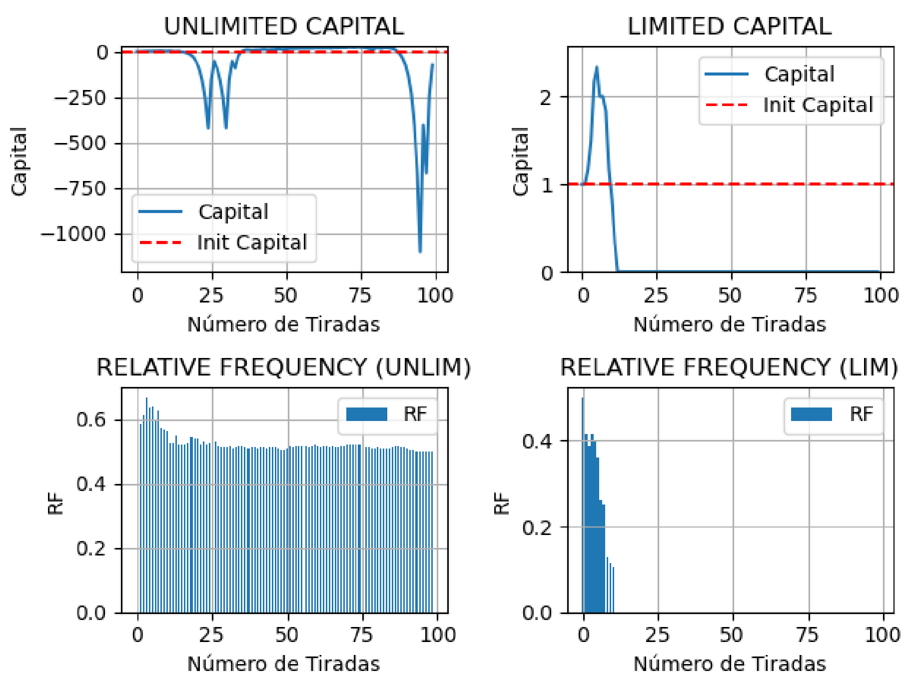


Figura 4: Fibonacci con N=100, capital inicial=1, tamaño de apuesta inicial=1 — Promedio entre 6 resultados de juego.

5.3. Resultados Estrategia Sofovich

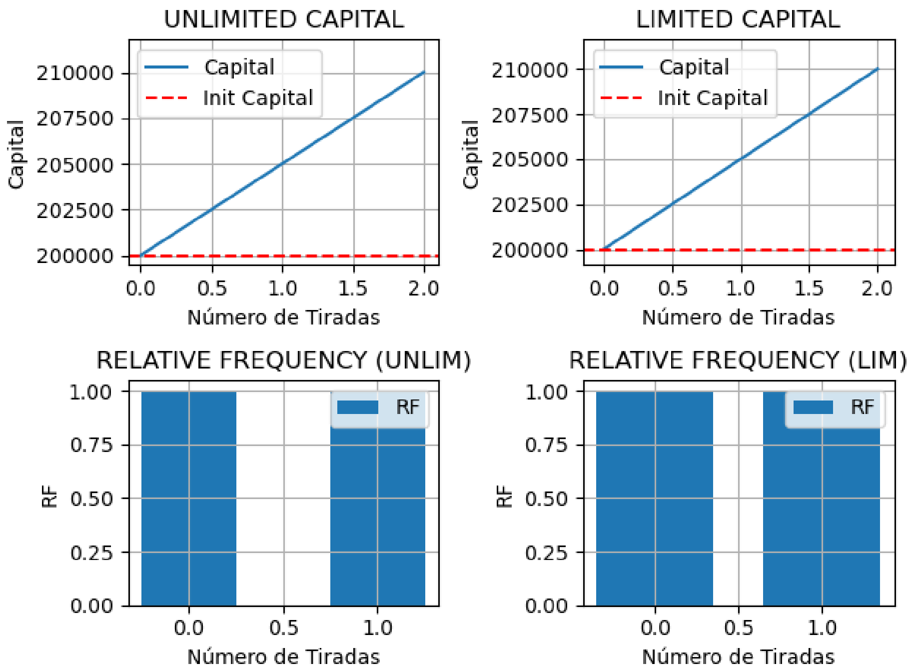


Figura 5: Sofovich con N=3, capital inicial=200000, tamaño de apuesta inicial=5000 — Un solo resultado de juego.

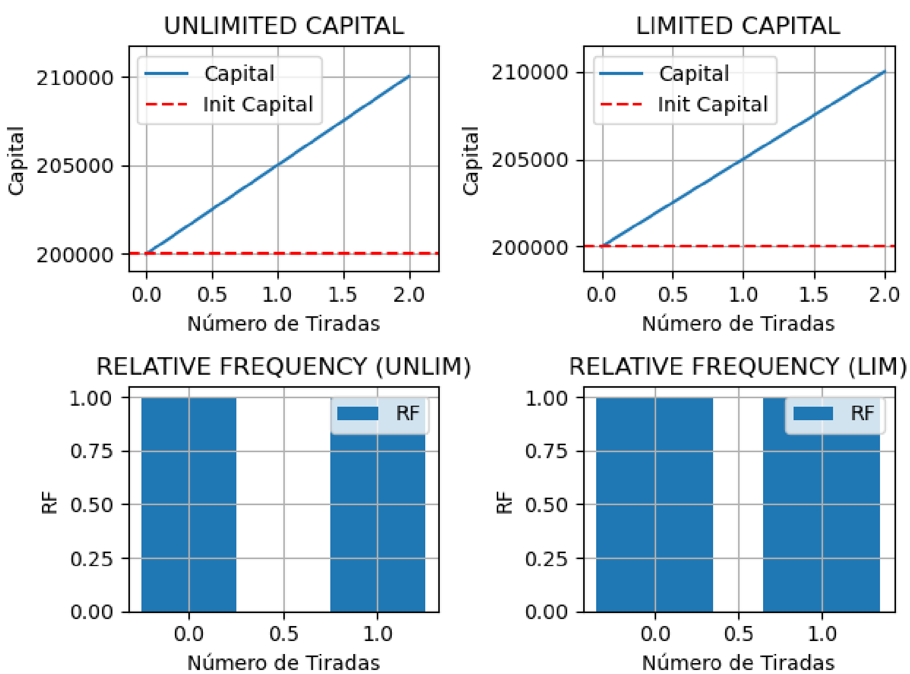


Figura 6: Sofovich con N=3, capital inicial=200000, tamaño de apuesta inicial=5000 — Promedio entre 6 resultados de juego.

5.4. Resultados Estrategia SantE (Original)

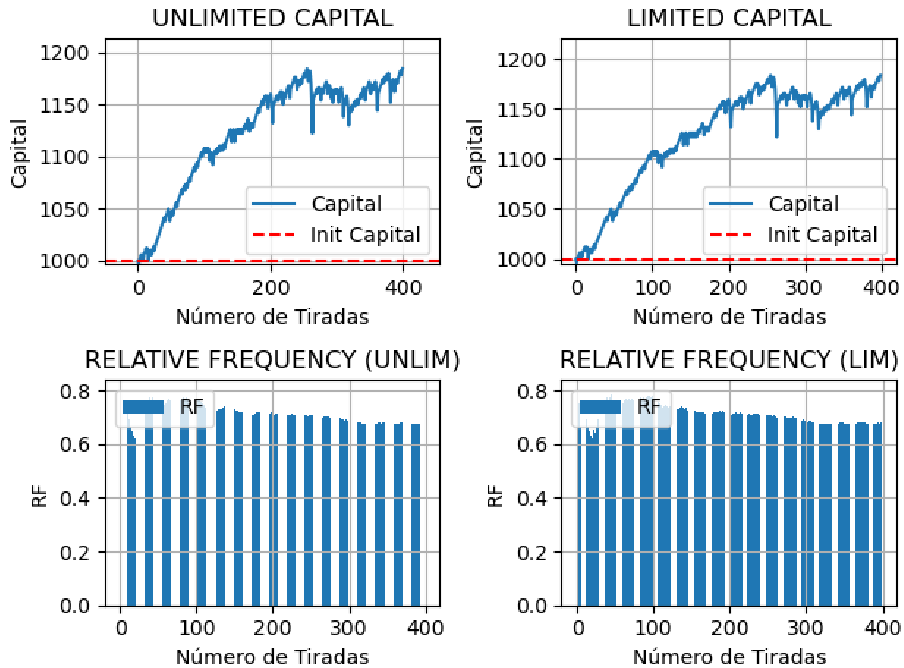


Figura 7: SantE con N=400, capital inicial=1000, tamaño de apuesta inicial=2 — Un solo resultado de juego.

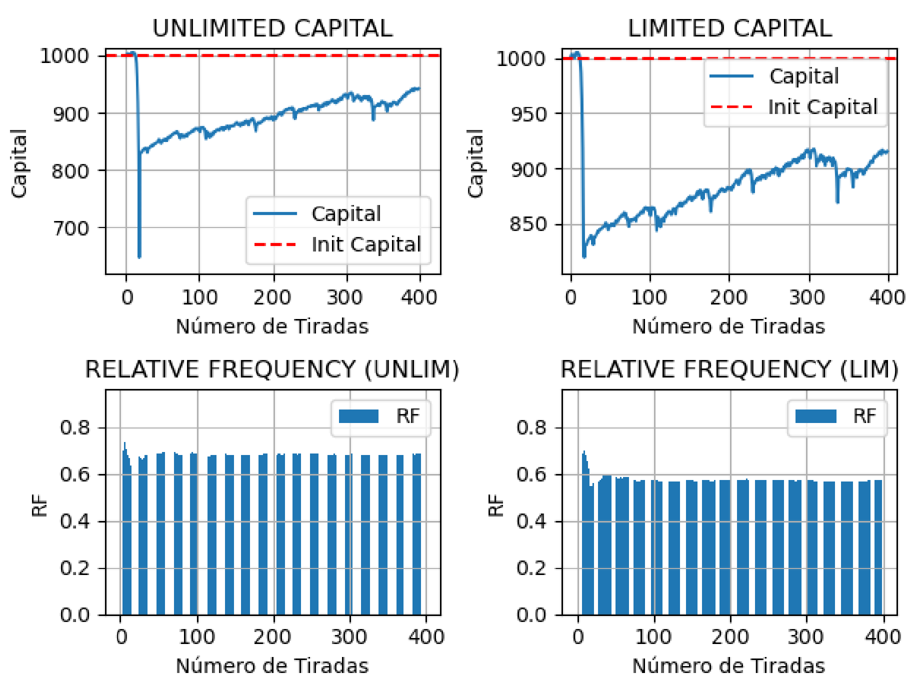


Figura 8: SantE con N=400, capital inicial=1000, tamaño de apuesta inicial=2 — Promedio entre 6 resultados de juego.

5.5. Resultados Estrategia Martingala Cásica, Factor 1:20 (Bajo)

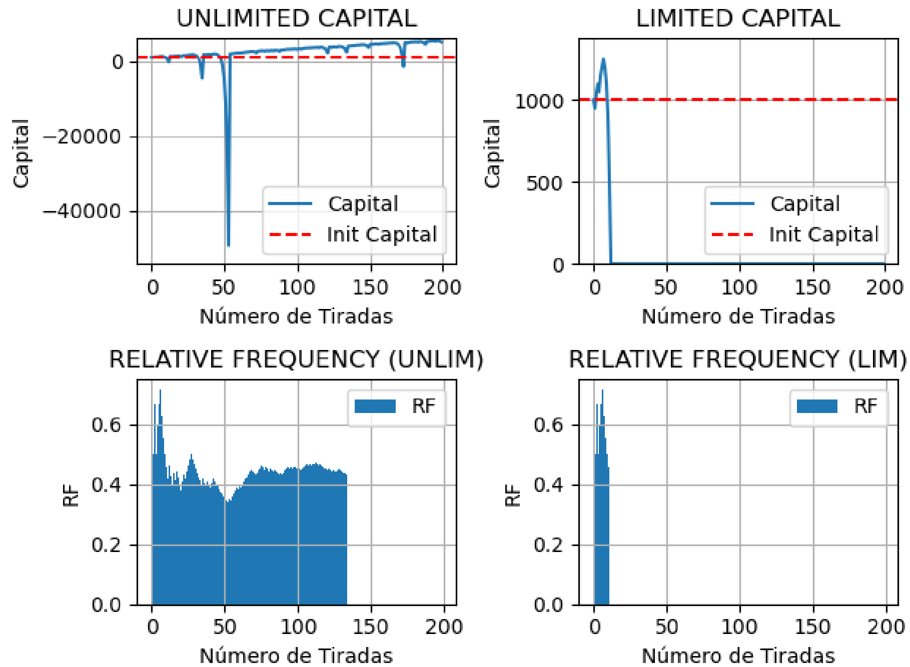


Figura 9: Martingala Clásica con N=200, capital inicial=1000, tamaño de apuesta inicial=50, factor apuesta:capital de 1:20 (bajo) — Un solo resultado de juego.

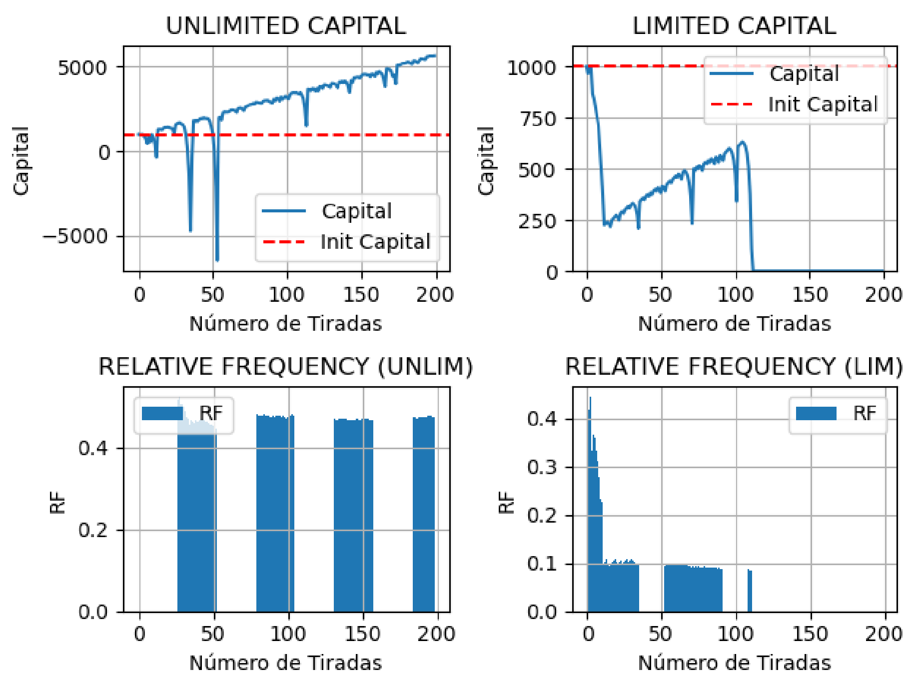


Figura 10: Martingala Clásica con N=200, capital inicial=1000, tamaño de apuesta inicial=50, factor apuesta:capital de 1:20 (bajo) — Promedio entre 6 resultados de juego.

5.6. Resultados Estrategia Martingala Cásica, Factor 1:1000 (Alto)

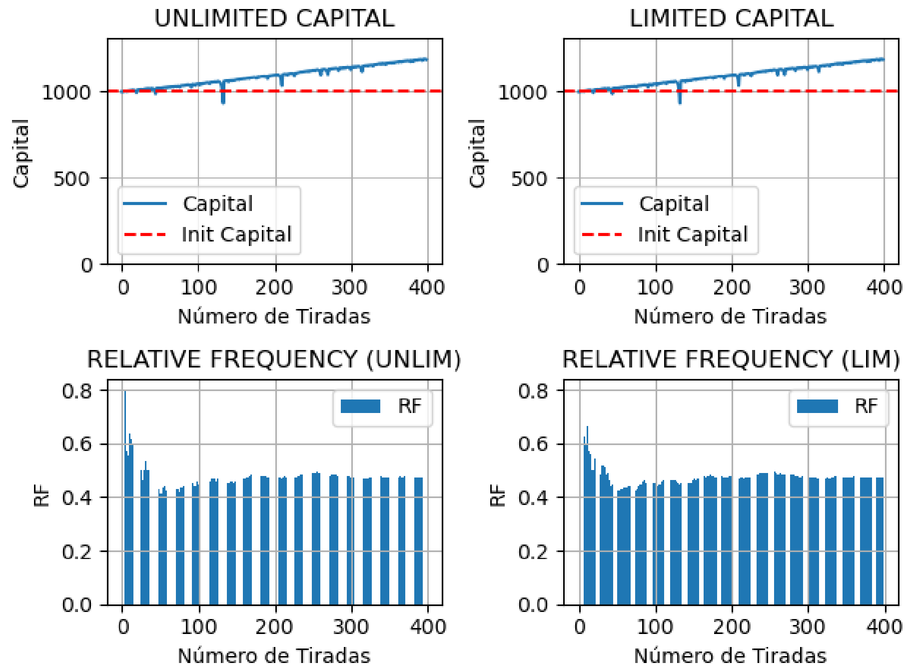


Figura 11: Martingala Clásica con N=400, capital inicial=1000, tamaño de apuesta inicial=1, factor apuesta:capital de 1:1000 (alto) — Un solo resultado de juego.

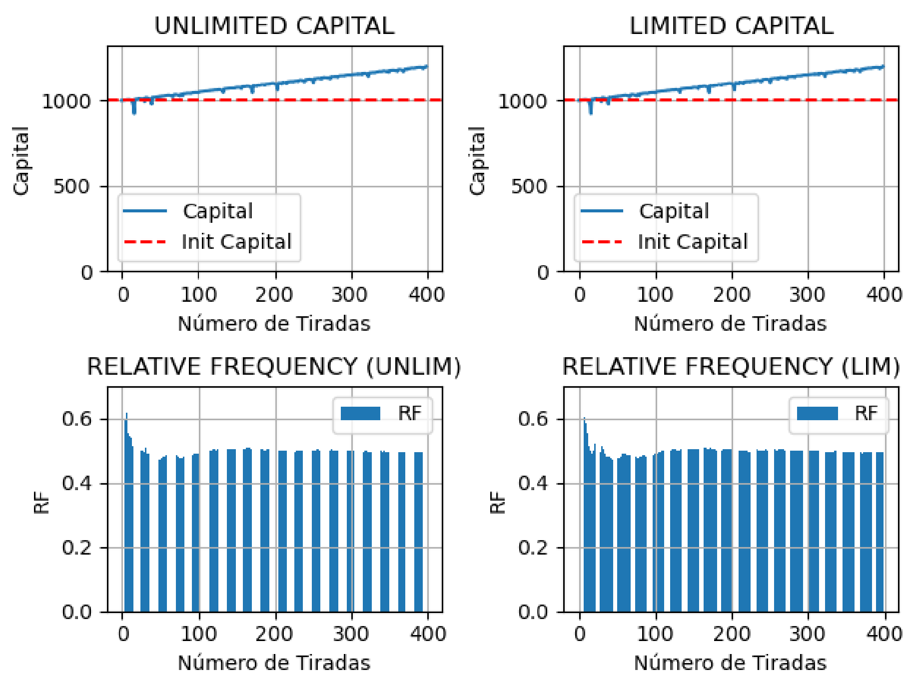


Figura 12: Martingala Clásica con N=400, capital inicial=1000, tamaño de apuesta inicial=1, factor apuesta:capital de 1:1000 (alto) — Promedio entre 6 resultados de juego.

5.7. Resultados Estrategia Martingala Modificada, Factor 1:1000 (Alto)

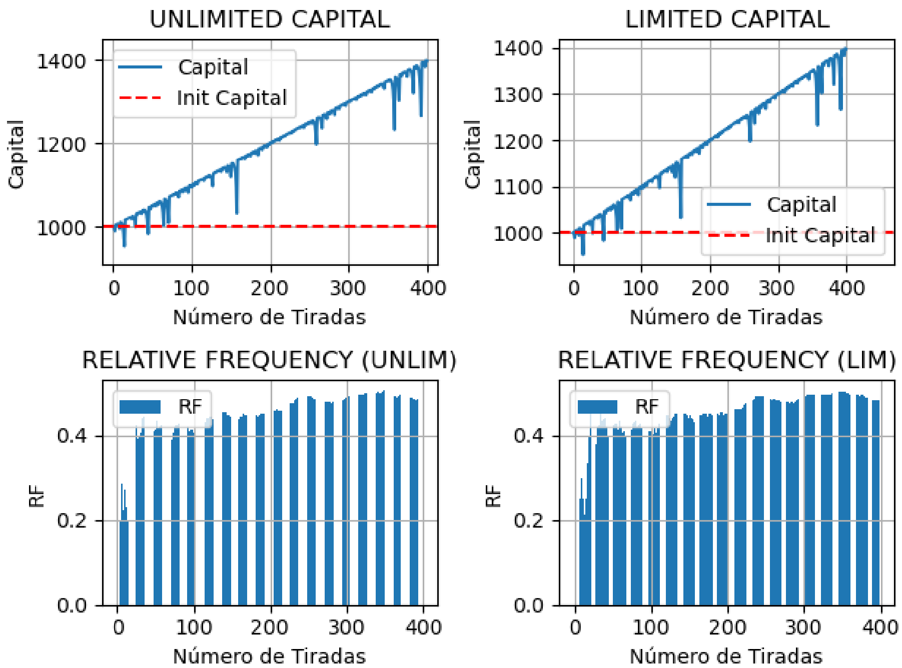


Figura 13: Martingala Modificada con N=400, capital inicial=1000, tamaño de apuesta inicial=1, factor apuesta:capital de 1:1000 (alto) — Un solo resultado de juego.

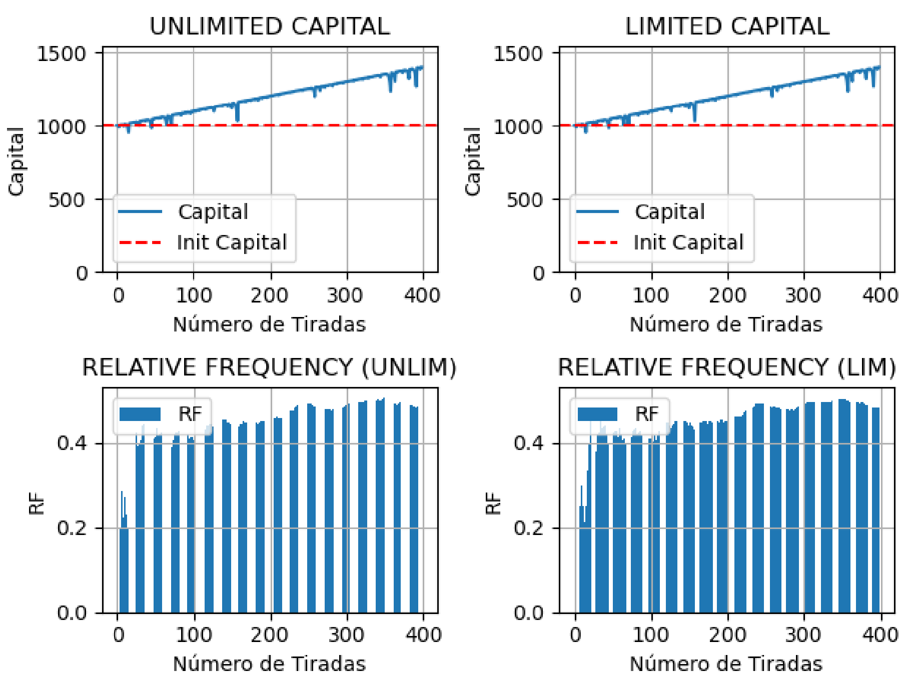


Figura 14: Martingala Modificada con N=400, capital inicial=1000, tamaño de apuesta inicial=1, factor apuesta:capital de 1:1000 (alto) — Promedio entre 6 resultados de juego.

6. Conclusiones

Podemos ver fácilmente que apostar a un número simple(pleno) o bien a un color no tiene buena rentabilidad. Esto ocurre principalmente porque, por ejemplo para el color, no tenemos el 50 % de probabilidades pues el número cero (0) queda fuera tanto del rojo como del negro, y a la larga, esto nos jugaría en contra.

Sin embargo, podemos apostar por una estrategia más completa o más sofisticada variando la cantidad apostada en cada tirada o bien la forma en sí de la apuesta, para ello emplearemos alguna de las estrategias explicadas anteriormente para así apreciar resultados más interesantes.

Comenzando con la famosa estrategia que utilizaba **Gerardo Sofovich**, sabemos que tenemos una muy alta probabilidad de ganar, pero no una muy alta ganancia. Al ser la apuesta alta en valor neto de por sí, podríamos decir que esta estrategia sirve para emplearla en un N considerablemente chico, es decir que el tiempo de juego sería corto, y llevarnos un buen pozo de ganancia (ejemplo: apostar $10.000 a 35 de los 37 números de la ruleta solamente 2 veces, es decir con nuestro N=2, considerando ambas tiradas como victoria, estaríamos apostando un total de $350.000, pero nos habríamos llevado en cuestion de pocos minutos unos $20.000). No nos convendría en un N grande por el simple hecho de que se arriesga un monto muy alto en capital y de perder, todo queda para la casa.

Con la estrategia de **D'Alembert**, se tienen serios problemas con respecto a las probabilidades de ganar, pues no siempre se esperan buenos resultados. Aunque la ventaja que tiene es que si sale mal, las pérdidas no son totales.

La de **Fibonacci** nos parece una curiosidad meramente matemática, y bastante moderada en el crecimiento de la apuesta. Es más que nada subjetivo utilizarla en vez de otras como Martingala.

Para la estrategia de **SantE**, como se mencionó, se entrevistó a una persona que tiene una *estrategia original* para jugar a la ruleta en la realidad, y si bien mejora a la Martingala en riesgo a apostar, las ganancias son menores. Se la puede considerar una estrategia más segura y que requiere mucho más paciencia, ideal para la mayoría de los jugadores que no les interesa ganar mucho dinero, sino solamente ir a la casa de apuestas como forma de entretenimiento.

En la estrategia **Martingala** (o su versión **modificada**), si bien observamos un avance lento, el capital va en aumento constante con unas probabilidades muy bajas de perderlo todo como hemos calculado en la sección de estrategias (con una proporción apuesta:capital de 1:100 era solo de 1,13 %, y con 1:1000 de 0,17 %).

Por lo que con un factor 1:1000 y con N = 100 aproximadamente en una martingala modificada lo más probable es percibir una ganancia del 10 % del capital inicial. Esto resulta interesante.

Sin embargo, aunque la Martingala modificada parezca una verdadera solución factible para jugar al casino, es menester tener el cuenta otro recurso: **EL TIEMPO**. Investigando, encontramos que en un casino con un rango de 3-4 a 8-9 personas jugando a la vez, es decir, ni vacía ni muy conglomerada de gente, el tiempo que transcurre desde el momento en que todos los jugadores hacen sus apuestas, se tira la ruleta, ésta se frena y se pagan las apuestas, es en promedio **tres minutos**[7]. Esta información también la agregamos a la simulación para ver cuánto tiempo tardaría aproximadamente en la realidad realizar *todo el juego completo deseado*.

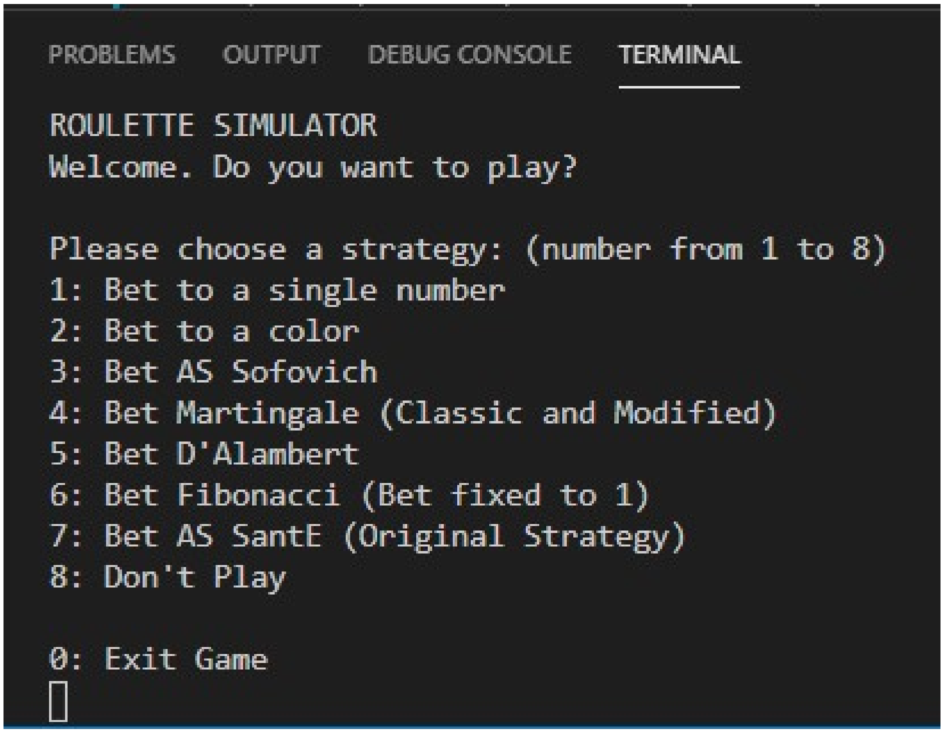
Esto hace que en el ejemplo que dimos con N=100, 100 x 3 = 300 minutos = 5 horas para aumentar un 10 % el capital, y arriesgando siempre 1:1000 como tope. Teniendo en cuenta esto, y que los casinos ponen topes de apuesta máxima y perjudica escalar esta etrategia a un capital inicial muy elevado a utilizar, hacen que esta estrategia sea muy poco rentable.

Aún reduciendo el tiempo de apuesta utilizando una ruleta electrónica o un sitio de apuestas virtual, las ganancias de Martingala no son realmente *considerables* a menos que pasen cantidades muy grandes de tiempo, como pueden ser horas y horas.

Desde un punto de vista ingenieril, podemos finalizar diciendo que ninguna estrategia es realmente rentable ya sea económica o temporalmente, o que de serlo, poseen un riesgo elevado que aumenta en el tiempo. Es factible utilizar algunas de ellas siendo completamente conscientes de que no nos haremos ricos, pues si bien podemos poner a nuestro favor las probabilidades en la casa de apuestas, esto lleva a un aumento del riesgo de alto capital perdido, o bien consumir muchísimo tiempo para tan poca ganancia neta.

Por supuesto, la estrategia de *no jugar* siempre estará disponible, siendo ésta la más recomendable.

7. Anexo: Salidas de pantalla



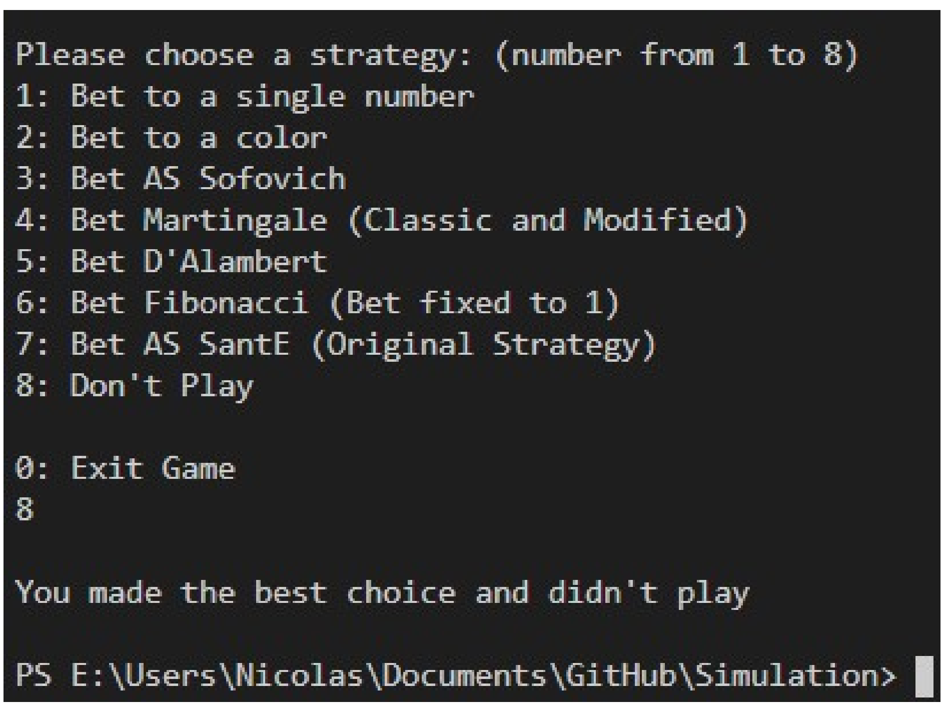
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

ROULETTE SIMULATOR
Welcome. Do you want to play?

Please choose a strategy: (number from 1 to 8)
1: Bet to a single number
2: Bet to a color
3: Bet AS Sofovich
4: Bet Martingale (Classic and Modified)
5: Bet D'Alambert
6: Bet Fibonacci (Bet fixed to 1)
7: Bet AS SantE (Original Strategy)
8: Don't Play

0: Exit Game
```

Figura 15: Menú principal de la ruleta simulada.



```
Please choose a strategy: (number from 1 to 8)
1: Bet to a single number
2: Bet to a color
3: Bet AS Sofovich
4: Bet Martingale (Classic and Modified)
5: Bet D'Alambert
6: Bet Fibonacci (Bet fixed to 1)
7: Bet AS SantE (Original Strategy)
8: Don't Play

0: Exit Game
8

You made the best choice and didn't play

PS E:\Users\Nicolas\Documents\GitHub\Simulation>
```

Figura 16: Si la decisión fue no jugar[8].

```
4

How many games do you want to play?: 200
Define your initial capital: 1000
Bet Value?: 50
Please choose a color ("red" or "black"): red
There are two types of Martingale. Classic and Modified. The default is Classic. Do you want to change to Modified? (Y/N): n

Final capital:  5050
Your total play time would be about: 600 min

Final capital:  5850
Your total play time would be about: 600 min

Final capital:  5600
Your total play time would be about: 600 min

Final capital:  5950
Your total play time would be about: 600 min

Final capital:  5500
Your total play time would be about: 600 min

Final capital:  6000
Your total play time would be about: 600 min

graph_martingale_classic_200iter_1000cap_50bet_one_result.png guardado correctamente

graph_martingale_classic_200iter_1000cap_50bet_all_results_average.png guardado correctamente
```

Figura 17: Ejemplo de configuración de una estrategia: Martingala, y el tiempo aproximado que requeriría realizar cada jugada en la realidad.

Referencias

[1] Wikipedia ES. Definición de Ruleta.
https://es.wikipedia.org/wiki/Ruleta

[2] Raúl Katz - Pablo Sabatinelli (2018). Probabilidad y Estadística: Variables Aleatorias Discretas y algunas Distribuciones de Probabilidad

[3] Wikipedia ES. Sucesión de Fibonacci.
https://es.wikipedia.org/wiki/Sucesión_de_Fibonacci

[4] Casino 888. Estrategia Martingala.
https://www.888casino.es/blog/el-sistema-de-la-martingala-la-historia-de-un-sistema-de-apuestas-pop

[5] Casino News. Estrategia D'Alembert.
https://www.casinonewsdaily.es/guia-la-ruleta/el-sistema-dalembert-de-la-ruleta/

[6] Betsson. Estrategia Fibonacci.
https://www.betsson.es/blog/tutoriales/casino/como-funciona-la-estrategia-fibonacci/

[7] Quora. Tiempo entre jugadas en la ruleta.
https://www.quora.com/How-much-time-do-I-place-my-bets-on-a-roulette

[8] Mr. Vitto. Pensar en no jugar como una estrategia.

[9] Python Doc. Python 3.8.2 | Documentación Oficial.
https://docs.python.org/3/

[10] Microsoft (VSC). Visual Studio Code Official Webpage
https://code.visualstudio.com/

[11] Numpy Doc. Numpy 1.18 | Documentación Oficial.
https://numpy.org/doc/1.18/

[12] Pyplot Doc. Pyplot 3.1.1 | Documentación Oficial.
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.html