

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

ESTUDIO COMPARATIVO DEL DESARROLLO ESTADÍSTICO DE UN JUEGO DE MINORÍA Y LA CONTRAPARTE TEÓRICA Y SIMULADA DEL PLANTEAMIENTO CUÁNTICO DEL PROBLEMA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
FÍSICO**

PROYECTO DE INVESTIGACIÓN

JORGE LUIS ENRÍQUEZ MAYORGA

jorgenriquez309@hotmail.com

DIRECTOR: CESAR COSTA VERA, Ph. D.

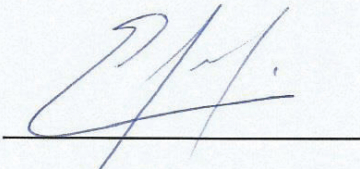
cesar.costa@epn.edu.ec

Quito, junio 2017

DECLARACIÓN

Yo Jorge Luis Enríquez Mayorga, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

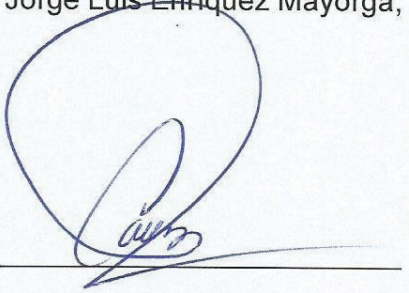
A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Jorge Luis Enríquez Mayorga

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Jorge Luis Enríquez Mayorga, bajo mi supervisión.

A handwritten signature in blue ink, consisting of a large, stylized 'C' followed by 'Vera', written over a horizontal line.

César Costa Vera, Ph. D.

DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco a Dios por permitirme vivir en el tiempo adecuado cada etapa de mi vida, por haber programado nuestro universo de tal manera que cada suceso sea digno de ser estudiado brindándonos la emoción al encontrar la explicación de las cosas.

A Geova por la compañía, apoyo e impulso incondicional, por estar en cada momento mostrando su preocupación para que logremos terminar este paso, siendo un gran apoyo en mi vida y sabiendo cómo llenarla en cada instante.

Al Doc Cesar que me apoyó con su paciencia, conocimiento y disponibilidad, quien además durante mi vida universitaria fue un gran ejemplo de profesional y profesor.

A mi padre que ahora me mira desde el cielo, pero que con sus lecciones de vida aún me guía y protege cada día.

Gracias a Katy y Ely por estar ahí en nuestras vidas, velando y cuidando por nosotros.

A los familiares y amigos que continuamente me han acompañado y motivado a terminar este trabajo.

DEDICATORIA

A mi madre quien de manera especial ha sido el cimiento para construir mi vida profesional, quien sembró en mí las bases de responsabilidad y deseos de superación. Y con este paso pueda ver cómo han crecido esos valores y sigamos cosechando juntos nuevos logros.

A mi hermano Carlos Bolívar, quien ha sabido apoyarme en toda mi vida estudiantil desde muy pequeño llevándome a la escuela de la mano, hasta ahora ser mi ejemplo para seguir estudiando. Y en quién conjuntamente con el recuerdo del nombre de mi padre al desarrollar este trabajo he dedicado un cambio al nombre de los actores tradicionales de la literatura.

A mi hija quién ha sido el motor de mi vida, que lo tome como un ejemplo de que con esfuerzo, constancia y trabajo todo se puede alcanzar.

CONTENIDO

Índice de Figuras	i
Índice de Tablas	ii
Declaración de Autoría	iii
Introducción	1
1. CAPÍTULO I: Fundamentos de la Teoría de Juegos	5
1.1. Conceptos Básicos de la Teoría de Juegos	5
1.2. Ejemplos de Juegos Clásicos	8
1.2.1. “Cara o Sello” (Juego Simétrico 2x2)	8
1.2.2. “Piedra, Papel o Tijera”	9
1.2.3. “El dilema del prisionero”	10
1.2.4. Resumen de las características de los juegos clásicos:	11
1.3. Juegos de Minoría	11
1.3.1. Versión Analítica del Juego:	13
2. CAPÍTULO II: Cuantización de la Teoría de Juegos	16
2.1. Elementos de Mecánica Cuántica Para la Teoría de Juegos	16
2.1.1. Matrices de Pauli	18
2.1.2. Rotaciones en el Sistema de Espín del Electrón	19
2.2. Definición de Juegos Cuánticos	21
2.2.1. “Cara o Sello Cuántico”.	21
2.2.2. Descripción Detallada del Juego.	21
2.2.3. Esquema Utilizado para Cuantizar el Problema.	23
3. CAPÍTULO III: Simulación Computacional	26
3.1. Esquema General de una Simulación Computacional	26
3.2. Simulación de Monte Carlo y ventajas del uso de SAS®	29
3.2.1. Generación un Número Pseudo-Aleatorio	30

3.3.	Simulación de Monte Carlo	31
3.3.1.	Ejemplo de una Simulación de Monte Carlo:	31
3.3.2.	Simulación de Monte Carlo con Variables Discretas:	34
3.3.3.	Generación de Números Aleatorios Provenientes de Otras Distribuciones.....	36
4.	CAPÍTULO IV. Resolución de un juego de minorías	37
4.1.	Modelación Algebraica de un Juego Cuántico	37
4.2.	Simulación del Juego Clásico	40
4.3.	Aproximación Analítica	44
4.4.	Simulación del juego cuántico	48
5.	CAPÍTULO V: Conclusiones	56
	Referencias bibliográficas	57
	Anexos	63

ÍNDICE DE FIGURAS

Figura 1.- Esquema de un juego con estrategias y pagos asociados al jugador....	5
Figura 2.- Representación en notación de Dirac del spin de un electrón.	17
Figura 3.- Esquema de solución al juego de cara o sello clásico con 3 momentos de cambio.....	22
Figura 4.- Etapas para implementar una simulación.	28
Figura 5.- Ejemplo de tabla de frecuencias acumuladas.....	32
Figura 6.- Gráfico de frecuencias absolutas (probabilidades) del Ejemplo 3.3.1..	33
Figura 7.- Promedio de la variable número de consultas por día en 100 (a) y 100.000 repeticiones (b).....	34
Figura 8.- Distribución de frecuencias para la simulación de Monte Carlos con variables discretas.....	34
Figura 9.- Valores estadísticos obtenidos para diferentes números de tarjetas. (a) 100, (b) 150, (c) 200, (d) 250, (e) 300.	35
Figura 10.- Resultado comparativo de las simulaciones realizadas para el ejemplo de tarjetas.....	35
Figura 11.- Varianza de los jugadores que escogieron asistir (Opción “1”)......	41
Figura 12.- Varianza reudcida σ^2N vs número de historias por jugador $2mN$	41
Figura 13.- Gráfico de la varianza distribuido por regiones.	42
Figura 14.- Evolución de los resultados de la simulación de un juego de minorías con estrategias puras. (a) Aleatoria, (b) Insistente, (c) Determinística, (d) Aleatoria Cuántica, (e) Aleatoria Cuántica, y (f) Determinística Cuántica.....	54

ÍNDICE DE TABLAS

Tabla 1.- Matriz de Pago 2x2 del juego simétrico de suma cero y de información completa llamado “Cara o Sello”	9
Tabla 2.- Matriz de Pago 3x3 en un juego simétrico y de información completa “Piedra, Papel o Tijera”	9
Tabla 3.- Matriz de Pago 2x2 en un juego simétrico de información completa “Dilema del prisionero”	10
Tabla 4.- Resumen de los conceptos presentados en los ejemplos de juegos clásicos.	11
Tabla 5.- Ejemplo de estrategias típicas de un participante del juego de minorías	14
Tabla 6.- Función de Pago juego Cara o Sello cuantizado asumiendo que no se realiza el movimiento en el momento 4 por parte del jugador B.	24
Tabla 7.- Distribuciones y función de densidad de probabilidad más comunes.	36
Tabla 8.- Posibles historias de un jugador en el juego de minorías cuántico.	37
Tabla 9.- Posibles estados finales del sistema y su espin asociado.	39
Tabla 10.- Efecto de los Intrusos determinísticos cuánticos y en el sistema determinístico clásico.	55

DECLARACIÓN DE AUTORÍA

El trabajo que se presenta ha sido desarrollado en su gran mayoría por Jorge Luis Enríquez. De manera detallada los aportes realizados se citan a continuación:

El desarrollo del modelo teórico basado en las matrices de rotación de Pauli vistas como posibles estrategias de los jugadores del sistema en un juego cuántico tomando como espín de un electrón a similitud de una “moneda cuántica” fue adaptado y conceptualizado.

El planteamiento y programación en lenguaje SAS del esquema de juego de minorías tradicional, cuántico y la posibilidad de introducir un estado mixto para verificar la evolución del mismo con los distintos tipos de jugadores, así como la definición de cada uno de estos tipos de jugadores incluidos en la simulación final.

Las ejecuciones de la simulación fueron desarrolladas y ejecutadas en un ambiente de SAS Enterprise Guide 7.1 *OnDemand for Academics*, bajo licencia de uso educativo para un usuario de responsabilidad del estudiante y el código utilizado se lo detalla en los anexos para que pueda ser reproducido.

El marco conceptual inicial fue desarrollado en la materia proyecto bajo la revisión del Doctor César Costa, y la revisión de este documento se realizó con su apoyo y acompañamiento.

El detalle escrito fue realizado en su totalidad por el estudiante tomando conceptos y referencias debidamente citados en la bibliografía. Los gráficos obtenidos son en su totalidad con datos generados y obtenidos por parte del estudiante.

INTRODUCCIÓN

Mediante la teoría de juegos podemos estudiar de una manera formal y abstracta las decisiones que pueden tomar diversos intervinientes en un conflicto interpretado como “Juego” [1] [2] [3]. Esta definición se realizará mediante modelos matemáticos que describan el conflicto, la cooperación y la posible competencia entre entes inteligentes que son quienes toman decisiones también llamados “Jugadores” estas definiciones se mantienen desde el planteamiento original de la teoría de juegos por Von Neumann [4]. Las decisiones que pueden ser tomadas serán las estrategias posibles de los jugadores. Finalmente en cada juego al terminarse el mismo se interpretará una función de pago que determinará la existencia o no de uno o varios ganadores y que será función de las estrategias adoptadas por cada uno de los jugadores.

Mediante el presente trabajo se espera incorporar a este esquema de Jugador-Estrategia-Función de Pago una mejora al momento de establecer mayores estrategias accesibles a los jugadores mediante la cuantización del juego la cual va a ser considerada a través de un modelo computacional que permita simular los resultados del juego, tanto cuando los jugadores tienen acceso a estrategias tradicionales como cuando los jugadores dispongan acceso a estrategias definidas como cuánticas, este campo fue explorado inicialmente con el apareamiento de los ordenadores para el juego de minorías [5] [6]. Desde entonces se han realizado varios trabajos aprovechando las mejoras tecnológicas [7] [8].

La teoría de juegos es capaz de ofrecer resultados de interés para varios tipos de ramas de las Ciencias Sociales y la Biología [9] [10] [11], así como técnicas para tomar decisiones efectivas en el campo de los negocios. Aunque propiamente la palabra “juego” podría tener connotaciones lúdicas o relativas al azar, con el uso de la teoría de juegos no estudiaremos para esta ocasión los juegos de salón, aunque sí están contemplados dentro del dominio de esta rama de la ciencia. De esta manera para evitar la confusión mencionada, la teoría de juegos también se la conoce como “análisis matemático de conflictos” o como “toma interactiva de decisiones” e incluso, existen algunas aproximaciones en estos campos incluyendo las versiones cuánticas de los juegos [12].

De manera general los jugadores los identificaremos como entes decidores que se consideran racionales, no necesariamente humanos, ya que se ha realizado estudios en los que tendencias o fenómenos biológicos se podrían ser explicados por medio de la Teoría de Juegos [9] y de fenómenos físicos que de igual manera incorporan este formulismo para determinar la evolución del sistema [13].

Como ejemplos típicos de juegos se puede citar por ejemplo conflictos militares, modelos de evolución biológica, campañas políticas, campañas publicitarias o de marketing, y otras situaciones que generan competencia entre empresas [4] [14].

El principio básico para hallar la solución de un juego de decisiones simultáneas, donde los jugadores poseen información completa e idéntica, es hallar el conjunto de estrategias que al variarse por uno o varios de los participantes el sistema no pueda recibir una mejor función de pago [15]. También es posible analizar juegos donde la toma de decisiones se da de forma dinámica, es decir como respuesta a la decisión tomada por el jugador anterior generando un análisis evolutivo del sistema en general [16].

Para poder estudiar de una manera más amplia el tema propuesto se utilizará en el presente elementos de simulación computacional que permitan observar de una manera más efectiva los resultados al variar el sistema [17] [18].

El desarrollo de los computadores y programas computacionales durante los últimos años ha permitido que se los aplique para la resolución de problemas científicos, por lo que se ha denominado a la simulación computacional también como una tercera metodología para la investigación científica [19]. Este método de carácter complementario o alternativo a los modos tradicionales de hacer ciencia (el experimental y el teórico) ha ejercido un fuerte impacto en casi todos los campos de la ciencia. El objetivo de la simulación computacional es resolver los modelos teóricos con su complejidad, mediante la resolución numérica de las ecuaciones involucradas, haciendo uso intensivo de computadoras. En el área de la física la simulación computacional ha sido introducida como una herramienta a partir de los años cincuenta [20]. Posterior a este trabajo pionero se realizaron varios estudios en mecánica estadística clásica, especialmente en el estudio de líquidos, mediante los cuales esta forma de hacer ciencia se fue incorporando en el estudio de otras áreas de la física como la cromodinámica cuántica, la física de fluidos, relatividad

general, física del plasma, materia condensada, física nuclear, ciencia de materiales [21] [22] [23].

En la actualidad podemos considerar que las computadoras duplican su velocidad aproximadamente cada dieciocho meses, por lo que la simulación computacional se puede convertir en una herramienta de cálculo esencial tanto para proyectos experimentales como para teóricos [17].

Mediante la utilización de un modelo computacional adecuado, no sólo se puede replicar experimentos realizados en un laboratorio, sino que además gracias a que se pueden variar libremente los parámetros utilizados, se podría realizar cambios al experimento que de otra manera podrían ser muy costosos, o que incluso nos permitirían identificar mejoras al diseño experimental. También permite solventar conflictos entre explicación teórica y la observación real [24] [22].

Los métodos más utilizados en la física en la actualidad son la dinámica molecular [25] que resuelve los problemas mediante un carácter determinista, y el método de Montecarlo [21] que es de carácter probabilístico. Los dos métodos consisten en generar configuraciones diferentes de un sistema de partículas (para objetivos de este estudio trataremos a los jugadores como partículas de un sistema sin perder la generalidad de las ventajas que supone el utilizar la simulación computacional), es decir como puntos en el espacio de fases compatibles con las condiciones externas.

Los dos métodos a los que se ha hecho referencia han sido empleados exitosamente en la simulación de gases, líquidos y sólidos [22], ampliando su uso y desarrollo de técnicas específicas de manera paralela al avance tecnológico de las computadoras. Los sistemas estudiados van desde cientos a miles y últimamente incluso a decenas de millones de átomos. Los aspectos estudiados incluyen propiedades estructurales, termodinámicas, mecánicas y cinéticas. Hay que tener en cuenta que las técnicas de simulación tienen aplicaciones mucho más amplias, por ejemplo el método de Monte Carlo que es el que se ha seleccionado para este estudio también se ha empleado con éxito en Física de partículas [21] [22].

En este trabajo se presenta una simulación estadística computacional de un juego de minorías que respeta las propiedades de la solución teórica del problema y que

permita realizar una comparación decisiva con una representación cuántica del juego, y se encuentra esquematizado con la valoración del juego de una manera clásica, las definiciones necesarias para poder considerar cuantizado al juego, con esto proceder a realizar la simulación computacional de los puntos de equilibrio de los 2 esquemas propuestos y finalmente verificamos la ventaja de utilizar una estrategia cuántica por los jugadores debido a que esto representa una función de pago más estable y beneficiosa para el sistema. Parte del proceso presentado incluye la investigación de las bases matemáticas de la teoría de juegos, con énfasis en el juego de minorías, el estudio de los parámetros susceptibles de ser cuantizados y las condiciones necesarias para lograr esto en un juego de minorías de igual manera se presentan los pasos generados para implementar simulaciones con muchos agentes y realizar estas simulaciones con el software adecuado usando métodos de Monte Carlo [26].

Este documento está estructurado de la siguiente manera:

En el Capítulo I se introducen los conceptos básicos de la teoría de juegos haciendo un énfasis especial en el juego de minorías explorando sus características y planteamientos generales.

En el Capítulo II se plantea los fundamentos de la mecánica cuántica que son necesarios para plantear los juegos cuánticos, en particular las matrices de rotación de Pauli cómo una posible estrategia para rotar el espín del electrón y su similitud con una moneda cuántica.

En el Capítulo III se detalla la metodología computacional que se siguió y las ventajas de utilizar SAS® para realizar la simulación.

Los resultados obtenidos se los señala en el capítulo IV, tanto la resolución matemática del juego de minorías, como la ejecución de la simulación clásica y cuántica del sistema.

Finalmente en el capítulo V se exponen las conclusiones halladas en este estudio.

1. CAPÍTULO I: FUNDAMENTOS DE LA TEORÍA DE JUEGOS

1.1. Conceptos Básicos de la Teoría de Juegos

En los siguientes capítulos se hará mención a varios conceptos tradicionales de la teoría de juegos que han sido compilados y adaptados de textos fundamentales sobre el tema [1] [3] [4] y que se detallan a continuación:

1.1.1. Conceptos Generales

Para mantener uniformidad en el desarrollo del trabajo se citan los diferentes conceptos y la notación que se utilizará durante este trabajo, la misma ha sido consolidada para evitar conflictos de nomenclatura.

Juego: Es la combinación de 3 conjuntos, determinados como Jugadores, Reglas (Estrategias) y Pagos. En el cual cada uno de los N jugadores puede adoptar una de las S_j Estrategias la cual genera un Pago asociado a ella.

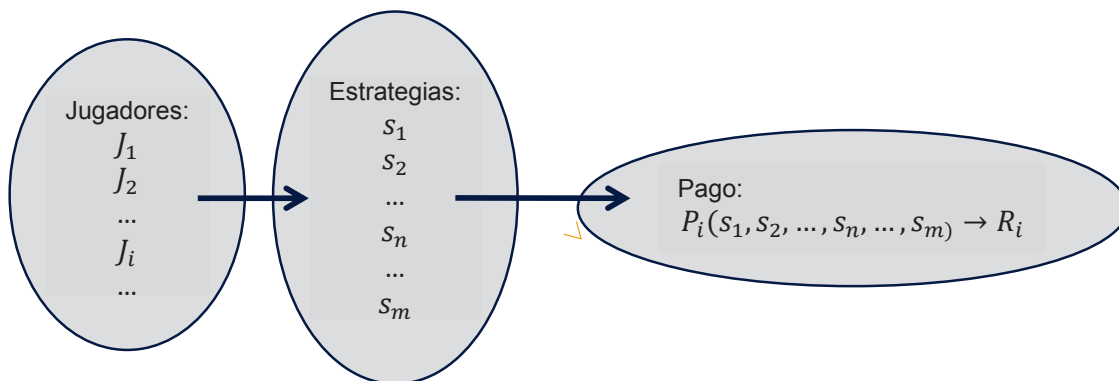


Figura 1.- Esquema de un juego con estrategias y pagos asociados al jugador.

Para cada conjunto de Jugadores J existe un conjunto de estrategias S disponible para el cual se asocia una función de pago P que genera un resultado R .

Pago o utilidad: Es una medida de la deseabilidad de una estrategia hacia los jugadores, lo que le llevará al mismo a definirse por una u otra situación. Existe un posible resultado final del juego asociado a cada conjunto de estrategias obtenidos por los jugadores participantes.

Acción o Movimiento: Se define como la posibilidad de un jugador de escoger o cambiar su estrategia.

Estrategia: Es la acción que toma el jugador dependiendo de la situación presentada en el juego. En esta convergen todas las acciones o movimientos elaborados por el jugador durante el juego.

Evolución Temporal: Se entiende por evolución temporal a la secuencia de estados del sistema basado en la repetición previa del juego o a realizar referencias a los resultados anteriores del juego.

1.1.2. Tipos de estrategia

El conjunto de estrategias que puede ser adoptado por un jugador tiene los siguientes tipos que difieren principalmente en las diferentes acciones y movimientos que puede tomar un jugador y en los criterios que va a tomar para cambiar o mantener su decisión:

Estrategia pura: Es un tipo de estrategia que se presenta cuando existe un solo movimiento dada una determinada situación del juego, generalmente se la denomina sólo estrategia.

Estrategia mixta: Es la estrategia que consiste cuando una o varias acciones de los jugadores son generadas de manera aleatoria.

Estrategia dominante: Se dice que una estrategia es dominante para un jugador cuando la misma presenta una mayor función de pago que cualquier otra, independientemente de las acciones tomadas por otros jugadores.

Se dice que: S_k es dominante para el jugador k sí:

$$\forall S_j, j \neq k, P_k(S_1, \dots, S_2, \dots, S_N) \geq P_k(S_1, \dots, S_{k'}, \dots, S_N), \quad \forall S_{k'} \quad (\text{Ec. 1})$$

Estrategia evolucionaria estable (EEE). Una estrategia S es estable evolucionaria ante otra estrategia S' sí $\forall \varepsilon > 0$ pequeño, S se comporta mejor que S' aplicando una estrategia mixta $(1-\varepsilon) S + \varepsilon S'$.

Una EEE es una estrategia que es estable evolucionaria con respecto a todas las otras posibles estrategias. En términos prácticos: una población que conoce y aplica una EEE es resistente contra una invasión de un pequeño grupo que maneja otra estrategia.

1.1.3. Tipos de juegos

Existen diferentes tipos de juegos que se encuentran clasificados por sus distintas propiedades de la siguiente manera:

Juego $N_j \times N$: Tipo de juego en el cual existe N jugadores donde el jugador j -ésimo tiene n_j posibles estrategias.

Juego de suma cero: Se trata de un juego en el cual la sumatoria de las funciones de pago P_k es constante independientemente de la estrategia S_j escogida por los jugadores generalmente se puede igualar a cero. Esto quiere decir que lo que uno gana lo pierde el otro.

Juego de información completa: Es aquel en el cual toda la información (posiciones, estrategias, funciones de pago) es conocida antes de que los jugadores decidan sus estrategias.

Juego simétrico: Cuando todos los agentes (jugadores) tienen el mismo tipo de pago ante la misma acción.

1.1.4. Puntos de Equilibrio

Un punto de equilibrio se encuentra definido como la solución de un juego cuando un jugador no tiene una estrategia diferente o mejor para sí mismo o para el sistema de acuerdo a diferentes criterios que son los más importantes son mencionados a continuación:

Equilibrio de Nash¹ (EN): Resultado del juego en el cual ningún jugador puede mejorar su pago ante un movimiento unidireccional. Se dice que una estrategia cumple el equilibrio de Nash cuando:

$$S = \{S_1, \dots, S_N\} \text{ es EN sí } \forall k, P_k(S_1, \dots, S_k, \dots, S_N) \geq P_k(S_1, \dots, S_{k'}, \dots, S_N) \forall S_{k'} \text{ (Ec. 2)}$$

Punto Focal: Dado que no existe una forma de definir la ventaja de un determinado EN se puede fijar uno de ellos mediante el gusto o criterio psicológico de los jugadores.

¹ El equilibrio de Nash suele convertirse en un tema puntual y largo de estudio detallado claramente en Myerson [65].

Maximin: Es un punto de equilibrio del sistema donde cada jugador maximiza el mínimo de pago que pudiera recibir. Esto es que cada jugador asume que los otros jugadores escogerán la mejor opción para sí mismo.

Es ideal para considerarlo en juegos de suma cero pero falla en ocasiones cuando se requiere de cooperativismo.

Óptimo de Pareto (OP). Resultado de un juego en el cual ningún jugador puede mejorar su pago sin perjudicar a los otros. Se define un óptimo de Pareto:

$$\forall k \exists l \text{ t. q. } P_k(S_1, \dots, S_{k'}, \dots, S_l, \dots, S_N) > P_k(S_1, \dots, S_k, \dots, S_l, \dots, S_N) \quad (\text{Ec. 3})$$

entonces, $P_k(S_1, \dots, S_{k'}, \dots, S_l, \dots, S_N) < P_k(S_1, \dots, S_k, \dots, S_l, \dots, S_N)$

1.2. Ejemplos de Juegos Clásicos

A continuación citaremos algunos de los juegos que son considerados clásicos. Los mismos que han sido estudiados a profundidad y nos permiten distinguir claramente los conceptos y propiedades de la teoría de juegos [1] [3].

Se los expone a detalle para en el próximo capítulo indicar la mejora que se puede presentar al introducir una variación cuántica de estos juegos.

1.2.1. “Cara o Sello” (Juego Simétrico 2x2)

El juego de “cara o sello” es un juego tradicional [27] en el cuál se arroja una moneda al aire mientras se pregunta a uno de los jugadores cuál es su decisión, si la misma coincide con el lado de la moneda que termina hacia arriba el jugador “gana”. Si conociéramos con exactitud la fuerza, el lugar exacto en que fue impulsada la moneda, las condiciones ambientales, el tiempo de vuelo, el peso y estructura de la misma, podríamos mediante cálculos hallar el resultado final y “ganar” en todos los lanzamientos. Dado que esto se torna en un complicado problema computacional, asumiremos que los resultados de nuestra moneda son equiprobables entre cara y sello.

Para modelizar este juego con las consideraciones antes realizadas, otorgaremos como función de pago $P = \frac{1}{2}$ si la elección del jugador concuerda con el resultado del lanzamiento y $P = -\frac{1}{2}$ si no lo hace. Tomando en cuenta que el resultado lo

podríamos hacer mutuamente excluyente, es decir si el jugador A (Ana) escoge cara, el jugador B (Bolívar)², automáticamente ha escogido sello.

Podríamos resumir este juego como una matriz como la indicada en la Tabla 1.

		Bolívar	
		Cara	Sello
Ana	Cara	$(\frac{1}{2}, -\frac{1}{2})$	$(-\frac{1}{2}, \frac{1}{2})$
	Sello	$(-\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, -\frac{1}{2})$

Tabla 1.- Matriz de Pago 2x2 del juego simétrico de suma cero y de información completa llamado “Cara o Sello”.

1.2.2. “Piedra, Papel o Tijera”

En este juego en cambio se deberá tomar en cuenta el criterio del jugador, en el cual cualquiera de las acciones que este tome tiene las mismas opciones ante los movimientos que realice su oponente, como se ve en la Tabla 2.

		Bolívar		
		Piedra	Papel	Tijera
Ana	Piedra	(0,0)	$(-\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, -\frac{1}{2})$
	Papel	$(\frac{1}{2}, -\frac{1}{2})$	(0,0)	$(-\frac{1}{2}, \frac{1}{2})$
	Tijera	$(-\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, -\frac{1}{2})$	(0,0)

Tabla 2.- Matriz de Pago 3x3 en un juego simétrico y de información completa “Piedra, Papel o Tijera”.

Se puede observar que existe simetría en las matrices de pagos ya que:

$$P(A) = P(B)^T \quad (\text{Ec. 4})$$

$$P(A) = \begin{vmatrix} 0 & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \end{vmatrix}; P(B) = \begin{vmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{vmatrix}; P(B)^T = \begin{vmatrix} 0 & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \end{vmatrix} \quad (\text{Ec. 5})$$

² Tradicionalmente se toma el jugador B como Bob [1], sin embargo por los motivos descritos en la dedicatoria de este trabajo se ha realizado el cambio de nombre del jugador B.

1.2.3. “El dilema del prisionero”

El dilema del prisionero es uno de los juegos más estudiados debido a sus connotaciones psicológicas de cooperar o competir dadas ciertas condiciones iniciales [28] [29] [30] [31].

En una situación en la cual se han capturado a 2 prisioneros que han participado juntos en una fechoría (un robo por ejemplo) y se les otorga 2 oportunidades las cuales son cooperar con la investigación es decir acusar al otro prisionero que ha sido capturado o desistir de cooperar guardando silencio. Obtendríamos la matriz de pagos siguiente, en la cual se describe las situaciones para los dos jugadores en caso de cooperar o desistir:

		Bolívar	
		C	D
Ana	C	$(-3,-3)$	$(0,-5)$
	D	$(-5,0)$	$(-1,-1)$

Tabla 3.- Matriz de Pago 2x2 en un juego simétrico de información completa “Dilema del prisionero”.

Claramente se ve que este juego no puede ser expresado como de suma cero, ya que no existe directamente un ganador entre los 2 participantes. Además se ha considerado como negativa la función de pago para de manera práctica considerarlo como los años que se están aplicando como condena para los participantes dependiendo de su selección.

Al tratarse de un juego de información completa suponiendo que el otro jugador va a responder de una manera llevada por la teoría de juegos podríamos considerar que la mejor opción o equilibrio de Nash para el sistema consiste en elegir la opción en la cual un cambio de uno de los jugadores no represente una mejora a su función de pago, así la opción de desistir le asegura un año de prisión pero si cambia de opción tendría 5 años. Mientras que si se hubiera escogido la opción de cooperar representa para el jugador un mínimo de 3 años de prisión con el mismo máximo de 5 años. Por lo que basado en la estrategia *Minimax* sugiere escoger la opción de desistir ya que en este caso el castigo sería entre 0 y 1 año. Por otra parte el escoger la opción de cooperar representa un óptimo de Pareto considerando que es el escenario en el cuál la única manera de mejorar su función de pago consiste en perjudicar la función de pago del otro jugador.

La teoría de juegos no siempre presenta una solución única como se puede ver en este caso, en el cuál se presentan los conflictos entre el bien común al cooperar entre los involucrados (Equilibrio de Nash) [15] [28], o el competir entre ellos (Óptimo de Pareto) para generar un bienestar personal [32] [33].

1.2.4. Resumen de las características de los juegos clásicos:

La siguiente matriz resume las características de un número de juegos clásicos en función de las definiciones hechas arriba y permite clasificarlos según estas.

Juego	N	S	P	Sim	S _{mix}	S _{dom}	Z ₀	IP	EN	OP	PF	Mm	EEE
Cara o Sello	2	2	2x2	X	X		X	X			X		
Piedra, Papel o Tijera	2	3	3x3	X	X		X	X			X		
Dilema del Prisionero	2	2	2x2	X		X		X	X	X	X	X	X

Tabla 4.- Resumen de los conceptos presentados en los ejemplos de juegos clásicos.

En donde:

N = Número de Jugadores.

S = Posibles estrategias para cada jugador.

P = Dimensiones de la matriz de pagos.

Sim = Simetría.

S_{mix} = Estrategia mixta.

S_{dom} = Estrategia dominante.

Z₀ = Suma cero.

IP = Información perfecta.

EN = Equilibrio de Nash.

OP = Óptimo de Pareto.

PF = Punto Focal.

Mm = Maxmin.

EEE = Estrategia Evolucionaria Estable.

1.3. Juegos de Minoría

En el punto anterior hemos revisado juegos básicos para identificar los conceptos fundamentales de la teoría de juegos, ahora procederemos a entender el juego con el que se realizó el presente estudio.

Un juego de minoría por concepto es el conjunto de **N** participantes que tienen 2 opciones, la estrategia accesible para los participantes es escoger una de las 2 opciones y la función de pago se calcula una vez que todos los participantes han tomado su decisión y se determina como ganadores a los jugadores que han escogido la opción que menos participantes la escogieron. En una primera iteración el juego es de información incompleta ya que ninguno de los participantes conoce la intención de los otros, sin embargo para realizar la siguiente decisión cada

jugador tiene acceso al estado anterior del sistema y a la función de pago asociado. Esta evolución fue estudiada por primera vez en 1997 por Challet y Zhang [5] hasta trabajos más actuales como el de Kets [34].

Se definirá como las posibles acciones para la evolución del sistema el 0 el mantenerse sin tomar la decisión o el 1 para cuando sí se tome la acción³. Para explicar el acierto de los ganadores, y en razón a que los individuos que estamos considerando cuentan con una racionalidad limitada, tomaremos los últimos dígitos, dicho de otro modo, la última combinación exitosa que dio como ganadores a los jugadores. De esta forma la combinación estaría compuesta por los tres últimos dígitos, por ejemplo en la secuencia 0, 1, 0: *hace tres iteraciones quedaron en minoría los que eligieron 0, hace dos los que eligieron 1 y en la última jugada los que eligieron 0*. Los jugadores tienen que adoptar estrategias para tomar la mejor decisión en el siguiente movimiento del juego.

Originalmente, Schelling [35] planteó el juego de minorías aplicado a ejemplos de tránsito, asistentes a una playa, etc. Allí exponía las estrategias que podrían utilizar los participantes involucrados para evitar ser parte de la mayoría. Posteriormente el problema fue abordado por Brian Arthur [36], quien lo hizo famoso a través del ejemplo de un bar llamado “El Farol” en California. Según se describía el bar tenía un show de música irlandesa los días jueves por la noche al que asistía el propio Arthur con sus compañeros del Santa Fe institute [37]. El lugar era de dimensiones pequeñas con una capacidad cercana a 60 personas sentadas. De esta manera si los asistentes pensaban que ese día asistirían más de 60 personas (incluyéndose ellos mismo) preferirían no asistir debido a que iban a estar incómodos e incomodar al resto de asistentes. Así que en ese escenario se le ocurrió a Arthur pensar que cada uno de los aproximadamente 100 asistentes habituales del espectáculo tenía la misma forma para determinar la incomodidad en la que incurrirían si asisten o no, partiendo de la idea inicial de que ninguno de los participantes sabía cuál era la decisión de los otros.

³ De manera más general se puede considerar para el análisis la posibilidad de que el jugador que no toma la decisión su estrategia se la represente con $-\frac{1}{2}$ y al que la toma con $\frac{1}{2}$ siendo de esta manera más relacionable con el concepto que se desea introducir posteriormente en relacionarlo.

De esa manera cada uno establecería sus tendencias y patrones de comportamiento o lo que hemos definido como “estrategia” para la teoría de juegos, mediante los cuales podría determinar si el día seleccionado debería ir al bar o quedarse en casa. Luego se analizaría las posibles diferentes estrategias que marca un participante y validaría cuál es la más efectiva. Se estima que las estrategias posibles se referían a efectos condicionales de lo que ha pasado en días anteriores así por ejemplo, si la semana pasada fueron menos (más) de 60 personas entonces este día jueves asistirían más (menos) personas. Si bien el hecho de establecer este tipo de estrategias para los participantes podría resultar en un sesgo estadístico al considerar que son sólo 100 los clientes habituales es decir se trata de un sistema cerrado, o que las estrategias de los participantes no van a variar temporalmente.

Comportamientos similares a los de este ejemplo podemos evidenciarlos en la vida cotidiana. El problema del tránsito automotor por ejemplo: Todos los días los conductores deciden por qué calle o autopista van a realizar su viaje, especulando con la saturación de los caminos y utilizando también reglas que tienen en cuenta el clima, huelgas, reparaciones, fines de semana largo, día de la semana, inicio de vacaciones, etc. La elección de destinos turísticos es otro ejemplo. Tanto en salidas vacacionales o por ejemplo la visita de compras a Colombia mucha gente prefiere evitar los momentos demasiado poblados y por lo tanto pretendemos buscar alternativas que a priori se consideren convenientemente tranquilas. Las reglas que podemos utilizar para predecir el “nivel de tranquilidad” del destino son variadas: la publicidad, la experiencia en visitas anteriores, la época del año, el clima, la tasa de cambio, etc.

1.3.1. Versión Analítica del Juego:

A partir de la idea de “El Farol, Challet y Zhang [38] generan un modelo abstracto de decisión binaria que considera la posibilidad de aprendizaje inductivo basado en que cada jueves corresponda a una jugada, las decisiones de “ir” o “quedarse” las representaremos con 0 y 1 y la división pasa de ser 60-40 a 50-50. Es decir se modifica el juego de tal manera que el juego pueda ser simétrico y la minoría (mayoría) pueda ser interpretada intuitivamente. El resultado de cada jugada será representado como 0 si la minoría de asistentes va al bar y como 1 si la minoría

decide quedarse en casa. En base a esta información se marca como ganadores a los jugadores que se encuentran en la minoría. Las estrategias posteriormente se las va definiendo como una lista combinatoria de los resultados en las últimas jugadas, esto es, con todas las posibles combinaciones de 0 y 1 dentro de una cadena de m caracteres, donde m representa la “historia” o “memoria” de los agentes. De esta manera debido a que existen 2^m posibles historias, cada estrategia puede asignar a cada historia la predicción 0 o 1 para la próxima jugada.

Estrategia	$m = 3$	Predicción
1	000	1
2	000	0
3	001	1
4	001	0
5	010	1
6	010	0
7	011	1
8	011	0
9	100	1
10	100	0
11	101	1
12	101	0
13	110	1
14	110	0
15	111	1
16	111	0

Tabla 5.- Ejemplo de estrategias típicas de un participante del juego de minorías

En la Tabla 5 podemos ver una estrategia típica cuando $m = 3$. Por lo que podemos ver hay 2^{2^m} posibles opciones de las que cada agente toma aleatoriamente s estrategias (se permite repetir selecciones entre los participantes), se considera que cada participante elige una estrategia única al principio del juego de esta manera se dispone de heterogeneidad de los individuos en el juego. Posterior a esto se considera que el método de selección de cada participante es el siguiente: al final de cada jugada a aquellas estrategias que han predicho correctamente el lado minoritario (0 o 1) se les otorga un punto (independiente de que hubieran sido las utilizadas en la decisión o no); para la jugada siguiente, cada agente utiliza aquella estrategia que más puntos recibió en el anterior punto de pago desde el comienzo del juego o elige aleatoriamente de entre las que empaten en el mayor puntaje.

Para dar inicio el juego es necesario definir una historia inicial (una de las 2^m posibles) la cual la planteamos aleatoriamente asignando a cada participante s estrategias (tomamos $s = 2$ para este caso) de las 2^{2^m} posibles. Los resultados

esperados en este modelo estarían relacionados con la coordinación que se presenta de entre los participantes mediante la cual, para ciertos valores de memoria m la varianza de las distribuciones de las decisiones es sensiblemente inferior a la que obtendría por decisiones aleatorias.

2. CAPÍTULO II: CUANTIZACIÓN DE LA TEORÍA DE JUEGOS

2.1. Elementos de Mecánica Cuántica Para la Teoría de Juegos

Para introducir la mecánica cuántica a la teoría de juegos es conveniente trabajar con la notación *bra-ket* de Dirac, la misma que unifica en una misma simbología la descripción de los operadores y las cantidades observables que podemos utilizar cuando las matrices proceden a actuar como operadores [39] [40] [41] [42] [43]. El planteamiento se lo ha realizado adaptando los conceptos generales de la notación de Dirac [44] [45] y personalizado en nomenclatura y funcionalidad para poder ser introducida de una manera adecuada a la teoría de juegos.

Vamos a considerar la representación del vector $x = (x^1, x^2, x^3)$ de tal forma que está expresado a través de vectores unitarios⁴.

$$\begin{aligned} x &= x^1(1 \ 0 \ 0) + x^2(0 \ 1 \ 0) + x^3(0 \ 0 \ 1) \\ x &= (x^1 \ 0 \ 0) + (0 \ x^2 \ 0) + (0 \ 0 \ x^3) \end{aligned} \quad (\text{Ec. 6})$$

La misma representación puede ser realizada a través de vectores⁵ verticales en lugar de vectores horizontales:

$$x = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} x^1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ x^3 \end{bmatrix} = x^1 |1\rangle + x^2 |2\rangle + x^3 |3\rangle \quad (\text{Ec. 7})$$

La notación en color rojo en la tercera parte de la Ecuación 7, formada por una línea vertical a la izquierda y un paréntesis angulado a la derecha también se lo conoce como un **ket**. Los kets en esencia representan los estados y son básicamente una etiqueta conveniente para representar los mismos.

En la notación de Dirac podemos referenciar un estado general de un sistema cuántico, es una combinación lineal de vectores, *ket*, individuales:

⁴ Los superíndices representan dimensión y no representan exponentes.

⁵ Al tratarse de vectores que también pueden ser expresados como matrices de dimensiones $N \times 1$ o $1 \times N$ se emplea indistintamente el término vector o de manera más general el de matriz.

$$a|\psi_1\rangle + b|\psi_2\rangle \quad (\text{Ec. 8})$$

Para el sistema representado en la Ecuación 8 la probabilidad de que bajo una medición particular del sistema se lo encuentre primer estado, $|\psi_1\rangle$, es: $\frac{|a|^2}{|a|^2+|b|^2}$ y la probabilidad de que se dé el estado simbolizado por el segundo ket es: $\frac{|b|^2}{|a|^2+|b|^2}$.

Si el estado está normalizado se tiene que $|a|^2 + |b|^2 = 1$. Note que a y b son de manera general números complejos.

De manera general, cuando sumamos dos o más kets lo que tenemos es una superposición de estados que genera una nueva situación. Por ejemplo, un estado arbitrario de espín de un electrón estaría representado como:

$$|\otimes\rangle = a|\uparrow\rangle + b|\downarrow\rangle$$

$$|a|^2 + |b|^2 = 1$$

Figura 2.- Representación en notación de Dirac del spin de un electrón.

Una medición realizada sobre este estado para determinar el espín del electrón (por ejemplo con un aparato de Stern-Gerlach [46]), solamente puede dar lugar a una de las dos opciones indicadas, está arriba o abajo. Esto en mecánica cuántica se conoce como el colapso de la función de onda. La probabilidad de encontrar uno u otro de los dos únicos resultados posibles sería, como se dijo antes, $|a|^2$ ó $|b|^2$, respectivamente. Es decir, que cuando colapsemos la función de onda del espín del electrón podrá ser únicamente el visualizado en el *ket* 1 o el visualizado en el *ket* 2 a pesar de que existen infinitas combinaciones de a y b posibles. Sin embargo la probabilidad de encontrar al sistema en el estado 1 o en el estado 2 es complementaria y mutuamente excluyente por lo que su sumatoria es la certeza.

El producto interno de dos vectores (estados, o *kets*): a y b es:

$$a \cdot b = (\overline{a_1} \quad \overline{a_2} \quad \overline{a_3} \dots) \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{pmatrix} = \overline{a_1}b_1 + \overline{a_2}b_2 + \overline{a_3}b_3 + \dots \quad (\text{Ec. 9})$$

El producto interno de dos kets no es conmutativo, ya que:

$$\overline{b \cdot a} = a \cdot b \quad (\text{Ec. 10})$$

En la notación de Dirac el producto escalar se representa como:

$$\langle \psi_i | | \psi_j \rangle = \langle \psi_i | \psi_j \rangle \quad (\text{Ec. 11})$$

Donde, de nuevo:

$$\overline{\langle \psi_i | \psi_j \rangle} = \langle \psi_j | \psi_i \rangle \quad (\text{Ec. 12})$$

De manera frecuente en la mecánica cuántica los componentes utilizados a modo de vectores suelen ser ortogonales⁶ estando cada uno de ellos normalizado⁷ es decir, son vectores orto-normales. En ese caso:

$$\langle \psi_j | \psi_i \rangle = \delta_{ij} \quad (\text{Ec. 13})$$

En esta definición se ha utilizado el delta de Kronecker en el que: $\delta_{ij} = 1$ cuando $i = k$, y $\delta_{ij} = 0$ cuando $i \neq k$. Nótese que para todo *ket* se puede asignar un *bra* que es un miembro del espacio *dual* correspondiente. Para un *ket*, podemos obtener su *bra* dual tomando la transpuesta conjugada compleja del *ket*.

A continuación discutiremos las propiedades de este sistema de espín del electrón ya que el mismo puede ser asociado al comportamiento de una “moneda cuántica” por su posibilidad de que al colapsar la función de onda tenemos dos posibles estados: espín arriba o espín abajo haciendo el símil de cara o sello respectivamente.

2.1.1. Matrices de Pauli

Tomando en cuenta el espín de un electrón como la función a ser analizada que nos permitirá aproximar la rotación de espín del electrón [47] [48] [49], introduciremos matrices hermitianas $\sigma_x, \sigma_y, \sigma_z$ que satisfacen relaciones de conmutación:

$$\begin{aligned} \sigma_x \sigma_y - \sigma_y \sigma_x &= 2i \sigma_z \\ \sigma_y \sigma_z - \sigma_z \sigma_y &= 2i \sigma_x \\ \sigma_z \sigma_x - \sigma_x \sigma_z &= 2i \sigma_y \end{aligned} \quad (\text{Ec. 14})$$

⁶ Independientes entre sí.

⁷ Longitud igual a la unidad.

Además introduciremos el concepto de spinores el efecto de que al colapsar el sistema pueda terminar

Las decisiones en la teoría de juegos se modelan regularmente como el estado final de un sistema que en el mundo clásico podría ser una moneda, mientras que el símil en el mundo cuántico es el espín de un electrón, por este motivo es necesario introducir el concepto de matrices de rotación como posibles estrategias de los jugadores al acceder a una estrategia cuántica.

2.1.2. Rotaciones en el Sistema de Espín del Electrón

Para poder entender las rotaciones estableceremos por convención que el eje positivo z será en la dirección del campo magnético. En este sentido será directo que una rotación alrededor de ese eje no tendrá ningún efecto sobre estas probabilidades, por lo tanto procederemos a realizar una rotación infinitesimal de un ángulo ε alrededor del eje z con lo que $|c_1|^2$ y $|c_2|^2$ no debería cambiar, es decir:

$$U_R = I - \frac{i}{2} \varepsilon \sigma_z \quad (\text{Ec. 15})$$

La matriz U_R es diagonal, por lo tanto σ_z también debe ser diagonal. Además tomando la traza de la primera línea de la ecuación 17 obtenemos:

$$\text{Tr}(\sigma_z) = 0 \quad (\text{Ec. 16})$$

Y dado que $\text{Tr}(AB) = \text{Tr}(BA)$, entonces σ_z tiene la forma:

$$\sigma_z = \begin{pmatrix} \lambda & 0 \\ 0 & -\lambda \end{pmatrix}, \text{ con } \lambda \in R \quad (\text{Ec. 17})$$

En este punto introduciremos matrices auxiliares no hermitianas σ_+ y σ_- :

$$\sigma_+ = \sigma_x + i\sigma_y, \quad \sigma_- = \sigma_+^T = \sigma_x + i\sigma_y \quad (\text{Ec. 18})$$

Que cumplen las siguientes reglas de conmutación:

$$\begin{aligned} [\sigma_z, \sigma_+] &= 2\sigma_+ \\ [\sigma_z, \sigma_-] &= -2\sigma_- \\ [\sigma_+, \sigma_-] &= 4\sigma_z \end{aligned} \quad (\text{Ec. 19})$$

Estas matrices nos ayudarán a determinar σ_x y σ_y , para lo cual pondremos:

$$\sigma_+ = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (\text{Ec. 20})$$

Con los valores de a , b , c y d para ser determinados mediante sustitución como sistema de ecuaciones entre las ecuaciones 21, 24 y la primera de 23, obtenemos como resultado:

$$\begin{pmatrix} 0 & 2\lambda b \\ -2\lambda c & 0 \end{pmatrix} = 2 \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (\text{Ec. 21})$$

Donde $a = d = 0$ y por otra parte $b(\lambda - 1) = 0$ y $c(1 + \lambda) = 0$. Sin perder generalidad λ puede tomar tanto el valor de 1 cómo -1 por la definición que hemos tomado de valor propio para el spin. Para este caso tomaremos el valor positivo lo que nos lleva a determinar la forma de σ_+ y σ_- como:

$$\sigma_+ = \begin{pmatrix} 0 & b \\ 0 & 0 \end{pmatrix}, \sigma_- = \sigma_+^T = \begin{pmatrix} 0 & 0 \\ b & 0 \end{pmatrix} \quad (\text{Ec. 22})$$

Sustituimos estos resultados en el sistema de ecuaciones 19. Para hallar finalmente la forma de las matrices de spin de Pauli:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (\text{Ec. 23})$$

Las mismas que están completamente determinadas en el orden de filas y columnas y corresponden a la elección de $\lambda = +1$ y a las relaciones de conmutación planteadas en la ecuación 23 y por la elección que hemos realizado para los spinores.

Estas matrices cumplen con las siguientes propiedades:

$$\begin{aligned} \sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 1 \\ \sigma_x \sigma_y = i \sigma_z, \sigma_y \sigma_z = i \sigma_x, \sigma_z \sigma_x = i \sigma_y \end{aligned} \quad (\text{Ec. 24})$$

Con estas matrices acopladas a la matriz unitaria se puede construir el espacio 2×2 de la manera que cualquier matriz A en el espacio de las matrices 2×2 puede ser expresada como:

$$A = \lambda_0 I + \lambda_x \sigma_x + \lambda_y \sigma_y + \lambda_z \sigma_z = \lambda_0 I + \lambda \sigma \quad (\text{Ec. 25})$$

Si A es hermitiana, todos los coeficientes son reales.

Como se indicó, las matrices de Pauli aquí definidas también pueden ser consideradas como posibles rotaciones de un sistema de espín que son como las utilizaremos para la cuantización del juego de minorías.

Al introducir este tipo de estrategias se permitirá que además de rotar o dejar en el mismo estado al sistema se pueda acceder a 3 nuevas estrategias, las mismas que por sí solas no representan una ventaja para el jugador, pero pueden generar nuevas historias para influir en la resolución de los estados clásicos del sistema.

2.2. Definición de Juegos Cuánticos

Una vez definidos los conceptos fundamentales de la teoría de juegos revisaremos la posibilidad de incorporar el modelamiento mecánico cuántico a través de los operadores al cambio de estrategias, funciones de pago y puntos de equilibrio en un juego se han tomado de varios autores [50] [51] [52] y se presenta aquí una compilación.

2.2.1. “Cara o Sello Cuántico”.

Uno de los dispositivos más simples para realizar un juego es aquel que permite la presentación de 2 estados como lo que representa una moneda, o de manera más general para el propósito de cuantizar el juego la veremos como un electrón con spin positivo o spin negativo [16].

En el punto que es necesario realizar una rotación al sistema clásicamente sólo tendríamos acceso a 2 opciones rotar la moneda o mantenerla en el mismo estado, sin embargo si incorporamos las posibles rotaciones expresadas a través de las matrices de Pauli tendríamos acceso a 3 acciones adicionales con el sistema.

2.2.2. Descripción Detallada del Juego.

Al inicio cada jugador define su función de pago como el resultado del estado final de la “moneda”.

El jugador A que la denominaremos Ana puede preparar una moneda en el estado cara o sello y la pone en una caja. La caja utilizada para estas situaciones permitirá a cualquiera de los jugadores realizar una rotación sin que el otro pueda identificar si fue o no realizada una rotación. Bolívar (jugador B) por su parte al recibir la caja decide rotar o no la moneda. Una vez más Ana decide realizar o no una rotación y devolver la caja a Bolívar. En este momento Bolívar tiene una última opción de rotar el sistema. Una vez realizado todo el proceso anterior se destapa la caja y se determina el ganador. No existe manera alguna de determinar el resultado final de

la operación previamente, ya que con una moneda clásica ambos jugadores tienen iguales probabilidades de ganar.

Para esquematizar el juego clásico consideraremos que Ana escogió el resultado cara y Bolívar el resultado sello.

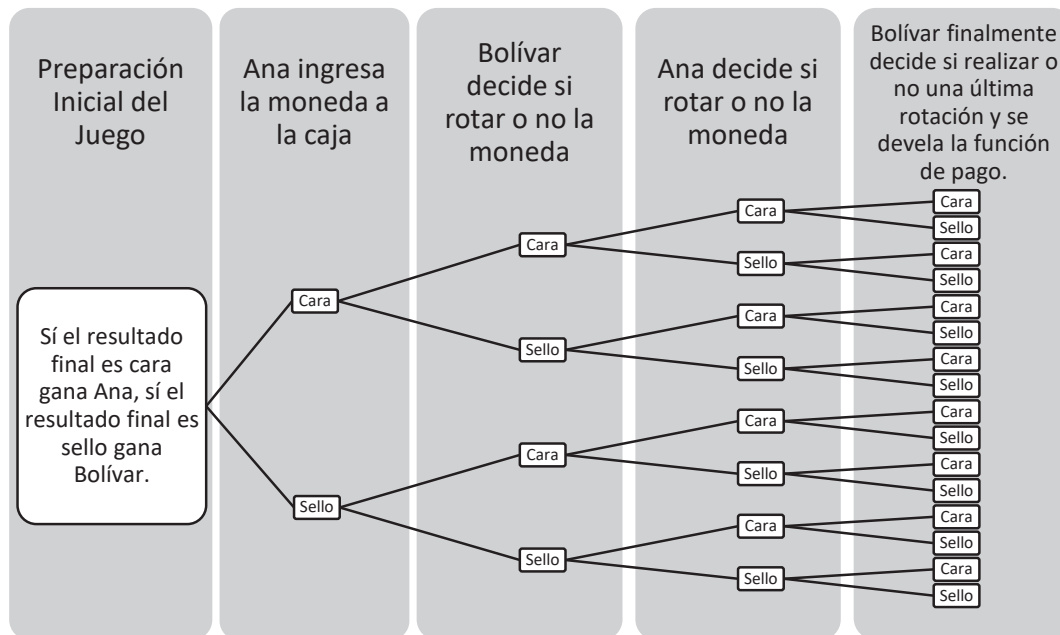


Figura 3.- Esquema de solución al juego de cara o sello clásico con 3 momentos de cambio.

Se analiza cada uno de los 5 momentos del juego y el árbol de posibles estados dependiente de las estrategias tomadas en el evento anterior.

En el ejemplo presentado en la figura 12 vemos que existen un total de 2^m caminos posibles que pudo haber tenido este juego, donde m es la cantidad de posibles rotaciones antes de procesar la función de pago, sin embargo si consideramos las decisiones de rotar o no la moneda como equiprobables ya que después del momento 1 en el cuál Ana fija un estado inicial⁸ de la moneda en todos los momentos siguientes es cada decisión tiene una probabilidad de 0.5. Por lo que no podría presentarse una estrategia ganadora por ninguno de los participantes.

⁸ Existe la posibilidad de corregir la estabilidad de este juego desde el momento uno cuando se permite elegir al jugador que no fijó el estado inicial de la moneda desde el momento uno y la elección se produce después de terminar el juego no antes.

2.2.3. Esquema Utilizado para Cuantizar el Problema.

Para cuantizar este juego procedemos a reemplazar la moneda por un sistema mecánico cuántico que tenga dos posibles estados [53], una posibilidad es usar el spin semi-entero de una partícula [54].

Supongamos que Bolívar tiene la posibilidad de hacer rotaciones cuánticas mientras Ana se encuentra restringida a los movimientos clásicos.

El estado entonces del sistema estará dado por:

$$\sigma_c = \sigma_{cara} = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_s = \sigma_{sello} = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (\text{Ec. 26})$$

El resultado estará determinado como el valor propio de la matriz resultante del sistema de tal manera que si es positivo el resultado será cara y si es negativo el resultado será sello:

$$|\sigma_c|^2 = \left(\sqrt{\frac{1}{2}(1 * 1 - 0 * 0)} \right)^2 = \frac{1}{2}, |\sigma_s|^2 = \left(\sqrt{\frac{1}{2}(0 * 0 - 1 * 1)} \right)^2 = -\frac{1}{2} \quad (\text{Ec. 27})$$

La jugadora que tiene acceso únicamente a la estrategia clásica podremos expresar sus decisiones como:

$$\sigma_r = \sigma_{rotar} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_{nr} = \sigma_{no\ rotar} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (\text{Ec. 28})$$

Que son básicamente la matriz I y la matriz $-I$. En cuanto al jugador B que puede incorporar las estrategias cuánticas sus posibilidades se amplían a las presentadas en la Ecuación 27. Aquí podemos validar que la decisión de rotar es equivalente a la rotación de la matriz x de Pauli y la decisión de no rotar es equivalente a la matriz de rotación I .

Como resultados previos podríamos anticipar que las posibles combinaciones que vamos a encontrar en el juego serían:

$$\sigma_c \sigma_r = \sigma_s; \sigma_c \sigma_{nr} = \sigma_c \quad (\text{Ec. 29})$$

$$\begin{aligned}
\sigma_c \sigma_x &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2} \sigma_x \\
\sigma_c \sigma_y &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \frac{1}{2} \sigma_y \\
\sigma_c \sigma_z &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2} \sigma_z \\
\sigma_c \sigma_{nr} &= \sigma_c \\
\sigma_s \sigma_x &= \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \sigma_c \\
\sigma_s \sigma_y &= \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} \\
\sigma_s \sigma_z &= \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\
\sigma_s \sigma_{nr} &= \sigma_s
\end{aligned} \tag{Ec. 30}$$

Al poder evidenciar una estrategia dominante a partir del momento 2 después de la rotación cuántica se plantea el momento 3 en la siguiente tabla considerando que el momento 4 de parte del jugador B consistirá en no rotar el sistema.

Estado Inicial momento 1	Estrategia momento 2	Estado momento 3	Rotación Aplicada	Matriz Resultante	Resultado Asociado	Ganador
Cara	σ_x	$\frac{1}{2}\sigma_x$	σ_r	σ_c	$\frac{1}{2}$	Cara (Ana)
Cara	σ_y	$\frac{1}{2}\sigma_y$	σ_r	$\frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_z	$\frac{1}{2}\sigma_z$	σ_r	$\frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_{nr}	σ_c	σ_r	σ_s	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_x	σ_c	σ_r	σ_s	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_y	$\frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$	σ_r	$\frac{1}{2} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_z	$\frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	σ_r	$\frac{1}{2} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Cara	σ_{nr}	σ_s	σ_r	σ_c	$\frac{1}{2}$	Cara (Ana)
Sello	σ_x	$\frac{1}{2}\sigma_x$	σ_{nr}	$\frac{1}{2}\sigma_x$	$-\frac{1}{2}$	Sello (Bolívar)
Sello	σ_y	$\frac{1}{2}\sigma_y$	σ_{nr}	$\frac{1}{2}\sigma_y$	$-\frac{1}{2}$	Sello (Bolívar)
Sello	σ_z	$\frac{1}{2}\sigma_z$	σ_{nr}	$\frac{1}{2}\sigma_z$	$-\frac{1}{2}$	Sello (Bolívar)
Sello	σ_{nr}	σ_c	σ_{nr}	σ_c	$\frac{1}{2}$	Cara (Ana)
Sello	σ_x	σ_c	σ_{nr}	σ_c	$\frac{1}{2}$	Cara (Ana)
Sello	σ_y	$\frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$	σ_{nr}	$\frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Sello	σ_z	$\frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	σ_{nr}	$\frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	$-\frac{1}{2}$	Sello (Bolívar)
Sello	σ_{nr}	σ_s	σ_{nr}	σ_s	$-\frac{1}{2}$	Sello (Bolívar)

Tabla 6.- Función de Pago juego Cara o Sello cuantizado asumiendo que no se realiza el movimiento en el momento 4 por parte del jugador B.

En la tabla 6 se puede evidenciar que los resultados quedan establecidos en el momento en el cuál el jugador B tiene acceso a una estrategia cuántica en un momento del juego, en este caso en el momento 2 permitiéndole realizar una

rotación con las matrices σ_y o σ_z se asegura que la función de pago finalmente será el estado definido como sello dentro del sistema. En el caso de que el jugador B su decisión sea la opción cara basta con realizar una rotación final en el momento 4 para asegurar el resultado deseado.

Esto se explica debido a que el jugador B tiene acceso a un plano adicional de transición donde las decisiones (rotaciones) sobre el eje en el que se está analizando el sistema se anulan.

Los resultados obtenidos en este ejemplo de juego cuántico serán útiles en el planteamiento cuántico del juego de minorías que se presenta en el capítulo IV.

3. CAPÍTULO III: SIMULACIÓN COMPUTACIONAL

3.1. Esquema General de una Simulación Computacional

Para poder estudiar la evolución de un sistema determinado, como por ejemplo: flujo de caja en un banco, niveles de migración en una ciudad, capacidad utilizada en un sistema de transporte, o en el caso específico que estamos analizando un juego de minorías genérico [26]. Continuamente es necesario determinar las acciones que el sistema estará ejecutando. Estas acciones deben estar consideradas de tal manera que el sistema satisfaga de la mejor manera la evolución esperada. Por lo tanto es necesario saber cómo responde el sistema ante una determinada acción. Esto lo podemos hacer inicialmente con el método experimental sobre el sistema mismo; lo cual implica varios factores que no podemos controlar como son costos, seguridad, tiempo entre otros que hacen que esta opción no sea viable normalmente. Con la idea de sobrepasar estos inconvenientes, pretendemos reemplazar el sistema real por otro sistema que en la mayoría de los casos podría ser una versión simplificada del mismo en la que podamos analizar directamente los efectos que estamos suponiendo que se ven afectados. El último sistema lo definiremos como el modelo que deseamos ocupar para llevar a cabo las repeticiones necesarias sin los inconvenientes planteados. A este proceso de experimentar con un modelo se lo denomina simulación. Al proceso de diseño del plan para la experimentación que nos permita adoptar la mejor decisión se lo conoce como optimización. Sí el plan de experimentación se desea llevar a cabo sólo con la idea de aprender las respuestas del sistema, entonces estamos tratando con un entrenamiento o capacitación. Aquí es conveniente introducir ciertas definiciones [55]:

Sistema: Está dado por el conjunto de objetos o ideas que están interrelacionados entre sí como una unidad para la consecución de un fin. También podremos definir el sistema como un subconjunto del universo que será objeto de simulación permitiéndonos separarlo cuando sea necesario en un subconjunto de entrenamiento que nos permita entender el comportamiento del mismo y un subconjunto de validación que nos permita ajustar los resultados obtenidos.

Modelo: Relacionaremos al objeto X como un modelo del objeto Y definiendo un observador el cuál puede emplear X para responder cuestiones que le interesan acerca de Y.

Simulación: Es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, esperando aprender el comportamiento del sistema y evaluar diversas estrategias para su funcionamiento o evolución.

Podemos considerar que utilizar simulación es conveniente cuando:

- No existe una formulación matemática analíticamente resoluble. Debido a que muchos sistemas reales no podrían ser modelados o su modelamiento resultaría muy complejo.
- Existe una formulación matemática, pero su solución analítica resulta complicada. Los modelos matemáticos ocupados para modelar un reactor nuclear son un ejemplo en el cuál no podemos resolverlo sin incorporar gran cantidad de suposiciones y simplificaciones.
- No existe el sistema real. En el caso de desarrollo de ingeniería resultaría muy costoso fabricar cierto sistema nuevo con todos los materiales posibles para experimentar sobre él. Incorporar una simulación previa facilita la decisión óptima del material adecuado y realizar pruebas reales con menos sistemas.
- Los experimentos son imposibles debido a impedimentos económicos, de seguridad, de calidad o éticos. En este caso diremos que el sistema real está disponible para experimentar, sin embargo, existe algún otro tipo de dificultad social o ideológica que no nos permite alterarlo directamente. Podríamos citar como ejemplo en este caso la imposibilidad de provocar fallas en un avión real para evaluar la conducta del piloto, como tampoco podríamos variar el valor de un impuesto sólo con el fin de evaluar la reacción del mercado.
- El sistema tiene una evolución muy lenta o muy rápida. La temporalidad en la que deseamos inmiscuirnos podría resultar un conflicto como por ejemplo cuando se analiza los cambios climáticos que llevan años en estabilizarse. En cambio otros fenómenos como por ejemplo una explosión que en apenas segundos ya evolucionó completamente.

Por otra parte también resulta conveniente citar los problemas que podríamos tener al realizar simulaciones:

- El desarrollo de un modelo puede ser costoso, laborioso y lento.
- Existe la posibilidad de cometer errores. No está por más notar que en la elaboración de un modelo no estamos incorporando todas las posibilidades y generalmente tomamos al sistema como cerrado por lo que una influencia externa podría invalidar la simulación.
- No podemos conocer el grado de imprecisión de los resultados. Ya que por lo general el modelo se utiliza para experimentar situaciones nunca planteadas en el sistema real, por lo que no existe información previa para estimar el grado de correspondencia entre la respuesta del modelo y la del sistema real.

En el proceso de realizar e implementar una simulación podemos distinguir las siguientes etapas [56]:

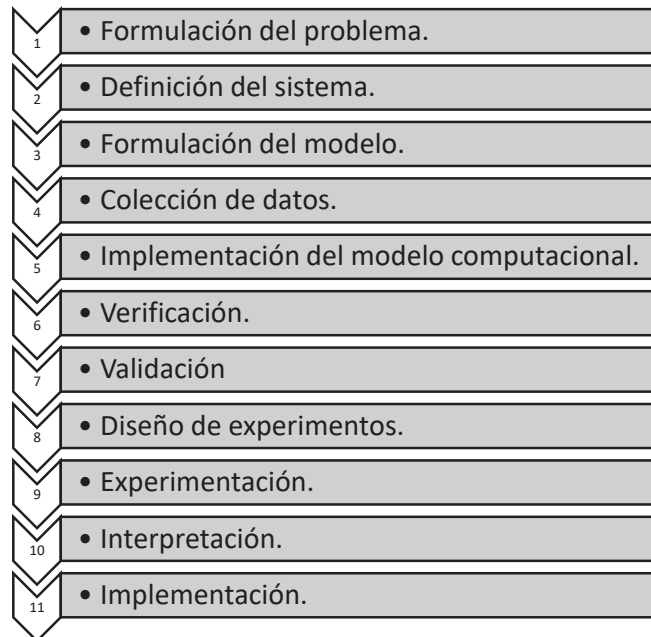


Figura 4.- Etapas para implementar una simulación.

Esquema funcional que se tomará en cuenta para realizar la programación de la simulación planteada, los pasos representan un esquema lógico, sin embargo el orden de realización puede no ser secuencial.

Las mismas que representan un orden lógico sin embargo no es necesario que se realicen en el orden mencionado y omitiremos el detalle de cada paso, sin embargo es adecuado mencionarlo ya que de esta manera se dispone de una metodología secuencial y simplificada que permite el desarrollo de una simulación que además de ser matemáticamente correcta, es funcional y visualmente atractiva para que pueda ser utilizada por alguien sin conocimiento completo de programación.

3.2. Simulación de Monte Carlo y ventajas del uso de SAS®.

La simulación de Montecarlo es una técnica que nos permite combinar conceptos estadísticos⁹ con la capacidad que tienen los computadores para generar números pseudo-aleatorios y automatizar los cálculos [17] [25] [26].

Esta técnica proviene originalmente a trabajos realizados por Stan Ulam y John Con Neumann a finales de los 40 en el laboratorio de los Álamos cuando se encontraban investigando el movimiento aleatorio de los neutrones [19]. En años siguientes la simulación de Monte Carlo se ha aplicado a varios ámbitos como alternativa a los modelos matemáticos exacto e incluso a veces como único método para estimar soluciones de problemas complejos. De esa manera; en el presente es posible encontrar modelos que utilizan simulación de Monte Carlo en las áreas de informática, economía, e incluso ciencias sociales [14]. Como podemos ver la simulación de Monte Carlo se encuentra presente en los hábitos en los que se puede evidenciar un comportamiento aleatorio o probabilístico, el mismo que desempeñe un papel fundamental, en realidad incluso el nombre de Monte Carlo proviene de una ciudad de Mónaco, donde abundan los casinos y los juegos de azar, la probabilidad y el comportamiento aleatorio son un estilo de vida.

Al igual que la estadística aplicada ha venido ganando mucha importancia en el mundo de la información, posiblemente debido a la mejora continua de los ordenadores y en concreto de los paquetes estadísticos. Paralelamente los paquetes estadísticos también se han ido desarrollando y adaptando a nuevas necesidades de manera que ahora podemos hablar de “paquetes estadísticos”. Para este estudio utilizaremos SAS®, el cual su nombre en un inicio significaba “Statistical Analysis System”¹⁰ pero a partir de su creación en 1960 además de contar con su propio lenguaje de programación este programa ha ido incorporando nuevas capacidades la mayoría de las cuales no pueden ser consideradas propiamente estadísticas, sin embargo de mucha utilidad para los usuarios del mismo. SAS® pasó a convertirse en una sola palabra. En la actualidad, los diferentes módulos de SAS® nos permiten considerarlo un software de los llamados

⁹ Principalmente el muestreo aleatorio.

¹⁰ Sistema de Análisis Estadístico.

“de inicio a fin”. Ya que permite importar y consolidar bases de diferentes fuentes, crear gráficos, trabajar como hoja de cálculo, compilar programación avanzada en su propio flujo, además de construir interfaces para el usuario, herramientas para tratar el manejo de datos y realizar Data-mining. Además de disponer de sistemas formales de capacitación y soporte para los programadores.

Se considera a SAS® una herramienta adecuada para realizar estudios y simulaciones utilizando debido a su versatilidad, en su capacidad de recalculer valores y sobretodo en las posibilidades que se ven presentes al momento de poder permitir a los usuarios codificar sus necesidades libremente con su lenguaje de programación, el mismo que además con el acceso a las herramientas de desarrollo de aplicaciones puede generar una simulación destinada a un usuario final sin problema.

3.2.1. Generación un Número Pseudo-Aleatorio

Los lenguajes de programación estándar son capaces de generar números pseudo-aleatorios que provienen de una distribución uniforme entre 0 y 1. Este tipo de números pseudo-aleatorios son los elementos básicos a partir de los cuales se desarrollan las simulaciones por computador.

En SAS® la función *RANUNI()* genera un número que es generado de una distribución uniforme en el intervalo (0,1) utilizando el módulo primo generador $2^{31}-1$ y factor multiplicador 397204094 [57].

En caso de que necesitemos una distribución con otro multiplicador o con otra longitud podemos utilizar:

$$variable_{aleatoria} = a * ranuni(semilla) + b \quad (\text{Ec. 31})$$

Con lo cual obtendremos un número generado de una distribución uniforme en el intervalo $(b, a+b)$. La semilla es por definición un valor entero menor que el módulo primo generados y que representa el valor en el cuál empezaremos a generar los números aleatorios. Esto podría ser de utilidad para utilizando el mismo código generar una secuencia de número aleatorios pero igual en caso de que requiramos repetir la ejecución. Por defecto la función utiliza el valor de semilla 0 el cual toma una secuencia numérica basada en el reloj interno del computador.

Los números que son generados por la función RANUNI tienen dos propiedades por las que podrían ser considerados pseudo-aleatorios:

1. En cada iteración se genera cualquier número real entre 0 y 1 con la misma probabilidad de ser generado.
2. Lo diferentes números generados son estadísticamente independientes unos de otros.

3.3. Simulación de Monte Carlo

Como hemos repasado anteriormente la simulación de Monte Carlo es una técnica cuantitativa que utiliza la estadística y los computadores para imitar, mediante modelos matemáticos, el comportamiento aleatorio generado en sistemas reales no dinámicos, o sistemas que en algún punto de su evolución recurren a estrategias que pueden ser descritas por un fenómeno aleatorio [26].

La clave en la simulación de Monte Carlo entonces consistirá en crear el modelo matemático para el sistema, proceso o actividad que se requiere analizar, identificando claramente aquellas variables (inputs del modelo) mediante las cuales el comportamiento aleatorio de las mismas determina o explica el comportamiento global del sistema. Una vez planteados los Inputs se llevará a cabo un experimento que consiste en que con ayuda del computador se puedan generar muestras aleatorias pero fijas para dichos inputs y analizar el comportamiento del sistema ante los valores generados. Este proceso lo repetimos n veces para evaluar mediante observaciones el comportamiento del sistema, lo cual nos permitirá el funcionamiento del mismo, como es de esperarse por la ley de los grandes números mientras mayor sea el número n de experimentos que llevemos a cabo el valor u observación obtenida será más precisa.

3.3.1. Ejemplo de una Simulación de Monte Carlo¹¹:

Tenemos el resultado de analizar durante 200 días el número de consultas de saldo a la cuenta de ahorros de un cliente. La segunda tabla de la figura 14 incluye un

¹¹ El código utilizado para este ejemplo se detalla en el Anexo 1. Se aclara en este punto que la utilización de SAS® se debe a todas las bondades que representa el software, además de que su versión estudiantil está al acceso de la comunidad educativa. Sin embargo el código es fácilmente interpretable y puede ser migrado a otros programas como R.

número de consultas diarias (entre 0 y 5) junto con las frecuencias absolutas (número de días que se producen 0,1,...,5 consultas), las frecuencias relativas ($10/200 = 0.05, \dots$) y las frecuencias acumuladas.

total_consultas			
200			
Salto de página			
consultas	frecuencia	frecuencia_absoluta	frecuencia_acumulada
0	80	0.400	0.400
1	50	0.250	0.650
2	40	0.200	0.850
3	15	0.075	0.925
4	10	0.050	0.975
5	5	0.025	1.000

Figura 5.- Ejemplo de tabla de frecuencias acumuladas.

Se puede interpretar la frecuencia relativa como la probabilidad de que el suceso asociado ocurra, aquí la probabilidad de un determinado número de consultas, por ejemplo que el cliente realice 4 consultas tiene una probabilidad de 5% por lo que la tabla nos proporciona la distribución de probabilidad asociada a una variable aleatoria discreta (la variable aleatoria sería el número de consultas que sólo puede tomar valores enteros entre 0 y 5).

Consideremos que se desea conocer el número esperado de consultas por día. Utilizando la teoría de probabilidad:

Tomando X la variable aleatoria que representa el número diario de consultas sabemos que:

$$E[X] = \sum_{i=0}^5 x_i \cdot P(X = x_i) = 0 * 0.4 + 1 * 0.25 + 2 * 0.2 + 3 * 0.075 + 4 * 0.05 + 5 * 0.025 = 1.2 \quad (\text{Ec. 32})$$

De esta manera hemos calculado el valor de una manera exacta utilizando la teoría de probabilidad, sin embargo ello no siempre es factible. Para este ejemplo procederemos a utilizar la simulación de Monte Carlo para estimar el número esperado de consultas diarias.

Ya que conocemos la distribución de probabilidad asociada a la variable aleatoria discreta dada en la columna de frecuencias relativas acumuladas para obtener los intervalos de números aleatorios asociados a cada suceso, para este caso los intervalos serían:

- $[0, 0.4)$ para el suceso 0.
- $[0.4, 0.65)$ para el suceso 1.
- $[0.65, 0.85)$ para el suceso 2.
- $[0.85, 0.925)$ para el suceso 3.
- $[0.925, 0.975)$ para el suceso 4.
- $[0.975, 1]$ para el suceso 5.

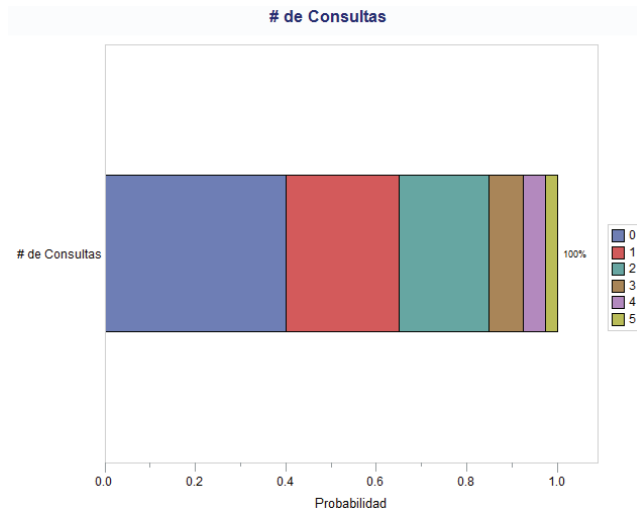


Figura 6.- Gráfico de frecuencias absolutas (probabilidades) del Ejemplo 3.3.1.

En la Figura 6 nos muestra cada una de las probabilidades sobre el número de consultas. En él se aprecia claramente la relación existente entre la probabilidad de cada suceso y el área que ocupa.

Esto significa que al generar un número pseudo-aleatorio con el computador proveniente de la distribución uniforme en el intervalo $(0,1)$ estaremos llevando a cabo un experimento con resultado obtenido de manera aleatoria y según la distribución de probabilidad asociada a un suceso. Por ejemplo si el computador nos proporciona 0.231 como número pseudo-aleatorio podremos suponer que ese día se han producido 0 consultas.

Entonces crearemos una tabla de datos con por ejemplo 100 registros y definiremos los condicionales necesarios para cumplir con la distribución de probabilidades y finalmente obtendremos el resultado del proceso para obtener estadísticos basándonos esta vez únicamente en el número de registros y en el promedio. Repetimos este proceso para una muestra mucho más grande por ejemplo 100.000 registros los resultados los podemos ver en la tabla 7.

Variable de análisis : suceso	
N	Media
100	1.1300000

(a)

Variable de análisis : suceso	
N	Media
100000	1.1999800

(b)

Figura 7.- Promedio de la variable número de consultas por día en 100 (a) y 100.000 repeticiones (b).

En este caso en la primera simulación con una población más pequeña hemos obtenido un valor de 1.13 mientras que con la muestra de 100.000 registros nuestro valor es de 1.9998 bastante más cercano al valor teórico. Definiendo como N el número de iteraciones podemos deducir que:

$$Observable_{real} = \lim_{N \rightarrow \infty} (Observable_{simulado}) \quad (\text{Ec. 33})$$

3.3.2. Simulación de Monte Carlo con Variables Discretas¹²:

Dado que no siempre los datos que deseamos simular obedecen a una distribución de probabilidad continua es conveniente identificar las diferencias que se debe tomar en cuenta cuando se pretende simular utilizando variables discretas.

Para este ejemplo vamos a suponer que es necesario determinar el número de tarjetas de cierto tipo que deben ser pre-emitidas para ciertos clientes, las mismas que podrán ser comercializadas y entregadas a su portador, pudiendo existir la posibilidad de que el cliente no acepte la tarjeta. La emisión de cada tarjeta tiene un valor de 7.5 dólares mientras que el precio con el cuál se le otorga al cliente es de 10 dólares. En caso de no aceptación por parte del cliente de la tarjeta la misma puede ser enviada al proveedor por un reembolso de 2.5 dólares por cada una, además se tiene como restricción que las tarjetas pueden ser pedidas en bloques de 50 con un pedido inicial mínimo de 100. Basados en datos históricos, se puede determinar la distribución de probabilidades presentada en la Figura 8.

tarjetas_compradas	probabilidad_venta	probabilidad_acumulada	intervalo_inferior	intervalo_superior
100	0.30	0.30	0.00	0.30
150	0.20	0.50	0.30	0.50
200	0.30	0.80	0.50	0.80
250	0.15	0.95	0.80	0.95
300	0.05	1.00	0.95	1.00

Figura 8.- Distribución de frecuencias para la simulación de Monte Carlos con variables discretas.

¹² El código utilizado para esta simulación se encuentra en el Anexo 2.

Con la información levantada construimos el modelo utilizando la probabilidad de venta y los 5 posibles escenarios (entre 100 y 300) calculando para cada escenario el número de tarjetas vendidas como el mínimo entre el número de tarjetas de la iteración de la simulación y el número de tarjetas compradas ya que bajo ningún concepto se podría vender más tarjetas que las adquiridas. Por otra parte calcularemos los beneficios obtenidos en cada caso como los ingresos por venta más los ingresos por devolución menos el costo de elaboración de las tarjetas.

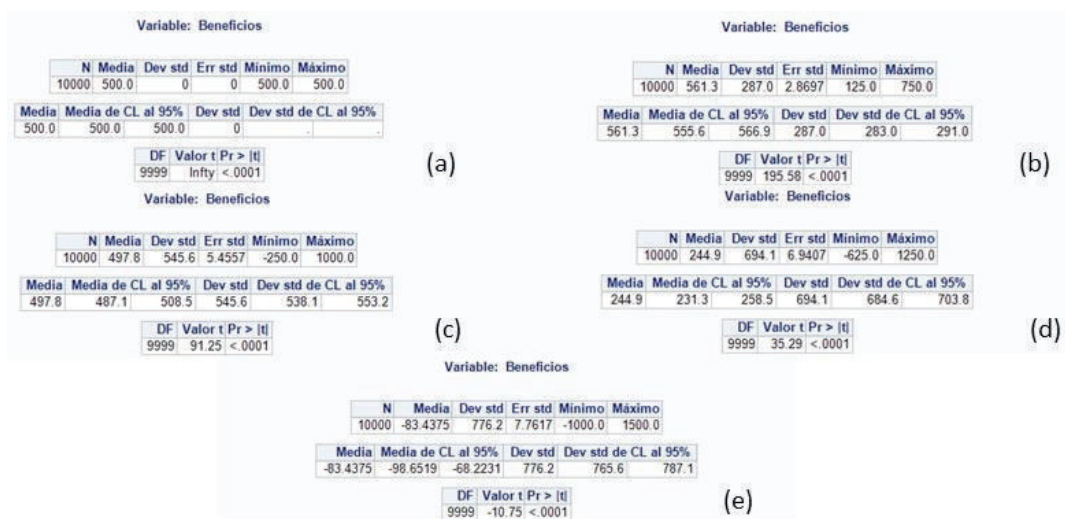


Figura 9.- Valores estadísticos obtenidos para diferentes números de tarjetas. (a) 100, (b) 150, (c) 200, (d) 250, (e) 300.

Principales estadísticos obtenidos para validar la condición resultante del sistema con diferentes condiciones iniciales, podemos apreciar que para un mayor número de tarjetas la desviación aumenta.

Finalmente obtenemos un resultado comparativo de las simulaciones realizadas:

Comparación de resultados obtenidos

tarjetas_compradas	beneficios	desviacion
100	500.000	0.000
150	561.250	286.968
200	497.813	545.566
250	244.938	694.070
300	-83.438	776.167

Figura 10.- Resultado comparativo de las simulaciones realizadas para el ejemplo de tarjetas.

En el resultado obtenido se aprecia que en la simulación con 150 tarjetas se maximiza el beneficio minimizando la desviación.

En el cual podemos ver que la decisión que mayor beneficio genera es la de comprar 150 tarjetas y además es la más estable (excluyendo la de 100 tarjetas).

3.3.3. Generación de Números Aleatorios Provenientes de Otras Distribuciones.

Para los casos en los que es necesario generar un número pseudo-aleatorio con otro tipo de distribución podemos utilizar la función RAND que nos permite generar números aleatorios de varias distribuciones continuas y discretas. Siempre que sea posible se utiliza la forma más simple de la distribución deseada.

La función RAND utiliza el generador mezclado de números aleatorios de Mersenne (RNG¹³) que fue desarrollado por Matsumoto y Nishimura [58]. El generador de números aleatorios tiene un período bastante grande ($2^{19937} - 1$) y cumple adecuadamente las propiedades estadísticas. Este período es un número primo de Mersenne de ahí que toma su nombre para el método de determinar el número estadístico. El algoritmo es una mezcla desplazada generalizada con retroalimentación (TGFSR¹⁴) método que da la otra parte del nombre del método. Las TGDSR dan a los RNG un alto orden de equidistribución (623 dimensiones con exactitud de 32 bits) lo que significa que existe una correlación muy pequeña entre 623 números pseudo-aleatorios.

Podemos con el algoritmo Mersenne-Twister RNG despreciar este efecto ya que el método nos garantiza que los primeros 2^{32} valores serán distintos. A diferencia de una muestra aleatoria uniforme que en un tamaño de 10^5 la probabilidad de encontrar al menos un duplicado es mayor al 50%.

De entre las distribuciones existentes para generar números aleatorios citaremos únicamente las que son de utilidad en este trabajo:

Distribución	Función de densidad de probabilidad	Observaciones
Exponencial	$f(x) = e^{-x}$	Definida para $x > 0$
Normal	$f(x) = \frac{1}{\lambda\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2\lambda^2}}$	Definida para $-\infty < x < \infty$ θ y λ parametrizables generalmente 0 y 1
Uniforme	$f(x) = 1$	Definido para $0 < x < 1$

Tabla 7.- Distribuciones y función de densidad de probabilidad más comunes.

El desarrollo de esta temática es importante para los resultados finales ya que nos permite definir la manera en la cual se plantea la programación de la simulación.

¹³ Por su nombre en inglés "Random Number Generator".

¹⁴ Sus siglas del nombre inglés: "Twisted Generalized Feedback Shift Register".

4. CAPÍTULO IV. RESOLUCIÓN DE UN JUEGO DE MINORÍAS

Hasta el momento se ha presentado las herramientas necesarias para poder proceder a cuantizar el juego de minorías. En este capítulo se presentarán el planteamiento algebraico para la aproximación cuántica del problema y la solución computacional para la aproximación clásica y cuántica del sistema.

4.1. Modelación Algebraica de un Juego Cuántico

Conforme a los conceptos introducidos en la primera parte iremos analizando uno por uno las definiciones y conceptos cuantizables.

Juego: El juego estará compuesto de la siguiente manera:

Jugadores: Se considerará un sistema cerrado de N individuos en donde cada uno puede encontrarse en un estado de spin $\frac{1}{2}$ o $-\frac{1}{2}$ al momento de observarlo, el momento de observación estará determinado como la traza de la matriz 2x2 de estado del jugador. Definiendo inicialmente los estados como las matrices similares a las de la ecuación 30:

$$\sigma_{\uparrow} = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_{\downarrow} = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (\text{Ec. 34})$$

Historia: Cada jugador parte de tener una historia asociada a las decisiones tomadas por él mismo en distintos m momentos¹⁵ anteriores al inicio del juego, los mismos que son construidos aleatoriamente para empezar el juego:

	Momento 1	Momento 2	Momento ...	Momento m	
Tipo de Selección	Determinado aleatoriamente	Determinado aleatoriamente	...	Determinado aleatoriamente	Definido por la "historia"
Valores Posibles	$\frac{1}{2}$ o $-\frac{1}{2}$	$\frac{1}{2}$ o $-\frac{1}{2}$...	$\frac{1}{2}$ o $-\frac{1}{2}$	2^m posibles estados
Ejemplo	$\frac{1}{2}$	$-\frac{1}{2}$...	$\frac{1}{2}$	$\begin{bmatrix} 1 & -1 & 1 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$

Tabla 8.- Posibles historias de un jugador en el juego de minorías cuántico.

Esquema para determinar el posible estado de un jugador cuántico dados un momento 1 y 2 determinados aleatoriamente.

¹⁵ El término momento se lo utiliza descriptivamente como un punto temporal o de transición donde se evalúa la función de pago de los jugadores.

El estado del jugador para posteriores instantes se determinará colapsando la función de pago (obteniendo la traza de la matriz 2×2).

Acciones: Las acciones están definidas por las matrices de rotación de Pauli más la matriz identidad. Al final de cada iteración cada jugador tendrá la posibilidad de mantener su acción o escoger una nueva acción, mediante la incorporación de los pagos virtuales cada acción será más o menos deseable para los jugadores siendo los puntos acumulados por cada estrategia el peso que se le otorga para la selección aleatoria de la estrategia de los jugadores

Estrategias: Cada uno de los individuos tienen acceso a las 4 acciones con las que podrá ir construyendo estrategias dadas por las matrices de rotación de Pauli y la matriz identidad lo cual permitiría tener una mayor diversidad de opciones para los jugadores que en el esquema clásico.

Pagos: La función de pago será determinado para cada jugador dependiendo de su estado final determinando como ganadores a aquellos jugadores que se encuentren en el estado menos poblado. Además de la función de pago propia para cada jugador se pretende establecer una función de pago para cada una de las posibles acciones que las definiremos como pagos virtuales.

A continuación analizaremos las posibles estrategias que pueden darse como la combinación de todas las acciones que puede tomar un jugador:

Para el estado inicial σ_{\uparrow} :

En el momento 1 los posibles estados resultantes serían:

$$\begin{aligned}\sigma_{\uparrow} \sigma_x &= \sigma_{\uparrow x} = \frac{1}{2} \sigma_x \\ \sigma_{\uparrow} \sigma_y &= \sigma_{\uparrow y} = \frac{1}{2} \sigma_y \\ \sigma_{\uparrow} \sigma_z &= \sigma_{\uparrow z} = \frac{1}{2} \sigma_z \\ \sigma_{\uparrow} I &= \sigma_{\uparrow}\end{aligned}\tag{Ec. 35}$$

En el momento 2 los posibles estados resultantes serían:

$$\begin{aligned}
\sigma_{\uparrow} \sigma_x \sigma_x &= \sigma_{\uparrow xx} = \frac{1}{2} \sigma_{\uparrow} & \sigma_{\uparrow} \sigma_x \sigma_z &= \sigma_{\uparrow xz} = -\frac{1}{2} i \sigma_y \\
\sigma_{\uparrow} \sigma_y \sigma_x &= \sigma_{\uparrow yx} = -\frac{1}{2} i \sigma_z & \sigma_{\uparrow} \sigma_y \sigma_z &= \sigma_{\uparrow yz} = \frac{1}{2} i \sigma_x \\
\sigma_{\uparrow} \sigma_z \sigma_x &= \sigma_{\uparrow zx} = \frac{1}{2} i \sigma_y & \sigma_{\uparrow} \sigma_z \sigma_z &= \sigma_{\uparrow zz} = \sigma_{\uparrow} \\
\sigma_{\uparrow} I \sigma_x &= \sigma_{\uparrow x} = \frac{1}{2} \sigma_x & \sigma_{\uparrow} I \sigma_z &= \sigma_{\uparrow z} = \frac{1}{2} \sigma_z \\
\sigma_{\uparrow} \sigma_x \sigma_y &= \sigma_{\uparrow xy} = \frac{1}{2} i \sigma_z & \sigma_{\uparrow} \sigma_x I &= \sigma_{\uparrow x} = \frac{1}{2} \sigma_x \\
\sigma_{\uparrow} \sigma_y \sigma_y &= \sigma_{\uparrow yy} = \frac{1}{2} i \sigma_{\uparrow} & \sigma_{\uparrow} \sigma_y I &= \sigma_{\uparrow y} = \frac{1}{2} \sigma_y \\
\sigma_{\uparrow} \sigma_z \sigma_y &= \sigma_{\uparrow zy} = -\frac{1}{2} i \sigma_x & \sigma_{\uparrow} \sigma_z I &= \sigma_{\uparrow z} = -\frac{1}{2} \sigma_z \\
\sigma_{\uparrow} I \sigma_y &= \sigma_{\uparrow y} = \frac{1}{2} \sigma_y & \sigma_{\uparrow} II &= \sigma_{\uparrow}
\end{aligned} \tag{Ec. 36}$$

Podemos extrapolar los resultados para expresarlos de manera simplificada como:

$$\sigma_{\uparrow m} = \frac{1}{2} \sigma_{\uparrow} \prod_{i=(1,m)} \sigma_i \tag{Ec. 37}$$

$$\sigma_{\downarrow m} = \frac{1}{2} \sigma_{\downarrow} \prod_{i=(1,m)} \sigma_i \tag{Ec. 38}$$

Considerando esto y los posibles resultados luego de la primera iteración podemos validar que existen 14 estados finales posibles:

Estado	Representación Matricial	Espín Asociado
σ_{\uparrow}	$\frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\frac{1}{2}$
σ_{\downarrow}	$\frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\frac{1}{2}$
$\frac{1}{2} \sigma_x$	$\frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$-\frac{1}{2}$
$-\frac{1}{2} \sigma_x$	$\frac{1}{2} \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$	$\frac{1}{2}$
$\frac{1}{2} i \sigma_x$	$\frac{1}{2} \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$	$\frac{1}{2}$
$-\frac{1}{2} i \sigma_x$	$\frac{1}{2} \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}$	$-\frac{1}{2}$
$\frac{1}{2} \sigma_y$	$\frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$-\frac{1}{2}$
$-\frac{1}{2} \sigma_y$	$\frac{1}{2} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	$-\frac{1}{2}$
$\frac{1}{2} i \sigma_y$	$\frac{1}{2} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$\frac{1}{2}$
$-\frac{1}{2} i \sigma_y$	$\frac{1}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	$\frac{1}{2}$
$\frac{1}{2} \sigma_z$	$\frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$-\frac{1}{2}$
$-\frac{1}{2} \sigma_z$	$\frac{1}{2} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	$-\frac{1}{2}$
$\frac{1}{2} i \sigma_z$	$\frac{1}{2} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$	$\frac{1}{2}$
$-\frac{1}{2} i \sigma_z$	$\frac{1}{2} \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}$	$\frac{1}{2}$

Tabla 9.- Posibles estados finales del sistema y su espín asociado.

En los estados accesibles mostrados vemos que el juego sigue siendo simétrico ya que existe la misma cantidad de estados que apuntan al spin $\frac{1}{2}$ como los que apuntan a $-\frac{1}{2}$.

En la tabla 9 podemos apreciar lo que sucede al colapsar la función de onda en cada uno de los estados posibles generados por las rotaciones dadas por las matrices de Pauli. Este resultado lo utilizaremos en la simulación discreta del problema como resultado una vez analizados los posibles estados finales después de ciertas rotaciones.

4.2. Simulación del Juego Clásico

Para realizar este juego se tomará en cuenta el problema de “El Farol”, un bar que cada jueves brinda un espectáculo en vivo, sin embargo la capacidad de personas sentadas que pueden ver el mismo es de apenas 60, este bar tendrá 101^{16} clientes regulares. Los mismos tendrán accesibilidad a 2 estrategias, la varianza σ^2 va a disminuir a medida que m aumenta, es decir sería menor a la que correspondería a decisiones aleatorias $\sigma_r^2 = \frac{N}{4}$ a partir de $m = 4$, alcanza el valor mínimo entre $m = 5$ y $m = 6$ y crece acercándose asintóticamente a $\frac{N}{4}$ para cuando $m \rightarrow \infty$.

Para iniciar el juego los participantes seleccionarán aleatoriamente en cada jugada la estrategia a seguir entre las s disponibles ya que todas las posibles estrategias inician con el mismo puntaje de estrategia. A medida que van ganando y perdiendo, las estrategias irán acumulando puntos que son los que utilizará el agente que tiene las estrategias por lo que se podría decir que el sistema está “aprendiendo” inductivamente a jugar [5]. El método de selección de los lados determinados como minoritarios que realizan los jugadores según las estrategias asignadas, se repite un número T de veces. El modelo descrito con $N = 101$ jugadores tiene un tiempo transitorio cercano a 1.000 eventos. Pasado este transitorio, durante el cual los resultados se asemejan a un comportamiento de decisiones aleatorias, el resultado emergente se estabiliza en otros comportamientos que podrían mejorar o empeorar el esquema aleatorio, depende del número de estrategias, el tamaño de la población y la longitud de la historia. Este estado estacionario es el que

¹⁶ Se ha considerado un número impar de clientes para evitar posibles empates en los resultados.

relacionaremos con coordinar actividades, es decir los participantes conocen el comportamiento de los otros y “deciden” cuando debe asistir cada uno al lugar del evento.

De aquí que para la población de $N = 101$ jugadores con 2 estrategias posibles cada uno, la varianza σ^2 disminuye a medida que aumenta m , lo que va haciendo que cada vez sea menor a la que correspondería a decisiones aleatorias $\sigma^2 = \frac{N}{4}$ a partir de $m = 4$, alcanzando el mínimo entre $m = 5$ y $m = 6$ y crece acercándose asintóticamente a $\frac{N}{4}$ para cuando $m \rightarrow \infty$, como muestra la figura 11.

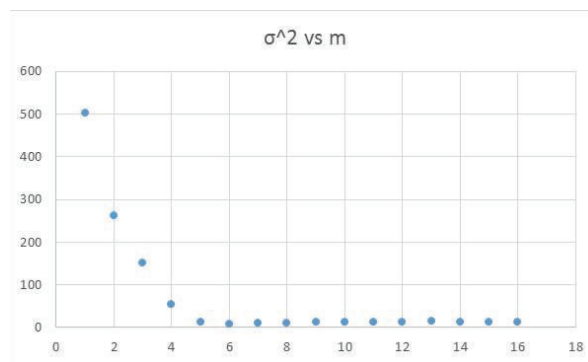


Figura 11.- Varianza de los jugadores que escogieron asistir (Opción “1”).

Comparación del resultado de la varianza de los resultados obtenidos por los jugadores que en el juego de minoría de “El Farol” decidieron asistir.

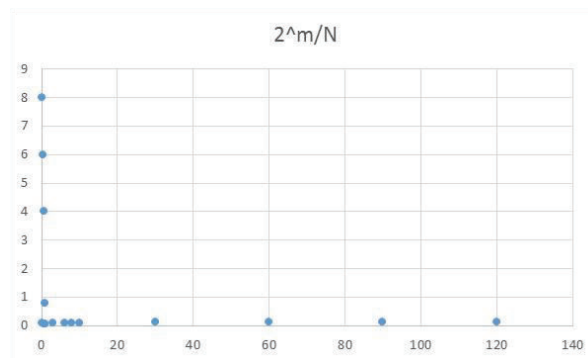


Figura 12.- Varianza reducida $\frac{\sigma^2}{N}$ vs número de historias por jugador $\frac{2^m}{N}$.

Se incorpora el criterio de varianza reducida, que hace referencia a la figura 11 pero su valor se divide para el número de jugadores. Lo cual permite estudiar las propiedades del sistema independiente de la población.

Se toma únicamente la posibilidad de que cada jugador escoja de entre 2 estrategias posibles ya que un mayor número de éstas no aportaría cualitativamente nuevas propiedades cooperativas al modelo, más bien podría

disminuir la coordinación, lo cual podría generar que el tiempo de aprendizaje de los jugadores sea mayor y el tiempo computacional para la simulación también incremente [59].

Uno de los problemas que se pueden presentar es que exista una dependencia de la evolución del sistema con variables como el tamaño de la población, la memoria de los jugadores y la varianza de la función de pago previa, por lo que para independizar la evolución del sistema se introduce parámetros reducidos como el número de historias posibles por jugador y la varianza reducida.

$$z = \frac{2^m}{N}, \mu = \frac{\sigma^2}{N} \quad (\text{Ec. 39})$$

Esta definición de los parámetros nos permite hallar el mínimo de μ de la siguiente manera

$$\begin{aligned} \mu &\cong 0.2 \text{ cuando } z_c \cong 0.35 \\ \mu &\cong \frac{1}{4} \text{ cuando } z \rightarrow \infty \end{aligned} \quad (\text{Ec.40})$$

Lo cual es válido de manera general en todo M , m y σ^2 con $s = 2$.

Normalmente en el análisis del juego podemos dividir el comportamiento de la figura 26. En tres regiones [60], La primera corresponde a m pequeño, donde $\sigma^2 > \frac{1}{4}$, la segunda zona comprende entre: $\sigma^2 = \frac{1}{4}$ y σ^2 mínimo y el último desde σ^2 mínimo en adelante, como lo podemos visualizar en la figura 19.

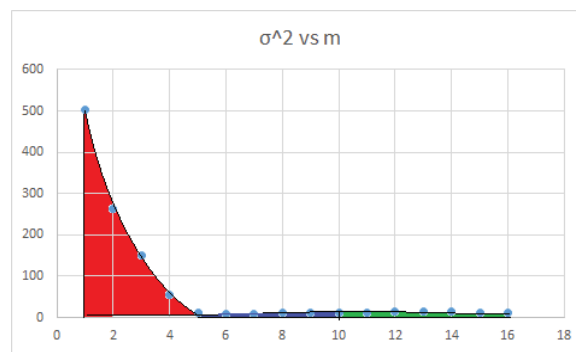


Figura 13.- Gráfico de la varianza distribuido por regiones.

Se aprecian las 3 regiones presentes por la cantidad de historias consideradas, en la zona de color rojo con una gran pendiente decreciente hasta el primer punto de inflexión, la zona azul con pendiente creciente hasta el segundo punto de inflexión, y la zona verde con pendiente negativa hasta el máximo de m .

En el estudio realizado no arroja determinadas características de cada región, sin embargo las propiedades de la región eficiente concuerda con lo hallado por Manuca, Li, Riolo y Savit [60], para $m = 2$ (4 historias posibles) y $s = 2$ se encuentra que en las situaciones impares de una historia dada, las desviaciones son leves (podríamos argumentar que la población termina distribuida de una manera estable), mientras que en las ocurrencias pares esta desviación es mayor. Podemos interpretar de aquí que al tener un m pequeño se puede dar la situación en la cual un jugador tenga sus dos estrategias similares.

Para respaldar esta observación tomaremos un jugador dado que disponga de dos estrategias similares. Para esto bastaría hallar un jugador con estrategias que se diferencien únicamente en una historia. La primera vez que aparezca el mismo en el juego sus estrategias estarán empatadas. La elección por una de las dos vendría a ser aleatoria y una de las estrategias resultaría ganadora y la otra perdedora. Hasta que volvamos a ver esa historia, las estrategias del jugador acumularán igual cantidad de puntos ya que hacen las mismas predicciones y por lo ganarían y perderían equitativamente. La segunda vez que aparezca en la historia mencionada la estrategia que escogió de manera aleatoria la primera vez tendría un punto adicional por lo que tendría mayor probabilidad de ser seleccionada por el jugador. Ahora consideremos que no era un único el jugador con esta particularidad, sino más bien habría un gran grupo con lo mismo: los jugadores con estrategias que difieran sólo en una determinada historia y hubieran ganado en la ocurrencia anterior esta vez repetirían su jugada y los que hubieran perdido ahora tomarían la otra estrategia. De esta manera se podría ocasionar que muchos jugadores se encuentren del lado que habría resultado minoritario en la ocurrencia impar anterior de la historia. La desviación de los resultados podría ser grande y la estrategia ganadora ahora sería la estrategia que habría perdido en la ocurrencia impar, por lo tanto, los puntajes acumulados de las estrategias estarían empatados otra vez y la tercera ocurrencia (que vendría a ser la segunda impar) volvería a darse de carácter aleatoria y daría una distribución de población similar a la media en cada grupo y el ciclo se repetiría. Esto nos diría que, leves desviaciones de la media en las apariciones impares de una historia (similares a una jugada aleatoria) vendrían seguidas por grandes desviaciones en las ocurrencias pares y el lado que era minoritario en la ocurrencia par resulta ser el opuesto al anterior impar. De una

manera cualitativa esta dinámica de período 2 persiste mientras exista la posibilidad razonable de que el puntaje relativo entre las estrategias de un jugador no varíe entre dos ocurrencias sucesivas de una dada historia.

Este modelo también es analizado en el trabajo de Manuca, Li, Riolo y Savit [60] mediante modelos de caminata aleatorio en donde se demuestra que para m relativamente pequeños este comportamiento es esperable, y se convierte en crecientemente improbable a medida que m aumenta, también el número esperado de jugadores que toman nuevamente la misma historia con el puntaje relativo entre estrategias sin modificación respecto a la ocurrencia anterior es de orden $\frac{N}{(2^m)^{\frac{1}{2}}}$. De

hecho si imaginamos un juego de minorías con $s = 2$ donde las historias con más visitas tienen idéntica probabilidad (comúnmente denominamos a esta situación ergodicidad [61]), entonces cada historia tendrá una visita cada 2^m jugadas. Por esto, para cada jugador dado que tenga ambas estrategias con empate en puntos virtuales de la aparición de cierta historia, la probabilidad de que en la próxima aparición de la historia mencionada vuelva a tener las estrategias empatadas se correspondería directamente con una caminata aleatoria unidimensional que pase por 0 exactamente cada 2^m pasos, esto se puede expresar como:

$$P = \left(\frac{2^m}{2^{m-1}} \right) 2^{-2m} \approx \sqrt{\frac{2}{2^m \pi}} \quad (\text{Ec. 41})$$

Y de aquí podemos determinar que el número esperado de jugadores que se encuentren con la misma historia sería:

$$N \sqrt{\frac{2}{2^m \pi}} \quad (\text{Ec. 42})$$

4.3. Aproximación Analítica

Existen varias versiones para realizar el análisis que hacen referencia a la formalización del juego mediante el uso de técnicas de física estadística o vidrios de espín [6] [62] [63]. Los cuales en su mayoría fueron realizados sobre el comportamiento en la región de información eficiente, en la cual $z < z_c$. Aquí analizaremos una aproximación del juego de minorías cuantitativamente útil para la región de información eficiente a partir de una versión de Juego de Minoría de

estrategia completa (FSGM¹⁷) tomando como referencia principal la presentada en el trabajo de Caridi y Ceva [64]. A este planteamiento se ha realizado algunas variaciones que permiten simplificar los resultados, como aplicar las ideas a otras variantes del juego, para lo que citaremos las principales definiciones de una manera formal con la notación y definiciones requeridas.

Sea $\#C$ el número de elementos presentes en el conjunto C lo llamaremos el cardinal del conjunto.

Definición 1: Dado un m tal que $2 \leq m \in \mathbb{N}$, llamaremos historia a toda cadena de longitud m formada por ceros y unos. Definiendo con $H = 2^m$ a la cantidad de cadenas diferentes para un m dado y con:

$$S_H = \{h_1, \dots, h_H\} \quad (\text{Ec. 43})$$

el conjunto completo de historias, para el cual $\#S_H = H$.

Definición 2. Una estrategia de un juego de minorías es una función del conjunto S_H en el conjunto $\{0,1\}$. Es decir la estrategia única asigna unívocamente una respuesta binaria a cualquier historia dada. Si definimos como S_L al conjunto con todas las estrategias posibles podemos hallar que $\#S_H = 2H$. De manera general se utilizará la notación $L = 2H$.

Definición 3. Dado un s tal que $2 \leq s \in \mathbb{N}$, un jugador del juego de minorías estará definido como el conjunto de s elementos tomados (con repetición de S_L). Se definirá S_N al conjunto completo de jugadores posibles diferentes y N al cardinal $\#S_N$. Si dos jugadores se pueden representar como conjuntos semejantes, entonces diremos que uno es una copia del otro.

Definición 4. Dado un s , $2 \leq s \in \mathbb{N}$, y un $N \in \mathbb{N}$, una instancia del juego de minorías es un conjunto I de N jugadores tomados (con repetición de S_N). Para cada elemento J del conjunto I se tiene además un arreglo $(v_1, \dots, v_s) \in \mathbb{N}_0^s$ que acumula la puntuación virtual de cada estrategia y una función de decisión:

$$D: \mathbb{N}_0^s \rightarrow J \quad (\text{Ec. 44})$$

¹⁷ Por las siglas en inglés de Full Strategy Minority Game.

La misma que determina la estrategia a utilizar como aquella que posee el máximo de los puntos virtuales (si existe un único l tal que: $v_l = \max\{v_i\}_{1 \leq i \leq s}$) o un elemento al azar (uniforme) de entre aquellos que alcanzan el máximo en otro caso.

Para los siguientes análisis tendremos en cuenta que $s = 2$ para simplificar el análisis. Con lo que tendremos que:

$$N = \#S_N = \binom{L}{2} + L \quad (\text{Ec. 45})$$

En las instancias I del juego de minorías generadas aleatoriamente puede suceder que algunos jugadores potenciales no participen del juego o que múltiples copias del mismo jugador puedan participar. En el FSMG normalmente se excluyen ambos casos. Elegiremos el número de jugadores tal que sea exactamente N y una sola versión de cada jugador potencial será permitida.

Definición 5. Definiremos el FSMG como la única instancia del juego de minorías en la cual se verifica que $I = S_N$.

Siguiendo el trabajo de Cardi y Ceva [64], ciertas simetrías se pueden ver presentes sólo parcialmente en el juego de minorías, donde las instancias se general aleatoriamente, esto puede ser explotado en el FSMG con el fin de conducirnos a resultados teóricos satisfactorios.

En lo anterior encontramos las definiciones generales para un juego de minoría. Para continuar con la estructura del juego que deseamos analizar en este estudio introduciremos las siguientes definiciones.

Definición 6. Dada una instancia I del juego de minorías, una ejecución E del juego es un par $E = \{\widehat{h}^0, I\}$ con la secuencia:

$$X_E = \{x_{-m+1}, \dots, x_0, x_1, x_2, \dots\} \quad (\text{Ec. 46})$$

Donde $x_k = 0,1$, definimos la cadena de resultados minoritarios obtenidos al ejecutar el juego partiendo de una instancia E y con:

$$S_E = \{\widehat{h}^0, \widehat{h}^1, \dots\} \quad (\text{Ec. 47})$$

Donde la secuencia de historias se encuentra dada por:

$$\widehat{h}^t = (x_{t-m}, x_{t-m+1}, \dots, x_t) \quad (\text{Ec. 48})$$

De esta forma $\widehat{h}^k \in S_H$ para cualquier k , es decir $\widehat{h}^k = h_i$ para algún $1 \leq i \leq H$. Además para \widehat{h}^0 , coincide con las historias iniciales aleatorias y dadas \widehat{h}^k y \widehat{h}^{k+1} diremos que la primera es la predecesora de la segunda y que la segunda es la sucesora de la primera.

Notemos que la información dada por X_E es equivalente a la dada por S_E puesto que cada secuencia puede ser reconstruida de otra y que E no determina unívocamente a X_E en la versión estándar del juego de minoría, ya que no se trata de un juego determinístico, sin embargo se puede dar la situación simplificada. Notemos también que para este marco formal una instancia I del juego de minoría o aún una realización E no contiene información acerca del modo en que el juego se llevará a cabo en lo que a las reglas respecta. Es decir considerando los jugadores que toman decisiones en base al sistema de puntos virtuales acumulados en las estrategias. Nuestro interés se centrará en estudiar el caso más general que es el juego de minorías tradicional.

Definición 8. Llamaremos regla de actualización RA del juego a la función $RA: N \rightarrow S_H$ para la que en cada paso k del juego nos indica la historia que es escogida por los jugadores a la hora de hacer la decisión durante el paso de selección k . A menos que sea definitorio no se hará mención a la regla que podría utilizar el jugador.

Definición 9. Dado un juego de minorías con una regla de actualización y un paso del juego $k \in N$, se llamará historia actual a $RA(k) = h_p^k$ o simplemente h_p si el contexto es claro para identificar que nos encontramos en el paso k para decidir la elección. Finalizada la fase de elección y actualizados los puntos virtuales el juego se halla en el paso $k+1$, en donde los jugadores podrán utilizar la información h_p^{k+1} .

Estas definiciones nos permitirán de manera más clara en la simulación verificar que todos los puntos necesarios para modelar un juego de minorías se estén cumpliendo.

4.4. Simulación del juego cuántico

Para la simulación del juego tomaremos las definiciones dadas en la parte anterior para realizar las adaptaciones adecuadas de tal manera que se pueda realizar la simulación del juego con las consideraciones cuánticas.

La implementación de la simulación fue realizada en lenguaje SAS utilizando el ambiente educativo de SAS Educational Software, con la intención de aprovechar las posibilidades del lenguaje SAS principalmente en los temas referentes a la orientación a objetos. En particular, las capacidades de herencia y polimorfismo de los lenguajes orientados a objetos fueron muy importantes para mantener acotado el volumen de la implementación. La implementación aquí mencionada se basa en un programa principal desarrollado de tal forma que el mismo código permite ejecutar las variantes clásica y cuántica del juego de minorías, por medio de las configuraciones apropiadas basándose en una interfaz propia del usuario para referirse a la cantidad de historias, jugadores, iteraciones. Además de simplificar el esfuerzo destinado a la implementación, esta característica del código permite asegurar que los experimentos destinados a comparar distintas parametrizaciones del juego se realizan en igualdad estricta de condiciones [18] [26].

La simulación del juego de minorías se la ha estructurado alrededor de las clases Población y Jugador. Una instancia de la clase Población representa a la población de una instancia del juego, compuesta por un conjunto de jugadores no necesariamente homogéneos. Por su parte, la clase Jugador es una clase abstracta que reúne el comportamiento común a todos los jugadores, y se implementan subclases para aclarar los distintos tipos de jugadores y estrategias involucrados en el juego. Hay que notar que las estrategias que conforman la base de las decisiones de los jugadores también pueden ser consideradas como subclases apropiadas de Jugador dentro del juego, y puesto que el comportamiento de una estrategia no cambia del comportamiento de un agente. Esta característica del diseño permite, por ejemplo, incorporar fácilmente jugadores con una sola estrategia generando como instancia directamente la clase que representa a una estrategia, que en este caso también responde a la interfaz y funcionalidad de un jugador. Por otra parte, puesto que los jugadores mantienen un registro de los puntos que han obtenido en el juego, no es necesario generar código adicional para

el registro de los puntos virtuales de las estrategias, sino más bien en cada registro se irá almacenando los resultados de su propia estrategia y puntos acumulados al seleccionarla.

A continuación se realiza a una descripción funcional de las clases consideradas para la simulación:

Población: Una instancia de la clase Población representa una instancia del juego de minorías, con el conjunto de los agentes que participan en el juego y la información relevante a la ejecución de la simulación. Contiene un arreglo de Jugador que, debido a los requerimientos del lenguaje SAS en cuanto al polimorfismo, se implementa como un arreglo de variables asociados al registro de cada Jugador. Durante la construcción de una instancia de Población, se construyen en memoria dinámica las instancias de las subclases de Jugadores especificadas por la configuración de la ejecución y se agregan a este registro los resultados de las instancias. El tipo de los jugadores que se agregan aclara la variante del juego de minorías que se ejecutará.

En caso de que la configuración requiera que se utilicen jugadores con acceso a estrategias cuánticas, el constructor de Población ubica los jugadores cuánticos con una variable llamada observación la que nos permite determinar el tipo de jugador seleccionado, y de manera aleatoria se marca esta variable para los agentes que serán interactivos, de acuerdo al número o la probabilidad de que un jugador tenga acceso a estrategias cuánticas, especificada también en la configuración. Una vez seleccionados los jugadores que tendrán acceso a las estrategias cuánticas, se almacena la información de cada jugador si es cuántico o no y. Este mecanismo permite a los jugadores abstraerse de la configuración en forma de tabla, omitiendo la posibilidad de que exista algún patrón de interacción.

La clase Población cuenta con el método de selección de estrategia, que se ejecuta en cada paso de la simulación. Durante cada paso, se permite a los jugadores que ejecuten este proceso de selección. Cada subclase de Jugador implementa distintos comportamientos en esta fase. Una vez realizada la elección, este método decide cuál es la decisión minoritaria e informa a los agentes el resultado del juego. Además de este método, la clase cuenta con métodos para consultar las cantidades de jugadores, sus pesos y puntajes obtenidos.

Jugador: La clase Jugador representa un jugador genérico, implementando la funcionalidad común a todos los jugadores. Es una clase abstracta, lo cual quiere decir que algunos de los métodos no pueden implementarse directamente sino más bien deben implementarse en las subclases, y por este motivo no es posible generar directamente esta clase directamente dentro del código. La interfaz principal de *Jugador* está dada por los siguientes métodos: *PrimeraRonda*. En este método se implementa la primera fase de decisión, que corresponde a seleccionar qué opción tendrá el jugador para la primera ejecución del juego. En función de la historia del juego y las características del jugador. Lo llamamos método protegido, virtual y abstracto *TomarDecision* que no se implementa en Jugador y debe, por lo tanto, implementarse en las subclases con funcionalidad específica de cada tipo de agente.

Decisión Segunda Ronda: Este método retorna la decisión final de la elección, que se implementa por medio del método protegido, virtual y abstracto *tomarDecisionFinal* que, nuevamente, se debe definir en las subclases.

DecisionGanadora: Este método permite informarle al jugador cuál fue la decisión minoritaria en la ronda anterior, y es utilizado por las subclases de Jugador cuya decisión se basa en la historia del juego para actualizar la historia. Al finalizar cada ronda de elección, la instancia de Población informa a todos sus agentes cuál fue la decisión ganadora mediante este método. Además de otros métodos de consulta, la clase Jugador mantiene como datos la cantidad de rondas del juego, los puntos obtenidos por el jugador a lo largo del juego, el peso del jugador y, en caso de ser cuántico la observación acerca de esta particularidad. Estos datos se actualizan dentro de los métodos implementados por el Jugador, dado que esta funcionalidad es común a todas las versiones de jugadores.

Las subclases de Jugador implementan los distintos tipos de jugador que se consideran dentro de la aplicación, y que fueron utilizados a lo largo de este trabajo. Las subclases que se consideran son las siguientes:

Determinístico: Esta subclase representa a una estrategia individual. El constructor de esta subclase toma como parámetro el tamaño de la muestra y genera aleatoriamente los resultados para cada salida del juego. La

implementación del método *TomarDecision* sencillamente retorna la indicación de la estrategia de acuerdo con la historia actual del juego.

Aleatorio: Esta subclase corresponde a un agente que toma una decisión aleatoria en cada ronda, con lo cual la implementación de *TomarDecision* genera un número aleatorio y retorna con probabilidad 1/2 una de las dos posibles decisiones.

Insistente: Esta subclase corresponde a un jugador que en cada paso selecciona la misma estrategia.

Para la simulación cuántica se consideran las mismas 3 subclases de jugadores refiriéndolos como el nombre inicial con una Q al final (*DeterminísticoQ*, *AleatorioQ*, *InsistenteQ*). El análisis que se lleva a cabo toma en cuenta los resultados comparativos al ejecutar distintas poblaciones de cada una de las clases mencionadas.

Bajo esta metodología se presenta la programación que para facilidad de la secuencia de análisis se la automatizó en un proyecto de SAS Enterprise Guide ejecutable en el cual se puede llevar a cabo las simulaciones con variación de parámetros¹⁸. Tomando en cuenta como principales parámetros y su descripción los siguientes:

N_Jugadores: Determina la cantidad de jugadores que se incluyen en el sistema teniendo como restricción que debe ser un número entero e impar de jugadores como mínimo 1 y como valor dado por defecto y utilizado para las obtener los resultados la cantidad de 101.

Tipo_Jugadores: Se citan los 6 tipos de jugadores posibles (*Aleatorio*, *AleatorioQ*, *Determinístico*, *DeterminísticoQ*, *Insistente*, *InsistenteQ*) que pueden ser parte del sistema y se solicita el número de cada uno de ellos que serán parte del sistema, la suma total de los jugadores entre los 6 tipos debe ser igual al número total de jugadores, en caso de que el número sea menor se asigna como *Determinístico* a la cantidad adicional de jugadores presentes, en caso de que exista un mayor número de jugadores al sumar los distintos tipos de jugadores que los definidos en

¹⁸ El proyecto mencionado se lo adjunta en el Anexo 3 de este trabajo.

el $N_{Jugadores}$, se reasigna el valor de $N_{jugadores}$ a la sumatoria de los tipos de jugador solicitados.

N_Iteraciones: especifica el número de veces que se va a realizar la simulación, como mínimo deben realizarse las 3 primeras iteraciones (1 aleatoria, 1 para escoger la opción de los diferentes tipos de jugadores y 1 para consolidar las estrategias).

Condiciones de estado inicial:

Estado_0: determinado de manera aleatoria entre 1 o 0 como la condición de asistir o no al evento. Para esta función se ha tomado el generador de números aleatorios entre 0 y 2 y se ha tomado el valor entero de este resultado.

Resultado_0: determinado por las condiciones dadas en el Estado_0 se obtiene la opción que resultaba como ganadora en la iteración inicial del juego.

Condiciones transitorias del juego:

Puntaje_ganador: se almacena el valor numérico de jugadores que seleccionaron la decisión ganadora en la última iteración registrada del juego.

Opcion_ganadora: se almacena la decisión que resultó ganadora en la última iteración registrada del juego.

Una vez determinadas todas las opciones que puede tomar un jugador se observa la evolución de las mismas y se busca determinar si existe alguna estrategia dominante.

Se analizó el comportamiento de cada una de las estrategias planteadas para los jugadores como si fueran puras obteniendo los siguientes resultados para la estabilidad del sistema los mismos que se observan en la Figura 14 y se detallan a continuación:

Estrategia Aleatoria: Se puede apreciar que para un juego dominado por una estrategia aleatoria no se encuentra un equilibrio en el paso del tiempo, debido a que para cada una de las iteraciones, el sistema no logra generar un aprendizaje adecuado con la situación anterior, y sería el estado base del que partiríamos en caso de que no existiere ningún tipo de inteligencia que ingresemos al juego, ya que cada momento de la evolución representaría un nuevo momento inicial.

Estrategia Insistente: En este caso los jugadores toman una estrategia que evita que nuevos jugadores vuelvan a ganar el juego manteniendo el estado de equilibrio del mismo siempre como ganadora la opción inicial. Es decir cada jugador mantendrá la decisión inicial, sea que eso represente un ganar o perder directamente, sino más bien depende del sistema que se estabiliza a una selección rápidamente.

Estrategia Determinística: El aprendizaje de los jugadores en las distintas iteraciones genera que la evolución de los resultados sea más equitativa. Observándose esto en que la distribución de ganadores tiende a estabilizarse, sin embargo aún existen fallas o errores en momentos consecutivos de la simulación.

Estrategia Aleatoria Cuántica: En esta estrategia en cuanto a la evolución no se aprecia una considerable mejora con respecto a su símil clásico, ya que a pesar de brindarles nuevas oportunidades o características a los jugadores, los mismos no las aprovechan directamente sino sólo parten de un momento inicial cada vez que se genera la simulación.

Estrategia Insistente Cuántica: Para esta estrategia notamos que la evolución con respecto a la estrategia clásica similar corresponde a que en este caso los jugadores se inclinan por otorgar la ganancia a la estrategia opuesta a la original aleatoria, de esta manera vemos que hay un aprendizaje con el momento inicial de parte de los jugadores, sin embargo luego no se presenta una variación.

Estrategia Determinística Cuántica: Conforme a lo esperado cuando toda la población aplica una estrategia determinística cuántica, es decir va aprendiendo de la memoria de los sucesos que se han presentado los resultados de la evolución son estables en cuanto a que en cada iteración gana una de las opciones.

Conforme a lo esperado cuando toda la población aplica una estrategia determinística cuántica, es decir va aprendiendo de la memoria de los sucesos que se han presentado los resultados de la evolución son estables en cuanto a que en cada iteración gana una de las opciones, y de esta manera queda confirmado que la opción de una estrategia cuántica permite una mejora para el sistema general.

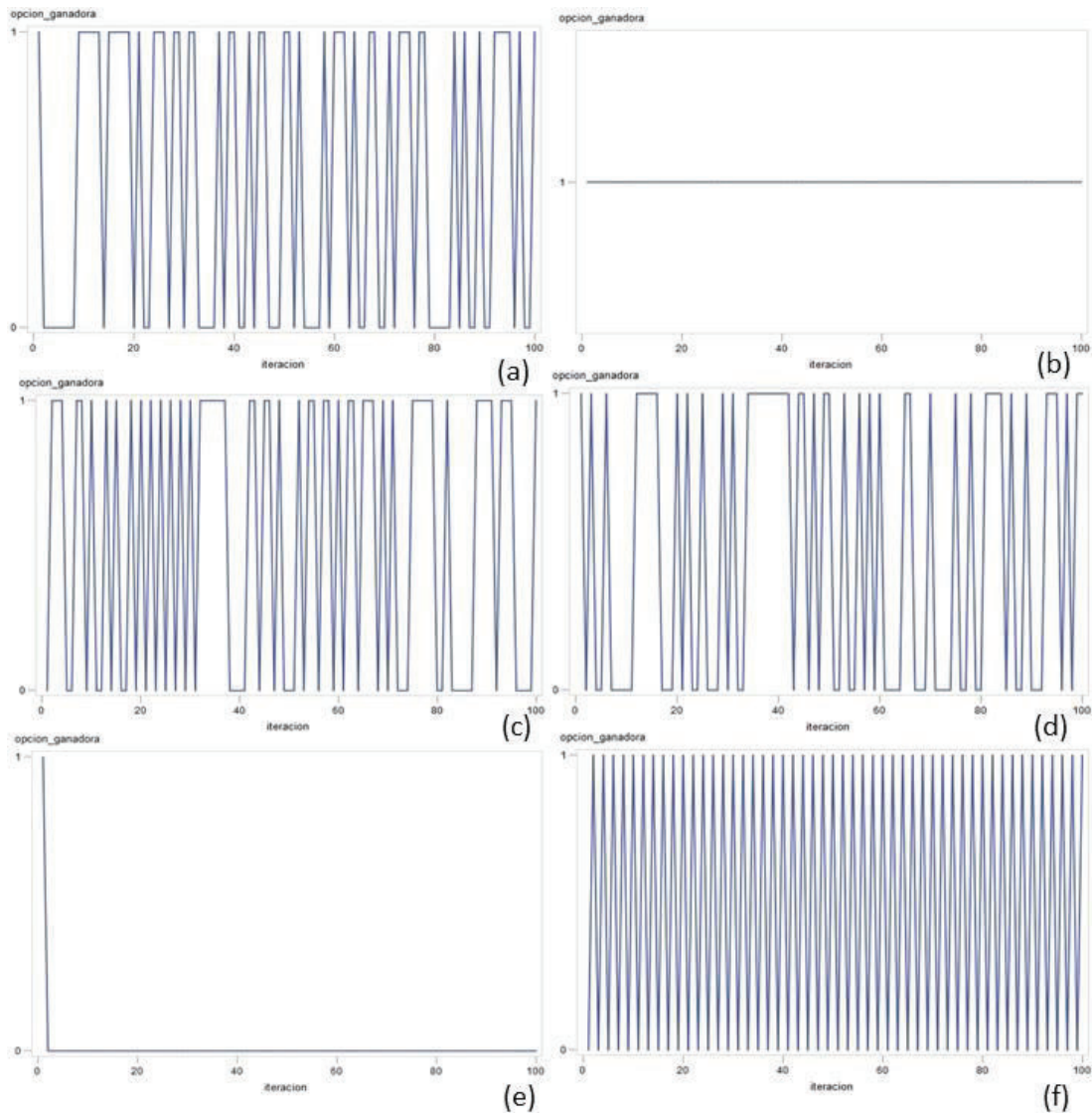


Figura 14.- Evolución de los resultados de la simulación de un juego de minorías con estrategias puras. (a) Aleatoria, (b) Insistente, (c) Determinística, (d) Aleatoria Cuántica, (e) Aleatoria Cuántica, y (f) Determinística Cuántica.

Resultados de la simulación obtenidos para las diferentes estrategias, se puede observar que las estrategias (a), (c) y (d) presentan muchas irregularidades, mientras que la estrategia (b) mantiene siempre el mismo resultado ganador, la estrategia (e) mantiene como ganadora a la opción contraria a la inicial del sistema, la estrategia (f) muestra una variabilidad ideal, haciendo que el juego sea más estable.

Para determinar el efecto de introducir poblaciones ajenas al sistema puro previamente indicado, se efectuó la simulación con la estrategia determinística clásica introduciendo jugadores cuánticos hasta encontrar la población mínima requerida para estabilizar el sistema. Se tomó en cuenta la población total y máxima de 101 jugadores.

Intrusos	Estabilidad	Errores
10	93	15
15	58	10
20	64	9
25	62	6
30	62	3
35	51	3
40	54	1
51	1	0

Tabla 10.- Efecto de los Intrusos determinísticos cuánticos y en el sistema determinístico clásico.

Cantidad de jugadores necesario para afectar a la estabilidad de población, en donde se aprecia que a mayor cantidad de intrusos en el sistema menos irregularidades o errores se presentan en el mismo.

En la Tabla 10 podemos ver los resultados de introducir jugadores que siguen una estrategia determinística cuántica, en una población pura de jugadores que disponían de una opción determinística clásica. De manera general al tener una población mayoritaria la estrategia cuántica envuelve al sistema y permite llegar al equilibrio desde la primera iteración, sin embargo el efecto de la presencia de estos jugadores se nota claramente desde los 30 intrusos en donde prácticamente no se observan errores o imperfecciones en el sistema.

5. CAPÍTULO V: CONCLUSIONES

Fue posible elaborar una solución de un juego de minorías con muchos jugadores, estudiar su evolución clásica haciendo referencia a las posibles historias que puede acceder el jugador, introducir la cuantización para las estrategias posibles y encontrar una solución de la experiencia de la simulación que tiene un grado de similitud adecuado con la solución teórica encontrada, de esta manera nos fue factible hallar nuevos puntos de equilibrio del sistema más estables por la presencia de las estrategias cuánticas.

Se investigó las bases matemáticas de la teoría de juegos, en especial sobre los juegos de minorías, lo que permitió lograr un planteamiento teórico y conceptual adecuado del sistema y de las variables que influyen a su evolución.

Se han analizado los parámetros de un juego de minorías que podrían ser cuantizados, planteando las condiciones necesarias para que la unidad de medida sea similar al espín de un electrón, la base algebraica correspondiente a la opción de introducir como estrategia posible para un jugador una rotación del tipo cuántico basada en las matrices de Pauli y la función de pago interpretada como el colapso de la función de onda del sistema.

Mediante la utilización de SAS Enterprise Guide, con un esquema ordenado y adecuado de simulación computacional, fue posible el realizar una simulación con muchos jugadores, en específico fue posible realizar una simulación de la evolución de los juegos de minorías para muchos jugadores utilizando métodos de Monte Carlo para la simulación, tanto en su versión clásica, como en su versión cuántica, con la posibilidad de generar múltiples escenarios con diferentes tipos de jugadores.

Las simulaciones implementadas permitieron acceder a una respuesta numérica que facilita un entendimiento teórico del problema planteado, mediante la evaluación de las diferentes estrategias puras de cada tipo de jugador. Pudimos también identificar que al buscar estrategias dominantes la opción cuántica en particular es preferida a cualquier otra estrategia a disposición de los jugadores, sobre todo cuando las condiciones iniciales no permiten definir una historia para el jugador.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Binmore, K., Teoría de Juegos, Madrid: McGraw-Hill, 1994.
- [2] Sanchez-Cuenca, I., Teoría de Juegos, Madrid: Centro de Investigaciones Sociológicas, 2009.
- [3] Davis, M., Introducción a la Teoría de juegos, Madrid: Alianza Universidad, 1986.
- [4] Neumann, J., Morgenstern, O., Theory of games and economic behavior, Princenton: Princenton University Press, 1947.
- [5] Challet, D. Zhang, Y., «Emergence of Cooperation and Organization in an Evolutionary Game.,» *Physica A: Statistical Mechanics and Its Applications*, vol. 246, nº 8, 1997.
- [6] Challet, D., Zhang, Y., «On the minority game: analytical and numerical studies,» *Physica A*, vol. 256, nº 514, 1998.
- [7] Kets, W., «Learning With Fixed Rules: the Minority Game,» *Journal of Economic Surveys*, 2011.
- [8] Du, J., Li, H., Xu, X., Shi, M., Wu, J., Zhou, X., Han, R., «Experimental Implementation of Quantum Random Walk Algorithm,» *Physical Reviews A*, vol. 67, nº 4, 2001.
- [9] Hammerstein, P., Selten, R., «Game theory and evolutionary biology,» *Handbook of game theory*, vol. 2, pp. 929-993, 1994.
- [10] Swedberg, R., «Sociology and game theory: contemporary and historical perspectives,» *Theory and Society*, pp. 301-335, 2001.

- [11] Becker, G., «Crime and Punishment: An Economic Approach,» *Journal of Political Economy*, nº 78, pp. 168-217, 1968.
- [12] Levine, D., «Quantum games have no news for economists,» *Quarterly 45 Pp*, pp. 208-211, 2005.
- [13] Hauert, C., Szabó, G., «Game theory and physics,» *American Journal of Physics*, vol. 5, nº 73, 2005.
- [14] Becker, M., «Statistics in Business and Social Science,» *The Journal of American Statistical Association*, vol. 95, pp. 628-629, 2000.
- [15] Nash, J., «Non-cooperative games,» *The Annals of Mathematics*, vol. 2, p. 54, 1951.
- [16] Benjamin S., Hayden P., «Multiplayer Quantum Games,» *Physical Review A*, vol. 64, 2001.
- [17] García, A., Ortega, M., Izquierdo, D., *Elementos de simulación*, Madrid: Terra, 2012.
- [18] Gutierrez, G., *Elementos de Simulación computacional: Dinámica molecular y método de Monte Carlo*, Santiago de Chile: Universidad de Santiago de Chile, 2001.
- [19] Metropolis, N., «The beginning of the Monte Carlo Method,» *Los Alamos Science Special Issue*, 1987.
- [20] Metropolis N., Rosenbluth A., Rosenbluth M. and Teller E., *Journal Chem. Phys.*, nº 21, 1953.
- [21] Binder, K., *Montecarlo Method in Statistical Physics*, Berlin: Springer, 1986.
- [22] Cicotti, G., Frenkel, D. and Mc Donald, I., *Simulation of Liquids and Solids*, Amsterdam: North-Holland, 1987.

- [23] Challet, D., Zhang, Y., «Emergence of Cooperation and Organization in an Evolutionary Game,» *Physica A: Statistical Mechanics and Its Applications*, vol. 246, nº 8, 1997.
- [24] Tarifa, E., Teoría de modelos y simulación: Introducción a la simulación, Jujuy: Universidad Nacional de Jujuy, 1988.
- [25] Heerman, D., Computer Simulation Methods in Theoretical Physics, Springer-Verlag, 1986.
- [26] Fan, X., Felsövályi, A., Sivo, S., Keenan, S., SAS for Monte Carlo Studies: A Guide for Quantitative Researchers, The SAS Institute, 2001.
- [27] Wierzbicki, A., «A mathematical basis for satisficing decision making,» *Mathematical Modelling*, nº 3, pp. 391-405, 1982.
- [28] Nash, J., «Equilibrium points in n-person games,» *Proceedings of the National Academy of Sciences*, nº 1, pp. 48-49, 1950.
- [29] Axelrod, R., La evolución de la cooperación, Madrid: Alianza Editorial, 1986.
- [30] Fernandez, J., Teoría de Juegos: su aplicación en economía, México, D.F.: Solar, 2002.
- [31] Pérez, J., Jimeno, J., Cerdá, E., Teoría de Juegos, Madrid: Ibergarceta, 2013.
- [32] Van Veldhuizen, D., Lamont, G., «Evolutionary Computation and Convergence to a Pareto Front,» *memories of the Third Annual Conference of Genetic Programming*, pp. 22-25, 1998.
- [33] P. Warr, «Pareto optimal redistribution and private charity,» *Journal of Public Economics*, vol. 1, nº 19, pp. 131-138, 1982.
- [34] Kets, W., «Learning With Fixed Rules: the Minority Game,» *Journal of Economic Surveys*, 2011.

- [35] Schelling, T., *Micromotives and Macrobehavior*, Norton, 1978.
- [36] Arthur, B., «Inductive Reasoning and Bounded Rationality,» *American Economist Association Papers and Procedures*, vol. 84, n° 406, 1994.
- [37] Arthur, W., *The Economy as an Evolving Complex System II*, Santa Fe: Westview Press, 1997.
- [38] Challet, D. and Zhang, C., *Physica A*, vol. 407, n° 246, 1997.
- [39] McMahon, D., *Quantum Mechanics Demystified*, Londres: McGraw-Hill, 2006.
- [40] Shankar, R., *Principles of Quantum Mechanics*, USA: Kluwer Academic/Plenum Publishers, 1994.
- [41] Zettili, N., *Quantum Mechanics concepts and Applications*, USA: Wiley, , 2009.
- [42] Rae, A., *Quantum Mechanics*, IOP Publishing Ltd., 1980.
- [43] Einstein, A., Podolsky, B., Rosen, N., «Can quantum-mechanical description of physical reality be considered complete?,» *Physical Review*, 1935.
- [44] Neylor, A., Sell, G, , *Linear Operator Theory in Engineering and Science*, New York: Rinehart and Winston, 1971.
- [45] Cohen-Tannoudji, C. Diu, B., Laloe, F., *Quantum Mechanics*, New York: John Wiley and Sons, 1977.
- [46] Feynman, R., Leighton, R., Sands, M., *Física: Volumen III*, Adisson-Wesley Iberoamericana, 19897.
- [47] Arfken, G., *Mathematical Methods for Physicist*, Orlando: Academic Press, 1985.
- [48] Griffiths, D., *Introduction to Elementary Particles*, New York: Wiley, 1987.

- [49] Landau, L. Lifschitz, E., Quantum Mechanics, Oxford: Pergamon Press, 1991.
- [50] Lee, C., Johnson, N., Quantum Game Theory, Oxford: Center of Quantum Computation and Physics Department Laboratory, 2006.
- [51] Deutsch, D., «Quantum theory of probability and decisions,» *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, n° 455, pp. 3129-3137, 1988.
- [52] Meyer, D., «Quantum Games and Quantum Algorithms,» *AMS Contemporary Mathematics*, pp. 1-10, 2000.
- [53] Du, J., Li, H., Xu, X., Shi, M., Wu, J., Zhou, X., Han, R., «Experimental implementation of the quantum random-walk algorithm,» *Phys. Rev. A*, vol. 67, 2003.
- [54] Benjamin, S., & Hayden, P., «Multiplayer quantum games,» *Physical Review A*, n° 64, pp. 1-4, 2001.
- [55] Shannon, R., Simulación de Sistemas. Diseño, desarrollo e implementación, México: Trillas, 1988.
- [56] Banks, J., Carson, J., Nelson, B., Discrete-Event System Simultacion, New Jersey: Prentice-Hall, 1996.
- [57] Fishman, G. Moore L., «A Statistical Evaluation of Multiplicative Congruential Generators with Modulus (231-1),» *Journal of the American Statistical Association*, vol. 1, n° 77, pp. 29-136, 1982.
- [58] Matsumoto, M. Nishimura, T., «Mersenne Twister: A 623-Dimensionally Equidistributed uniform Pseudo-Random Number Generator,» *ACM Transacions on Modeling and Computer Simulation*, vol. 8, pp. 3-30, 1998.
- [59] Challet, D., Marsili, M., Zhang, Y., Minority Games, Oxford University Press, 2005.

- [60] Manuca, R., Li, R., Riolo, R. and Savit, R., *Physica A*, vol. 282, n° 559, 2000.
- [61] Einsiedler, M., Ward, T., *Ergodic Theory: with a view towards Number Theory*, Berlin: Springer, 2009.
- [62] Marsili, M., Challet, D., «Continuum time limit stationary states of the Minority Game,» *Physical Review E*, vol. 64, n° 056138, 2001.
- [63] Challet, D., Marsili, M., Zecchina, R., «Statistical mechanics of systems with heterogeneous agents: Minority Games,» *Phys. Rev. Lett.*, vol. 84, n° 1824, 2000.
- [64] Cardi, I., Ceva, H., «Minority game: a mean-field-like approach,» *Physica A*, vol. 317, n° 574, 2002.
- [65] Myerson, R., «Nash Equilibrium and the history of economic theory,» *Journal of Economic Literature*, vol. 37, pp. 1067-1082, 1999.

ANEXOS

Anexo 1.- Ejemplo simulación frecuencia acumulada

Código SAS utilizado en el archivo: "Ejemplo Frecuencia Acumulada.sas"

```
data datos; input consultas frecuencia; datalines;
```

```
0 80
1 50
2 40
3 15
4 10
5 5
```

```
;run;
```

```
proc sql; select sum(frecuencia) as total_consultas into : total_observaciones
from datos; quit;
```

```
data frecuencia_acumulada; set datos;
frecuencia_absoluta = frecuencia/&total_observaciones;
frecuencia_acumulada + frecuencia_absoluta;
ejemplo = "# de Consultas"; run;
```

```
proc print data=frecuencia_acumulada noobs;
var consultas frecuencia frecuencia_absoluta frecuencia_acumulada; run;
```

```
Legend1FRAME POSITION = (MIDDLE RIGHT OUTSIDE) LABEL=( "" );
Axis1 STYLE=1 WIDTH=1 LABEL=( "" );
Axis2 STYLE=1 WIDTH=1 MINOR= (NUMBER=1) LABEL=( "Probabilidad");
TITLE; TITLE1 "# de Consultas"; FOOTNOTE;
```

```
PROC GCHART DATA=frecuencia_acumulada;
HBAR ejemplo / SUMVAR=frecuencia_absoluta SUBGROUP=consultas CLIPREF
FRAME DISCRETE TYPE=SUM OUTSIDE=PCT LEGEND=LEGEND1 COUTLINE=BLACK MAXIS=AXIS1
RAXIS=AXIS2; RUN; QUIT; TITLE; FOOTNOTE;
```

```
data datos_simulados;
do i = 1 to 100; aleatorio = ranuni(1);
if 0 le aleatorio lt 0.4 then suceso = 0;
if 0.4 le aleatorio lt 0.65 then suceso = 1;
if 0.65 le aleatorio lt 0.85 then suceso = 2;
if 0.85 le aleatorio lt 0.925 then suceso = 3;
if 0.925 le aleatorio lt 0.975 then suceso = 4;
if 0.975 le aleatorio le 1 then suceso = 5;
output; end; run;
```

```
title "Resultado para una simulación con 100 iteraciones";
proc means data=datos_simulados n mean;
var suceso; run; title;
```

```
data datos_simulados_2;
do i = 1 to 100000; aleatorio = ranuni(1);
if 0 le aleatorio lt 0.4 then suceso = 0;
if 0.4 le aleatorio lt 0.65 then suceso = 1;
if 0.65 le aleatorio lt 0.85 then suceso = 2;
if 0.85 le aleatorio lt 0.925 then suceso = 3;
if 0.925 le aleatorio lt 0.975 then suceso = 4;
if 0.975 le aleatorio le 1 then suceso = 5;
output; end; run;
```

```
title "Resultado para una simulación con 100.000 iteraciones";
proc means data=datos_simulados_2 n mean;
var suceso; run; title;
```

Anexo 2.- Ejemplo simulación selección de escenario

Código SAS utilizado en el archivo: "Ejemplo Elaboración de Tarjetas.sas"

```

data tarjetas; input tarjetas_compradas probabilidad_venta; datalines;
100 0.3
150 0.2
200 0.3
250 0.15
300 0.05
;run;

data frecuencia_tarjetas; set tarjetas;
probabilidad_acumulada + probabilidad_venta;
intervalo_inferior = lag (probabilidad_acumulada);
if intervalo_inferior = . then intervalo_inferior = 0;
intervalo_superior = probabilidad_acumulada; run;

proc print data=frecuencia_tarjetas noobs; run;

%macro tarjetas;
%do tarjetas_a_comprar = 100 %to 300 %by 50;
%let costo_tarjeta = 7.5;
%let ingreso_por_venta = 10;
%let ingreso_por_devolucion =2.5;

data simulacion_tarjetas;
do i = 1 to 10000; aleatorio = ranuni(1);
if 0 le aleatorio lt 0.3 then tarjetas_compradas = 100;
if 0.3 le aleatorio lt 0.5 then tarjetas_compradas = 150;
if 0.5 le aleatorio lt 0.8 then tarjetas_compradas = 200;
if 0.8 le aleatorio lt 0.95 then tarjetas_compradas = 250;
if 0.925 le aleatorio le 1 then tarjetas_compradas = 300;
tarjetas_vendidas = min(1*&tarjetas_a_comprar, tarjetas_compradas);
ingresos_venta = tarjetas_vendidas * &ingreso_por_venta;
ingresos_devolucion = tarjetas_vendidas * &ingreso_por_devolucion;
costos = 1*&tarjetas_a_comprar * &costo_tarjeta;
Beneficios = sum(ingresos_venta, ingresos_devolucion, -costos);
output; end; run;

title "Resultado para una simulación con 10.000 iteraciones y compra &tarjetas_a_comprar tarjetas";

/*proc means data=simulacion_tarjetas n mean stddev min max; var Beneficios; run;*/

PROC TTEST DATA=simulacion_tarjetas ALPHA=0.05;
VAR Beneficios; RUN; title;

proc sql noprint;
select mean(Beneficios) into : resultado_&tarjetas_a_comprar
from simulacion_tarjetas; Quit;

proc sql noprint;
select std(Beneficios) into : desviacion_&tarjetas_a_comprar
from simulacion_tarjetas; Quit;
%put resultado_&tarjetas_a_comprar;
%put &&resultado_&tarjetas_a_comprar;
%end;

data tarjetas_2; set tarjetas;
%do i = 100 %to 300 %by 50;
if tarjetas_compradas = 1*&i then do;
beneficios = 1*&&resultado_&i;
desviacion = 1*&&desviacion_&i; end;

```

```

%end; run;

title "Comparación de resultados obtenidos";
proc print data=tarjetas_2 noobs;
var tarjetas_compradas beneficios desviacion; run; title;

%mend tarjetas;

%tarjetas

```

Anexo 3.- Simulación para diferentes tipos de jugadores problema el Farol.

Código SAS utilizado en el archivo: "simulación final.sas"

```

data el_farol; do i = 1 to &n_jugadores;
estado_0 = int(ranuni(0)*2);
output; end; run;

data el_farol (drop=Aleatorio Insistente DeterministicoQ AleatorioQ InsistenteQ Deterministico);
set el_farol; format tipo_jugador $15.;
Aleatorio = 1*&Aleatorio;
Insistente = 1*&Insistente + sum(aleatorio);
DeterministicoQ = 1*&DeterministicoQ + sum(/*aleatorio,*/ Insistente);
AleatorioQ = 1*&AleatorioQ + sum(/*aleatorio, Insistente,*/ DeterministicoQ);
InsistenteQ = 1*&InsistenteQ + sum(/*aleatorio, Insistente, DeterministicoQ,*/ AleatorioQ);
Deterministico = sum(1*&n_jugadores,1);
if _N_ le Aleatorio then tipo_jugador = "Aleatorio";
if Aleatorio lt _N_ le Insistente then tipo_jugador = "Insistente";
if Insistente lt _N_ le DeterministicoQ then tipo_jugador = "DeterministicoQ";
if DeterministicoQ lt _N_ le AleatorioQ then tipo_jugador = "AleatorioQ";
if AleatorioQ lt _N_ le InsistenteQ then tipo_jugador = "InsistenteQ";
if InsistenteQ lt _N_ le Deterministico then tipo_jugador = "Deterministico";
run;

proc sql;
create table resultado_0 as select estado_0, count(*) as cuantos
from el_farol group by 1 order by 2; Quit;

proc sql;
create table resultado_0_por_tipo as select tipo_jugador, estado_0, count(*) as cuantos
from el_farol group by 1,2 order by 3; Quit;

proc sql;
create table puntaje_ganador as select min(cuantos) as ganador
from resultado_0; quit;

proc sql noprint;
select estado_0 into :opcion_ganadora from resultado_0
where cuantos in (select ganador from puntaje_ganador); quit;

data evolucion_0; iteracion = 1; opcion_ganadora = 1*&opcion_ganadora; run;

data evolucion; set evolucion_0; run;

data el_farol_i; set el_farol; resultado_0 = 0;
if estado_0 = &opcion_ganadora then resultado_0 = 1;
if tipo_jugador = "Aleatorio" then do;
decision_1 = int(ranuni(0)*2);
if estado_0 = 1 and decision_1 = 0 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 1; end;
if tipo_jugador = "Insistente" then do;
decision_1 = int(ranuni(0)*2);

```

```

if estado_0 = 1 and decision_1 = 0 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 1; end;
if tipo_jugador = "DeterministicoQ" then do; decision_1 = int(ranuni(0)*4);
if estado_0 = 1 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 3 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 3 then estado_1 = 1; end;
if tipo_jugador = "AleatorioQ" then do; decision_1 = int(ranuni(0)*4);
if estado_0 = 1 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 3 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 3 then estado_1 = 1; end;
if tipo_jugador = "InsistenteQ" then do; decision_1 = int(ranuni(0)*4);
if estado_0 = 1 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 1 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 3 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 2 then estado_1 = 1;
if estado_0 = 0 and decision_1 = 3 then estado_1 = 1; end;
if tipo_jugador = "Deterministico" then do; decision_1 = int(ranuni(0)*2);
if estado_0 = 1 and decision_1 = 0 then estado_1 = 1;
if estado_0 = 1 and decision_1 = 1 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 0 then estado_1 = 0;
if estado_0 = 0 and decision_1 = 1 then estado_1 = 1; end; run;

/*Desde aquí parametrizar*/
proc sql;
create table resultado_1 as select estado_1, count(*) as cuantos
from el_farol_i group by 1 order by 2; Quit;

proc sql;
create table puntaje_ganador as select min(cuantos) as ganador
from resultado_1; quit;

proc sql noprint;
select estado_1 into :opcion_ganadora from resultado_1
where cuantos in (select ganador from puntaje_ganador); quit;

data evolucion_i; iteracion = 2; opcion_ganadora = 1*&opcion_ganadora; run;

data evolucion; set evolucion evolucion_i; run;

data el_farol_j; set el_farol_j; resultado_2 = 0;
if estado_1 = &opcion_ganadora then resultado_2 = 1;
if tipo_jugador = "Aleatorio" then do; decision_2 = int(ranuni(0)*2);
if estado_1 = 1 and decision_2 = 0 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 1; end;
if tipo_jugador = "Insistente" then do; decision_2 = decision_1;
if estado_1 = 1 and decision_2 = 0 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 1; end;

```

```

if tipo_jugador = "DeterministicoQ" then do; if resultado_2 = 1 then decision_2 = decision_1;
if resultado_2 = 0 then decision_2 = int(ranuni(0)*4);
if estado_1 = 1 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 3 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 3 then estado_2 = 1; end;
if tipo_jugador = "AleatorioQ" then do; decision_2 = int(ranuni(0)*4);
if estado_1 = 1 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 3 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 3 then estado_2 = 1; end;
if tipo_jugador = "InsistenteQ" then do; decision_2 = decision_1;
if estado_1 = 1 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 1 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 3 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 2 then estado_2 = 1;
if estado_1 = 0 and decision_2 = 3 then estado_2 = 1; end;
if tipo_jugador = "Deterministico" then do; if resultado_2 = 1 then decision_2 = decision_1;
if resultado_2 = 0 then decision_2 = int(ranuni(0)*2);
if estado_1 = 1 and decision_2 = 0 then estado_2 = 1;
if estado_1 = 1 and decision_2 = 1 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 0 then estado_2 = 0;
if estado_1 = 0 and decision_2 = 1 then estado_2 = 1; end; run;

```

%macro simulacion;

```
%do i = 3 %to &n_iteraciones; %let k = %eval(&i-1); %let j = %eval(&i+1);
```

```
proc sql;
create table resultado_&k as select estado_&k, count(*) as cuantos
from el_farol_i group by 1 order by 2; Quit;
```

```
proc sql;
create table puntaje_ganador as select min(cuantos) as ganador
from resultado_&k; quit;
```

```
proc sql noprint;
select estado_&k into :opcion_ganadora from resultado_&k
where cuantos in (select ganador from puntaje_ganador); quit;
```

```
data evolucion_i; iteracion = &i; opcion_ganadora = 1*&opcion_ganadora; run;
```

```
data evolucion; set evolucion evolucion_i; run;
```

```
data el_farol_j; set el_farol_i; resultado_&i = 0;
if estado_&k = &opcion_ganadora then resultado_&i = 1;
if tipo_jugador = "Aleatorio" then do; decision_&i = int(ranuni(0)*2);
if estado_&k = 1 and decision_&i = 0 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 1; end;
if tipo_jugador = "Insistente" then do; decision_&i = decision_&k;
if estado_&k = 1 and decision_&i = 0 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 1; end;
```

```

if tipo_jugador ="DeterministicoQ" then do; if resultado_&i = 1 then decision_&i = decision_&k;
if resultado_&i = 0 then decision_&i = int(ranuni(0)*4);
if estado_&k = 1 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 3 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 3 then estado_&i = 1; end;
if tipo_jugador ="AleatorioQ" then do; decision_&i = int(ranuni(0)*4);
if estado_&k = 1 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 3 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 3 then estado_&i = 1; end;
if tipo_jugador ="InsistenteQ" then do; decision_&i = decision_&k;
if estado_&k = 1 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 1 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 3 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 2 then estado_&i = 1;
if estado_&k = 0 and decision_&i = 3 then estado_&i = 1; end;
if tipo_jugador ="Deterministico" then do; if resultado_&i = 1 then decision_&i = decision_&k;
if resultado_&i = 0 then decision_&i = int(ranuni(0)*2);
if estado_&k = 1 and decision_&i = 0 then estado_&i = 1;
if estado_&k = 1 and decision_&i = 1 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 0 then estado_&i = 0;
if estado_&k = 0 and decision_&i = 1 then estado_&i = 1; end; run;

%end;

%let n_menos_uno = %eval(&n_iteraciones-1);
proc sql;
%do m = 1 %to &n_menos_uno; drop table resultado_&m;
%end; quit;

%mend;
%simulacion

SYMBOL1 INTERPOL=JOIN HEIGHT=10pt VALUE=NONE LINE=1 WIDTH=2 CV = _STYLE_;
Axis1 STYLE=1 WIDTH=1 MINOR=NONE ;
Axis2 STYLE=1 WIDTH=1 MINOR=NONE ; TITLE;
TITLE1 "Resultado evolución para &Deterministico Jugadores Determinísticos,
&Aleatorio Jugadores Aleatorios, &Insistente Jugadores Insistentes
&DeterministicoQ Jugadores Determinísticos Cuánticos,
&AleatorioQ Jugadores Aleatorios Cuánticos,
y &InsistenteQ Jugadores Insistentes cuánticos";

PROC GPLOT DATA = evolucion; PLOT opcion_ganadora * iteracion / VAXIS=AXIS1 HAXIS=AXIS2
FRAME; RUN; QUIT; TITLE; FOOTNOTE; GOPTIONS RESET = SYMBOL;

```



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

CARRERA DE FÍSICA

ORDEN DE EMPASTADO

De acuerdo a lo estipulado en el artículo 27 del Instructivo para la Implementación de la Unidad de Titulación en las Carreras y Programas vigentes de la Escuela Politécnica Nacional, aprobado por Consejo de Docencia en sesión extraordinaria del 29 de abril de 2015 y una vez verificado el cumplimiento del formato de presentación establecido, se autoriza la impresión y encuadernación final del Trabajo de Titulación presentado por el señor **JORGE LUIS ENRÍQUEZ MAYORGA**.

Fecha de autorización: Quito, D.M., 12 de junio de 2017.

Una firma manuscrita en tinta azul que parece decir "Lascano".
Un sello circular azul que contiene el escudo de la Escuela Politécnica Nacional y el texto "ESCUELA POLITÉCNICA NACIONAL" y "FACULTAD DE CIENCIAS" en el borde superior, y "DECANATO" en el borde inferior.

Dr. Luis Lascano
DECANO
FACULTAD DE CIENCIAS