

**UNIVERSIDAD DE TARAPACÁ
ESCUELA DE INGENIERÍA INDUSTRIAL,
INFORMÁTICA Y DE SISTEMAS**



**PROPUESTA DE DISEÑO DE UN SIMULADOR
PARA EL PROTOCOLO DE CRIPTOGRAFIA
CUANTICA B92 EN UN AMBIENTE
DISTRIBUIDO**

**Memoria para optar al título de:
Ingeniero Civil en Computación e
Informática**

Alumno:
Enzo Videla Farías

Profesor Guía:
Dr. Raúl Herrera Acuña

**Arica – Chile
2019**

AGRADECIMIENTOS

Primero quiero agradecer a mi familia, a mi madre y a mi padre por el tiempo que se han dado en educarme y enseñarme como ser un buen hijo, un buen alumno y una buena persona.

Segundo agradecer a los profesores Raúl Herrera, Patricio Collao, Luis Cáceres y mi profesor informante Diego Aracena, siempre atendieron a mis dudas y lograron encaminarme hacia el éxito y termino de este trabajo.

Tercero agradecer al profesor Edmundo Lazo, quien me ayudo en todo el ámbito de los fundamentos físicos y matemáticos de la teoría cuántica. Me ayudo a encontrar la motivación necesaria para sacar adelante este trabajo y sin el yo aún estaría divagando en un callejón sin salida.

Finalmente agradecer a los guerreros más importantes durante este camino y esta batalla final, mis compañeros y únicos amigos que lucharon junto a mí a lo largo de toda esta aventura.

El gran maestro Kebi, Gran maestro Cachorro, Gran maestro Jou, Gran maestro Cheremy y el gran maestro Talon.



GRAN MAESTRO CACHORRO

CONTENIDOS

Agradecimientosii

Contenidos.....iii

Resumen.....vii

Indice de Figurasviii

Indice de Tablasxi

Nomenclaturaxii

Capítulo 1: introducción..... 1

 1.1. Descripción del problema 3

 1.2. Solución propuesta..... 5

 1.3. Objetivos 8

 1.3.1. Objetivo General 8

 1.3.2. Objetivos Específicos..... 8

 1.3.3. Actividades..... 9

 1.4. Planificación..... 10

 1.5. Estructura 11

Capítulo 2: Marco Teórico 12

 2.1. Introduccion a la mecánica cuántica 12

 2.1.2. Dualidad onda-partícula y Partícula-onda 12

 2.1.3. Postulados de De-Broglie 1924..... 14

 2.1.4. Experimento doble rendija 15

 2.1.5. Principio de Incertidumbre de Heisenberg..... 19

 2.1.6. Postulados de Heisenberg 1927 21

 2.1.7. Entrelazamiento cuántico 23

 2.1.8. Polarización de fotones 25

 2.1.9. Polarización y superposición de estados 31

 2.1.10. Superposición de estados y codificación de bits 37

 2.1.11. Fundamento matemático del entrelazamiento y la superposición de
estados 38

 2.1.12. Experimento de teletransportación de Alice y Bob..... 39

 2.1.13. Teorema de no clonación 1982 41

2.1.14. El Qubit.....	43
2.1.15. Ejemplo de comunicación entre Alice y Bob.....	48
2.2. Criptografía	49
2.2.1. Cifrado simétrico o de clave privada	50
2.2.2. DES	50
2.2.3. AES	51
2.2.4. One Time Pad, Cifrado de Vernam.....	51
2.2.5. Cifrado asimétrico o de clave publica.....	52
2.2.6. RSA.....	53
2.3. Criptografía cuántica.....	55
2.3.1 El computador cuántico.....	55
2.3.2. Algoritmos de distribución de claves cuánticas	56
2.3.3. Principio de Incertidumbre de Heisenberg aplicado en criptografía.....	58
2.3.4. Entrelazamiento cuántico aplicado en criptografía	58
2.4. Protocolos de criptografía cuántica.....	59
2.4.1. Protocolo BB84.....	59
2.4.2. Protocolo B92.....	63
2.5. Conclusiones del capítulo	71
Capítulo 3: Descripción de la solución	72
3.1. Descripción general de la solución	72
3.2. Metodología de trabajo	73
3.3. Arquitectura de software para el sistema	76
3.3.1. Arquitectura cliente-servidor	76
3.3.2. Arquitectura distribuida Java - RMI	77
3.4. Plataforma y lenguaje de programación a utilizar.....	81
3.5. Recursos necesarios	81
3.5.1. Factibilidad técnica	82
3.5.2. Factibilidad operativa.....	82
3.5.3. Factibilidad económica	83
3.6. Aporte.....	84
3.7. Conclusiones del capítulo	85

Capítulo 4: Desarrollo	86
4.1. Análisis de requerimientos.....	86
4.1.1. Requerimiento y objetivo principal del sistema B92	86
4.1.2. Requisitos funcionales	87
4.1.3. Requerimientos no funcionales.....	87
4.1.4. Restricciones	87
4.2. Casos de uso	88
4.2.1. Diagrama de casos de uso del sistema completo	88
4.2.2. Caso de uso: Definir variables del sistema	89
4.2.3. Caso de uso: Generar secuencia binaria (Alice)	89
4.2.4. Caso de uso: Generar esquema de polarizaciones (Alice)	89
4.2.5. Caso de uso: generar secuencia binaria (Bob)	90
4.2.6. Caso de uso: Generar esquema de polarización (Bob).....	90
4.2.7. Caso de uso: transmitir esquema de polarizaciones.....	90
4.2.8. Caso de uso: Comparar esquemas de polarizaciones de Alice y Bob.....	91
4.2.9. Caso de uso: Generar la Raw Key.....	91
4.2.10. Caso de uso: Generar la Sifted Key	92
4.2.11. Casos de uso: Transmitir la Sifted Key de Bob	92
4.2.12. Caso de uso: intercambiar Hashes.....	93
4.2.13. Casos de uso: Transmitir mensaje cifrado con AES	93
4.4. Diagramas de secuencia	94
4.4.1. Diagrama de secuencia: Definir variables del sistema.....	94
4.4.2. Diagrama de secuencia: Generar secuencia binaria Alice	95
4.4.3. Diagrama de secuencia: Generar esquema de polarización Alice	96
4.4.4. Diagrama de secuencia: Generar secuencia binaria Bob	97
4.4.5. Diagrama de secuencia: Generar esquema de polarización Bob	98
4.4.6. Diagrama de secuencia: Transmitir esquema de polarizaciones.....	99
4.4.7. Diagrama de secuencia: Comparar esquemas de polarización	100
4.4.8. Diagrama de secuencia: Generar la Raw Key.....	101
4.4.9. Diagrama de secuencia: Generar la Sifted Key.....	102
4.4.10. Diagrama de secuencia: Transmitir la Sifted Key.....	103

4.4.11. Diagrama de secuencia: Intercambiar Hashes.....	103
4.4.12. Diagrama de secuencia: Enviar mensaje cifrado AES.....	104
4.4.13 Diagrama de secuencia: Recibir mensaje cifrado AES.....	104
4.5. Diagrama de clases.....	105
4.6. Implementación.....	106
4.7. Ejecución de la aplicación.....	113
4.8. Conclusiones del capítulo	116
Capítulo 5: Resultados	117
5.1. Requerimientos de las pruebas.....	117
5.2. Especificación de los equipos utilizados.....	118
5.3. Pruebas de la aplicación.....	118
5.3.1. Caso de prueba 1	119
5.3.2. Caso de prueba 2	123
5.3.3. Análisis asintótico y orden de complejidad del algoritmo	129
5.4. Evaluación de requisitos	132
5.5. Conclusiones del capítulo	134
Capítulo 6: Conclusiones y Trabajo Futuro	136
6.1. Resumen.....	136
6.2. Conclusiones sobre los resultados.....	137
6.3. Trabajo Futuro.....	141
Referencias.....	143

RESUMEN

El presente trabajo es un estudio científico, que presenta las bases para el desarrollo de un protocolo de criptografía cuántica. Un estudio profundo sobre los fundamentos físicos y matemáticos de la criptografía cuántica. Un recuento breve de los hechos históricos que dieron origen a la mecánica cuántica tal cual como se conoce hoy en día. También es un trabajo práctico de diseño e implementación de un protocolo de criptografía cuántica, para el cual se desarrollaron los programas necesarios y se realizaron las pruebas de su funcionamiento.

El primer capítulo habla sobre el objetivo principal del proyecto, el cual es desarrollar e implementar un simulador para el algoritmo de criptografía cuántica B92 en un ambiente distribuido y realizar las pruebas para simular su funcionamiento. Esta simulación consiste en emular el funcionamiento de una comunicación mediante fotones polarizados, enviados por un canal cuántico desde un emisor Alice a un receptor Bob, el ángulo de polarización de los fotones sirve para codificar bits, o en este caso qubits, de manera de que, lo que viaja por el canal no son los bits propiamente tal, sino la polarización de los fotones, la cual se decodifica por el receptor en el otro extremo de la comunicación obteniendo los qubits sin siquiera ser enviados por Alice. El resultado esperado son los programas y la aplicación en java que simule el funcionamiento del protocolo de manera fidedigna.

El segundo capítulo define el marco teórico del proyecto, en el cual se definen las bases y el fundamento matemático detrás de la aplicación práctica. El capítulo es un recuento de la historia de la física de finales del siglo XIX.

En el tercer y cuarto capítulo, se define el problema específico y la metodología para resolver dicho problema. La resolución consiste en una serie de programas en lenguaje de programación Java, que simulan el funcionamiento cuántico del protocolo, de igual forma se realiza todo el análisis de software relacionado con las aplicaciones, tanto análisis de requerimientos como casos de uso, diagramas e implementación.

En los capítulos quinto y sexto se prueba la aplicación y se realizan las mediciones de los resultados, para finalmente concluir con el análisis final de los resultados y conclusiones futuras.

INDICE DE FIGURAS

Figura 1.1. Aumento de cantidad de transistores en procesadores Intel según la Ley de Moore en los últimos 50 años..... 2

Figura 1.2. Diagrama de comunicación con un canal cuántico..... 4

Figura 1.3. Diagrama de solución propuesto..... 6

Figura 1.4. Diagrama de flujo de tareas..... 9

Figura 1.5. Grafica Gantt para el flujo de tareas..... 10

Figura 2.1. Experimento doble rendija real..... 16

Figura 2.2. Interferencia de los electrones..... 17

Figura 2.3. Comparativa entre un fenómeno de partículas y ondulatorio. 18

Figura 2.4. Distintas formas de modelar la posición de un objeto..... 20

Figura 2.5. Generación de un par de partículas entrelazadas [18]..... 25

Figura 2.6. Campos magnéticos y eléctricos de una onda. 26

Figura 2.7. Polarización de un fotón en ángulos: a) = 90° b) = 0° c) = 45°. 27

Figura 2.8. Polarización lineal a) circular b) y elíptica c)..... 28

Figura 2.9. Filtro polarizador que permite solo el paso de la luz vertical. 28

Figura 2.10. Polarización lineal rectilínea horizontal y vertical. 29

Figura 2.11. Fotones atravesando un filtro paralelo a su polarización y otro perpendicular..... 30

Figura 2.12. Una onda electromagnética polarizada ante distintos filtros 31

Figura 2.13. Luz no polarizada, luego polarizada y un filtro..... 32

Figura 2.14. Los fotones que logran pasar el filtro a 90° tienen un 50% de probabilidad de pasar el filtro a 45°..... 33

Figura 2.15. Fotones logran pasar un polarizador perpendicular..... 34

Figura 2.16. Esfera de Bloch 45

Figura 2.17. Mecanismo de comunicación con canales clásico y cuántico [29] ... 57

Figura 2.18. Codificación de qubits para los ángulos de polarización. 60

Figura 2.19. Bases de polarización y sus ángulos. 62

Figura 2.20. Bases de polarización rectilíneas 64

Figura 2.21. Bases de polarización diagonales..... 65

Figura 2.22. Polarizador emisor de Alice. Fotón Polarizado a 0° rectilíneo para el qubit $|0\rangle$ y a 45° diagonal para el qubit $|1\rangle$ 66

Figura 2.23. Filtro detector de Bob. Fotón polarizado a 135° diagonal para el qubit $|0\rangle$ y a 90° rectilíneo para el qubit $|1\rangle$ 67

Figura 3.1. Modelo de desarrollo en cascada con prototipo secuencial [33]..... 73

Figura 3.2. Metodología de trabajo para las versiones del prototipo..... 74

Figura 3.3. Modelado de los diagramas UML 75

Figura 3.4. Arquitectura cliente/servidor	77
Figura 3.5. Componentes de un sistema RMI.....	78
Figura 3.6. Modelo de 4 capas RMI.....	79
Figura 4.1. Diagrama de casos de uso del sistema.....	88
Figura 4.2. Diagrama de secuencia Definir variables del sistema	94
Figura 4.3. Diagrama de secuencia Generar secuencia binaria (Alice)	95
Figura 4.4. Diagrama de secuencia Generar esquema de polarización (Alice) ...	96
Figura 4.5. Diagrama de secuencia Generar secuencia binaria (Bob)	97
Figura 4.6. Diagrama de secuencia Generar esquema de polarización Bob.....	98
Figura 4.7. Diagrama de secuencia Transmitir esquema de polarizaciones.....	99
Figura 4.8. Diagrama de secuencia Comparar esquemas de polarización de Alice y Bob.....	100
Figura 4.9. Diagrama de secuencia Generar la Raw Key.....	101
Figura 4.10. Diagrama de secuencia Generar la Sifted Key	102
Figura 4.11. Diagrama de secuencia Transmitir la Sifted Key	103
Figura 4.12. Diagrama de secuencia Intercambiar Hashes.....	103
Figura 4.13. Diagrama de secuencia Enviar mensaje cifrado AES	104
Figura 4.14. Diagrama de secuencia Recibir mensaje cifrado AES	104
Figura 4.15. Diagrama de clases del sistema	105
Figura 4.16. Código de las clases <i>main</i>	107
Figura 4.17. Código de la clase secuencia	107
Figura 4.18. Código de las clases esquema.....	108
Figura 4.19. Código clase Generar_secuencia_polarizacion.....	109
Figura 4.20. Método Generar_nro(), programa Emisor y Receptor	109
Figura 4.21. Código de las clases InterCanalEmi InterCanalRec y CanalEmi.	110
Figura 4.22. Código de la clase VentEnt.	111
Figura 4.23. Código de la clase VentIng.....	111
Figura 4.24. Clase VentB92, muestra los resultados por pantalla.....	112
Figura 4.25. Código de la clase Principio_de_incertidumbre	112
Figura 4.26. Ejecución del comando <i>rmiregistry</i> para iniciar el proceso RMI. .	113
Figura 4.27. Ventana principal para los programas Emisor y receptor.....	113
Figura 4.28. Ingreso de datos en el menú principal.	114
Figura 4.29. Diálogos de inicio de cifrado y termino de cifrado en la aplicación Emisor.	114
Figura 4.30. Salida del programa emisor Alice en modo consola	114
Figura 4.31. Salida del programa emisor Alice en modo interfaz grafica	115
Figura 4.32. Salida del programa Receptor Bob en modo consola.....	115
Figura 4.33. Salida de la aplicación Receptor Bob en modo interfaz grafica....	116
Figura 5.1. Ejecución de los programas en <i>Localhost</i>	120

Figura 5.2. Grafico Tamaño final de la clave 122

Figura 5.3. Grafito tiempo de ejecución..... 122

Figura 5.4. Porcentaje de bits que conforman la clave final 123

Figura 5.5. Diagrama de red LAN..... 124

Figura 5.6. Ejecución de los programas en red distribuida 124

Figura 5.7. Grafica tamaño final de la clave..... 126

Figura 5.8. Grafico tiempo de ejecución 126

Figura 5.9. Grafico porcentaje de clave final 126

Figura 5.10. Comparación Tamaño final caso de prueba 1 y caso de prueba 2 128

**Figura 5.11. Comparación Tiempo de ejecución caso de prueba 1 y caso de
prueba 2..... 128**

Figura 5.12. Comparación porcentaje de bits clave final 128

Figura 5.13. Grafica de las clases de complejidad conocidas 131

**Figura 5.14. Grafica de funciones de complejidad junto a la variable tiempo de
ejecución..... 132**

INDICE DE TABLAS

Tabla 2.1. Estados de polarización para representar Qubits..... 48

Tabla 2.2. Fotones polarizados que viajaran por un canal cuántico. 48

Tabla 2.3. Polarización para los fotones de Alice y los respectivos qubits 66

Tabla 2.4. Polarización para el detector de Bob y sus respectivos qubits..... 67

Tabla 2.5. Trasmisión y recepción de los bits en B92 67

Tabla 3.1. Primitivas deseables para la especificación de requisitos..... 76

Tabla 3.2. Costos involucrados en la factibilidad económica..... 84

Tabla 4.1. Descripción caso de uso generar variables del sistema 89

Tabla 4.2. Descripción del caso de uso Generar secuencia binaria Alice..... 89

Tabla 4.3. Descripción del caso de uso Generar esquema de polarización Alice 90

Tabla 4.4. Descripción del caso de uso generar secuencia binaria Bob 90

Tabla 4.5. Descripción del caso de uso Generar esquema de polarización Bob .. 90

Tabla 4.6. Descripción casos de uso Enviar y recibir esquema..... 91

Tabla 4.7. Descripción caso de uso Comparar esquemas de polarización 91

Tabla 4.8. Descripción del caso de uso Generar Raw Key 92

Tabla 4.9. Descripción Caso de uso Generar la Sifted Key 92

Tabla 4.10. Descripción de los casos de uso Enviar y Recibir Sifted Key 93

Tabla 4.11. Descripción caso de uso Intercambiar Hashes 93

Tabla 4.12. Descripción del caso de uso Enviar y recibir mensaje cifrado AES. 93

Tabla 5.1. Especificación del equipo 1 Emisor 118

Tabla 5.2. Especificación del equipo 2 Receptor 118

Tabla 5.3. Especificación router 118

Tabla 5.4. Largo de las claves 119

Tabla 5.5. Resultados del caso de prueba 1 120

Tabla 5.6. Configuración de equipos para el caso de prueba 2..... 124

Tabla 5.7. Resultados del caso de prueba 2 125

Tabla 5.8. Clases de complejidad conocidas 131

Tabla 5.9. Requisitos funcionales..... 133

Tabla 5.10. Requisitos no funcionales 133

NOMENCLATURA

Alice: La parte emisora en la comunicación, transmite la clave secreta

Bob: La parte receptora en la comunicación, recibe la clave secreta

BB84: Protocolo de 4 estados de polarización creado en 1984 por Bennett y Bassard

B92: Protocolo de 2 estados de polarización creado en 1992 por Bennett

E91: Protocolo de criptografía cuántica que hace uso del espín del electrón

Eve: Un espía hipotético en la comunicación

P.I.H: Principio de incertidumbre de Heisenberg

QBER: *Quantum Bit Error Ratio*, tasa de error de bit cuántico

Raw Key: clave en bruto

Sifted Key: clave depurada

RMI: *Remote Method Invocation*

RPC: *Remote Procedure Call*

RSA: Rivest Shamir Adleman, es un protocolo de criptografía de clave pública

AES: *Advanced Encryption Standard*, es un protocolo de criptografía de clave privada o simétrica

DES: *Data Encryption Standard*, es un protocolo de criptografía de clave privada o simétrica, predecesor de AES

FIPS: *Federal Information Processing Standard*, son los estándares anunciados por estados unidos para utilización de manera publica

CAPÍTULO 1: INTRODUCCIÓN

Hoy en día el avance de las tecnologías de la información ha evolucionado a tal nivel que los requerimientos de manejo de la información aumentan cada día en un orden exponencial según la ley de Moore [1]. Las computadoras modernas trabajan a base de electricidad y movimiento de electrones, pero en un futuro no lejano las computadoras transportaran los datos a través de haces de luz. Las transacciones de negocios, compra venta de servicios, manejo de información y estudios de grandes volúmenes de datos, requieren un nivel de procesamiento y seguridad cada vez mayor. A medida que los procedimientos migran a internet es necesario que los procesos sean altamente confiables, eficientes y seguros.

Junto con la introducción del Internet en el día a día de las personas (Internet de las Cosas, *Internet Of Things-IOT*) y los avances en nuevas tecnologías de computación, nos acercamos a un punto en que las computadoras tal como las conocemos actualmente no podrán dar abasto con este constante crecimiento del entorno. Los requerimientos de velocidad, capacidad y seguridad, crecen día a día, generando la necesidad de migrar a un nuevo sistema de cómputo capaz de lidiar con los nuevos desafíos.

Con esta mayor demanda de cómputo, también va creciendo el poder de los procesadores y, con esto, los computadores son capaces de descifrar los algoritmos de seguridad en menor tiempo. Este aumento de cómputo y requerimiento se rige por la llamada Ley de Moore [1], la cual fue publicada en 1965 por Gordon Moore en la revista *Electronics* y estrictamente no es una ley, sino una observación. Esta ley expresa que aproximadamente cada dos años, se duplica el número de transistores en un circuito integrado. Esta ley lleva cumpliéndose aproximadamente unos 50 años (desde 1965 hasta el 2015) como se puede apreciar en la **Figura 1.1**. Cada chip que se produce es más barato, más poderoso y más pequeño que el anterior y esta tendencia parece seguir en curso durante los últimos años. Pero ¿tiene algún límite? Moore afirmó en 2010 que su ley estaba “muerta” ya que poseía un límite infranqueable que pronto sería alcanzado.

La dimensión física de la materia: los átomos. Considerando esta frontera, fue necesario que la ciencia propusiera un nuevo modelo capaz de mejorar aún más la capacidad de los computadores.

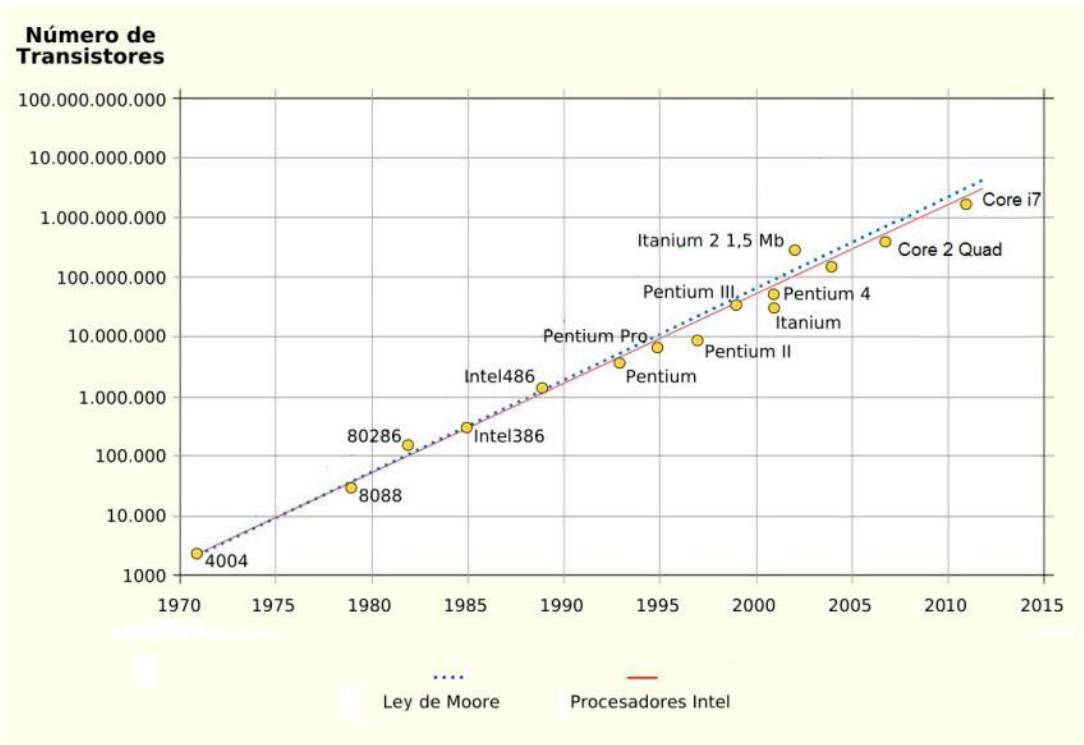


Figura 1.1. Aumento de cantidad de transistores en procesadores Intel según la Ley de Moore en los últimos 50 años

Surge así la computación cuántica como nuevo paradigma para solucionar estos requerimientos informáticos. Las ventajas de esta nueva tecnología radican en su alta potencia, precisión, y seguridad, centrándose en este último punto el estudio de la presente memoria.

Con la llegada de las computadoras cuánticas, se deberá redefinir los conceptos de seguridad de información dando origen a nuevos algoritmos de criptografía que ofrecerán una seguridad inalcanzable para la computación tradicional. Es por ello que se hace necesario entender a cabalidad como estos sistemas pueden ser aprovechados de manera óptima. La capacidad de procesamiento y el mayor volumen de manejo de datos que ofrecen las computadoras cuánticas, generara la degradación del sistema actual de encriptación (los algoritmos actuales podrían ser rotos fácilmente por una computadora cuántica). Según un artículo publicado por la NSA (National Security Agency) [2], la computación cuántica por su potencial, podría representar una amenaza bastante seria para cualquier sistema de seguridad, encriptación o cifrado actual.

El presente trabajo sugiere un acercamiento a los protocolos de criptografía cuántica BB84, E91 y B92, centrándonos en el algoritmo B92, los cuales son protocolos que sirven para distribuir y generar claves que puedan ser compartidas a través de medios de comunicación cuánticos. Los trabajos realizados hasta el momento, cuentan con una implementación para los protocolos BB84 y E91 en los trabajos de Miguel Pinto [3] y Roberto Fritis [4] respectivamente. También se realizó una comparativa entre estos dos protocolos en el trabajo de Patricio Collao [5]. En estos trabajos se obtuvo como resultado práctico los simuladores para estos protocolos, así como una comparativa entre el rendimiento de estos dos, faltando solamente por analizar y simular el protocolo B92. En este trabajo se llevará a cabo el análisis y diseño de un simulador del protocolo B92, en un ambiente distribuido.

1.1. DESCRIPCIÓN DEL PROBLEMA

A medida que avanza la tecnología, el poder de cómputo de los procesadores va creciendo y, con esto, los computadores son capaces de descifrar los algoritmos de seguridad cada vez en un menor tiempo. Por esta razón, es importante estudiar nuevas formas de encriptar la información para que siga siendo segura. En este contexto, existen dos temas a tratar respecto a la criptografía, uno es el cifrado y el otro la distribución de claves. Para el segundo punto existen diversas alternativas, siendo los algoritmos de distribución más avanzados aquellos con orientación cuántica. En este caso hablamos de los algoritmos de distribución de claves cuánticas BB84 [6], E91 [7] y B92 [8], a los que nos referiremos como protocolos de criptografía cuántica.

Los protocolos de criptografía cuántica ofrecen una mayor seguridad que los algoritmos convencionales, debido a las propiedades de la física cuántica que poseen. Un protocolo de este tipo no puede ser descifrado con computadoras normales. Para lograrlo se necesita una computadora cuántica, además de un canal cuántico. La mecánica cuántica no proporciona un método de cifrado como tal, sino que proporciona un protocolo de distribución de clave seguro. Esto permite la implementación de canales de comunicación más seguros para los sistemas actuales y así solucionar la posibilidad futura de la descryptación de información por computadoras cuánticas.

La **Figura 1.2**, muestra un diagrama de red consistente de dos canales de comunicación; por un lado un canal tradicional con protocolos TCP/IP y por otro lado el canal cuántico para distribuir la clave. Este segundo canal puede ser, por ejemplo, una fibra óptica donde se envían fotones polarizados. El nuevo nivel de seguridad que ofrecen estos protocolos, radica en sus propiedades ligadas a la física cuántica. Según este paradigma, un elemento cuántico no puede ser interceptado u observado, porque en el momento en que esto suceda, este cambia su estado, por lo tanto, el intento de ataque queda frustrado, según el principio de incertidumbre de Heisenberg [9]. El problema concreto es la falta del simulador para el protocolo B92, habiendo sido implementados ya los simuladores para los protocolos BB84 y E91. Es importante implementar el protocolo B92 ya que su funcionamiento es más simple respecto a los protocolos BB84 y E91, lo cual simplifica el mecanismo de codificación de bits utilizando solo dos estados no ortogonales en vez de cuatro. Además, este es el último protocolo que falta por simularse de la familia de protocolos de Bennett, para luego, en trabajos futuros poder hacer una comparativa entre los tres.

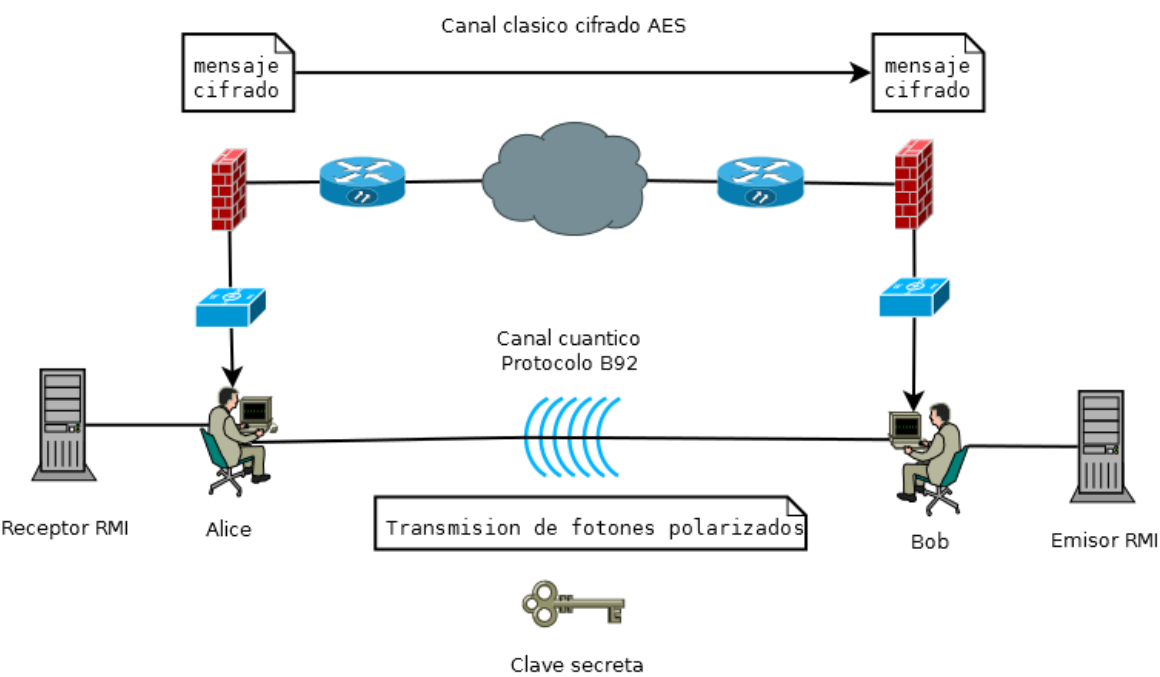


Figura 1.2. Diagrama de comunicación con un canal cuántico.

1.2. SOLUCIÓN PROPUESTA

Se sugiere diseñar un simulador para el protocolo B92, en un ambiente distribuido controlado, en el cual se puedan observar y probar las características cuánticas del protocolo para mejorar la seguridad en la comunicación de datos e información. Para esto se desarrollaran los programas en java que representaran el algoritmo, los cuales deben poder comunicarse en una red distribuida para simular su funcionamiento. Las características cuánticas del medio de comunicación también serán simuladas, utilizando medios comunes, como los estándares de comunicación TCP/IP Ethernet.

Justificación

En el trabajo de Peter Shor [10], se presenta un algoritmo de factorización de números primos de gran tamaño, modelo matemático en el cual está basada la seguridad del algoritmo RSA [11], actualmente el algoritmo más utilizado en internet. Encontrar una solución matemática en un tiempo de orden no exponencial, permite romper la seguridad de los protocolos actuales de cifrado en un tiempo razonable (el tiempo de cómputo para romper una clave RSA de 2048 bits con tecnología actuales es del orden de trillones de años, mientras que con un algoritmo cuántico se reduce a unas 8 horas considerando millones de Qubits [12]). Esto significa que con una computadora lo suficientemente potente se puede romper el algoritmo matemático que está detrás de la mayoría de protocolos de encriptamiento de Internet (el algoritmo de factorización de números grandes). Según el trabajo de Scolnik [13], postula: *“Hoy en día todavía no se sabe si el problema de descomponer un entero a sus factores primos tiene complejidad sub exponencial o polinomial”* (utilizando computadoras normales). Existe un interés práctico y es que si se pudiese factorizar enteros grandes entonces se quebrantarían las firmas digitales hechas con el algoritmo más popular RSA. En el trabajo de Pinto, Collao y en el trabajo realizado por Luis Cáceres [14] se presentó un simulador para el protocolo BB84, E91 y una comparativa entre los rendimientos de ellos. Sin embargo, el algoritmo B92 es el más simple tanto en diseño como en implementación, por lo tanto promete ser un buen candidato para ser la alternativa a utilizar en el futuro.

DIAGRAMA DE SOLUCION

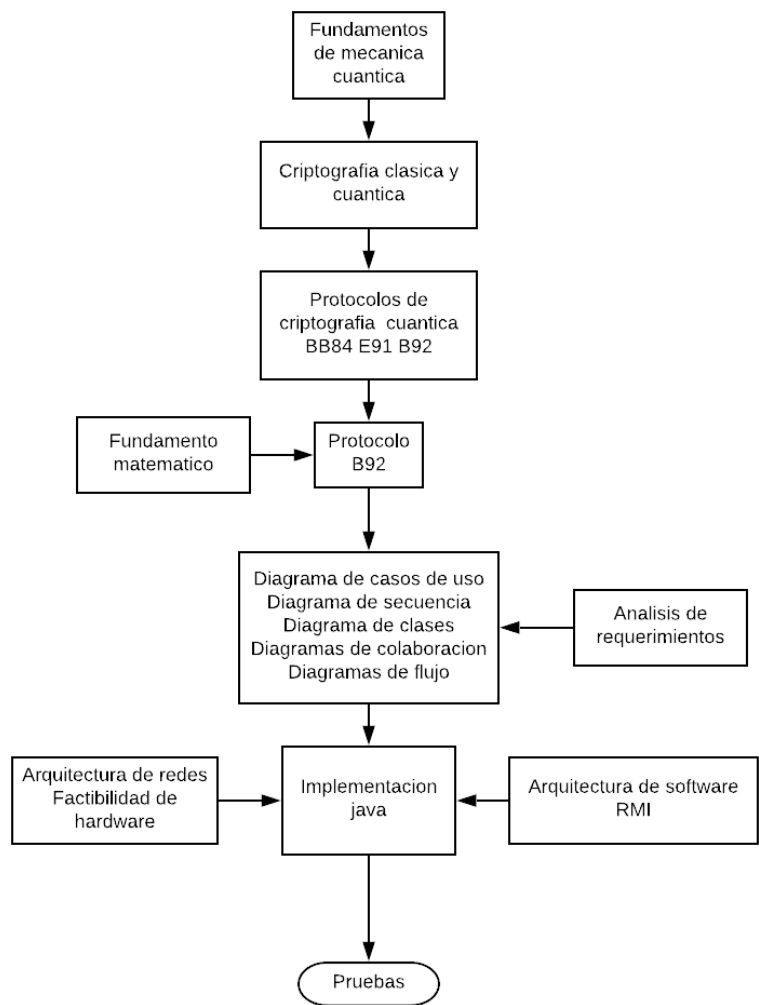


Figura 1.3. Diagrama de solución propuesto.

La estructura del estudio y posterior tesis puede apreciarse en la **Figura 1.3**, en la cual se muestra las etapas a considerar para el proyecto. La primera etapa consiste en realizar un estudio teórico de los problemas que afronto la física en los años 1900 y como esto culmino en el nacimiento de una nueva rama de la ciencia, la mecánica cuántica.

La segunda etapa consta de los fundamentos de la mecánica cuántica y el uso de esta en la criptografía. Se estudian conceptos matemáticos vitales para comprender los comportamientos de los simuladores. Esta sección se divide en:

- Fundamentos de mecánica cuántica
- Computación cuántica

- Representación de información a través de canales cuánticos
- Criptografía cuántica

En la tercera etapa se analizan los protocolos de criptografía cuántica más importantes, aquellos basados en el principio de incertidumbre de Heisenberg y las desigualdades de Bell. El protocolo B92 es el que se implementara como prototipo y se realizara un análisis matemático de sus fundamentos. Esta sección se divide en:

- Protocolos de criptografía cuántica BB84 y E91
- Protocolo de criptografía cuántica B92
- Prototipo de software para el protocolo B92

En la cuarta etapa del proyecto, se realiza el análisis de requerimientos para la aplicación de software. También se realizan los diagramas correspondientes a cada etapa de análisis. Esta etapa se divide en:

- Requisitos del sistema
- Requisitos funcionales
- Requisitos no funcionales
- Restricciones y limitaciones
- Diagramas de casos de uso, Secuencia, Clases , Colaboración

La quinta etapa es el desarrollo del software propiamente tal y la creación de la aplicación Java. En esta etapa se implementan los casos de uso diseñados en la etapa de requisitos. Se definen los pasos para la aplicación y cada módulo que se debe programar. Las actividades realizadas son las siguientes:

1. Generar las secuencias aleatorias de bits tanto en el emisor como en el receptor de manera independiente
2. Generar las secuencias de polarizaciones en el emisor y el receptor, de acuerdo a la secuencia de bits generada previamente
3. Enviar el esquema del emisor Alice al receptor Bob y guardarlo localmente en este ultimo
4. Comparar los esquemas de polarizaciones de Alice y Bob

5. Obtener la clave en bruto o *Raw Key* guardando solo los bits en los que Bob detecto el fotón que se entiende como los campos que no concuerdan en los esquemas de Bob y Alice
6. Obtener la clave depurada o SIFTED KEY eliminando los bits que colapsan en la componente incorrecta según el principio de incertidumbre de Heisenberg. Para cada bit existe un 50% de probabilidad de ser eliminado en este proceso
7. Intercambiar la clave depurada entre Alice y Bob
8. Crear la interfaz gráfica para mostrar los valores al usuario
9. Utilizar la clave generada con el protocolo B92 y usar un canal clásico para enviar mensajes encriptados mediante el algoritmo AES
10. Crear una red de laboratorio con IP's estáticas para simular la comunicación distribuida

1.3. OBJETIVOS

1.3.1. Objetivo General

Diseñar un simulador en java para el protocolo de criptografía cuántica B92 en un ambiente local y distribuido. En el cual se debe compartir una clave secreta entre el emisor y el receptor y luego poder usarla para cifrar mensajes en un canal clásico.

1.3.2. Objetivos Específicos

1. Estudiar e investigar la teoría matemática y física específicamente aquellos conceptos necesarios para el desarrollo de un algoritmo de criptografía cuántico.
2. Definir los fundamentos matemáticos del algoritmo B92 y los mecanismos de propagación de información sobre canales cuánticos
3. Definir los requerimientos funcionales y no funcionales para el desarrollo de un simulador del protocolo B92
4. Diseñar los diagramas UML necesarios para el análisis de software y funcionamiento de los protocolos.
5. Diseñar un prototipo de simulador para el protocolo
6. Implementar el prototipo en java RMI
7. Ejecutar los programas y medir los resultados en un ambiente distribuido

1.3.3. Actividades

Lista de actividades:

- Estudio de la teoría cuántica
- Estudio de los protocolos de criptografía cuántica BB84 y E91
- Estudio del protocolo de criptografía cuántica B92
- Análisis del protocolo B92 y definición matemática-física
- Análisis de requerimientos de la solución
- Implementación de la aplicación en java
- Instalación de la red distribuida

La **Figura 1.4**, muestra el diagrama de flujo de tareas, donde se aprecian las tareas necesarias para cumplir los objetivos.

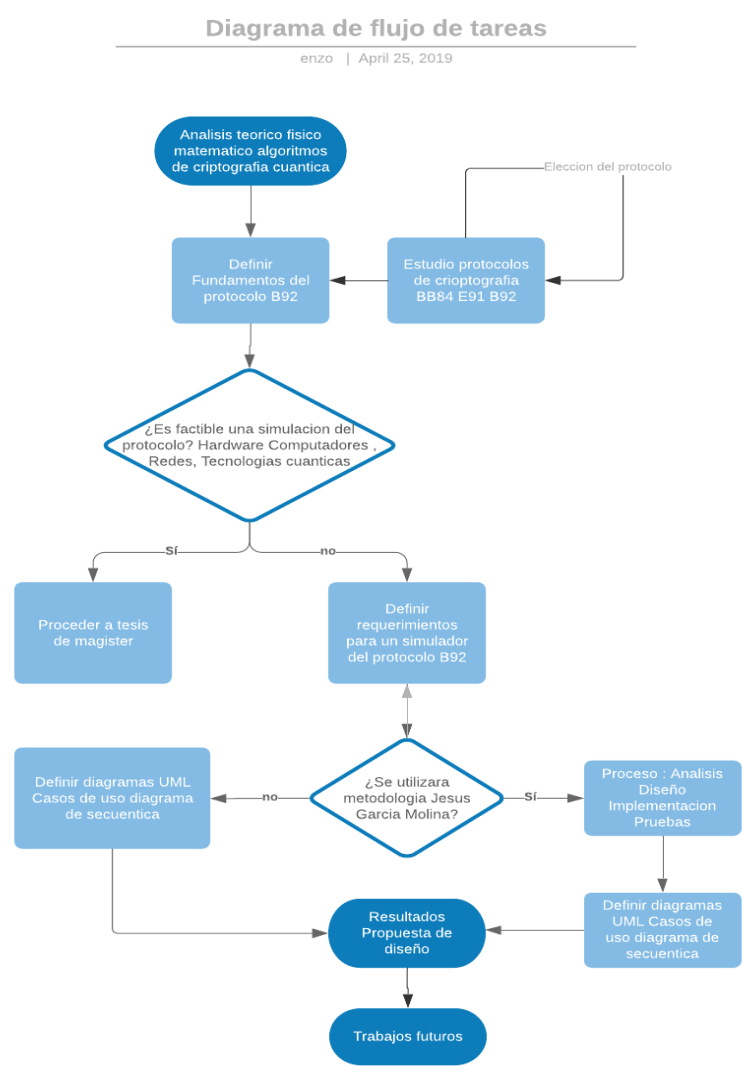


Figura 1.4. Diagrama de flujo de tareas

1.4. PLANIFICACIÓN

La planificación para el año 2019, se realizó considerando todo el año desde marzo hasta diciembre. Cada actividad se realizara en una determinada cantidad de meces, permitiendo también varias actividades ocurriendo al mismo tiempo.

MESES	marzo		abril		mayo	junio	julio	agosto	septiembr	octubre	noviembre	diciembre
ACTIVIDADES/SEMANAS	1	2	3	4	1	2	3	4	1	2	3	4
Estudiar e investigar el fundamento matematico y fisico												
Estudiar e investigar la criptografia clasica												
Estudiar e investigar los protocolos cuanticos existentes												
Analisis de requerimientos, casos de uso y diagramas												
Implementacion en java del protocolo B92												
Montar una red distribuida de prueba para el protocolo												
Probar el protocolo en un ambiente distribuido												
Conclusiones												

Figura 1.5. Grafica Gantt para el flujo de tareas

1.5. ESTRUCTURA

Capítulo 1 Introducción: Muestra la problemática de manera general, se definen los objetivos generales y específicos del sistema, se presenta el ámbito y entorno de trabajo de la memoria y se plantean los tiempos de ejecución del trabajo.

Capítulo 2 Marco teórico: Se presentan los fundamentos teóricos físicos y matemáticos para el funcionamiento del algoritmo. La física cuántica y los conceptos que son usados en criptografía. También se hace un recuento de los fundamentos de criptografía simétrica y asimétrica.

Capítulo 3 Descripción de la solución: Se describe la solución de manera más detallada y se definen todos los procedimientos y estructuras para el desarrollo del proyecto. Entre otros se tiene la definición del modelo de desarrollo de software, la arquitectura RMI, el lenguaje a utilizar, la arquitectura de red a utilizar.

Capítulo 4 Desarrollo: En este capítulo se desarrolla todo el análisis de ingeniería de software así como la implementación del algoritmo. Se definen los requisitos funcionales y no funcionales, se realiza el análisis de casos de uso y el diseño en diagramas del sistema, para luego pasar a la implementación y una prueba inicial de la aplicación.

Capítulo 5 Resultados: Una vez implementado el algoritmo, llega el momento de realizar las pruebas. En este capítulo se realizan las pruebas del programa en dos ambientes de prueba distintos, uno local y uno en red distribuida.

Capítulo 6 Conclusiones y trabajo futuro: este capítulo consta con las conclusiones finales del trabajo, las cuales están divididas en un resumen completo de todo lo realizado y luego las conclusiones de cada etapa del desarrollo. También se plantean trabajos futuros para esta memoria.

Anexos: Se tienen 7 anexos numerados de manera consecutiva, los cuales están indicados en el pie de página de la página correspondiente a su cita.

CAPÍTULO 2: MARCO TEÓRICO

2.1. INTRODUCCIÓN A LA MECÁNICA CUÁNTICA

El periodo histórico comprendido entre los años 1900 y 1925, conlleva el desarrollo de una nueva teoría física. Este conjunto de nuevas teorías contiene las soluciones revolucionarias a las que se enfrentaron los físicos del siglo veinte para describir adecuadamente los problemas no- resueltos de la física clásica del siglo diecinueve. Las contradicciones entre la teoría y el experimento motivan la búsqueda de nuevas ideas fuera de la intuición convencional. Las nuevas teorías resuelven los antiguos problemas, pero generan otros nuevos y así, de manera continua va avanzando la ciencia [15].

En este capítulo se estudian los conceptos históricos que dieron desarrollo a las primeras ideas cuánticas. Se analizan los conceptos más importantes para el uso en criptografía. También se desarrollan las bases matemáticas y físicas para el protocolo E92.

2.1.2. Dualidad onda-partícula y Partícula-onda

Dualidad onda-partícula

A raíz de los experimentos explicados en el efecto fotoeléctrico, se tenía constancia de que la luz (onda electromagnética) poseía características de partícula como cuando choca un cuanto de luz con la superficie del metal.

Según el trabajo de Maxwell con sus ecuaciones, sabemos que la luz es una onda electromagnética y como tal posee los comportamientos típicos de una onda como por ejemplo, la difracción, es decir que cambia la forma del frente de onda cuando se encuentra con algún obstáculo. Por otra parte cualquier fenómeno físico queda completamente identificado como una onda si existe un patrón de interferencia.

Una característica de un fenómeno corpuscular es el momentum, definido como el producto de la masa y la velocidad. Se sabe que la luz no posee masa y las ondas

electromagnéticas no pueden transportar materia, sin embargo la luz tiene la capacidad de generar momentum cuando choca contra un material. ¿Cómo es posible?

Para identificar las características de una onda, primero definimos una onda como una perturbación física que viaja en el tiempo y el espacio, puede moverse en un medio o en el vacío, puede transportar energía momentum lineal y momentum angular, pero no puede transportar materia. Una onda puede ser:

- Onda mecánica: necesita de un medio material para propagarse
- Onda electromagnética: no necesita de un medio de propagación
- Ondas longitudinales: el medio material se mueve en la misma dirección en que viaja la onda
- Ondas transversales: el medio material se mueve en dirección perpendicular a la dirección de propagación de la onda
- Ondas longitudinales y transversales.

Las ondas electromagnéticas como la radiación y la luz se rigen por las leyes de Maxwell. Sin embargo cuando se utilizaban estas leyes para describir otros fenómenos, la teoría se contradice con los experimentos. Cuando se envió luz monocromática a chocar con una superficie metálica en el efecto fotoeléctrico, se produjo una interacción entre “dos partículas”, el electrón de la superficie y el fotón asociado a la onda luminosa. Es en estos experimentos donde queda en evidencia la característica dual de la luz que al ser una onda también puede comportarse como partícula.

Dualidad partícula-onda

Hemos visto que las ondas poseen un fenómeno de difracción e interacción que las identifican como tales, y que también por los experimentos de Einstein y el efecto Compton la luz puede manifestarse con características corpusculares, veremos ahora si la dualidad se cumple inversamente, si es posible encontrar partículas que se comporten como ondas.

En el año 1924 el físico Luis de Broglie postula lo que llamo “ondas de materia”. Una teoría en la que la materia, las partículas podían comportarse como una onda, esto es que una partícula sea capaz de difractarse o generar interferencia, sin embargo la comprobación de esta teoría no fue encontrada experimentalmente sino hasta el año 1927 con el experimento de la doble rendija. Para que un fenómeno tenga características corpusculares, se encuentran las magnitudes energía y momentum mientras que en un experimento ondulatorio se encuentran magnitudes como frecuencia o longitud de onda. Si una partícula posee longitud de onda o frecuencia entonces estamos en presencia de un fenómeno partícula-onda.

De Broglie planteo que las partículas tienen asociadas una onda y que esta onda posee una longitud de onda λ igual a la constante de Planck dividida por el momentum de los fotones asociados a esa partícula, se basó en la idea de que el movimiento estable de los electrones en torno al átomo está regido por números enteros cuantizados (como indico Bohr) y hasta la fecha los únicos fenómenos que involucraban números enteros en la física de ese entonces eran los fenómenos de interferencia y módulos normales de vibración, con esta idea surge la necesidad de asignarle a los electrones una periodicidad (una frecuencia). Su hipótesis le mereció el premio nobel de física de 1929.

La naturaleza ondulatoria de la materia se ve en manifiesto cuando la longitud de onda de las partículas es un poco mayor que las dimensiones de los obstáculos que atraviesan ya que en ese caso las ondas asociadas a esas partículas se difractan con facilidad, por lo tanto para presenciar un fenómeno de partícula ondulatoria se necesitan masas muy pequeñas como las partículas atómicas para que la longitud de onda de estas masas sea lo suficientemente grande como para que se puedan observar los fenómenos.

2.1.3. Postulados de De-Broglie 1924

- Para las partículas y para las ondas, la energía total E de cualquiera de estas dos magnitudes físicas está relacionada con la frecuencia γ de la onda asociada con su movimiento según la ecuación

$$E = h\gamma$$

- El momento lineal p de cualquiera de ellas está relacionado con la longitud de onda λ de la onda asociada según la ecuación

$$p = \frac{h}{\lambda}$$

En estos postulados se ve que los aspectos asociados a una partícula como energía E y momentum p están asociados, a través de la constante de Planck h , con los aspectos ondulatorios frecuencia γ y longitud de onda λ .

La ecuación del momento lineal puede escribirse despejando la longitud de onda de la siguiente manera

$$\lambda = \frac{h}{p}$$

La cual se conoce como la longitud de onda de De-Broglie asociada a una partícula en movimiento con momentum p . Predice la existencia de ondas de materia.

La teoría de De-Broglie demuestra la otra faceta de la luz. Su dualidad depende del tipo de experimento al que se le someta, la luz puede comportarse como onda o como partícula pero nunca como ambas al mismo tiempo. Esto se conoce como *el principio de complementariedad de Bohr*, el cual dice que “los modelos de onda y de partícula son complementarios, de manera que si una medición prueba el carácter ondulatorio de la radiación o la materia, entonces es imposible probar el carácter corpuscular en la misma medición del experimento y viceversa”.

Los postulados de De-Broglie no pudieron ser comprobados hasta el año 1927 en el experimento de Davidson y Gartner llamado el experimento de la doble rendija.

2.1.4. Experimento doble rendija

La mecánica clásica explica los fenómenos que ocurren en una escala intermedia, pero no sirve para explicar modelos cosmológicos como el modelo del Big Bang, la

gravitación cuántica, las m-brasas, la teoría de súper cuerdas, o los modelos del universo cíclico. Tampoco explica porque un electrón gira y no emite radiación.

La construcción de una teoría que explique el dominio de las masas, longitudes y tiempos muy pequeños o muy grandes exige un cambio radical en las leyes y en las ideas clásicas fundamentales. Por ejemplo: al pasar un haz homogéneo de electrones por un cristal, en el haz emergente se observa una figura formada por máximos y mínimos de intensidades separados por espacios, de igual que sucede con las ondas electromagnéticas. Es decir en ciertas condiciones las partículas electros se comportan como ondas. Este experimento se conoce como el experimento de la doble rejilla, y muestra dos conceptos fundamentales de la mecánica cuántica.

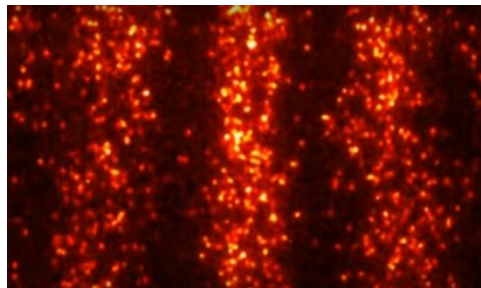


Figura 2.1. Experimento doble rendija real.

El experimento de la doble rendija pone de manifiesto dos características desconcertantes de ese mundo. La primera es que, a escala micro, los objetos físicos tienen una naturaleza dual: según las circunstancias, pueden comportarse como un conjunto de partículas o como una onda. Y la segunda consiste en que el hecho de observarlos hace que actúen de una manera o de otra. La **Figura 2.1** muestra el experimento de forma real.

Esta forma de aclarar estas idead respecto a la dualidad onda-partícula del electrón es mediante un experimento en el que se disparan electrones hacia una doble rejilla, como en la figura. Supongamos que el ancho de la rejilla es pequeño en comparación con la longitud de onda de los electrones, por lo que no es necesario preocuparse por máximos y mínimos de difracción. Lejos de las rejillas y a una distancia mayor que “ d ”, con d como la separación entre ellas, se ubica un detector de electrones. Si la pantalla del detector recibe electrones durante un tiempo largo, se halla un patrón de interferencia de ondas. Este patrón no debería esperarse si los electrones se

comportaran como si fueran partículas, el patrón refleja un comportamiento más distintivo a una onda.

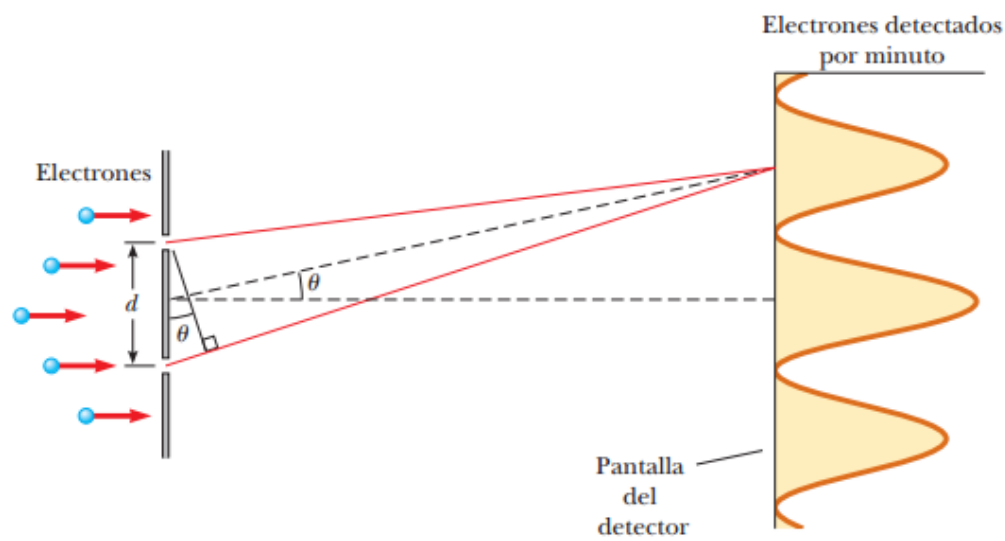


Figura 2.2. Interferencia de los electrones.

La interpretación estándar nos dice que el electrón se lanza y se recoge como una partícula, pero se propaga como una onda. Es decir, que durante su recorrido el electrón está distribuido o superpuesto en toda el área que ocupa su onda, por lo que atraviesa las dos rendijas a la vez e interfiere consigo mismo hasta impactar contra la segunda placa. En ese momento, como consecuencia del impacto, el electrón vuelve adoptar la naturaleza de partícula, en términos más precisos diríamos que colapsa su función de onda, situándose en uno de los múltiples puntos atravesados por la onda. Al comenzar el experimento los electrones se distribuirán por la segunda placa de una forma aparentemente aleatoria, pero al incrementar el número de impactos veremos cómo va formándose el “patrón de interferencia”. Es decir, que la posibilidad de impactar en uno u otro punto está determinada por la onda. La **Figura 2.2** muestra el experimento. Donde la separación “ d ” entre las rejillas es mucho mayor que el ancho de cada una y mucho menor que la distancia entre las rejillas y el detector [16].

El observador

¿Qué pasa cuando colocamos un detector para averiguar por qué rendija pasa el electrón? Pues que el “patrón de interferencia” desaparece y los electrones impactan en la segunda placa como si fuesen canicas. Es decir, que al tratar de observar el sistema, hemos actuado sobre él, obligando a nuestro electrón a comportarse como

una partícula. Los fotones que hemos enviado para detectarlo han interactuado con él y alterado el resultado del experimento.

Esto lleva al concepto de que al medir el experimento existe una incertidumbre sobre si realmente estamos midiendo lo correcto o hemos alterado la naturaleza del fenómeno.

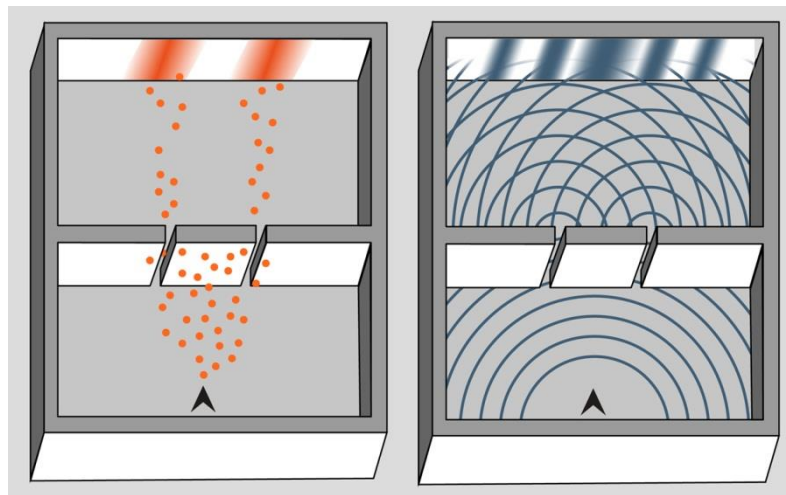


Figura 2.3. Comparativa entre un fenómeno de partículas y ondulatorio.

Niels Bohr, decía en los años 20 del siglo pasado, que ya no somos meramente observadores de lo que medimos sino también actores. De repente, una ciencia dura como la física comenzaba a cuestionar el paradigma de la objetividad: ¿podemos conocer la realidad sin interferir en ella y sin que ella interfiera en nosotros? Bohr plantea que la presencia del observador introduce una incertidumbre insoslayable. De acuerdo con Werner Heisenberg y su principio de incertidumbre formulado en 1927, es imposible conocer al mismo tiempo todas las propiedades de nuestra partícula porque, al observar una, estamos alterando el resto. Al querer conocer la posición exacta de un electrón, por ejemplo, su velocidad queda muy indeterminada. En la **Figura 2.3** se aprecian las dos franjas verticales correspondientes al experimento con naturaleza de partícula y en la derecha se ve el patrón de difracción de un fenómeno ondulatorio [17].

2.1.5. Principio de Incertidumbre de Heisenberg

Definición de error, incerteza o incertidumbre en una medición

En un sistema de coordenadas cartesianas se acostumbra representar la ubicación de algún objeto mediante sus coordenadas.

$$(x, y, z) = (2cm, 5cm, 3cm)$$

Sin embargo esta idealización no toma en cuenta la extensión completa del cuerpo como materia en el espacio. Supongamos que el objeto que estamos posicionando es un cubo con un grosor de 5mm, la posición sería:

$$(x, y, z) = (2cm \pm 0,5, 5cm \pm 0,5, 3cm \pm 0,5)$$

Esto es válido incluso para partículas muy pequeñas como el electrón, dado que aunque clásicamente se les considera como una partícula puntual, en realidad ninguna partícula puede estar confinada en un punto del espacio de dimensión cero. De esta forma a nivel microscópico una partícula esférica de diámetro Δx no se puede considerar ubicada en un cierto punto x .

Si se considerara una partícula como un objeto puntual, su posición podría representarse como en la parte izquierda de la **Figura 2.4**, en la cual su posición calza perfectamente con el sistema de coordenadas, sin embargo los objetos macroscópicos poseen masa y esta masa ocupa un espacio, por lo tanto una visión más realista de la posición de un objeto se refleja en la parte derecha de la **Figura 2.4**.

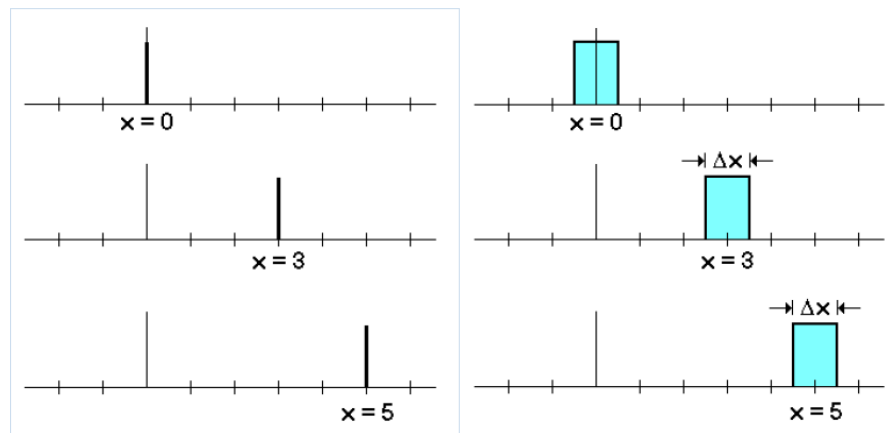


Figura 2.4. Distintas formas de modelar la posición de un objeto.

Si el objeto no se considera como un objeto puntual, no como si toda la masa estuviera concentrada en un punto único del espacio de dimensión cero, entonces debemos considerar un “error” en las mediciones o una incertidumbre en las mediciones debido a la inexactitud de los instrumentos de medición.

Este error o incertidumbre en la medición puede trasladarse al momentum de igual forma. No podemos definir el momentum como $p = mv$ si la posición de la partícula no es exacta.

$$p = \pm \Delta p$$

En mecánica clásica, las ecuaciones de movimiento de un sistema sometido a fuerzas conocidas se puede resolver de forma específica para todo tiempo t , conociendo las condiciones iniciales del sistema x_0 p_0 v_0 . De este modo se pueden obtener las posiciones y el momentum de la partícula para cada instante de tiempo. Vemos que en mecánica clásica el conocimiento preciso de la posición y del momentum de una partícula en $t = 0$ permite conocer el estado futuro del sistema de manera precisa.

Sin embargo hay que remarcar que en cada proceso experimental de medición, el observador interactúa con el sistema generando incertidumbres en las mediciones. Estas perturbaciones generadas por los instrumentos son despreciables ya que en el mundo macroscópico los objetos poseen gran masa entonces los efectos del acto de medición son insignificantes. En consecuencia, en mecánica clásica o más bien dicho en mediciones macroscópicas, se acepta de manera natural el hecho de que se pueden medir simultáneamente la posición x y el momentum p de una partícula con la misma precisión despreciando la incertidumbre. Estas ideas serían cambiadas con la llegada de la teoría cuántica para fenómenos microscópicos, donde la incertidumbre toma real importancia.

Supongamos el siguiente experimento: medir la posición de un electrón observándolo a través de un microscopio. Para “ver” al electrón la luz debe rebotar en él y luego llegar al microscopio, pero la luz está compuesta por ondas de fotones, los cuales son del tamaño del electrón, por lo tanto al chocar el fotón con el electrón y regresar al lente se produce un choque comparable al choque entre dos vehículos.

El fotón provoca un cambio significativo en el estado de movimiento del electrón que queríamos observar, es decir, la perturbación no es despreciable ya que al con el choque podemos obtener información precisa de la posición del electrón pero si quisiéramos conocer su momentum o su velocidad veríamos que el choque cambio totalmente su velocidad y su momentum.

Mientras más precisa sea la ubicación del electrón por medio del fotón, mayor alteración sufrirá la velocidad del electrón y viceversa, si tratamos de enviar un fotón que produzca el menor efecto sobre la velocidad del electrón, entonces se pierde información sobre la posición del electrón en el momento del choque con el fotón.

2.1.6. Postulados de Heisenberg 1927

La determinación experimental, en el mismo instante de la posición y el momentum de la materia o de una onda se puede realizar con una precisión no mayor que la permitida por el Principio de incertidumbre de Heisenberg, postulado en 1927 [9]

- En cualquier medición simultanea de la posición y del momentum lineal de una partícula se cumple la siguiente relación entre las incertezas cometidas en el proceso de medición

$$\Delta x \Delta p \geq \frac{\hbar}{2}$$

Donde el término Δx representa la incerteza cometida en la medición de la posición x y el término Δp la incerteza cometida en la medición del momentum lineal y \hbar es la constante de Planck

- Entre las mediciones de energía E y tiempo t necesario para su medición existe también una restricción en el producto de sus incertezas correspondientes

$$\Delta E \Delta t \geq \frac{\hbar}{2}$$

Donde ΔE es la incerteza en la medición de la energía del sistema y Δt es el intervalo de tiempo asociado al cambio de la energía del sistema

Las relaciones de incerteza se refieren a mediciones simultáneas del par de variables involucradas, por eso es un producto entre los errores individuales, y este resultado no depende de la calidad de la medición ni de la calidad de los instrumentos usados en el proceso de medida. Se debe destacar que es el producto de las incertezas lo que importa en el principio de incertidumbre, de modo que si en una medición simultánea logramos disminuir la incerteza asociada a una de las variables ($\Delta x \rightarrow 0$ *por ejemplo*), la incerteza asociada a la otra variable se hace muy grande ($\Delta p \rightarrow \infty$) de manera que el producto siempre se mantiene $\geq \frac{\hbar}{2}$.¹

Conclusión

Como conclusión el principio de incertidumbre establece dos premisas, la primera es que en el mundo subatómico no es posible conocer al mismo tiempo dos observables correlacionados como posición-momentum o energía-tiempo. Segundo, al llevar a cabo una medición, el estado original del sistema cuántico se altera de forma inevitable

La seguridad de los protocolos de criptografía cuántica BB84 y B92 hace uso del principio de incertidumbre para lograr la seguridad del protocolo, si un intruso quisiera leer un mensaje que viaja por un canal cuántico, debe medir los estados cuánticos de las partículas viajeras. Al hacer esto el sistema se modifica, así los usuarios legítimos pueden saber que cualquier cambio en el estado de la partícula significa la presencia de un intruso.

¹ Ver apéndice: microscopio de Heisenberg, El acto de medición, Ecuación de Schrödinger y Apéndice matemático

2.1.7. Entrelazamiento cuántico

Los trabajos presentados en ERP, Heisenberg y Schrödinger predijeron un fenómeno experimental conocido como *entanglement* o entrelazamiento cuántico. Inicialmente el término fue concebido por Schrödinger para describir sistemas cuyas partes han interactuado en el pasado y permanecen enlazados.

Cuando dos sistemas para los que se conocen los estados respectivos de cada uno entran en interacción física debido a fuerzas entre ellos y luego de un tiempo se separan, ya no es posible describirlos en términos de sus partes individuales.

El entrelazamiento puede explicar la no localidad de los experimentos EPR y ser usado como una fuente física de información para protocolos cuánticos o teleportación cuántica. El origen de este fenómeno proviene de la naturaleza intrínseca del estado de un sistema cuántico. Debido a que el estado cuántico puede ser matemáticamente una superposición de funciones de onda.

Como una ecuación diferencial posee una solución particular y general, las funciones de onda $\varphi()$ ósea el elemento que define el estado del sistema, son soluciones de la ecuación diferencial de Schrödinger, si $\varphi_1()$ es una función de onda solución de la ecuación de Schrödinger y $\varphi_2()$ también es una solución de la misma ecuación diferencial, entonces $\varphi_1() + \varphi_2() = \varphi_3()$ es solución también de la misma ecuación diferencial.

El entrelazamiento define las siguientes características:

1. Cualquier modificación aplicada a una de las partículas entrelazadas, por ejemplo sobre la partícula 1, implica una variación en su pareja entrelazada la partícula 2. La alteración en la partícula 2 depende de la modificación realizada sobre la partícula 1.
2. La alteración de la partícula 2 debido a la modificación de la partícula 1 no depende de la distancia entre ellos.
3. La partícula 2 se altera de forma instantánea una vez que la partícula 1 se modifica.

Los experimentos de entrelazamiento generalmente utilizan electrones como partícula elemental y el espín como el sistema para medir el estado cuántico.

Ejemplo: Supongamos que tenemos dos electrones con espines opuestos (\uparrow, \downarrow) o (\downarrow, \uparrow), uno de estos electrones se entrega a Alice y el otro a Bob (ambos fueron creados de la misma fuente de partículas en un proceso EPR).

Alice recibe su electrón herméticamente cerrado en una “caja” de manera que no haya ninguna interacción con él. Bob mantiene el suyo en una caja también.

En este punto ocurren dos situaciones al mismo tiempo:

1. Alice tiene espín up \uparrow y Bob tiene espín down \downarrow
2. Alice tiene espín down \downarrow y Bob tiene espín up \uparrow

Antes de abrir su caja, el electrón no está ni en estado espín up ni espín down, sino en un estado indeterminado representado como la superposición de ambos estados de ambos electrones.

Cuando Alice abre su caja, si ella encuentra su electrón con espín up, entonces Bob tiene su electrón con espín down, sin importar la distancia entre Alice y Bob, el acto de mirar en la caja por parte de Alice instantáneamente afecta el electrón de Bob.

La apertura de la caja equivale a realizar una medición en mecánica cuántica. Una medición es simplemente una observación de un elemento en particular en un sistema, no te dice el estado del sistema directamente pero se puede deducir el estado completo del par de partículas debido a que esta entrelazadas. La observación de Alice cambia el estado del sistema entrelazado de estar en “indeterminado” a “up o “down”, la medición perturba irremediablemente el sistema.

Esta propiedad cuántica permite saber si un intruso intercepta el mensaje en el camino, ya que con solo el hecho de observarlo se modifica el mensaje y el sistema en sí, cambiando el estado original del sistema haciendo inútil leerlo.

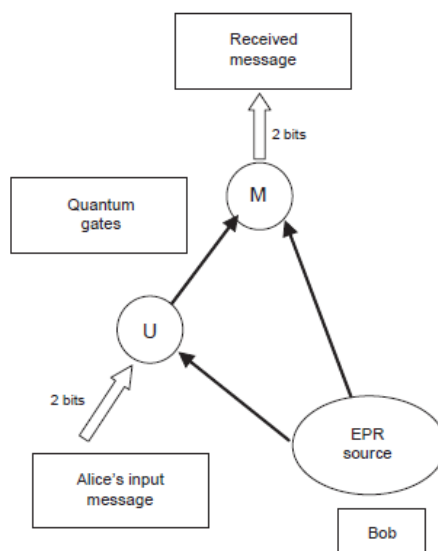


Figura 2.5. Generación de un par de partículas entrelazadas [18].

El entrelazamiento cuántico dio origen a varios descubrimientos relacionados con la computación e información, entre ellos tenemos la criptografía cuántica, el codificado denso y la teleportación cuántica de información. Todos estos avances están relacionados con el entrelazamiento cuántico y han sido demostrados en el laboratorio.²

La aplicación más interesante del entrelazamiento cuántico, se da en la criptografía cuántica, como se vio en el ejemplo de Alice y Bob y la **Figura 2.5**, se puede interpretar el fenómeno como el concepto de privacidad. Si la clave secreta esta entrelazada en dos partes entonces cualquier observación o interrupción en la distribución de la clave seria descubierto por las partes inmediatamente.

2.1.8. Polarización de fotones

La luz es una onda electromagnética y como tal, está compuesta por un campo eléctrico y un campo magnético oscilantes, perpendiculares entre si y perpendiculares a la dirección de propagación de la onda. La **Figura 2.6**, muestra los campos eléctricos y magnéticos oscilando mientras la onda se propaga por el eje x.

² Apéndice: El espín

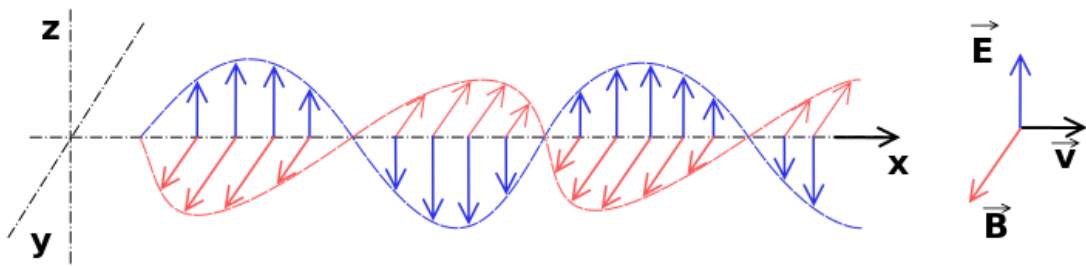


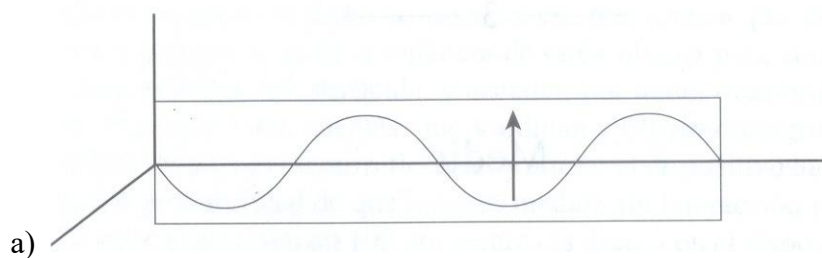
Figura 2.6. Campos magnéticos y eléctricos de una onda.

Se define la polarización como la propiedad que tiene una onda para oscilar en una determinada orientación. Un fotón polarizado puede pasar o no pasar a través de un filtro polarizador, si logra pasar entonces quedara alineado con el filtro sin importar su estado inicial.

La probabilidad de que un fotón polarizado pase por un filtro que este polarizado en el mismo ángulo que el fotón es del 100%, para un filtro polarizado de manera perpendicular al ángulo de polarización del fotón es del 0% y para un filtro polarizador a 45° es del 50%.

La polarización de un fotón se mide mediante el plano en el que oscila el campo eléctrico del fotón mientras se propaga, se ignora el campo magnético por ser más pequeño respecto al eléctrico y porque el vector de campo magnético se puede obtener a partir del vector campo eléctrico.

En la **Figura 2.7** se observan las distintas opciones de vibración de una onda, para cada uno de los ejemplos, existe un ángulo que representa la inclinación del plano en el que oscila la onda.



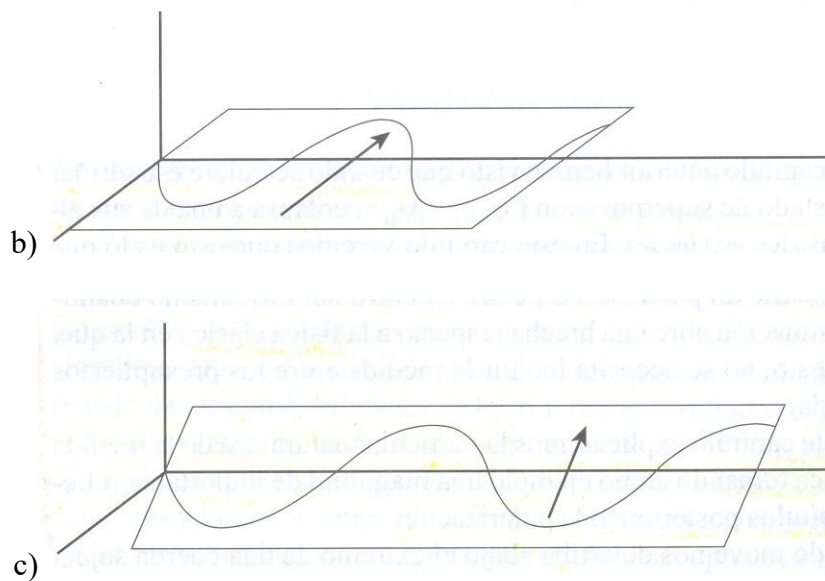


Figura 2.7. Polarización de un fotón en ángulos: a) = 90° b) = 0° c) = 45° .

Polarización lineal, circular y elíptica

Si descomponemos el vector campo eléctrico E en sus componentes E_x, E_y ambas son perpendiculares a la dirección de propagación y varían su amplitud con el tiempo y la suma de ambas componentes va trazando una figura geométrica.

- Si esa figura geométrica es una recta, entonces la polarización se denomina lineal. Figura a.
- Si la figura trazada es un círculo, entonces la polarización se denomina circular. Figura b.
- Si la figura trazada es una elipse, entonces la polarización se denomina elíptica. Figura c.

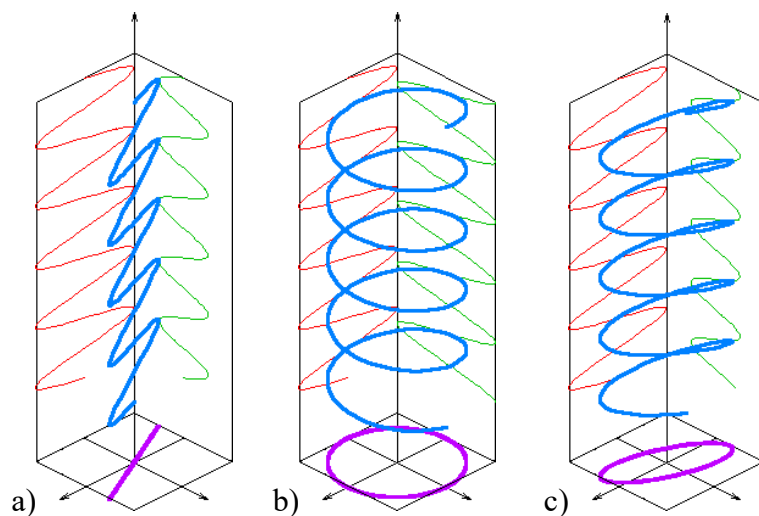


Figura 2.8. Polarización lineal a) circular b) y elíptica c).

La función de un filtro polarizador es que la onda electromagnética no polarizada que oscila en todas las direcciones perpendiculares a la propagación, después de pasar por el filtro quede oscilando solo en un determinado plano, ya sea de cualquiera de las tres formas vistas. Como se ve en la **Figura 2.9**, la luz oscila en muchas direcciones antes de pasar por el filtro, y luego de pasar el filtro todas las demás componentes que no sean paralelas al filtro son bloqueadas.

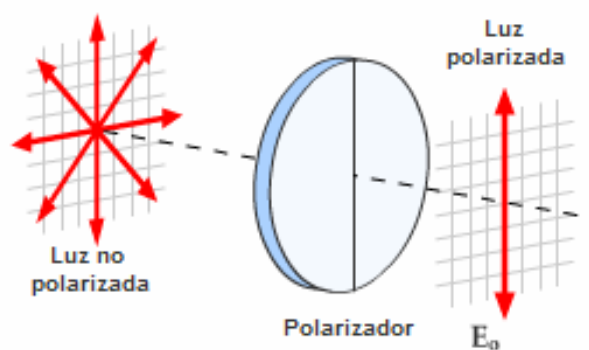


Figura 2.9. Filtro polarizador que permite solo el paso de la luz vertical.

En el caso de la polarización lineal, esta se produce cuando la oscilación del plano perpendicular a la dirección de propagación se produce a lo largo de una línea recta. Se puede representar cada oscilación descomponiéndola en dos ejes X e Y. La polarización lineal se produce cuando ambas componentes están en fase (con un ángulo de desfase nulo, cuando ambas componentes alcanzan sus máximos y mínimos simultáneamente) o en contrafase (con un ángulo de desfase de 180° , cuando cada una de las componentes alcanza sus máximos a la vez que la otra alcanza sus mínimos).

Considerando una polarización lineal, en la cual tenemos dos posibles direcciones de oscilación o vibración, vertical u horizontal a 0° y 90° o 180° y 270° , diremos que esta polarización es de tipo **lineal rectilínea**.

En el caso de una polarización lineal donde las posibles direcciones de oscilación son diagonales a 45° y 135° o 225° y -45° , diremos que esta polarización es de tipo **lineal diagonal**.

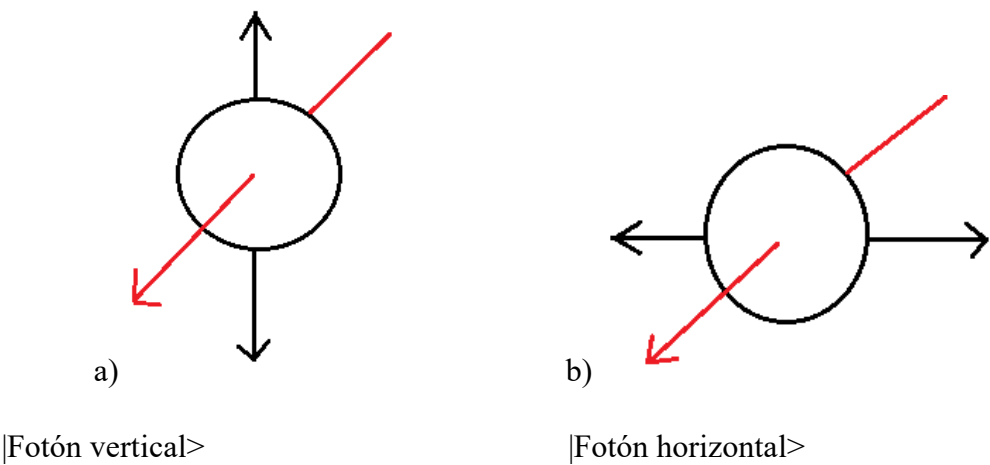


Figura 2.10. Polarización lineal rectilínea horizontal y vertical.

En la **Figura 2.10**, en a) el fotón vibra verticalmente a la dirección de movimiento, en la figura b) el fotón vibra horizontalmente respecto a la dirección del movimiento. La notación matemática para estos dos posibles estados cuánticos de polarización de un fotón, será mediante la notación de Dirac, para estados cuánticos.

Consideremos ahora un flujo de fotones de luz polarizada verticalmente, los cuales atraviesan un filtro polarizador paralelo respecto a la dirección de movimiento del flujo de fotones y luego un filtro perpendicular.

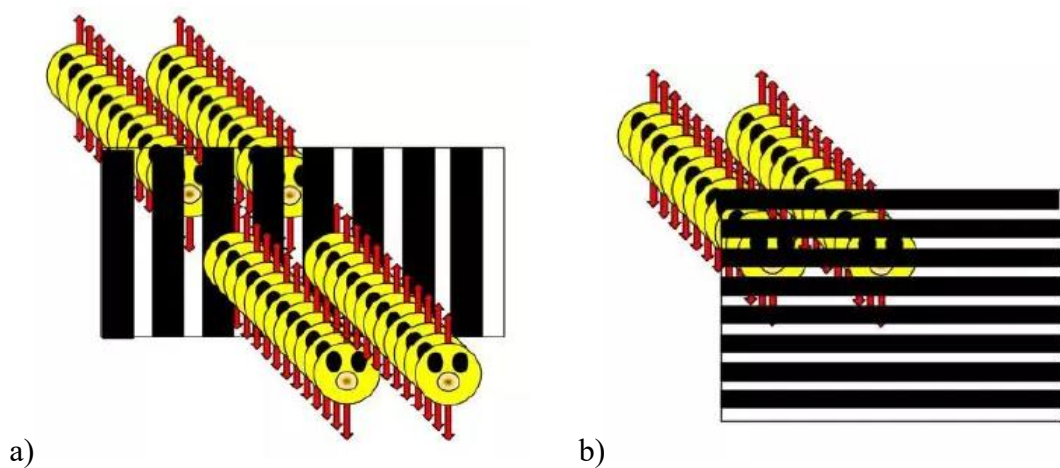


Figura 2.11. Fotones atravesando un filtro paralelo a su polarización y otro perpendicular.

En la **Figura 2.11**, en a) los fotones logran atravesar una pantalla polarizadora paralela y en b) no logran pasar una pantalla polarizadora perpendicular [19]. Para el filtro polarizador alineado de manera paralela a la polarización de los fotones como en la figura a), vemos que logran pasar el 100% de ellos, pero si se gira el filtro 90° como en la figura b) entonces se bloquean todos los fotones. En el caso de que se envíe un flujo de fotones polarizado a 0° horizontal y el filtro sea vertical a 90° entonces tampoco pasaran fotones.

En general siempre que el polarizador esté colocado con dirección perpendicular (formando un ángulo de 90°) a la dirección de la polarización que tienen los fotones, la luz no pasará. Si en cambio el polarizador se sitúa paralelamente a la dirección de polarización de la luz entonces pasará toda la luz. Un caso especial ocurre cuando el filtro polarizador está a 45° de diferencia respecto a la luz entrante.

En la **Figura 2.12** se ven los distintos casos de bloqueo y paso de luz, en el caso de un ángulo α , mientras más cerca esté ese ángulo de formar un ángulo perpendicular con la luz entrante, menos luz podrá pasar el filtro y mientras más se acerque el ángulo α a formar una paralela con la luz entrante, más luz pasará el filtro, de manera que la luz se volverá más o menos intensa dependiendo del ángulo α .

Cuando el ángulo α entre la luz entrante y el polarizador es exactamente 45° , solo pasará el 50% de la luz. Si la dirección del polarizador forma un ángulo cualquiera, con la polarización de la luz entrante, entonces pasa una fracción igual al cuadrado del coseno de dicho ángulo:

$$\cos(\alpha)^2 = \frac{1}{2}$$

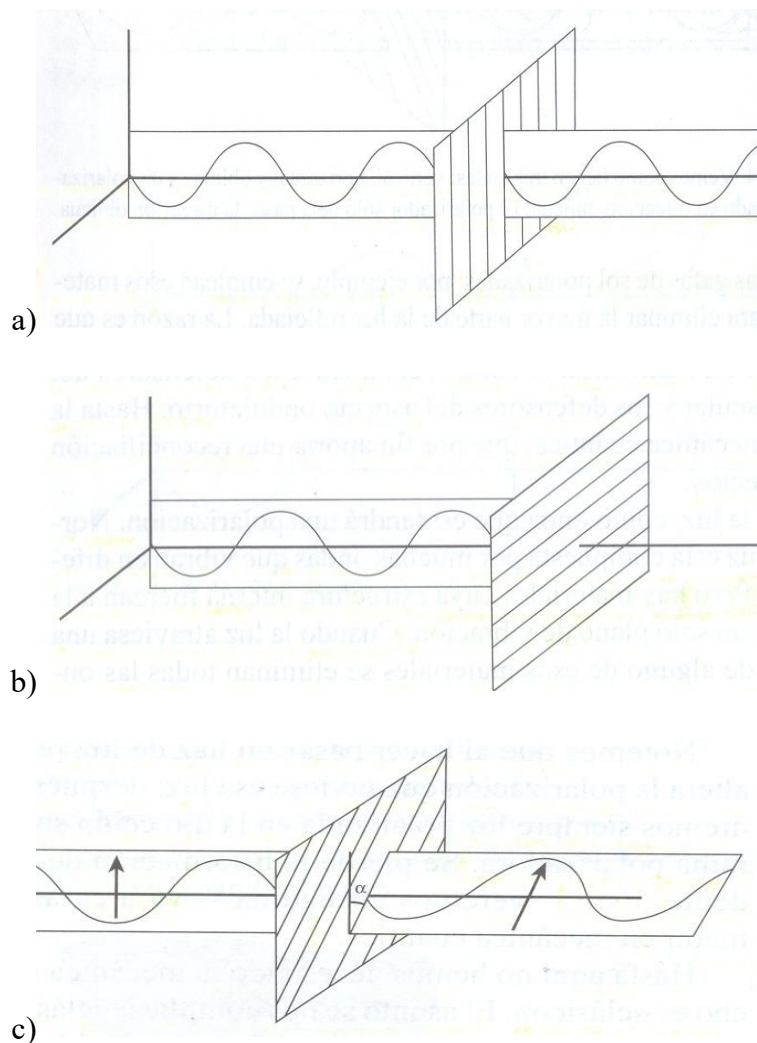


Figura 2.12. Una onda electromagnética polarizada ante distintos filtros

En la **Figura 2.12**, los fotones que están polarizados paralelamente al filtro logran pasar en un 100%, los fotones que están polarizados perpendicularmente al filtro no logran pasar y aquellos fotones que estén polarizados en un ángulo α respecto al filtro podrán pasar dependiendo que tan cercano este el ángulo a la paralela, mientras que en el caso de $\alpha = 45^\circ$ pasará exactamente el 50% de los fotones.

2.1.9. Polarización y superposición de estados

Así como los fotones pueden estar alineados a 90° o 180° vertical y horizontalmente, también pueden estar oscilando linealmente en cualquier ángulo. Si colocamos un filtro vertical u horizontal frente a esta luz no polarizada, el resultado es que solo pasara la luz que por naturaleza ya estaba oscilando en la dirección de ese filtro, ya sea horizontal o vertical.

Ahora, si se pone un segundo filtro perpendicular al primero, por ejemplo digamos que el primer filtro estaba a 90° vertical y el segundo estará a 180° horizontal, lo que ocurre es que toda la luz se bloquea como muestra la **Figura 2.13**. En a) tenemos la luz natural sin polarización, en b) ponemos un filtro a 90° , lo que ocasiona que solo pase la luz que originalmente estaba a polarizada en 90° , en c) se coloca un filtro perpendicular al primero y se bloquea toda la luz.

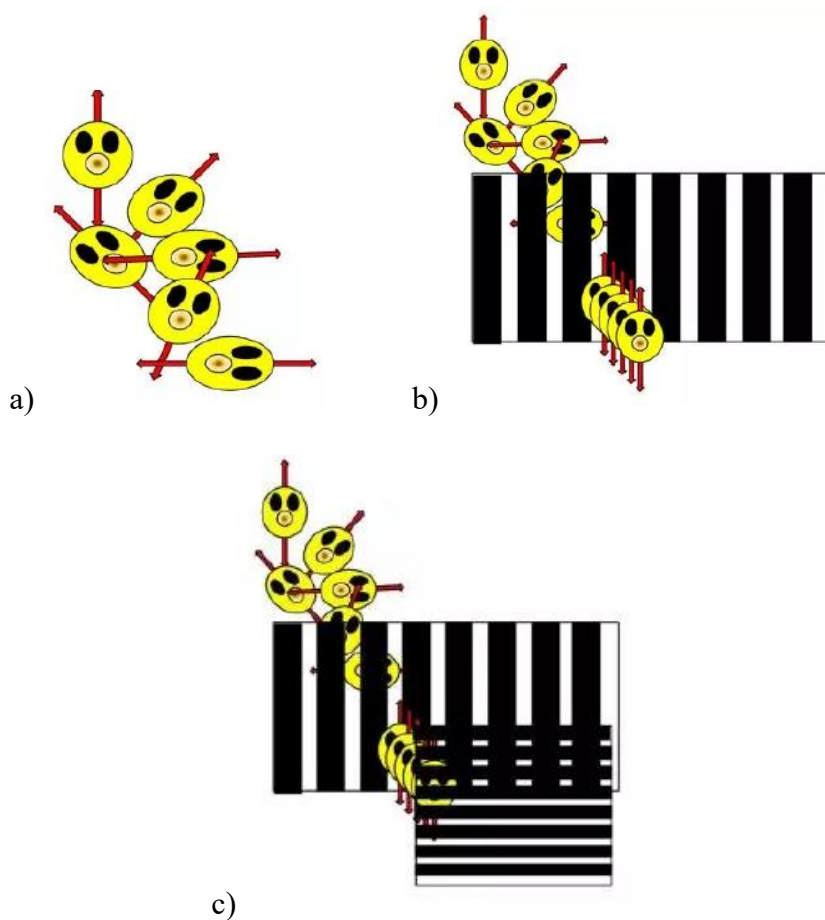


Figura 2.13. Luz no polarizada, luego polarizada y un filtro.

Supongamos que en este experimento colocamos un tercer filtro alineado a 45° respecto a las direcciones de los otros dos y se coloca entre medio de los dos anteriores. ¿Cuál será el resultado?

- El estado inicial del sistema es “luz no polarizada”
- Se coloca un filtro vertical y cada fotón cuya oscilación original era vertical lograra pasar el filtro
- Los fotones tendrán un estado $|\text{Fotón vertical}\rangle$
- Ahora se coloca un filtro en 45° respecto al vertical

Lo que ocurre es que solo algunos fotones logran pasar. El experimento pregunta a los fotones cuáles de ellos están polarizados a 45° respecto a la dirección de salida del anterior polarizador, los cuales vienen con un estado $|Fotón\ vertical>$. El secreto radica en que el estado vertical puede escribirse como una **combinación de otros estados**.

$$|foton\ vertical\ > = \frac{1}{2}|foton\ a\ 45^{\circ}\ respecto\ a\ la\ vertical\ > + \frac{1}{2}|foton\ a\ -45^{\circ}\ respecto\ a\ la\ vertical\ >$$

Lo que significa que un fotón polarizado verticalmente tiene un 50% de probabilidad de pasar un filtro a 45° respecto a la vertical. El estado de los fotones de salida será $|Foton\ a\ 45^{\circ}\ respecto\ a\ la\ vertical\ >$, también disminuye la intensidad de la luz en la mitad de la original. Quedando un efecto como en la **Figura 2.14**.

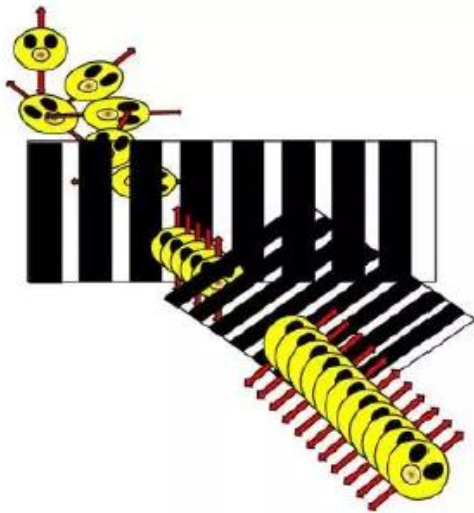


Figura 2.14. Los fotones que logran pasar el filtro a 90° tienen un 50% de probabilidad de pasar el filtro a 45°.

Ahora se coloca el tercer filtro perpendicular al primero, ósea a 180°. Le preguntamos a los fotones que vienen con estado $|Foton\ a\ 45^{\circ}\ respecto\ a\ la\ vertical\ >$ cuántos de ellos están polarizados horizontalmente.

$$|foton\ a\ 45^{\circ}\ > = \frac{1}{2}|foton\ vertical\ > + \frac{1}{2}|foton\ horizontal\ >$$

Por lo que los fotones a 45° tienen un 50% de probabilidades de pasar por el polarizador horizontal. La **Figura 2.15** muestra este efecto. Donde los fotones logran

pasar un polarizador perpendicular a la polarización original , gracias a que se colocó un filtro a 45° entre medio.

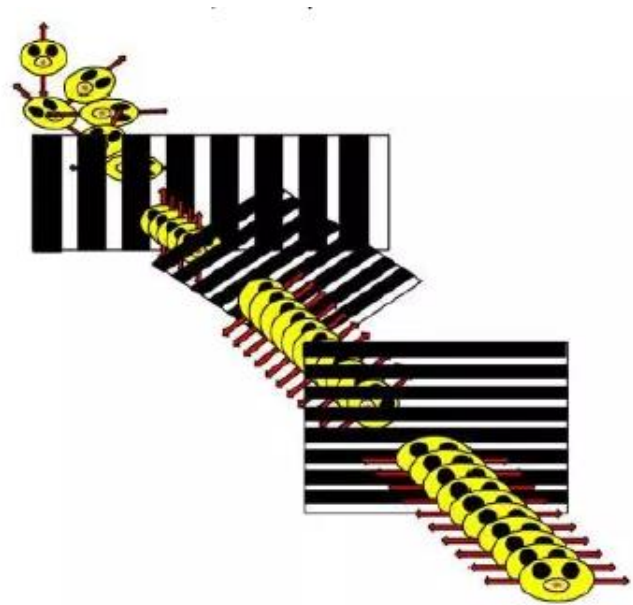


Figura 2.15. Fotones logran pasar un polarizador perpendicular.

Y es por esto que logra salir luz al poner un polarizador de 45° entre medio de un polarizador vertical y uno horizontal. De hecho, en condiciones ideales la luz que sale tiene una intensidad de salida que es un 25% de la intensidad que sale por el primer polarizador.

Conclusiones

La física cuántica describe los estados de un experimento como combinaciones lineales de otros estados con cierta probabilidad de ocurrencia. Esto es debido a la definición matemática de un estado en mecánica cuántica. Los estados cuánticos son funciones de onda que son solución a la ecuación de onda de Schrödinger, al ser una ecuación diferencial, las soluciones son una sumatoria de funciones que satisfacen la ecuación. En un experimento cuántico antes de la medición existen todas estas soluciones coexistiendo de manera conjunta, pero al realizar la medición o el experimento la función de onda colapsa en solo una de estas componentes en este caso una de las posibles polarizaciones.³

³ Apéndice: Ecuación de Schrödinger

$$\begin{aligned}
|foton\ vertical > &= \frac{1}{2} |foton\ a\ 45^\circ\ de\ la\ vertical > \\
&+ \frac{1}{2} |foton\ a\ -45^\circ\ de\ la\ vertical > \\
|foton\ a\ 45^\circ > &= \frac{1}{2} |foton\ vertical > + \frac{1}{2} |foton\ horizontal >
\end{aligned}$$

En esta combinación los argumentos indican con que probabilidad se puede ver uno de los estados que forman la combinación, por ejemplo: si el filtro en vez de estar a 45° hubiera estado a 80° hay más probabilidad de que la componente vertical se manifestase. Cuando se realiza la medición, es decir, cuando se coloca el filtro polarizador en frente de la luz, es entonces cuando solo uno de los componentes sobrevive.

1. Sobrevive el fotón orientado con el primer filtro $|Fotón\ vertical >$
2. Sobrevive el fotón orientado al filtro de 45°
 $|Fotón\ a\ 45^\circ\ respecto\ a\ la\ vertical >$
3. Sobrevive el fotón alineado al tercer filtro de 180° $|Fotón\ horizontal >$

La combinación de todos los posibles estados **colapsa** en uno de sus constituyentes.

Las probabilidades perpendiculares son excluyentes. Si tengo luz polarizada en la vertical, esta no pasara por un polarizador horizontal y viceversa.

El experimento funciona para cualquier par de direcciones que sean perpendiculares entre si y siempre que el polarizador que se introduce entre medio sea 45° respecto a ambos.

Este experimento es análogo al experimento mental propuesto por Schrödinger conocido como el gato de Schrödinger. En el cual un gato está encerrado en una caja sometido a un dispositivo de veneno cuya activación está determinada por la descomposición radiactiva de un átomo (proceso ciertamente cuántico). El estado del gato es una superposición de los estados $|vivo >$ y $|muerto >$.

$$|gato > = \frac{1}{\sqrt{2}} |gato\ vivo > + \frac{1}{\sqrt{2}} |gato\ muerto >$$

En general para una función de onda $\varphi()$ su estado puede ser una combinación de estados:

$$|\varphi\rangle = \alpha|\varphi_1\rangle + \beta|\varphi_2\rangle$$

Donde α y β generalmente son números complejos y los argumentos son $\frac{1}{\sqrt{2}}$ para que la norma sea igual a 1, ósea ambos argumentos suman 1 para dar el 100% de certeza.

$$|\alpha|^2 + |\beta|^2 = 1$$

Si definimos que la función 1 representara un uno y la función 2 representara un cero, $\varphi_1 = 1$ y $\varphi_2 = 0$, entonces tenemos la definición de un bit cuántico.

$$|\varphi\rangle = \alpha|1\rangle + \beta|0\rangle$$

Ecuación fundamental de estado

La mecánica cuántica debe respetar el principio de linealidad en el cual si se tienen un par de estados aceptables para un sistema $|\varphi_1\rangle = \text{vivo}$ $|\varphi_2\rangle = \text{muerto}$. Cualquier combinación lineal de los mismos es una configuración aceptable para el sistema. Además ambos estados constituyentes deben ser excluyentes.

La probabilidad de obtener una de las componentes como resultado, viene dada por el cuadrado del coeficiente que acompaña a cada componente.

$$|\text{gato}\rangle = \sqrt{0,5} |\text{gato vivo}\rangle + \sqrt{0,5} |\text{gato muerto}\rangle$$

$$|\text{gato}\rangle = \sqrt{0,7} |\text{gato vivo}\rangle + \sqrt{0,3} |\text{gato muerto}\rangle$$

$$|\text{gato}\rangle = \sqrt{0,15} |\text{gato vivo}\rangle + \sqrt{0,85} |\text{gato muerto}\rangle$$

El colapso de la función de onda es un proceso completamente aleatorio. Es decir, en el proceso de medida cualquier estado superpuesto, pierde todas las componentes de la superposición excepto aquella que ha sido seleccionada por la medida experimental. Este proceso es aleatorio y se rige por las probabilidades de cada componente, la teoría no predice cuál de los estados superpuestos será elegido, solo

una probabilidad de que alguno de ellos sea elegido. No hay teoría que describa este colapso dentro de un ámbito determinista y unitario.

Algunas explicaciones a este colapso son:

1. El ambiente puede introducir interacciones con los sistemas que son aleatorias en su conjunto. Cualquier fotón, molécula de aire o cambio en la temperatura podría ocasionar el colapso. Esto se conoce como la decoherencia de los estados cuánticos.
2. El colapso no ocurre, lo que ocurre es que el universo se desdobla en cada medición en tantas alternativas como estados superpuestos se tengan de inicio. Esta es la interpretación de los mundos múltiples de la mecánica cuántica.

2.1.10. Superposición de estados y codificación de bits

Matemáticamente, el principio de superposición permite descomponer un problema lineal en dos o más subproblemas más sencillos, de manera que el problema original se obtiene como suma de los subsistemas sencillos.

Cualquier solución a la ecuación de Schrödinger se considera una función vectorial compleja $\varphi(x, t)$ la cual se usa para representar estados de la forma $|\varphi\rangle$. Esta función de onda al ser un vector cumple con las propiedades de combinación lineal de vectores.

Cualquier combinación lineal de soluciones a la ecuación de Schrödinger es también una solución válida para la ecuación.

$$\varphi = \alpha_1\varphi_1 + \alpha_2\varphi_2 + \alpha_3\varphi_3 + \alpha_4\varphi_4 + \dots$$

$$|\varphi\rangle = \sum_i \alpha_i |\varphi_i\rangle$$

Para experimentos que miden el comportamiento de dos observables excluyentes, como el espín o la polarización, se asigna a cada uno de los dos observables los dos estados $|0\rangle$ y $|1\rangle$ quedando un estado $|\varphi\rangle$ que será combinación lineal de los dos subestados:

$$|\varphi\rangle = \alpha|1\rangle + \beta|0\rangle$$

α y β pueden ser generalmente complejos y representan la proporción que cada uno de los subestados aportan al estado total. Se debe cumplir:

$$|\alpha|^2 + |\beta|^2 = 1$$

condicion de normalizacion

Debido a que los índices representan las probabilidades de ocurrencia de cada subestados, la suma total debe dar 1 para obtener el 100% de certeza. Cuando se hace una medida sobre el sistema en el estado $|\varphi\rangle$ el resultado solo decantara en uno de los dos subestados posibles, ya sea $|0\rangle$ o $|1\rangle$, jamás en ambos a la vez. Se puede obtener $|0\rangle$ con una probabilidad de $|\alpha|^2$ y el estado $|1\rangle$ con una probabilidad de $|\beta|^2$.

Dependiendo el valor de α y β se puede estimar qué subestado sería elegido para el colapso de la función. Por ejemplo si $\alpha = 0,1$ y $\beta = 0,9$, todo indica que β sería elegido. Sin embargo siguen siendo probabilidades por lo que 0,1 aún puede ser elegido.

2.1.11. Fundamento matemático del entrelazamiento y la superposición de estados

Se suponen dos sistemas entrelazados que pueden estar en un estado superpuesto entre $|0\rangle$ y $|1\rangle$.

$$|\varphi_1\rangle = |0_1\rangle + |1_1\rangle$$

$$|\varphi_2\rangle = |0_2\rangle + |1_2\rangle$$

Para entrelazar los dos sistemas se calcula el producto entre ellos:

$$|\varphi_1\rangle |\varphi_2\rangle = |\varphi_1\varphi_2\rangle$$

$$|\varphi_1\varphi_2\rangle = (|0_1\rangle + |1_1\rangle)(|0_2\rangle + |1_2\rangle)$$

$$|\varphi_1\varphi_2\rangle = |0_10_2\rangle + |1_11_2\rangle + |0_11_2\rangle + |1_10_2\rangle$$

Si se tiene un estado $|\varphi_3\rangle = |0_11_2\rangle + |1_10_2\rangle$, este no puede reescribirse como el producto de un subsistema 1 y un subsistema 2.

$$|\varphi_3\rangle \neq |\varphi_1\varphi_2\rangle$$

$$|\varphi_1\varphi_2\rangle \neq |0_11_2\rangle + |1_10_2\rangle$$

Si se tiene el estado $|\varphi_1\varphi_2\rangle = |0_10_2\rangle + |1_11_2\rangle + |0_11_2\rangle + |1_10_2\rangle$ y se hace una medida sobre el subsistema 1 (sobre los elementos con subíndice 0_1 y 1_1) y el resultado es $|0_1\rangle$, entonces el subsistema 2 (los elementos son subíndice 0_2 y 1_2) podría estar en los estados $|0_2\rangle$ y $|1_2\rangle$.

$$|\varphi_1\varphi_2\rangle = |0_10_2\rangle + |1_11_2\rangle + |0_11_2\rangle + |1_10_2\rangle$$

Resultados del subsistema 1

Únicos valores posibles para el subsistema 2

En cambio si se tiene un estado $|\varphi_3\rangle = |0_11_2\rangle + |1_10_2\rangle$ y se mide el subsistema 1 (subíndices 1) y se obtiene $|0_1\rangle$, entonces el subsistema 2 solo puede estar en el estado $|1_2\rangle$ y viceversa, si el subsistema 1 está en $|1_1\rangle$ el subsistema 2 solo puede estar en $|0_2\rangle$.

$$|\varphi_3\rangle = |0_11_2\rangle + |1_10_2\rangle$$

Resultado obtenido para el subsistema 1

Únicos valores posibles para el subsistema 2

Conclusión

Un estado se dice entrelazado si no se puede escribir como producto de los estados de los subsistemas que lo conforman. [20]

2.1.12. Experimento de teletransportación de Alice y Bob

Volviendo a la comunicación entre Alice y Bob, supongamos que un subsistema 1 Alice y un subsistema 2 Bob, son creados al mismo tiempo en un laboratorio fruto del mismo experimento. Ambos entrelazan un par de partículas, esto quiere decir que el estado total de las dos partículas no se puede descomponer como el producto del

estado individual de una de ella por el estado individual de la otra. Alice y Bob son separados muy lejos, millones de años luz de distancia.

Existe un mecanismo para teletransportar información entre Alice y Bob, Alice tiene un sistema S el cual quiere enviar a Bob. El mecanismo sería:

1. Alice tiene un sistema S en un estado cuántico concreto y también tiene el subsistema que esta entrelazado con Bob.
2. Bob solo tiene su subsistema entrelazado con Alice a millones de años luz de distancia.
3. Alice hace una medida sobre su sistema S y el subsistema entrelazado. Al realizar esta medida se produce un cambio en el estado del sistema S y en el estado entrelazado con Bob.
4. Alice mira los resultados y deduce como ha cambiado la parte del subsistema entrelazado que tiene Bob, por ejemplo, si Alice mide el espín de la partícula en \uparrow entonces sabe con certeza que el espín de la partícula de Bob es \downarrow .
5. Alice **llama por teléfono** a Bob y le cuenta el resultado obtenido.
6. Una vez recibida la llamada Bob sabe en qué estado esta su partícula entrelazada con Alice sin siquiera tocarlo.
7. Con esa información recibida por teléfono Bob deduce las acciones que tiene que hacer para poner su sistema entrelazado en el estado S.

De esta forma se puede teletransportar el estado cuántico S de un sistema desde el laboratorio de Alice al de Bob.

¿Se transmite información instantáneamente desde Alice a Bob?

¿Se ha copiado el estado de Alice en el laboratorio de Bob?

Para la primera pregunta la respuesta es NO, el estado solo se recupera cuando Alice hace la llamada telefónica a Bob informándole su estado. Es decir, tiene que haber un medio clásico de comunicación entre Alice y Bob para que el teletransporte ocurra. Esta es la razón por la que Einstein estaba en contra del entrelazamiento y la “acción fantasmal a distancia”, él decía que la información no puede viajar más rápido que la luz, no puede violar la teoría de la relatividad.

Para la segunda pregunta la respuesta es NO, Alice al hacer la primera medida destruye su estado. El estado S ya no está en su estado original, ha sido perturbado por la medición de Alice, por lo tanto Bob reconstruye el estado de S en otra parte, pero el original ya no existe. No se viola el teorema de no clonación.

2.1.13. Teorema de no clonación

El teorema de no clonación fue publicado en la revista Nature en 1982 por William Wootters y Wojciech Zurek [21]. El teorema de no clonación declara que es imposible crear una copia idéntica de un estado cuántico desconocido arbitrario. El teorema impide crear copias de un estado cuántico dado, a menos que el estado ya sea conocido, es decir, que exista información clásica que lo describa. Sin embargo la física cuántica no prohíbe el teletransporte de un estado como se demostró en el tema anterior. El caso del teletransporte no es una clonación, ya que se pierde el estado original y por ello se denomina precisamente teletransporte. El poder copiar el estado de un sistema violaría el Principio de incertidumbre de Heisenberg, ya que al copiar un estado se podría medir la posición en el original y medir el momentum en la copia, permitiendo conocer ambos valores con 100% de certeza. Al tener varias copias de un estado se podría medir una propiedad distinta en cada una y así obtener con precisión todos los valores del sistema incluyendo aquellos conjugados.

Desarrollo matemático

Se tiene una máquina cuántica con dos ranuras etiquetadas A y B. La ranura A, ranura de datos, inicia en un estado cuántico desconocido pero puro $|\varphi_A\rangle$. Este es el estado que se copia en la ranura B, o ranura objetivo. Se asume que la ranura objetivo inicia en algún estado puro $|\gamma_B\rangle$. El conjunto de los sistemas A y B está descrito mediante un ket que es el producto tensorial de los dos kets anteriores. Entonces el estado inicial de la máquina copiadora es $|\varphi_A\rangle \otimes |\gamma_B\rangle$.

Si suponemos que podemos copiar el estado del sistema A, sea cual sea este estado, al sistema B, quiere decir que habrá una transformación, de modo que el estado final sea $|\varphi_A\rangle \otimes |\gamma_B\rangle$. Vamos a denominar por U al operador evolución correspondiente a dicha transformación, de modo que:

$$|\varphi_A\rangle \otimes |\gamma_B\rangle \xrightarrow{U} |\varphi_A\rangle \otimes |\varphi_B\rangle$$

Alguna evolución unitaria U efectúa ahora el procedimiento de copia de manera ideal.

Si se supone que este procedimiento de copia funciona para un nuevo estado $|\mu\rangle$, entonces se tiene:

$$U(|\varphi\rangle \otimes |\gamma\rangle) = |\varphi\rangle \otimes |\varphi\rangle$$

$$U(|\mu\rangle \otimes |\gamma\rangle) = |\mu\rangle \otimes |\mu\rangle$$

Tomando el producto interno entre estas dos ecuaciones, se tiene:

$$\langle\varphi|\mu\rangle = (\langle\varphi|\mu\rangle)^2$$

La únicas soluciones posibles para una ecuación del tipo $x = x^2$, son $x = 0$ y $x = 1$, entonces o $|\varphi\rangle = |\mu\rangle$ o $|\varphi\rangle$ y $|\mu\rangle$ son ortogonales, esto significa que su producto interno es igual a cero $\langle\varphi|\mu\rangle = 0$. Es decir que el dispositivo de clonación solo puede clonar estados ortogonales conocidos con resultado fiable, y este producto punto es imposible que de cero para dos estados arbitrarios. Por tanto la clonación de estados arbitrarios es imposible.

Para utilizar el teorema de no clonación como mecanismo de seguridad en protocolos de criptografía, hay que asegurarse que los estados usados sean **no ortogonales** así no será posible copiarlos.

Este teorema es una importante motivación para el desarrollo de protocolos y claves de seguridad. La restricción que supone no poder copiar estados cuánticos puede emplearse como una clara apuesta en el ámbito de la criptografía. En 1984 se propuso un esquema de distribución de clave cuántica por Bennett y Brassard [6]. La idea es que el emisor, Alice, mande fotones al receptor, Bob, con el objetivo de crear una secuencia aleatoria de 0's y 1's. Dicha cadena o secuencia puede emplearse como una clave secreta para encriptar y desencriptar mensajes. En el esquema propuesto, cada uno de los fotones de Alice está preparado aleatoriamente en uno de cuatro posibles estados de polarización rectilínea o diagonal: $\{0^\circ, 90^\circ, 45^\circ, 135^\circ\}$

Una espía, Eve, intentaría obtener una copia de cada fotón para ella misma, dejando a su vez pasar una copia exacta a Bob para pasar completamente desapercibida, ya que Bob y Alice podrían comprobar mediante secuencias si Eve ha alterado la señal enviada inicialmente. Por el teorema del no clonado, Eve no puede hacerlo de forma

exitosa. Sin embargo, sí puede hacer una buena aproximación al clonar la transmisión de Alice. Si el proceso de aproximación fuese óptimo, podría usarse de forma eficiente contra la criptografía cuántica. Afortunadamente, pueden ponerse límites teóricos en la fidelidad del esquema.

2.1.14. El Qubit

El bit se considera una unidad de información binaria discreta, que puede tomar solo los valores $\{0,1\}$ y es la base de la comunicación por medios físicos. Al ser un sistema binario son aplicables todas las matemáticas del álgebra de Boole, haciéndolo el sistema de codificación más importante de la humanidad.

Cualquier experimento o sistema cuyos observables puedan ser medidos o configurados en dos estados, es un buen sistema para codificar bits.

Un proceso de comunicación es útil solo si el sistema de comunicación teórico tiene un paralelismo con los sistemas físicos reales, por los que se realizara la comunicación eficazmente.

Uno de los problemas que deben afrontar los sistemas de implementación físicos es la **decoherencia** de la información, la cual es producto del ruido y las interacciones del sistema cuántico con su entorno, ya que ningún sistema puede estar completamente aislado.

La decoherencia hace que las combinaciones lineales de los objetos cuánticos microscópicos decaigan muy rápidamente y el sistema se vuelva clásico.

Representación de Qubits usando el espín

Un Qubit o quantum bit es un bit implementado mediante algún fenómeno o experimento cuántico, que pueda ser observado y medido. El espín del electrón es un ejemplo de un sistema cuántico, como la polarización de fotones y la descomposición de un átomo radiactivo. Recordemos que el espín es una propiedad de las partículas

subatómicas que posee el electrón como el protón, y representa el momentum angular intrínseco de una partícula. Su valor es $\pm \frac{1}{2}$ para el electrón y ± 1 para el protón. Esto significa que en presencia de campos magnéticos el espín puede alinearse solo de dos formas $+\frac{1}{2} \uparrow$ o $-\frac{1}{2} \downarrow$, espín up o espín down.

Denotemos por $|0\rangle$ al estado de espín up $+\frac{1}{2} \uparrow$ que se obtiene cuando el electrón se alinea paralelamente al campo magnético y $|1\rangle$ al estado de espín down $-\frac{1}{2} \downarrow$ cuando el electrón se alinea antiparalelamente al campo (esta asignación es arbitraria y podría ser al revés). La notación de estados utilizada es la notación braket de Dirac.

Al aplicar un campo magnético a las partículas cuánticas (como un haz de átomos de plata en el experimento Stern y Garlach) lo que hacemos es *pedirle* al haz de átomos que nos muestre una de las dos posibles alineaciones de su espín, es decir *medir* el estado del espín.

Antes de la medición el espín puede estar no solo en un valor u otro sino que en una combinación lineal de ambos, tal como se estudió en el apartado de Superposición de estados y codificación de bits.

El acto de medición del qubit superpuesto antes de ser medido es:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Al hacer una medición sobre el Qubit, se tiene una probabilidad del 50% de obtener el estado $|0\rangle$ o $|1\rangle$.

Sea α y $\beta = \frac{1}{\sqrt{2}}$ tal que:

$$|\varphi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

El estado $|\varphi\rangle$ es una función de onda cuántica de carácter vectorial, por lo tanto los subestados $|0\rangle$ y $|1\rangle$ también se consideran vectores, solo que en este caso son las bases del espacio vectorial, con las cuales se pueden generar todos los estados posibles mediante combinación lineal de ellos⁴.

⁴ Apéndice : Espacios vectoriales complejos Espacio de Hilbert

Como los vectores $|\varphi\rangle$ cumplen que $|\alpha|^2 + |\beta|^2 = 1$, se dice que tienen **norma**.

Geoméricamente pueden representarse estos vectores como puntos sobre la superficie de una esfera de radio 1, conocida como **esfera de Bloch**. La esfera de Bloch representa la interacción de una partícula de espín $\frac{1}{2}$ que decanta en $+$ o $-$ debido al efecto de un campo magnético en la región norte en e_3 . [22]

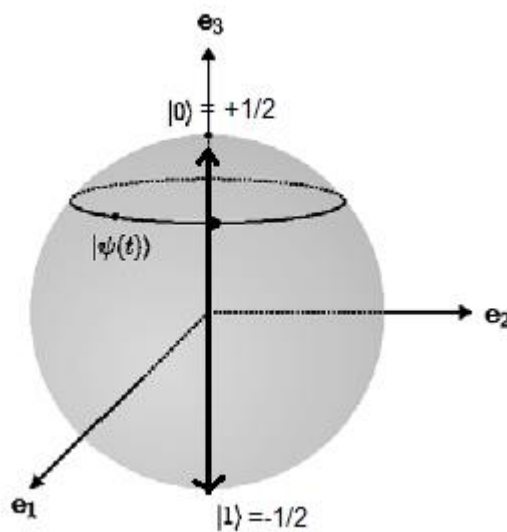


Figura 2.16. Esfera de Bloch

El vector $|0\rangle$ corresponde al polo norte de la esfera, mientras que $|1\rangle$ es el polo sur, también se pueden representar mediante flechas $\uparrow\downarrow$.

Como se muestra en la **Figura 2.16**, para un Qubit arbitrario que no esté perfectamente alineado a $|0\rangle$ o $|1\rangle$ en presencia de un campo magnético \vec{B} en la dirección del eje \vec{Z} por ejemplo, obliga al vector del Qubit arbitrario $|\varphi(t)\rangle$ a moverse alrededor de e_3 . El vector $|\varphi(t)\rangle$ se encuentra en un cono circundante al eje Z , cerca del campo magnético.

Se pueden obtener las componentes de cualquier vector arbitrario que no esté alineado perfectamente con la vertical, mediante una parametrización del vector $|\varphi(t)\rangle$ en términos de los ángulos θ y β de las coordenadas esféricas para $r = 1$.

$$|\varphi(t)\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\beta} \sin \frac{\theta}{2} |1\rangle \right)$$

Con γ número real arbitrario introducido por formalidad que en la práctica se puede ignorar. Cualquier punto sobre la esfera de Bloch podrá entonces representarse como una combinación del polo norte y el polo sur en términos de lo que valgan los parámetros φ y β .

La habilidad de los Qubits de estar simultáneamente en los estados clásicos 1 y 0 antes de medirlos, es útil para codificar mensajes que de ser interceptados u *observados*, dejan de reflejar la información original (porque han colapsado).

Cuando se trabaja con espines $\frac{1}{2}$, lo correcto es usar campos magnéticos para realizar operaciones lógicas, ya que esta clase de espín es en cierto modo como un imán diminuto (ante la presencia de un campo magnético se comporta como una brújula).

Como muestra la **Figura 2.16**, se tiene un campo magnético $\vec{B} = B_z e_z$ apuntando en la dirección del eje Z, $e_z = e_3$ es un vector de longitud 1 sobre la vertical apuntando hacia arriba.

Al colocar el Qubit $|\varphi(t)\rangle$ en la región del campo magnético, si el vector $|\varphi\rangle$ apunta originalmente en la misma dirección que el campo magnético \vec{B} es decir $|\varphi\rangle = |0\rangle$, entonces se mantendrá así y ese será su estado con espín = $\frac{1}{2}$.

Si por el contrario apunta a cualquier otra dirección (supongamos en algún punto de la circunferencia superior de la esfera), entonces empezara a danzar alrededor del vector e_3 describiendo un círculo con centro como la componente Z y paralelo a los planos XY.

Representación de Qubits usando fotones polarizados

Se pueden utilizar fotones para describir estados cuánticos como Qubits, por ejemplo, los vectores $|0\rangle$ y $|1\rangle$ podrían estar representados por alguna de las dos posibles polarizaciones lineales rectilíneas (vertical u horizontal). Las compuertas lógicas correspondientes estarían representadas por polarizadores ópticos que reorientan los fotones a distintos ángulos entre $|0\rangle$ y $|1\rangle$.

Si se mide la polarización de un fotón que pasa por un filtro polarizador orientado verticalmente, el fotón aparece del otro lado como verticalmente polarizado, a pesar de su dirección inicial de polarización.

Si se pone un segundo filtro orientado un cierto ángulo θ respecto al filtro vertical, hay cierta probabilidad de que el fotón pase por el segundo filtro, esta probabilidad depende del ángulo θ .

Si θ se acerca a la vertical, ósea su ángulo aumenta hasta 90° , entonces aumenta la probabilidad de que el fotón pase este segundo filtro (ya que el primer filtro es vertical y esta polarizado a 90°), cuando θ sea exactamente 90° , la probabilidad de pasar será del 100%.

Si θ baja acercándose a 0° o aumente más allá de los 135° acercándose a 180° donde estará prácticamente horizontal, entonces disminuye la probabilidad de que el fotón pase por el segundo filtro. La probabilidad será de 0% cuando θ sea igual a 0° y 180° .

Si θ es igual a 45° la probabilidad de que el fotón pase por el segundo filtro es del 50%.

Este mecanismo de filtro polarizador lo que hace es colapsar el estado superpuesto del fotón que antes de ser medido está en un estado $|\varphi\rangle = |0\rangle + |1\rangle$, permitiéndole pasar o no el filtro, con un estado $|0\rangle$ o $|1\rangle$ respectivamente.

Estos eventos son determinísticos (medir con un filtro polarizador) y pueden ser medidos en laboratorio, por lo tanto se puede utilizar una polarización para representar el bit 1 clásico y otra para representar el bit 0 clásico.

$$|0\rangle = 0 \quad |1\rangle = 1$$

La polarización de un fotón hace referencia al plano en el que oscila el campo eléctrico de la onda, mientras el fotón se propaga.

La correlación para medir Qubits usando fotones polarizados es la siguiente:





Bases de polarización / Bits	<i>Estado</i> $ \varphi\rangle = 0\rangle = 0$	<i>Estado</i> $ \varphi\rangle = 1\rangle = 1$
Polarización rectilínea Fotón polarizado en 0° o 90°		
Polarización diagonal Fotón polarizado en 45° o 135°		

Tabla 2.1. Estados de polarización para representar Qubits.

Una cadena de bits clásicos 1 1 1 0 1 1 0 se representaría de la siguiente forma:








Bit	1	1	1	0	1	1	0
Filtro	D	R	D	R	D	D	R
Fotón polarizado							

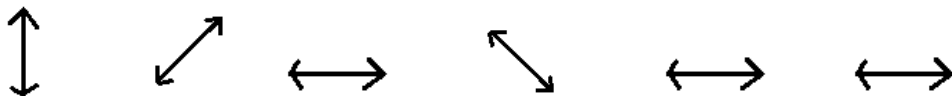
Tabla 2.2. Fotones polarizados que viajaron por un canal cuántico.

2.1.15. Ejemplo de comunicación entre Alice y Bob

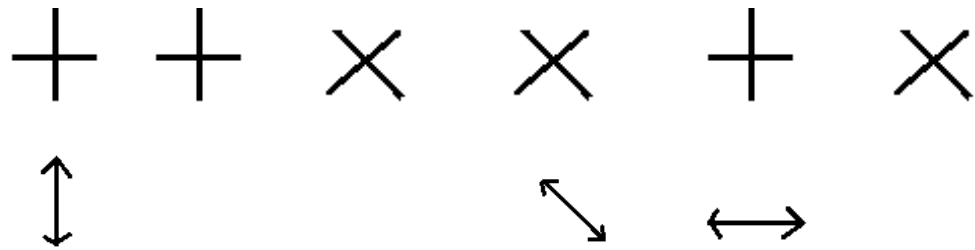
Alice es la emisora de un mensaje en bits que quiere enviar a Bob mediante fotones polarizados. Para ello polariza los fotones eligiendo de entre cuatro direcciones posibles: 0°, 90°, 45°, 135°.

Bob recibe los fotones y puede medirlos usando un filtro polarizador que puede alinearse a dos posibilidades, rectilíneo o diagonal. Si Bob pone su filtro en rectilíneo, podrá medir los fotones que vengan con ángulo 0° y 90° pero si bien un fotón con ángulo 45° o 135° no podrá medirlo.

Alice envía una colección de fotones:



Bob realiza las mediciones utilizando los siguientes filtros: R-R-D-D-R-D



Solo logran pasar tres fotones, luego Alice le da la respuesta a Bob de que filtros utilizo y cuáles deberían ser las mediciones correctas, y estas se traducen a 0 y 1. Bob y Alice guardan los resultados en los que concuerdan. De esta manera Alice y Bob generan una secuencia binaria que solamente **ellos conocen** y pueden mantenerla cuanto tiempo quieran.

Conclusiones

Los bits clásicos pueden ser representados mediante experimentos cuánticos, en el caso de átomos de plata que pasan a través de un campo magnético, el observable es el espín del electrón. En el caso de un experimento de fotones polarizados, el observable es el ángulo de orientación al pasar por un filtro. En ambos casos se obtienen representaciones para los bits cero y uno.

El ejemplo presentado con Alice y Bob es el mecanismo de **distribución de claves cuánticas** y es utilizado en los protocolos BB84 y B92 con modificaciones. Estos protocolos serán abordados más adelante.

En el área de la computación donde la comunicación se vuelve cada vez más rápida, se vuelve imprescindible crear sistemas de criptografía que sean seguros. Un sistema criptográfico bien implementado se vuelve imposible de ser intervenido sin ser detectado.

2.2. CRIPTOGRAFÍA

La palabra criptografía viene del griego *kryptos*, que significa esconder y *gráphein*, que significa escribir, es decir, escritura escondida. La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo

las personas autorizadas puedan entender el mensaje. Sus inicios se remontan a las culturas antiguas. En la cultura egipcia los sacerdotes usaban jeroglíficos incomprensibles para el común de las personas. En la babilonia usaban la escritura cuneiforme y en la griega se inventaron artefactos en los que dependiendo el ancho de un bastón sobre el que se enrollaba un cinturón de cuero, podría leerse o no la información escondida.

2.2.1. Cifrado simétrico o de clave privada

El punto débil de la cifra de Vigenere⁵ reside en la repetición de la palabra clave. La única posibilidad realmente segura era que la clave sea de la misma longitud del mensaje y totalmente aleatoria.

La criptografía monoclave, es decir que usa la misma clave para cifrar y para descifrar el mensaje, también se suele llamar sistema de **clave privada** o **sistema de criptografía simétrica** y no garantiza autenticidad.

El método consiste en aplicar diferentes funciones al mensaje que se quiere cifrar, de tal modo que solo conociendo una clave esta pueda aplicarse de forma inversa para poder descifrar el mensaje

La criptografía simétrica puede dividirse en:

- Criptografía simétrica por bloques (*Block cipher*)
- Criptografía simétrica de lluvia (*Stream cipher*)
- Criptografía simétrica de resumen (*Hash cipher*)

2.2.2. DES

El algoritmo DES (*Data Encryption Standard*) es un algoritmo de cifrado simétrico desarrollado por la NASA a petición del gobierno de los Estados Unidos, para que las empresas puedan proteger sus comunicaciones. DES fue escogido como estándar

⁵ Apéndice: Cifrado de Vigenere

FIPS (*Federal Information Processing Standard*) en el año 1976. Hoy en día DES es considerado inseguro dada su clave de 56 bits, insuficiente frente al poder computacional actual. En su variante Triple DES, el algoritmo se cree seguro.

DES es un algoritmo de criptografía simétrica por bloques. Se toma un bloque de una longitud fija de bits y lo transforma mediante una serie de operaciones básicas en otro bloque cifrado de la misma longitud. En el caso de DES, el tamaño del bloque es de 64 bits. La clave también es de 64 bits, pero 8 de estos bits se emplean para paridad, haciendo que la longitud efectiva de la clave sea de 56 bits.

En la actualidad no se ha podido romper el sistema DES desde la perspectiva de poder deducir la clave simétrica a partir de la información interpretada como en el caso del criptoanálisis, sin embargo, con un método a fuerza bruta probando alrededor de 2^{56} posibles claves, se pudo romper DES en 1999. Para fortalecer el tamaño de la clave y evitar los ataques de fuerza bruta se implementó Triple DES o TDES.

2.2.3. AES

EL algoritmo AES (*Advanced Encryption Standard*) fue el ganador del concurso convocado en 1997 por el NIST (*National Institute Standards and Technologies*) con el objetivo de escoger un nuevo algoritmo de cifrado que reemplazara a Des. En 2001 fue tomado como estándar FIPS y en 2002 se transformó en el estándar oficial. Desde el año 2008 es el algoritmo más popular en criptografía simétrica.

AES opera sobre una matriz de 4x4 bytes, mediante un algoritmo se reordenan los distintos bytes de la matriz. El algoritmo funciona mediante una serie de bucles que se repiten en una red de sustitución-permutación, 10 ciclos para claves de 128 bits, 12 ciclos para claves de 192 bits y 14 ciclos para claves de 256 bits.

2.2.4. One Time Pad, Cifrado de Vernam

Gilbert Vernam publicó en 1927 [23] un sistema de cifrado conocido como *one-time-pad* o cuaderno de un solo uso, es un tipo de algoritmo en el que el texto se combina con una clave aleatoria de la misma longitud que el mensaje y que solo se utiliza una vez. Si la clave es verdaderamente aleatoria, nunca se reutiliza y por supuesto se mantiene en secreto, se puede demostrar que el método de la libreta de un solo uso es irrompible.

En 1949 Shannon [24] demostró que la cifra de Vernam era totalmente segura en condición de que la clave no se usase dos veces. De aquí el nombre de cuaderno de un solo uso, la clave aleatoria debe escribirse en un libro de claves y usar una hoja distinta para cada mensaje. Algunos inconvenientes:

- Requiere que las libretas sean perfectamente aleatorias
- La generación e intercambio de las libretas debe ser segura y la libreta debe ser del mismo tamaño que el mensaje
- Debe deshacerse correctamente la clave luego de ser usada

2.2.5. Cifrado asimétrico o de clave pública

La criptografía asimétrica también conocida como de clave pública, es un sistema en el que cada usuario posee dos claves. Una es de dominio público llamada clave pública y cualquiera puede tenerla, y la otra es privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas.

El remitente usa la clave pública del destinatario para cifrar un mensaje y enviarlo, este mensaje solo podrá ser descifrado con la clave privada del destinatario, así solo el destinatario puede acceder a la información. De la misma forma el propietario puede usar su clave privada para cifrar un mensaje y ser descifrado con la clave pública. Si todo el mundo tiene acceso a las claves públicas ¿Qué sentido tendría hacer esto? Usando tu clave privada estás demostrando tu identidad, ya que solo tú posees esa clave privada.

El cifrado asimétrico se usa principalmente en dos ámbitos:

- Intercambios de claves privadas en sistemas simétricos

- Firma digital

Una firma digital se puede definir como una cadena de caracteres que se agrega a un archivo digital, que cumple el mismo rol que la firma convencional que se escribe en un documento de papel.

El mayor uso de los sistemas asimétricos es para la distribución de claves simétricas, pero agregando un mayor tiempo de proceso y aumentando el tamaño de las claves, el cifrado se vuelve más grande que el original. Lo que se hace normalmente es compartir las claves simétricas mediante cifrado asimétrico para posteriormente pasar a un cifrado simétrico más rápido y menos costoso.

En la actualidad la criptografía de clave pública se divide en tres familias según el problema matemático en el que basan su seguridad:

- La primera familia es la que basa su seguridad en el problema de factorización de números enteros grandes PFE [25], los sistemas que pertenecen a esta familia son el sistema RSA [11] el de Rabin Williams RW [26].
- La segunda familia es la que basa su seguridad en el problema del logaritmo discreto, a esta familia pertenecen el sistema de Diffie Hellman de intercambio de claves y el sistema DSA [27] de firma digital.
- La tercera familia es la que basa su seguridad en el problema del logaritmo discreto elíptico, en este caso hay algoritmos tanto de intercambio de claves como de firma digital, algunos son el DHE (Diffie Hellman elíptico), NRE (Nyberg-Rueppel), MQV (Menezes Qu Vantone).

2.2.6. RSA

RSA (Rivest, Shamir, Adleman) [11] es un algoritmo de cifrado asimétrico de clave pública desarrollado en 1977, cuyo problema matemático es el de la factorización de un número entero n muy grande (1024 bits que equivalen a un número de 308 dígitos) que es el producto de dos números primos p y q de la misma longitud. Entonces la clave pública es n y la clave privada son p y q .

Para crear el juego de llaves (pública y privada) se buscan dos números primos grandes p y q , luego se calcula $n = p * q$. Al mismo tiempo se buscan dos enteros d y e , de manera que se verifique que:

- d sea coprimo con $(p - 1)(q - 1)$
- e sea inverso modular de d , es decir que sea solución de $e * d = 1 \bmod (p - 1)(q - 1)$
- La llave pública es el conjunto de números $\{e, n\}$
- La llave privada es el conjunto de números $\{d, n\}$

Seguridad de RSA a futuro

La seguridad de RSA reside en la dificultad de encontrar la clave privada a partir de la pública, cosa que se lograría si se pudiera factorizar n (publico) en sus factores primos. Hasta ahora el método más eficiente de factorizar grandes números (el algoritmo de Euclides) necesita un tiempo de computo que aumenta exponencialmente con el número de dígitos de n . Es un problema de la clase NP aunque no se haya demostrado rigurosamente.

Si se descubriese un método de factorización que necesite un tiempo que aumente de forma polinómica con respecto al número de dígitos de n , la seguridad de RSA sería vulnerada.

Peter Shor demostró que un computador cuántico puede implementar un algoritmo de factorización polinómico en un tiempo razonable [10]. El algoritmo de Shor marca el final de la seguridad de RSA, siempre que exista una computadora cuántica.

Se vuelve imprescindible en un futuro cuando la computación cuántica sea posible, diseñar algoritmos de criptografía cuántica que reemplacen los algoritmos actuales.

2.3. CRIPTOGRAFÍA CUÁNTICA

Como se revisó en la cifra de Vernam *One time Pad*, un cifrado simétrico es secreto perfecto siempre y cuando se cumplan los dos requisitos:

- La llave debe ser aleatoria
- La llave debe usarse una sola vez

La llave tan larga como el mensaje y de un solo uso, debe de estar en posesión tanto del emisor como del receptor. El principal obstáculo de orden práctico es el de como compartir la clave, dado que si cae en manos de terceros el secreto se perdería. Es en este punto donde la mecánica cuántica hace su aparición aportando métodos seguros de distribución de llaves “Quantum Key Distribution”.

La seguridad de los mecanismos QKD reside en las características intrínsecas de la mecánica cuántica estudiadas en los apartados, Principio de incertidumbre de Heisenberg, Desigualdades de Bell y Entrelazamiento cuántico.

- El teorema de no clonación asegura que un estado cuántico $||\varphi\rangle$ no puede ser copiado sin destruirlo en el proceso.
- No se puede medir la polarización de un fotón en la base rectilínea y diagonal al mismo tiempo.
- Observar un sistema cuántico equivale a realizar una medida en él, lo que inevitablemente lo altera
- Si se utilizan las desigualdades de BELL y el fenómeno EPR de entrelazamiento, no se puede modificar una partícula entrelazada sin modificar al mismo tiempo a su par en posesión del emisor, detectando así a cualquier intruso

2.3.1 El computador cuántico

Dentro de los avances de la física cuántica aplicada a la ingeniería existe el computador cuántico. Concepto conocido como una maquina autónoma capaz de procesar bits cuánticos o qubits. Cada vez se incrementa la demanda por

procesamiento computacional, computadores más rápidos, más seguros, más eficientes y reducir cada vez más el tamaño de sus componentes acercándose a la escala del átomo, lugar en donde reinan las leyes cuánticas.

Según Gruska: "alrededor del año 2020 la computación se llevara a cabo en el nivel del átomo" [28]. Los computadores cuánticos podrían desarrollar ciertas tareas computacionales exponencialmente ms rápido que cualquier computador convencional. Dichas predicciones van de la mano con los desarrollos de algoritmos cuánticos que desde una base teórica aprovecharan las características de la teoría cuántica.

Dentro de los problemas que pueden ser resueltos con un computador cuántico se tienen:

- La transformada de Fourier, problema crucial para resolver algoritmos de ordenación y algoritmos de factorización.
- La búsqueda en bases de datos desordenadas.
- Factorización de números grandes de aproximadamente 400 dígitos, proceso que tomaría billones de años al supercomputador más potente actual, mientras que con una computadora cuántica tardaría aproximadamente 1 año.
- Tele portación de estados cuánticos.
- Distribución segura de claves criptográficas.

Es en este último punto en el que está centrada la seguridad de la criptografía. Analizar los componentes físicos que permitan la implementación de un prototipo simulador de uno de estos algoritmos de criptografía (distribución de claves secretas).

2.3.2. Algoritmos de distribución de claves cuánticas

El problema de los sistemas de criptografía simétricos es poder compartir la clave de manera segura. Es para resolver este problema que surgen los protocolos de distribución de claves, los que permiten mediante un canal cuántico intercambiar las claves de manera segura, para luego mediante un canal tradicional realizar la

comunicación simétrica. La generación de estas claves también forma parte del proceso de distribución de claves QKD (Quantum Key Distribution). La generación debe cumplir las siguientes condiciones

- Ningún intruso puede obtener la clave transmitida
- Cualquier intento de intromisión para obtener la clave transmitida puede ser detectado con alta probabilidad
- Los usuarios pueden estar seguros de que esta compartiendo la misma clave [28].

El modelo para los protocolos QKD consta de dos participantes Alice y Bob que quieren establecer una llave en conjunto, ambos tienen acceso a un canal de comunicación clásico y un canal cuántico. Como se ve en la **Figura 2.17**, un tercer participante llamado Eve posee acceso a los dos canales y posee ilimitada capacidad de cómputo.

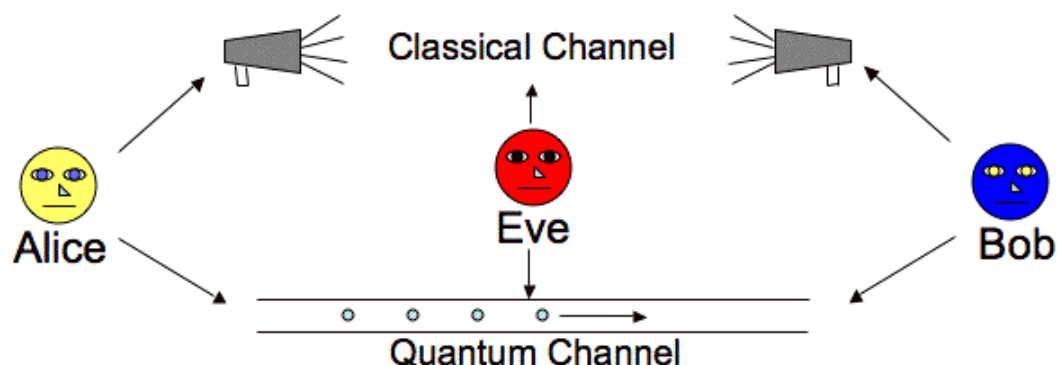


Figura 2.17. Mecanismo de comunicación con canales clásico y cuántico [29]

En la **Figura 2.17** se ve el modelo QKD el cual tiene como propósito que dos usuarios Alice y Bob que aún no intercambian mensajes puedan ponerse de acuerdo en una llave aleatoria que mantienen secreta de un tercer usuario Eve. Este puede escuchar y monitorear todo el tráfico en el canal clásico y se asume que posee gran capacidad de cómputo, también tiene acceso al canal cuántico, sin embargo cuando la información se transmite mediante estados no ortogonales en un sistema cuántico como un fotón polarizado en $0^\circ, 45^\circ, 90^\circ, 135^\circ$ se obtienen las propiedades de que las transmisiones no pueden ser copiadas, leídas ni observadas por el atacante. Eve no puede ganar ningún tipo de información de la transmisión sin perturbar el mensaje de manera que inevitablemente los usuarios legítimos sabrán de su presencia.

2.3.3. Principio de Incertidumbre de Heisenberg aplicado en criptografía

La seguridad de los sistemas de criptografía cuántica, reside en ciertos principios de la física cuántica y en cómo aplicarlos en la ingeniería. La propiedad cuántica fundamental es el principio de incertidumbre de Heisenberg, el fenómeno en el que cualquier medición en la polarización rectilínea de un fotón ($0^\circ, 90^\circ$) vuelve aleatorias las mediciones en la polarización diagonal del fotón ($45^\circ, 135^\circ$) y viceversa.

Solamente una de un par de propiedades puede ser conocida con certeza, en este caso la polarización rectilínea o diagonal. Se puede generalizar este principio a cualquier par de propiedades que estén conjugadas, como el momentum y la posición.

De manera que se pueden usar las distintas polarizaciones de un fotón como las bases conjugadas de un experimento para codificar Qubits, esto es gracias a que los fotones pueden transmitirse por fibra óptica y es tal vez el mejor sistema de transmisión cuántica hasta el momento. Además del principio de incertidumbre, se hace uso del teorema de no clonación, que establece que es imposible crear copias idénticas de un estado cuántico arbitrario desconocido. Sin el teorema de no clonación sería posible evitar el principio de incertidumbre, creando múltiples copias de un estado cuántico, permitiendo medir diferentes propiedades conjugadas en cada copia.

Los protocolos que utilizan el principio de incertidumbre y el teorema de no clonación son el BB84 y el B92.

2.3.4. Entrelazamiento cuántico aplicado en criptografía

La otra propiedad de la mecánica cuántica que puede ser usada en criptografía, es el entrelazamiento. Este fenómeno como ya se ha estudiado, define como dos partículas se encuentran entrelazadas como fruto de una fuente de partículas común. Al medir una propiedad en una de estas partículas, la otra partícula manifiesta el estado opuesto al medido de manera instantánea. Este proceso de comunicación usando estados entrelazados y ayudado por un canal de información clásico es conocido como teletransportación cuántica de información y es la base del protocolo E91 [7].

Mediante las desigualdades de Bell es posible cuantizar el entrelazamiento, de manera que al transmitir 2 partículas entrelazadas se pueden comparar los estados y verificar si se viola la desigualdad de Bell. Si el grado de violación no es el anticipado, eso significa que el estado cuántico interno ha sido alterado y la probabilidad de que estén espionando el canal de comunicación es extremadamente alta.

2.4. PROTOCOLOS DE CRIPTOGRAFÍA CUÁNTICA

2.4.1. Protocolo BB84

En la International Conference on Computers Bangalore 1984, Charles Bennett y Gilles Brassard publicaron el primer protocolo de distribución de claves cuánticas [6]. Está basado en el principio de incertidumbre de Heisenberg (P.I.H) y se le conoce como protocolo BB84 debido a los autores y el año de publicación. Es el protocolo más importante dentro de la familia de protocolos basados en el principio de incertidumbre y prácticamente todos los protocolos futuros son variantes del BB84.

La idea principal para todos los protocolos de la familia P.I.H es que Alice pueda transmitir a Bob una clave secreta aleatoria mediante el envío de una cadena de fotones por un canal cuántico, donde la información de la clave secreta está codificada en la polarización de los fotones. Un intruso al que llamaremos Eve tiene acceso total al canal cuántico y al canal público de comunicación y se presume posee ilimitada capacidad de cómputo, a pesar de tener acceso a ambos canales Eve no puede interceptar los fotones, medirlos y reenviarlos a Bob sin perturbar el estado de los fotones.

Polarización y definición de estados

Un bit puede ser codificado mediante el estado de polarización de un fotón, las posibles polarizaciones del fotón que se usan en el protocolo BB84 son: $0^\circ, 45^\circ, 90^\circ, 135^\circ$, estos cuatro tipos de polarización se dividen entre dos conjuntos que llamaremos “bases” y serán utilizados para representar los bits 0 y 1:

Polarización lineal en base rectilínea: fotón polarizado en 0° para el bit $|0\rangle$ y polarización en 90° para el bit $|1\rangle$.

Polarización lineal en base diagonal: fotón polarizado en 45° para el bit $|0\rangle$ y polarización en 135° para el bit $|1\rangle$.





	Polarización rectilínea	Polarización diagonal
Estado $0\rangle$ Fotón polarizado en 0° o 45°		
Estado $1\rangle$ Fotón polarizado en 90° o 135°		

Figura 2.18. Codificación de qubits para los ángulos de polarización.

Como muestra la **Figura 2.18**, el cero se codifica polarizando un fotón en un ángulo de 0° para un filtro de polarización rectilínea y en 45° para un filtro de polarización diagonal, el bit uno se codifica polarizando un fotón en 90° en un filtro de polarización rectilínea y en 135° en un filtro de polarización diagonal. [30]

Los estados cuánticos $|0\rangle$ y $|1\rangle$ son vectores complejos por lo tanto son aplicables todas las propiedades de los vectores, así como el producto escalar interno. Se dice que las bases rectilíneas y diagonales son bases **no ortogonales** porque el producto interno entre sus vectores de estado es distinto de cero y dos estados son ortogonales cuando su producto interno es igual a cero, lo que indica que los vectores forman un ángulo de 90°.

$$Base\ rectilínea\ |0\rangle \cdot Base\ rectilínea\ |1\rangle =$$

$$0^\circ \cdot 90^\circ = 0$$

“los dos estados en la misma base son ortogonales”

$$Base\ rectilínea\ |0\rangle \cdot Base\ diagonal\ |1\rangle =$$

$$0^\circ \cdot 135^\circ \neq 0$$

“los estados de distintas bases son no ortogonales”

Por ejemplo el estado $|0\rangle$ (0°) y $|1\rangle$ (90°) codificados en la misma base rectilínea son ortogonales porque sus vectores forman 90° y el producto interno entre ellos es cero. En cambio el estado $|0\rangle$ (0°) y $|1\rangle$ (135°) codificados en bases distintas (0 en rectilínea y 1 en diagonal) forman un ángulo distinto de 90° , por lo tanto son estados en bases no ortogonales. Esto es muy importante ya que define la seguridad del protocolo.

Si un intruso intercepta un fotón y esta codificado en una base ortogonal, basta con medir el fotón y descifrar uno de los ángulos de polarización del bit y al ser una base ortogonal sabrá inmediatamente que el ángulo que este a 90° del que midió será el que debe utilizarse para codificar el otro bit.

Si en cambio se usan bases no ortogonales para codificar los bits, el intruso no podrá descifrar el bit restante debido a que “medir” en física cuántica implica modificar el estado y en el caso de estados no ortogonales, medir la polarización de base rectilínea vuelve aleatoria y destruye la información de la base diagonal debido a que son bases conjugadas.

Este fenómeno es la expresión del principio de incertidumbre de Heisenberg que da la fortaleza al protocolo, cualquier medición en la polarización rectilínea del fotón ($0^\circ, 90^\circ$) vuelve aleatorias las mediciones de la polarización diagonal del fotón ($45^\circ, 135^\circ$) y viceversa.

Las dos bases rectilínea y diagonal también se llaman “bases conjugadas” porque al medir un fotón con un filtro polarizador distinto al que se usó para polarizarlo en un principio, se pierde la información original “si ocurre uno el otro ya no puede ocurrir”.

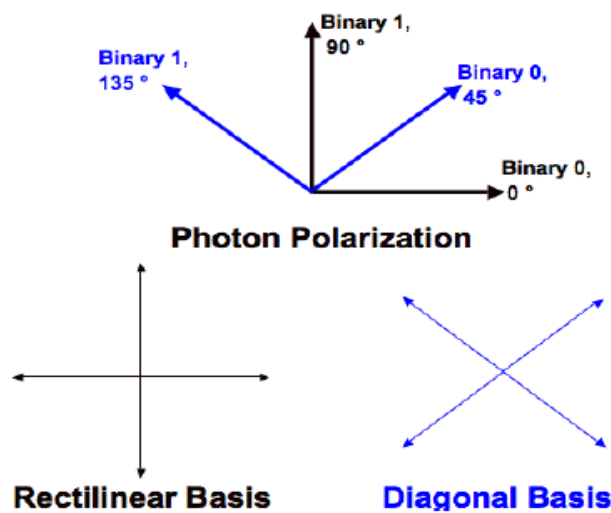


Figura 2.19. Bases de polarización y sus ángulos.

La base rectilínea polariza los fotones en 0° o 90° y al mismo tiempo utilizar un filtro rectilíneo para saber en qué ángulo viene polarizado un fotón hace que la información de la polarización original del fotón se pierda en caso que fuera diagonal. [29].

Polarización como combinación lineal de estados

La polarización diagonal también puede expresarse como una combinación de estados cuánticos de base rectilínea. Polarización diagonal como combinación lineal:

Polarización rectilínea: $|0\rangle = 0^\circ$ $|1\rangle = 90^\circ$

Polarización diagonal: $|0'\rangle = 45^\circ$ $|1'\rangle = 135^\circ$

$$|0'\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

$$|1'\rangle = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle$$

Esta combinación de estados fortalece el principio de incertidumbre en el que el ángulo 45° puede interpretarse como una combinación entre los ángulos 0° y 90° y al realizar la medición, este estado superpuesto colapsa en uno de los posibles valores 45° , 0° o 90° , este colapso depende del filtro que se use para medir el fotón. Si se envía un fotón polarizado a 0° en base rectilínea y se utiliza un filtro rectilíneo para detectarlo, hay 100% de que el fotón pase por el filtro (debido a que se está midiendo con la misma base). Por lo tanto cuando se usa la misma base para enviar y para recibir el fotón este lograra pasar el filtro, sin embargo cuando se usa una base

distinta para detectar el fotón en el lado receptor el fotón resulta alterado y pierde su polarización original.

Si el fotón viene polarizado a 45° y el receptor utiliza un filtro rectilíneo a 90° para detectarlo, entonces hay un 50 % de probabilidad de que el fotón se comporte como si estuviera polarizado a 90 ° y un 50% de probabilidad de que el fotón se comporte como si estuviera polarizado a 0°, esto es porque el fotón polarizado a 45° se comporta como una combinación lineal de los estados 0° y 90 °.

$$| \text{Fotón polarizado a } 45^\circ \rangle = \frac{1}{2} | \text{Fotón polarizado a } 0^\circ \rangle + \frac{1}{2} | \text{Fotón polarizado a } 90^\circ \rangle$$

De esta manera cuando se envía un fotón en polarización diagonal y se usa un detector en base rectilínea hay un 50% de obtener un bit correcto, por lo tanto los bits obtenidos con una base distinta a la del emisor deben ser descartados en el proceso de discusión pública (debido a que solo en el 50% de los casos emisor y receptor acertaran).

Por esta razón en el protocolo BB84 se obtienen dos claves o dos cadenas de bits, la primera son todos los fotones que fueron detectados sin importar si se usó la misma base para enviar y recibir. Todos estos bits forman la **RAW KEY** o clave en bruto. Luego Alice y Bob se ponen de acuerdo en conservar solo los bits en lo que ambos usaron la misma base para enviar y recibir. Esta será la **SIFTED KEY** o clave depurada. Ver ejemplo de funcionamiento en apéndice⁶.

2.4.2. Protocolo B92

En el mismo escenario anterior, Alice y Bob necesitan intercambiar una clave secreta para luego usarla en un algoritmo de criptografía ONE-TIME-PAD. La clave que intercambiaran debe ser una secuencia perfectamente aleatoria de unos y ceros. Como Alice y Bob no pueden juntarse a intercambiar información físicamente esta tarea es imposible usando medios clásicos, debido que no existe mecanismo que les permita saber si un intruso intercepto la comunicación.

⁶ Apéndice: Ejemplo de funcionamiento BB84

Para superar este problema se utiliza un nuevo paradigma en criptografía “la criptografía cuántica”. Utilizando fotones polarizados al igual que en BB84 se usa también un filtro detector por parte de Bob que les permitirá mediante el uso del principio de incertidumbre saber si hay un intruso espiando la comunicación. La primicia de B92 en contraste a BB84 radica en que 2 estados no ortogonales son suficientes para garantizar la detección de un intruso.

En 1987 Ivanovic demostró que dos estados cuánticos no ortogonales pueden ser distinguidos sin ambigüedad a cambio de añadir pérdidas al sistema [31]. Basado en este principio, Bennett propuso en 1992 una versión simplificada del protocolo BB84 en su Paper [8]. La principal diferencia con BB84 es que para cada base diagonal y rectilínea solo se utiliza un estado no ortogonal en lugar de dos, de manera que en total se necesitan solo dos polarizaciones en lugar de cuatro.

Para llevar a cabo esta tarea se usaran fotones polarizados igual que en BB84, lo que permitirá mediante el principio de incertidumbre de Heisenberg saber si hay un intruso espiando la comunicación. Los fotones transmitidos por este protocolo están polarizados en dos sistemas no ortogonales de referencias incompatibles, uno rectilíneo (0° o 90°) y otro diagonal (45° o 135°).

La polarización utilizada por Alice será en dos bases distintas rectilínea y diagonal. Para un fotón polarizado en base rectilínea existen dos ángulos de polarización posibles: 0° y 90° . El fotón polarizado en un ángulo de 0 grados se usara para representar un qubit $|0\rangle$ y los fotones polarizados en un ángulo de 90 grados se usaran para codificar el qubit $|1\rangle$.

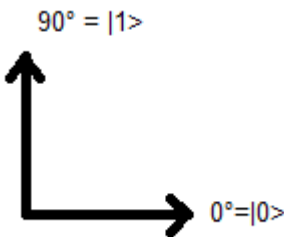


Figura 2.20. Bases de polarización rectilíneas

Para un fotón polarizado en base diagonal existen dos ángulos de polarización posibles: 45° y 135° (el ángulo 135° también puede considerarse como -45°). El fotón polarizado en un ángulo de 45° se usara para representar el qubit $|1\rangle$ y los fotones polarizados en un ángulo de 135° se usaran para representar el qubit $|0\rangle$.



Figura 2.21. Bases de polarización diagonales

Las bases rectilíneas y diagonales se dicen **NO ORTOGONALES** porque sus ángulos no forman 90° (por ejemplo un $|0\rangle$ en base rectilínea se polariza a 0° y en diagonal a 135° y un $|1\rangle$ en rectilínea se polariza a 90° y a 45° en diagonal). En cambio sí se usara la misma base para codificar los dos bits tanto el 1 como el 0 (por ejemplo el qubit $|1\rangle$ en base diagonal a 45° y el qubit $|0\rangle$ en base diagonal a 135°), estos formarían un ángulo de 90° volviéndolos ortogonales y esto eliminaría la seguridad del protocolo.

Los qubits deben formar un ángulo de 45° entre sí para poder usar el Principio de incertidumbre de Heisenberg como mecanismo de seguridad, ya que en el lado receptor Bob medirá los fotones usando la base opuesta o conjugada a la usada por Alice como emisora. Esto permite que los fotones sean detectados por Bob con un 50% de incertidumbre y gracias al P.I.H cualquier intruso que intercepte los fotones modificara su estado reduciendo aún más este 50% de detección.

De esta manera cuando Alice quiere codificar un bit clásico “1” usando un qubit $|1\rangle$ que representa un estado de polarización de un fotón ya sea en base diagonal o rectilínea, debe elegir una base y usar ese ángulo de polarización para su fotón, pero luego para codificar un bit clásico “0” deberá elegir un fotón polarizado en la base opuesta obligatoriamente.

Funcionamiento

El funcionamiento comienza con Alice transmitiendo a Bob una secuencia de fotones. Después de que Alice haya transmitido todos los fotones a Bob, se procede a evaluar de qué forma realizo la detección Bob y a mirar en donde hay coincidencias con respecto a Alice

1) Alice debe elegir codificar sus qubits $|0\rangle$ y $|1\rangle$ en bases distintas de manera que los fotones polarizados estén en bases no ortogonales. Por ejemplo Alice puede codificar el $|0\rangle$ usando una base rectilínea a 0° y el qubit $|1\rangle$ en base diagonal a 45° .

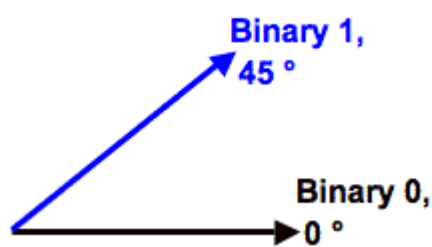


Figura 2.22. Polarizador emisor de Alice. Fotón Polarizado a 0° rectilíneo para el qubit $|0\rangle$ y a 45° diagonal para el qubit $|1\rangle$.

Bit o Qubit Alice	Angulo de polarización	Base
$ 0\rangle$	0°	Rectilínea
$ 1\rangle$	45°	Diagonal

Tabla 2.3. Polarización para los fotones de Alice y los respectivos qubits

2) Bob Debe preparar su detector para realizar las mediciones eligiendo entre las dos bases de manera aleatoria (entre diagonal y rectilínea), pero la interpretación de los fotones que le llegan y la codificación de los bits estarán en forma opuesta. Por ejemplo si Alice representa un $|0\rangle$ usando un fotón polarizando en base rectilínea a 0° , Bob deberá usar la base diagonal a 135° para representar el qubit $|0\rangle$.

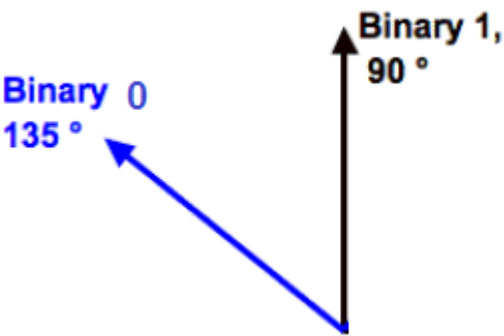


Figura 2.23. Filtro detector de Bob. Fotón polarizado a 135° diagonal para el qubit $|0 \rangle$ y a 90° rectilíneo para el qubit $|1 \rangle$.

Bit o Qubit Bob	Angulo de polarización	Base
$ 0\rangle$	135°	Diagonal
$ 1\rangle$	90°	Rectilínea

Tabla 2.4. Polarización para el detector de Bob y sus respectivos qubits.

Bits Alice	0	1	0	0	1	0
Base de polarización Alice	0° R	45° D	0° R	0° R	45° D	0° R
Fotón enviado por Alice	→	↗	→	→	↗	→
Base de polarización Bob	135°D	135° D	135° D	90° R	90° R	135° D
Pasa la base de detección?	SI	NO	SI	NO	SI	SI
Fotón detectado	↘	-	↘	-	↑	↗
Pasa el 50% de incertidumbre?	SI	-	SI	-	SI	NO
Discusión publica	SI		SI		SI	NO
Hubo detección?						
Bit finales SIFFTED KEY	0	-	0	-	1	-

Tabla 2.5. Trasmisión y recepción de los bits en B92

La **Tabla 2.5** muestra la transmisión y recepción de los bits para B92: Alice y Bob solo guardaran los bits para los que Bob haya detectado el fotón, para esto primero el fotón de Alice debe pasar la base de polarización de Bob y luego debe pasar el 50% de incertidumbre para que el fotón colapse en la componente correcta de la superposición. En la discusión pública se define si se detectó o no el fotón y en qué base.

Para cada fotón enviado por Alice, Bob establece su detector de manera aleatoria en una de las dos bases de polarización que tiene predeterminadas según la tabla. Pueden ocurrir dos casos:

a) La base de Bob coincide con la de Alice:

Cuando Bob utiliza la misma base para medir el fotón que la de Alice, entonces los ángulos de polarización del fotón y el detector forman 90° , bloqueando el paso del fotón. Por lo tanto el fotón no será detectado.

Ejemplo: Alice manda un $|1\rangle$ polarizando un fotón en un ángulo de 45° en base diagonal. Bob trata de medir el fotón usando la misma base diagonal, pero su detector estará alineado a 135° para esa base según su tabla, formando así un ángulo de 90° entre el fotón incidente y el detector, impidiendo que atraviese el filtro. Lo mismo pasa para la base rectilínea si se envía a 0° y se detecta a 90° .

Por lo tanto siempre que Bob mida usando la misma base con la que Alice envía, no habrá detección.

b) La base de Bob no coincide con la de Alice:

Cuando Bob mide usando una base distinta a la que uso Alice para polarizar el fotón, entonces los ángulos de polarización forman 45° y el fotón tiene un 50 % de probabilidad de pasar el filtro. Lo que ocurre es lo siguiente:

Según el principio de incertidumbre de Heisenberg hay un 50% de probabilidad de que un fotón polarizado a 45° logue pasar un filtro alineado a 90° , esto es porque un fotón a 45° se puede expresar como una combinación lineal o *superposición* de los estados 0° y 90° .

$$|\text{Fotón polarizado a } 45^\circ\rangle = \frac{1}{2} |\text{Fotón polarizado a } 0^\circ\rangle + \frac{1}{2} |\text{Fotón polarizado a } 90^\circ\rangle$$

$$\text{Diagonal} = 50\% \text{ Vertical} + 50\% \text{ Horizontal}$$

Al momento de realizar la medida con los instrumentos, el estado de superposición cuántico colapsa en solo una de sus componentes y el fotón se verá obligado a elegir si quedar polarizado a 0° o a 90° . De manera que en el 50% de los casos que el fotón elija colapsar a 90° pasara el filtro y cuando colapse a 0° se bloqueara.

Ejemplo: Alice envía un $|1\rangle$ polarizando un fotón a 45° en base diagonal y Bob lo mide usando una base rectilínea ajustando su filtro en 90° . Si Bob logra detectar un fotón con su filtro a 90° , él puede estar seguro de que Alice mando un $|1\rangle$ ya que el fotón a 45° tiene un 50% de probabilidad de pasar ese filtro mientras que un fotón polarizado a 0° tiene 0% de pasar el filtro. Todo lo que tiene que hacer Bob después de lograr detectar un fotón, es ver su tabla de codificación y ver que Qubit corresponde al ángulo de su filtro.

También puede que Alice mande un $|1\rangle$ a 45° y Bob no lo detecte con su filtro a 90° , esto es debido a que este fotón forma parte del 50% de fotones que colapsaran en un ángulo de 0° y no pasaran el filtro a 90° . Por esto cuando Bob elije la base opuesta a la emisora de Alice, solo se detecta el 50% de los fotones totales y el otro 50% de los fotones se perderán. Por esto B92 es más simple que BB84 pero menos eficiente debido a que se perderán muchos más fotones.

Por lo tanto cada vez que Bob recibe un fotón usando B92 se puede asegurar que se ha realizado una medida en la base correcta, de lo contrario no habrá detección en los instrumentos de Bob. En B92 no existe fase de reconciliación de bases y la clave intercambiada será directamente la SIFTED KEY (clave depurada).

3) En la fase de discusión pública, Alice informa a Bob mediante un canal público en que momento envió un fotón y Bob responde si detecto un fotón en ese instante de tiempo y en qué base lo detecto sin anunciar la polarización. Esta información no compromete la seguridad ya que no se informa el valor de cada bit sino el instante y la base. En promedio solo el 25% de los bits transmitidos son detectados con éxito lo que quiere decir que el 75 % restante de la secuencia original de bits se perderá. 50% de los bits totales se pierden en la elección de la base para el filtro de Bob y de ese 50% solo el 50% lograra pasar la incertidumbre en caso de ser medidos con la base adecuada.

Alice y Bob guardan aquellos bits en los casos que Bob detecto un fotón y todos los demás casos no serán detectados. Es decir que guardaran los bits para los cuales los fotones se codificaron en bases distintas. Esta secuencia de bits pasara a formar la clave secreta ya depurada.

4) Para el proceso de verificación de espionaje de forma similar a en BB84 por el canal público se intercambia una fracción de la clave, verificando para cada instante que el bit codificado sea idéntico y se define un valor límite para el Q.B.E.R (tasa de error de bit cuántico)

La fracción de bits que se pueden distinguir sin ambigüedad e Bob es igual a

$$1 - \cos \alpha$$

Siendo α el ángulo entre ambos estados de emisión y detección. En el caso de un ángulo de 45° el QBER es del 29 %. Lo que quiere decir que si la tasa de error de bit cuántico es menor al 29% por ejemplo un 27% entonces Eve podría pasar inadvertida.

El protocolo B92 es susceptible de sufrir un ataque de seguridad conocido como “*ambiguous state discrimination*” [32]. Este ataque consiste en que si las pérdidas en el canal son superiores al 71%, la presencia de Eve podría pasar totalmente desapercibida si esta compensara las pérdidas que genera introduciendo un canal sin ruido entre Eve y Bob. Por ello el empleo del protocolo B92 exige un cuidadoso análisis de pérdidas, con el objetivo de eliminar de la clave la información extraída por Eve mediante este ataque⁷.

⁷ Apéndice: Análisis B92 en presencia de un espía

2.5. CONCLUSIONES DEL CAPITULO

La hipótesis teórica de rendimiento para el protocolo B92 y todo el análisis matemático expuesto en este capítulo, predicen que del total de bits enviados, solo el 25% de ellos lograran ser usados en la clave final. Esto como se explicó en el capítulo anterior, es debido a la naturaleza aleatoria de la observación. Por un lado se pierde el 50% de los bits al elegir que polarización usar (si diagonal u horizontal) y de ese 50 % de bits que si lograron acertar la polarización solo un 50% lograra ser detectado, debido al principio de incertidumbre y el colapso de la superposición de onda.

En el siguiente capítulo se definen las estructuras de desarrollo para trabajar en la etapa de análisis e implementación del algoritmo.

CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN

3.1. DESCRIPCIÓN GENERAL DE LA SOLUCIÓN

La solución consistió en diseñar e implementar un simulador para el protocolo B92 en un ambiente distribuido controlado, en el cual se pudo observar y probar las características de seguridad cuánticas que ofrece el protocolo. Para esto se desarrollaron los programas en java que representan el algoritmo, los cuales deben poder comunicarse en una red cliente-servidor o punto a punto para simular su funcionamiento. Las características cuánticas del medio de comunicación (fotones polarizados) también fueron simuladas, utilizando medios comunes, como los estándares de comunicación TCP/IP Ethernet.

El problema global que se quiere abordar es que con la llegada de las computadoras cuánticas, la seguridad actual de los protocolos de cifrado asimétrico, estaría en peligro. Por esto, es de suma importancia comenzar a analizar y especificar protocolos de criptografía cuántica e implementarlos en algún lenguaje de programación robusto, como Java en este caso.

Parte de la solución fue diseñar el algoritmo matemático e implementar una aplicación simuladora en Java y también se construyó la red distribuida para realizar las pruebas de funcionamiento.

Se cuentan con los trabajos anteriores para los protocolos BB84 [3] y E91 [4], de los cuales el BB84 es el más cercano en funcionamiento al B92. Por lo tanto, fueron utilizados todos estos recursos de software como guía para diseñar el nuevo algoritmo.

3.2. METODOLOGÍA DE TRABAJO

La metodología de trabajo será la expuesta en la asignatura ingeniería de software de la universidad de Tarapacá, orientada a proyectos, considerando en su ciclo de desarrollo los siguientes puntos:

- Inicio proyecto de software
- Planificar un proyecto de software
- Estimar costos de un proyecto de software
- Ejecutar un proyecto de desarrollo de software
- Controlar un proyecto de software

Se utilizara un ciclo de vida con enfoque clásico en cascada, con prototipado secuencial y gestión de versiones. En la etapa de Ejecución del proyecto de software se definen las siguientes tareas:

- Definición del problema
- Análisis del problema
- Diseño de la solución
- Implementación
- Compilación ejecución y pruebas
- Documentación
- Capacitación y mantenimiento

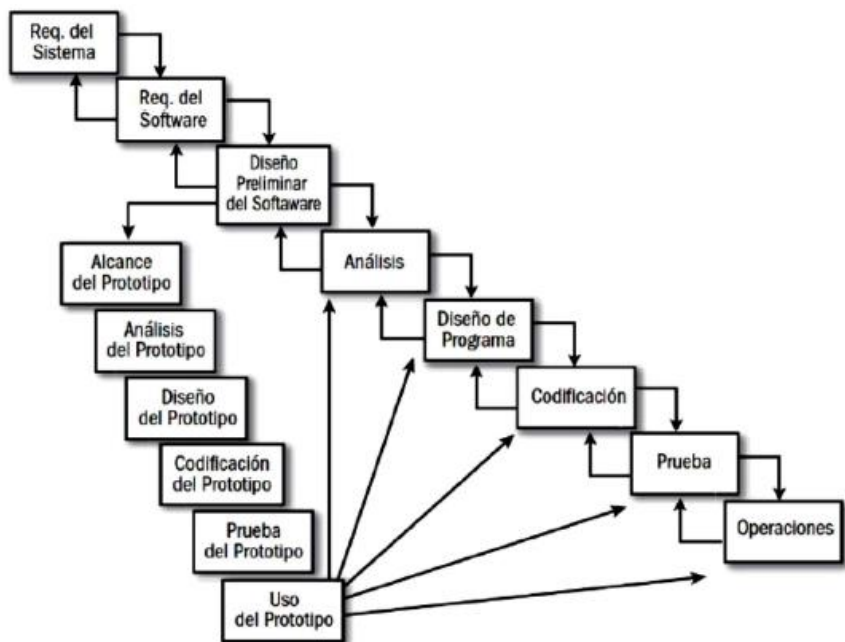


Figura 3.1. Modelo de desarrollo en cascada con prototipo secuencial [33]

La **Figura 3.1**, muestra el modelo de desarrollo de un proyecto de software con prototipo y la **Figura 3.2**, muestra el proceso de versiones. Se utilizará una metodología en cascada con recursión de versiones. De esta manera se realizarán consecutivamente procesos de análisis, diseño e implementación, para que en cada iteración se pueda pulir más el proceso.

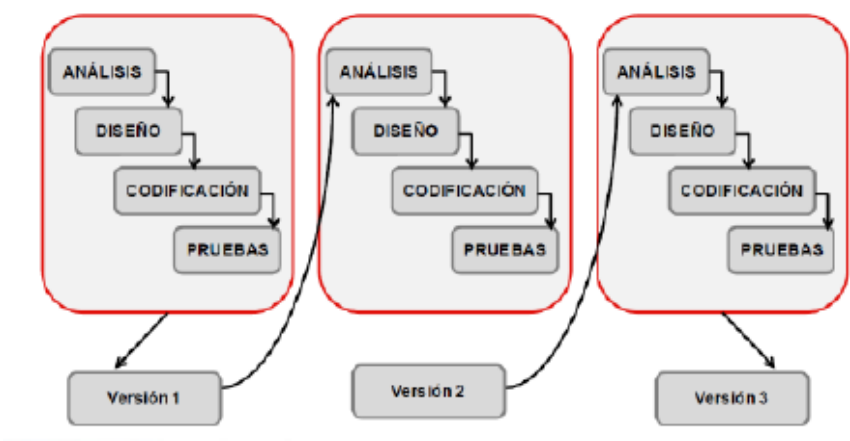


Figura 3.2. Metodología de trabajo para las versiones del prototipo

El proceso de análisis de requerimientos seguirá la estructura de Jesús García Molina estudiada en el curso de Ingeniería de software de la Universidad de Tarapacá [33]. Con la siguiente estructura:

- Educción de requisitos
- Análisis de requisitos
- Documentación (documento de requisitos según estándar)
- Validación de requisitos (entrevista con el cliente)

Para el modelado UML se diseñarán, para cada requisito los correspondientes diagramas de casos de uso, diagrama de secuencia, diagrama de clases. La **Figura 3.3**, muestra los diagramas que se deben diseñar en esta etapa.

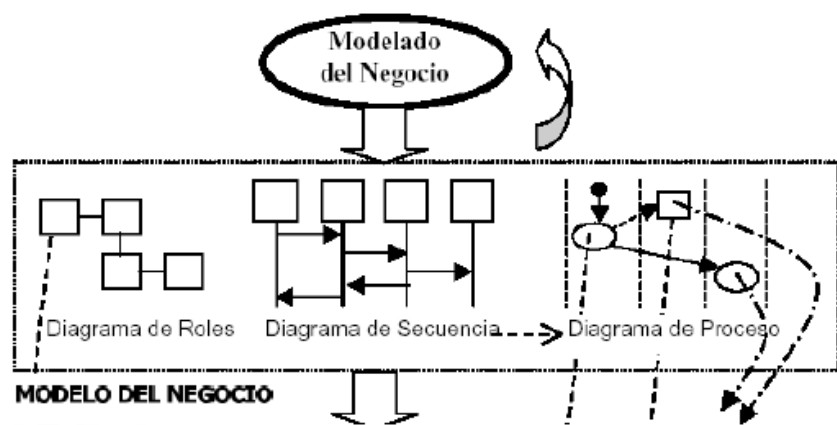


Figura 3.3. Modelado de los diagramas UML

Para medir los resultados se especificara una tabla de evaluación de requerimientos. Para cada requisito funcional y no funcional se deberá verificar si los requerimientos especificados y las primitivas de deseabilidad fueron implementados satisfactoriamente. La **Tabla 3.1**, muestra las primitivas que se deben evaluar para una buena especificación de requisitos [33].

CARACTERISTICAS DESEABLES DE UNA BUENA ERS	
No ambigua	Si todo requisito posee solo una interpretación
Completa	Si todo lo que se supone que el software debe hacer está incluido en la ERS. Por completitud, deberían describirse todas las posibles respuestas a todas las posibles entradas y en todas las situaciones. Además, la ERS no contendrá secciones del tipo “por determinar...”
Correcta	Todo requisito de la ERS contribuye a satisfacer una necesidad real (c/r al cliente)
Comprensible	Todo tipo de lectores (clientes, usuarios, desarrolladores, equipo de pruebas, gestores, etc.) entienden la ERS
Internamente consistente	No existen subconjuntos de requisitos contradictorios
Externamente consistente	Ninguno de los requisitos está en contradicción con lo expresado en documentos de nivel superior. Por ejemplo, en un sistema (hardware + software), lo requisitos del software no pueden contradecir los requisitos del sistema o estudio de factibilidad
Realizable	Si, dados los actuales recursos, la ERS es implementable
No redundante	Coda requisito se expresa en un solo lugar de la ERS. La redundancia, de todas formas, no es del todo mala si aumenta la legibilidad
A un nivel de detalle	Ni muy detallada ni muy vaga
Concisa	Debe ser lo más breve posible, sin que esto afecte al resto de atributos de calidad
Precisa	Si se hace uso de valores numéricos para precisar las características del sistema. La precisión es aplicable, ante todo, a los requisitos no funcionales. Por ejemplo, no es útil “El tiempo de respuesta será más bien rápido”, sino “el tiempo de respuesta será menos de dos segundos”. Esto en la medida de lo posible

Independiente del diseño	Existen más de un diseño e implementación que realizan la ERS. Para ello la ERS debería limitarse a describir el comportamiento externo del sistema
Trazable	Cada requisito se puede referenciar de forma unívoca. Es fundamental para precisar qué requisitos son implementados por qué componente del diseño, lo cual es imprescindible a la hora de realizar pruebas de dicho componente
Modificable	Los cambios son fáciles de introducir
Electrónicamente almacenada	Se encuentra en un archivo de texto, en una base de datos, o mejor aún, ha sido creada con una herramienta de gestión de requisitos
Anotada por relativa importancia	Si los requisitos se clasifican según su importancia. Como mínimo un requisito puede ser “obligatorios”, “deseable”, u “opcional”. Esto sirve para no asignar demasiados recursos a la implementación de requisitos no esenciales
Trazada	Si está claro el origen de cada requisito (quién o qué lo pide)
Organizada	Si el lector puede fácilmente encontrar la información buscada
Con referencias cruzadas	Si se utilizan referencias entre requisitos relacionados (trazabilidad intra-ERS) o entre secciones relacionadas

Tabla 3.1. Primitivas deseables para la especificación de requisitos.

3.3. ARQUITECTURA DE SOFTWARE PARA EL SISTEMA

3.3.1. Arquitectura cliente-servidor

La arquitectura de software a utilizar por la aplicación será en un principio local, utilizando el puerto Localhost 127.0.0.1. Luego, una vez probada la aplicación y depurados todos los errores, se migrara a una arquitectura distribuida cliente servidor.

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Un cliente realiza peticiones a otro programa servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multi-usuario distribuido a través de una red de computadoras.

La **Figura 3.4**, muestra una arquitectura cliente servidor, en la que un mismo tipo de hardware puede funcionar como cliente y servidor al mismo tiempo, dependiendo el servicio que se está ofreciendo.

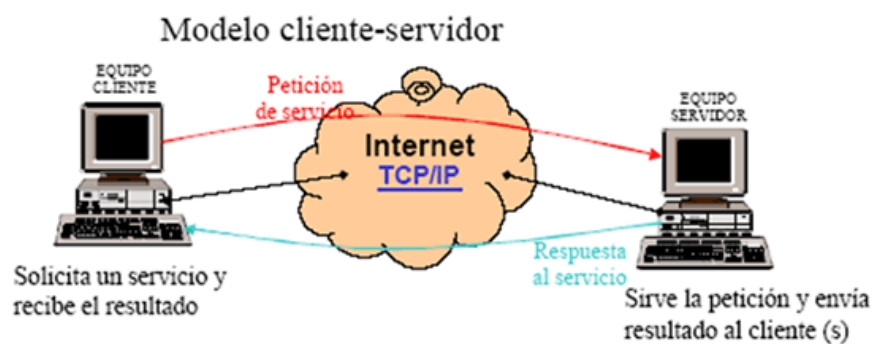


Figura 3.4. Arquitectura cliente/servidor

Se utilizara el patrón cliente-servidor, considerando cada máquina como cliente en un momento de la comunicación y como servidor en otro momento de la comunicación.

El desarrollo de aplicaciones cliente/servidor usando sockets conlleva el diseño de un protocolo consistente en un lenguaje común entre el cliente y el servidor. El diseño de dicho protocolo no siempre es sencillo y es una fuente de errores tales como el *deadlock* [34]. En vez de trabajar directamente con sockets, las aplicaciones cliente/servidor pueden ser desarrolladas mediante la invocación de métodos remotos en Java (JavaRMI), cuyo funcionamiento se detallara a continuación.

3.3.2. Arquitectura distribuida Java - RMI

Java-RMI es un paquete que puede ser usado para el desarrollo de sistemas distribuidos. Dicho paquete le permite a Java invocar métodos de otra máquina virtual Java (posiblemente en otro host). El sistema RMI es muy similar (y generalmente más fácil de usar) a la llamada a procedimientos remotos (RPC *Remote Procedure Call*) que podemos encontrar en otros sistemas. En RMI el programador tiene la sensación de realizar llamadas a métodos locales de una clase local, mientras el sistema es el encargado de pasar los argumentos, ejecutar el método y devolver los resultados de la máquina remota al objeto que ha realizado la llamada.

El conjunto de características soportadas por RMI son aquellas que hacen más fácil el desarrollo de sistemas distribuidos: transparencia en la invocación, recolector automático de basura distribuido, acceso conveniente a *Streams*, etc. La invocación remota es transparente ya que se realiza de idéntica manera que la llamada a un método local, y el recolector automático de basura distribuido nos libera de la necesidad de preocuparnos por la liberación de memoria de un objeto que ya no va a ser utilizado, con independencia de que éste sea local o remoto [35].

A través de RMI, un programa Java puede exportar un objeto, con lo que dicho objeto estará accesible a través de la red y el programa permanece a la espera de peticiones en un puerto TCP. A partir de ese momento, un cliente puede conectarse e invocar los métodos proporcionados por el objeto.

RMI permite exportar objetos como objetos remotos para que otro proceso remoto pueda acceder directamente como un objeto Java. Todos los objetos de una aplicación distribuida basada en RMI deben ser implementados en Java. Esta es una de las principales ventajas de RMI, ya que RMI forma parte del API de Java, con lo que la integración de objetos remotos en aplicaciones distribuidas se realiza sin necesidad de usar recursos adicionales (como por ejemplo, un lenguaje de descripción de interfaces o IDL). De hecho, se utiliza la misma sintaxis para una llamada a un objeto remoto o un objeto local.

La **Figura 3.5**, ilustra los componentes implicados en un sistema RMI: un interfaz remoto, un cliente, y uno o más servidores (objetos remotos) residentes en un host. Un host puede funcionar como servidor de un objeto remoto y al mismo tiempo funcionar como un cliente de un objeto remoto en otro host.

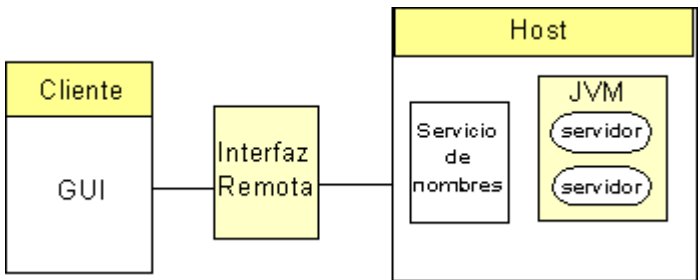


Figura 3.5. Componentes de un sistema RMI

El cliente invoca a los objetos remotos mediante la interfaz remota. Un servicio de nombres (registro RMI) reside en el host proporcionando el mecanismo que el cliente usa para encontrar uno más objetos remotos iniciales RMI.

La interacción con el objeto remoto se lleva a cabo a través de la interfaz remota. Esencialmente, ésta describe los métodos que pueden ser invocados de forma remota, y que el objeto remoto implementa. Cuando se obtiene una referencia a un objeto remoto, el objeto no se envía a través de la red al cliente que lo solicita. En su lugar, se genera un objeto *proxy* o *stub*, que constituye el *proxy* de la parte del cliente del objeto remoto. Todas las interacciones del cliente se realizarán con esta clase *stub*, la cual es responsable de gestionar los datos entre el sistema local y el remoto. Muchos clientes pueden tener referencias a un único objeto remoto. Cada cliente tiene su propio objeto *stub* que representa al objeto remoto, pero dicho objeto remoto NO se replica.

En la parte del servidor, una clase *skeleton* es la responsable de gestionar las llamadas al método y los datos enviados al objeto real referenciado. Éste es el proxy de la parte del servidor para el objeto remoto. El sistema completo puede verse como un modelo de cuatro capas, tal y como se ilustra en la **Figura 3.6**.

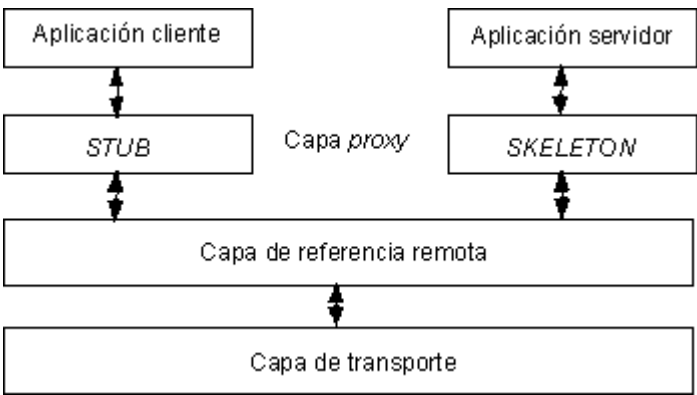


Figura 3.6. Modelo de 4 capas RMI

Capa 1: La primera capa es la de aplicación, y se corresponde con la implementación real de las aplicaciones cliente y servidor. Aquí tienen lugar las llamadas a alto nivel para acceder y exportar objetos remotos. Cualquier aplicación que quiera que sus métodos estén disponibles para su acceso por clientes remotos debe declarar dichos métodos en una interfaz que extienda *java.rmi.Remote*. Dicha interfaz se usa básicamente para "marcar" un objeto como remotamente accesible. Una vez que los métodos han sido implementados, el objeto debe ser exportado. Esto puede hacerse

de forma implícita si el objeto extiende la clase *UnicastRemoteObject* (paquete *java.rmi.server*), o puede hacerse de forma explícita con una llamada al método *exportObject()* del mismo paquete.

Capa 2: Es la capa proxy, o capa *stub-skeleton*. Esta capa es la que interactúa directamente con la capa de aplicación. Todas las llamadas a objetos remotos y acciones, sus parámetros y retorno de objetos tienen lugar en esta capa.

Capa 3: Es la de referencia remota, es responsable del manejo de la parte semántica de las invocaciones remotas. También es responsable de la gestión de la replicación de objetos y realización de tareas específicas de la implementación con los objetos remotos, como el establecimiento de las persistencias semánticas y estrategias adecuadas para la recuperación de conexiones perdidas. En esta capa se espera una conexión de tipo *stream* (*stream-oriented connection*) desde la capa de transporte.

Capa 4: Es la de transporte. Es la responsable de realizar las conexiones necesarias y manejo del transporte de los datos de una máquina a otra. El protocolo de transporte subyacente para RMI es JRMP (*Java Remote Method Protocol*), que solamente es "comprendido" por programas Java.

El desarrollo de Aplicaciones Cliente/Servidor en Java-RMI requiere 6 etapas:

1. Definir una interfaz remota.
2. Implementar dicha interfaz remota. (Servidor).
3. Implementar la aplicación que usa los objetos remotos. (Cliente).
4. Generar los *Stubs* (*Proxies* cliente) y *Skeletons* (entidades servidor).
5. Ejecutar el *Rmi-registry* (servidor de nombres).
6. Ejecutar el servidor y los clientes

3.4. PLATAFORMA Y LENGUAJE DE PROGRAMACIÓN A UTILIZAR

La principal característica y ventaja del lenguaje de programación Java es que se trata de un lenguaje multiplataforma ampliamente difundido en el mundo de las tecnologías, cualquier programa creado en java podrá funcionar correctamente en máquinas de todo tipo y con sistemas operativos distintos. Java proporciona una colección de clases y herramientas para su uso en aplicaciones de red, que permiten abrir sockets y establecer conexiones con servidores o clientes remotos, facilitando la creación de aplicaciones distribuidas.

Java es un lenguaje robusto diseñado para crear software altamente fiable. Para ello, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución, y además la recolección de basura elimina la necesidad de liberación explícita de memoria, propiedad que pueden heredar los sistemas RMI.

La arquitectura de este proyecto está basada en cliente servidor con tecnología Java RMI para la comunicación distribuida. Por esta razón y todas las ventajas que ofrece Java sobre otros lenguajes, es que se elige como la plataforma y el lenguaje a utilizaren el diseño, desarrollo grafico e implementación del protocolo de criptografía cuántica B92. Además los protocolos implementados en los trabajos anteriores también utilizan el lenguaje de programación Java.

3.5. RECURSOS NECESARIOS

Para gestionar los recursos que se necesitan en este proyecto, se realizara un estudio de factibilidad. Debido a la metodología aplicada en este trabajo (metodología de proyectos), es necesario definir que es la factibilidad en un ambiente de ingeniería de software.

La factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. Generalmente la factibilidad se determina sobre un proyecto. Definiendo restricciones y necesidades del sistema. Los puntos a estudiar serán:

- Factibilidad Técnica: ¿Existe la tecnología necesaria? ¿Está al alcance de la mano?
- Factibilidad Operativa: ¿Cuáles son las capacidades organizacionales para sostener el sistema?
- Factibilidad Económica: Relación Costo/Beneficio. ¿Se paga el costo? ¿Cuánto es necesario gastar para ejecutar el proyecto?

3.5.1. Factibilidad técnica

La factibilidad técnica permite evaluar si el equipo y software están disponibles y tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté planificando, también se consideran las interfaces entre los sistemas actuales (de existir) y los nuevos. Así mismo, estos estudios consideran si las organizaciones tienen el personal que posee la experiencia técnica requerida para diseñar, implementar, operar y mantener el sistema propuesto. De existir un sistema implementado previamente, la factibilidad técnica determina si es admisible modificar el sistema existente o cuales serían los costos asociados a dichas modificaciones. Para este proyecto se tiene contemplado crear un sistema nuevo, utilizando trabajos anteriores similares, con un equipo de trabajo de una persona y se posee la experiencia en la plataforma de software para desarrollar. Se consideraran los insumos de hardware y software siguientes:

- Computador portátil con conexión Ethernet y wifi, Intel core i2
- Computador portátil con conexión Ethernet y wifi, Intel core i2
- Router TP-Link con 2 salidas Ethernet de 100mbps y 1 salida coaxial para entrada del ISP
- Cableado de red UTP categoría 5, conectores RJ-45
- Java Development Kit
- Netbeans IDE
- La memoria mínima necesaria para los computadores es de 2 gb

3.5.2. Factibilidad operativa

La Factibilidad Operativa, tiene como objetivo comprobar que la empresa u organización será capaz de darle uso al sistema, que cuenta con el personal

capacitado para hacerlo o tiene los recursos humanos necesarios para mantener el sistema. La factibilidad operativa es la factibilidad de poner el sistema en funcionamiento o en operación. El sistema no debe ser complejo para los usuarios que lo operan, hay que evitar que el usuario ocupe el sistema de manera que pueda ocasionar errores o darle un uso indebido.

En el caso de este proyecto el usuario será el propio desarrollador por lo tanto la complejidad estará definida por y para el mismo individuo, no existirá incomodidad con sistemas anteriores ni complejidad de uso debido a que el sistema es nuevo. La empresa u organización involucrada son el alumno tesista y el profesor guía, ambos son los actores principales del proyecto. El personal ya se encuentra capacitado y ambos actores poseen conocimientos y recursos de software.

El usuario principal, al ser el mismo desarrollador, no puede ocasionar un mal uso o del sistema o errores en él. Para que el sistema sea más cómodo al usuario se utilizara una estructura de interfaz de usuario, con ventanas.

3.5.3. Factibilidad económica

La factibilidad económica estudia la capacidad monetaria del sistema, tanto en la relación costos/beneficios como los gastos asociados a los implementos que fueron especificados en la etapa de factibilidad técnica. Se debe definir cuanto es necesario gastar para ejecutar el proyecto.

Se tiene que comprobar que el proyecto es sustentable económicamente, justificar que los costos sean los mínimos para obtener los resultados requeridos, demostrar que si el sistema no cumple con su objetivo no habrá pérdidas económicas o serán las mínimas.

La **Tabla 3.2**, muestra los valores de los equipos y los materiales necesarios para la etapa de implementación y diseño de la red de simulación. Los elementos de software como licencias se consideran con un precio de 0 pesos, debido a que se utilizan solo programas de licencia libre. Los recursos bibliográficos también son de uso público. La mayoría de los libros son prestados de la biblioteca central de la Universidad de Tarapacá y los artículos científicos son sin fines lucrativos.

Debido a que este proyecto es de carácter científico no lucrativo, no existen beneficios económicos para las partes involucradas, no hay rentabilidad ni ganancia por publicación.

Insumo	Costo en pesos
Computador portátil Netbook Dell 2gb RAM, procesador Intel Core 2 Duo	\$145 000
Computador portátil Dell Notebook 2gb RAM, procesador i2	\$250 000
Router TP-Link velocidad 100mbps 4 puertos Ethernet	\$25 000
Cableado UTP categoría 5 (15 metros)	\$5 000
Conectores RJ-45 (4 unidades)	\$11 200
Licencias de software	\$0

Tabla 3.2. Costos involucrados en la factibilidad económica.

3.6. APORTE

Con la llegada de estas computadoras cuánticas se deberá redefinir los conceptos de seguridad de información dando origen a nuevos algoritmos de criptografía que ofrecerán una seguridad inalcanzable para la computación tradicional. La capacidad de procesamiento que ofrecen las computadoras cuánticas, generara la degradación del sistema actual de encriptación (los algoritmos actuales podrían ser rotos fácilmente por una computadora cuántica). Desarrollar el protocolo de criptografía cuántica B92 es un avance significativo en la comprensión de cómo se debe enfocar el paradigma criptográfico en los próximos años.

La salida directa de este proyecto es el simulador B92 y las pruebas realizadas sobre él, su rendimiento y los problemas de eficiencia que pueda tener.

El protocolo B92 en comparativa con BB84 es más simple de implementar, puesto que solo se necesitan 2 estados no ortogonales de polarización para codificar los bits, esta ventaja en simplicidad tiene un costo en rendimiento, ya que se pierden 75% aproximadamente de los fotones enviados, esto es a causa del principio de incertidumbre y a la elección aleatoria del filtro polarizador

Con el protocolo B92 diseñado e implementado, se puede planificar a futuro una comparativa entre los 3 protocolos implementados en el área de Computación e Informática de la Universidad de Tarapacá. Este estudio sería de gran importancia para verificar la eficiencia teórica de cada uno.

3.7. CONCLUSIONES DEL CAPITULO

La metodología utilizada fue el enfoque clásico de Jesus Garcia Molina presentado en la asignatura ingeniería de Software de la Universidad de Tarapaca. Los recursos necesarios son precarios, pero suficientes para realizar las pruebas en un ambiente distribuido.

CAPÍTULO 4: DESARROLLO

4.1. ANÁLISIS DE REQUERIMIENTOS

Los requerimientos o requisitos de un sistema, describen los servicios que ha de ofrecer un sistema, así como su funcionamiento y de qué manera se cumplirán los objetivos del sistema. Los requerimientos son características que el sistema debe tener o es una restricción que el sistema debe satisfacer para cumplir con su objetivo.

Los requerimientos funcionales representan la naturaleza del funcionamiento del sistema, todo lo que debe ser programado es un requerimiento funcional. Son las interacciones entre el ambiente (representado por los actores) y el sistema. Los actores incluyen al usuario, los miembros de la comunicación y cualquier otro sistema externo. Los requerimientos funcionales indican **que debe** hacer el sistema

Los requerimientos no funcionales son aspectos del sistema que son visibles para el usuario pero no aportan necesariamente una funcionalidad al sistema. Tienen que ver con las cualidades del sistema. Los requerimientos no funcionales definen **como debe** ser el sistema.

Las restricciones son requisitos que se imponen al sistema software a desarrollar (tecnología a usar, protocolos de comunicaciones, compatibilidad con navegadores, etc.). También son todas aquellas cualidades que no tienen capacidad de ser programadas, sino que son inherentes a la teoría del problema a resolver. Por ejemplo el colapso de la función de onda debe ser con un 50% de probabilidad para cada fotón.

4.1.1. Requerimiento y objetivo principal del sistema B92

El sistema debe permitir crear una clave secreta que sea compartida entre el emisor y el receptor, utilizando los conceptos de criptografía cuántica del algoritmo B92. Luego debe ser posible enviar mensajes mediante un canal inseguro utilizando la clave compartida.

4.1.2. Requisitos funcionales

1. Generar las secuencias binarias de (1 y 0) aleatorias que representaran los Qubits
2. Generar las secuencias o esquemas de fotones polarizados asociadas a cada bit
3. Enviar el pulso de fotones polarizados desde el emisor “Alice” hacia el receptor “Bob” y guardarlo en el programa receptor “Bob”.
4. Comparar la secuencia de fotones polarizados del emisor Alice con la secuencia generada en el receptor Bob.
5. Generar la clave en bruto o RAW KEY según los fotones que tengan coincidencias en el punto anterior
6. Generar la clave depurada o SIFTED KEY, comprobando si para cada foton se cumple el principio de incertidumbre de Heisenberg, con un 50% de probabilidad.
7. Intercambiar la SIFTED KEY generada entre Alice y Bob
8. Generar e intercambiar las tablas de hashes generadas para la clave depurada (son necesarias para la seguridad al enviar mensajes por medios no seguros)
9. Crear la interfaz gráfica para presentar los resultados
10. Enviar mensajes entre las partes usando un protocolo de cifrado simétrico (AES) que utilice la clave secreta compartida mediante el protocolo cuántico.

4.1.3. Requerimientos no funcionales

1. Se debe utilizar Java como lenguaje de programación y Java RMI como paquete de comunicación distribuida.
2. El programa debe poder funcionar en distintas máquinas de manera distribuida.
3. El sistema debe tener una presentación de ventanas e interfaz de usuario
4. La red de prueba debe utilizar el modelo Cliente-Servidor

4.1.4. Restricciones

1. La secuencia de bits y la secuencia de polarizaciones deben ser generadas de manera completamente aleatoria
2. La clave depurada SIFTED KEY debe generarse con el colapso de la función de onda en una de las dos polarizaciones posibles para el fotón con un 50% de probabilidad en ejecución
3. El tamaño de la llave no debe sobrepasar los 256 bits

4.2. CASOS DE USO

Una vez identificados los requerimientos, se deben definir los actores involucrados en la realización de estos requerimientos. Cada uno de estos actores del proceso desempeña un cierto papel cuando colabora con otros actores para llevar a cabo las actividades que conforman un caso de uso del sistema. En el caso del sistema B92 tenemos 3 actores: El emisor, El receptor y el usuario del simulador.

Primero se definirán todos los casos de uso relacionados al actor Emisor (al que llamaremos Alice) y luego los casos de uso relacionados al actor Receptor (Al que llamaremos Bob). Para cada caso de uso se define su tabla de descripción y su diagrama de secuencia correspondiente.

4.2.1. Diagrama de casos de uso del sistema completo

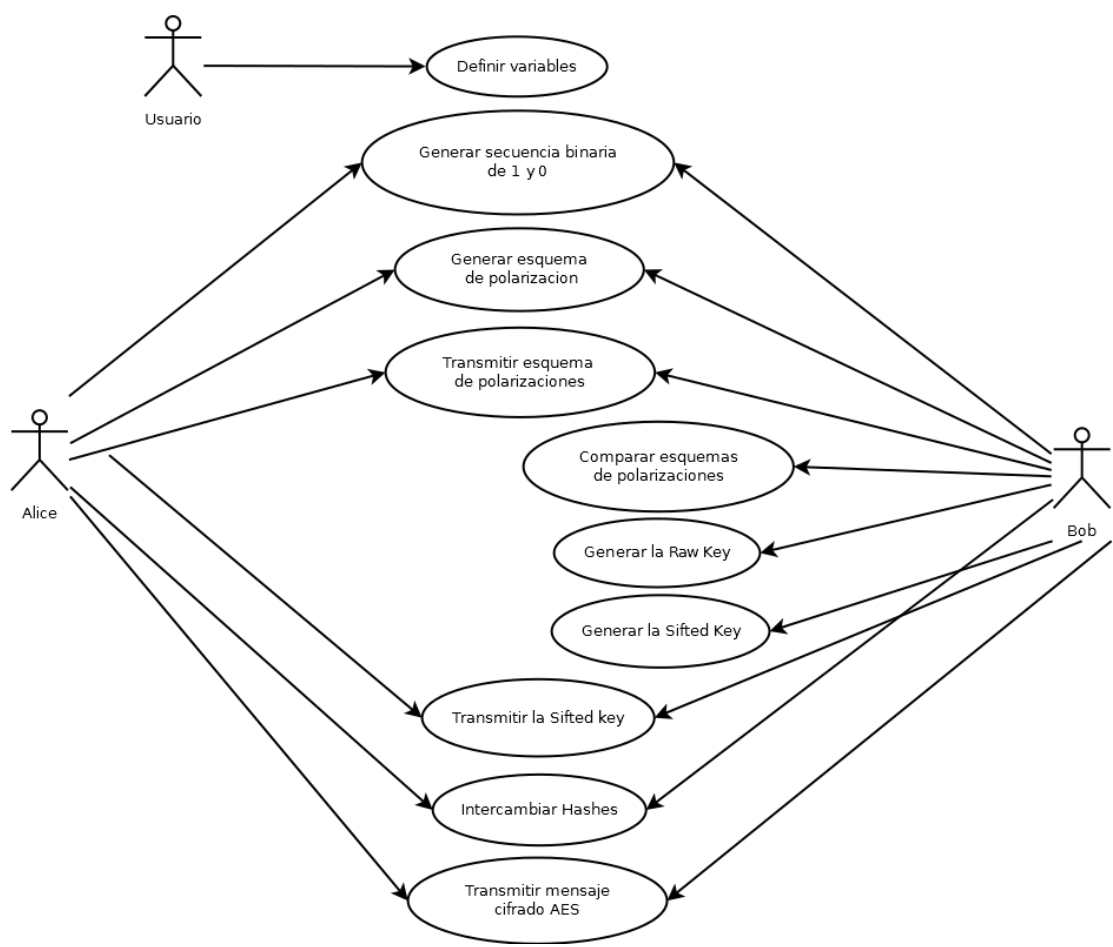


Figura 4.1. Diagrama de casos de uso del sistema

4.2.2. Caso de uso: Definir variables del sistema

Proceso de negocio	Definir variables del sistema
Objetivo	Definir el largo de la secuencia binaria y la ip, en los programas Emisor y Receptor
Descripción	1-Se define el largo de la clave en bits para el dispositivo emisor “Alice” 2-Se define la IP del dispositivo emisor “Alice” 3-Se define el largo de la clave en bits para el dispositivo receptor “Bob” 4-Se define la IP del dispositivo receptor “Bob”
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	5 segundos
Coste de ejecución	

Tabla 4.1. Descripción caso de uso generar variables del sistema

4.2.3. Caso de uso: Generar secuencia binaria (Alice)

Proceso de negocio	Generar secuencia binaria
Objetivo	Generar una secuencia aleatoria de 1 y 0
Descripción	1-Se define el tamaño de la secuencia binaria 2-El emisor genera de manera aleatoria con un 50% de probabilidad un 0 o un 1 3-Se guardan los bits generados en un arreglo
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	10 segundos (para 32 bits)
Coste de ejecución	

Tabla 4.2. Descripción del caso de uso Generar secuencia binaria Alice.

4.2.4. Caso de uso: Generar esquema de polarizaciones (Alice)

Proceso de negocio	Generar esquema de polarizaciones
Objetivo	Para cada bit generado aleatoriamente, se debe definir un fotón polarizado de manera rectilínea o diagonal
Descripción	1-Al momento de generar cada bit aleatorio, se genera al mismo tiempo una polarización que puede ser R o D para rectilíneo o diagonal. 2-El resultado R o D, se guarda en la variable “base3” 3-Luego la polarización se guarda en la clase “secuencia”

Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	5 segundos
Coste de ejecución	

Tabla 4.3. Descripción del caso de uso Generar esquema de polarización Alice

4.2.5. Caso de uso: generar secuencia binaria (Bob)

Proceso de negocio	Generar secuencia binaria
Objetivo	Generar una secuencia aleatoria de 1 y 0
Descripción	1-Se define el tamaño de la secuencia binaria, en la clase 2-El emisor genera de manera aleatoria con un 50% de probabilidad un 0 o un 1 en la clase 3-Se guardan los bits generados en un arreglo
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	10 segundos (para 32 bits)
Coste de ejecución	

Tabla 4.4. Descripción del caso de uso generar secuencia binaria Bob

4.2.6. Caso de uso: Generar esquema de polarización (Bob)

Proceso de negocio	Generar esquema de polarizaciones
Objetivo	Para cada bit generado aleatoriamente, se debe definir un fotón polarizado de manera rectilínea o diagonal
Descripción	1-Al momento de generar cada bit aleatorio, se genera al mismo tiempo una polarización que puede ser R o D para rectilíneo o diagonal. 2-El resultado R o D, se guarda en la variable “ base3 ” 3-Luego la polarización se guarda en la clase “ secuencia ”
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	5 segundos
Coste de ejecución	

Tabla 4.5. Descripción del caso de uso Generar esquema de polarización Bob

4.2.7. Caso de uso: transmitir esquema de polarizaciones

Proceso de negocio	Enviar y recibir el esquema de polarizaciones
Objetivo	Alice debe mandar su esquema de polarizaciones a Bob y Bob debe guardarlo de manera local
Descripción	1-Alice abre un canal de comunicación RMI con Bob, por el cual enviara su esquema de polarizaciones 2-Bob recibe el esquema usando el mismo canal y lo guarda en un arreglo local dentro de la clase “esquema”
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.6. Descripción casos de uso Enviar y recibir esquema

4.2.8. Caso de uso: Comparar esquemas de polarizaciones de Alice y Bob

Proceso de negocio	Comparar esquemas de polarizaciones
Objetivo	Comparar el esquema de polarizaciones creado por Alice con el esquema creado por Bob
Descripción	1-Bob guarda el esquema recibido por Alice de manera local en un arreglo 2-Bob compara el esquema creado por si mismo con el esquema recibido por Alice 3-Se crea un nuevo arreglo con las coincidencias en esta comparación 4-Si el fotón polarizado está en la misma base tanto en el esquema de Bob como en el de Alice, se llena el arreglo comparación con un 8 y en caso de ser distintas las bases se llena con el bit de Bob
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.7. Descripción caso de uso Comparar esquemas de polarización

4.2.9. Caso de uso: Generar la Raw Key

Proceso de negocio	Generar la Raw Key
Objetivo	Crear la clave en bruto o Raw Key eliminando todos

	los bits cuya base de polarización fue la misma para Alice y Bob
Descripción	1-Una vez obtenido el arreglo de coincidencias en el caso de uso anterior, se tiene un arreglo con 1,0 y 8 2-Los 8 representan una posición en donde las bases de polarización usadas por Alice y Bob coinciden y los 1 y 0 son los bits donde las bases de Alice y Bob no coinciden y por lo tanto comparten el mismo valor de bit. 3-Se eliminan todos los 8 del arreglo, quedando así la clave en bruto
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.8. Descripción del caso de uso Generar Raw Key

4.2.10. Caso de uso: Generar la Sifted Key

Proceso de negocio	Generar la Sifted Key
Objetivo	Generar la clave depurada
Descripción	1-El arreglo con la Raw key se recorre calculando para cada bit una función la cual le da a cada bit un 50% de probabilidad de ser eliminado 2-Los bits restantes conforman la Sifted Key
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.9. Descripción Caso de uso Generar la Sifted Key

4.2.11. Casos de uso: Transmitir la Sifted Key de Bob

Proceso de negocio	Enviar y Recibir la Sifted Key
Objetivo	Compartir la clave depurada entre Bob y Alice
Descripción	1-La clave depurada Sifted Key generada por bob se guarda y envía a Alice 2-Se abren los canales de comunicación en InterCanalEmi y InterCanalRec 3-Alice recibe la Sifted Key y la guarda localmente 4-Se utiliza esta clave en la comunicación mediante criptografía publica AES
Prioridad	Alta

Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.10. Descripción de los casos de uso Enviar y Recibir Sifted Key

4.2.12. Caso de uso: intercambiar Hashes

Proceso de negocio	Intercambiar tablas de Hashes
Objetivo	Compartir entre Alice y Bob una secuencia Hash de seguridad para la comunicación mediante AES
Descripción	1-El programa Bob prepara el Hash 2-El programa Alice prepara el Hash propio mediante la misma función de manera local 3-Alice y Bob envían sus hashes uno al otro, abriendo una comunicación RMI
Prioridad	Alta
Riesgos	
Posibilidades	
Tiempo de ejecución	30 segundos
Coste de ejecución	

Tabla 4.11. Descripción caso de uso Intercambiar Hashes

4.2.13. Casos de uso: Transmitir mensaje cifrado con AES

Proceso de negocio	Enviar y recibir mensaje cifrado en AES
Objetivo	Enviar un mensaje mediante el algoritmo AES, utilizando la clave creada en el proceso cuántico
Descripción	1-La clave depurada luego de pasar por el proceso de Hash, esta lista para ser usada por un algoritmo de criptografía simétrica 2-Alice cifra un mensaje utilizando la clave depurada en el algoritmo AES y se lo envía por un canal RMI a Bob 3-Bob recibe el mensaje cifrado y lo desenscriptar utilizando el algoritmo AES
Prioridad	Media
Riesgos	
Posibilidades	
Tiempo de ejecución	40 segundos
Coste de ejecución	

Tabla 4.12. Descripción del caso de uso Enviar y recibir mensaje cifrado AES

4.4. DIAGRAMAS DE SECUENCIA

4.4.1. Diagrama de secuencia: Definir variables del sistema

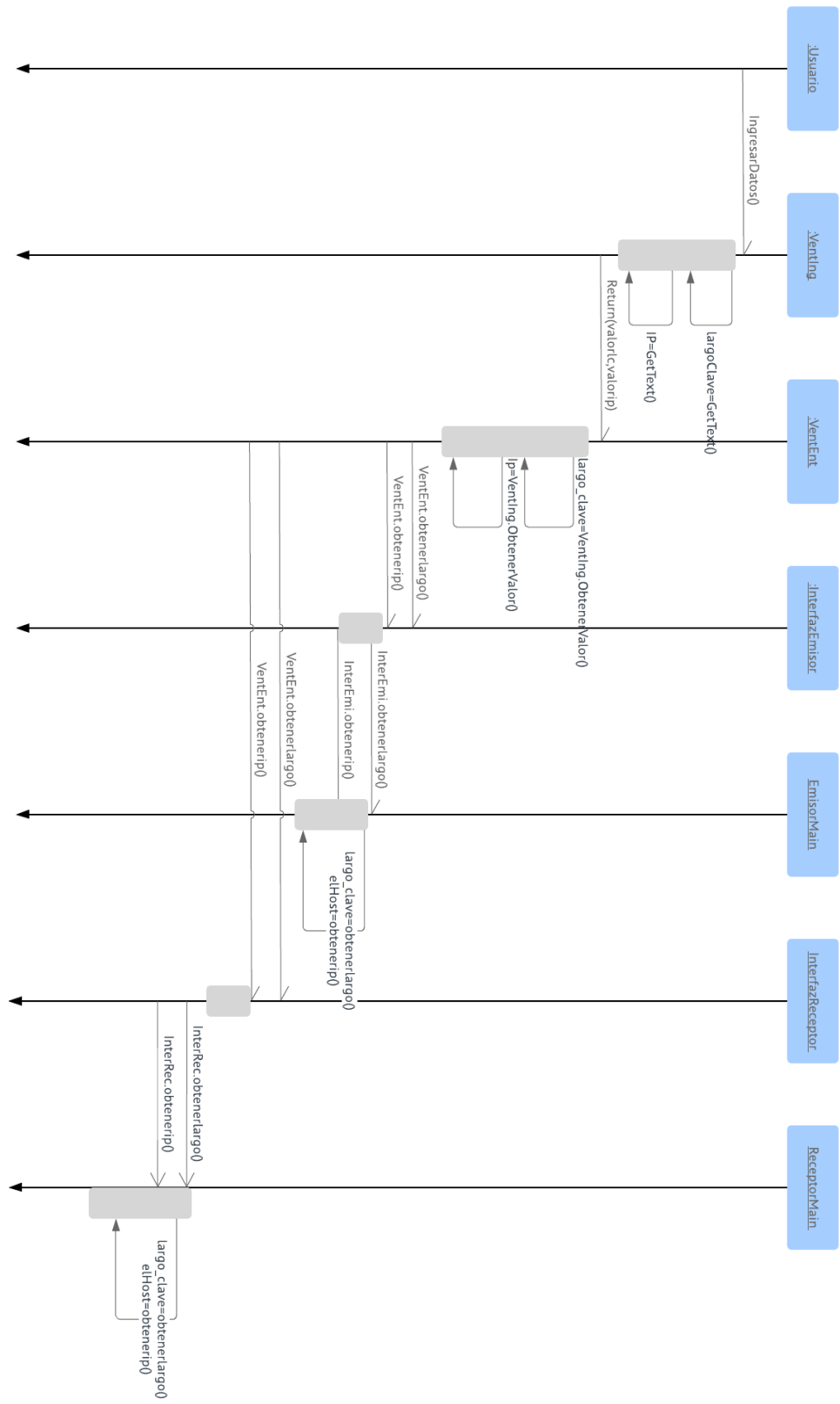


Figura 4.2. Diagrama de secuencia Definir variables del sistema

4.4.2. Diagrama de secuencia: Generar secuencia binaria Alice

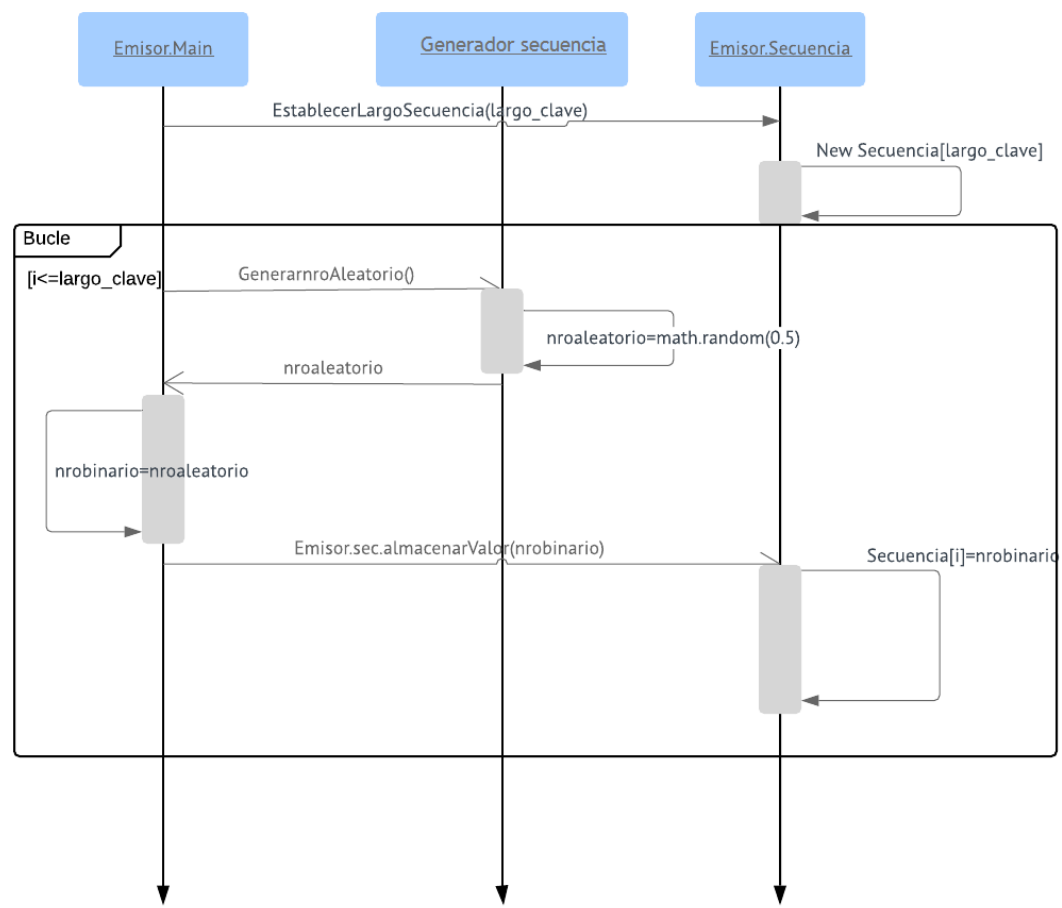


Figura 4.3. Diagrama de secuencia Generar secuencia binaria (Alice)

4.4.3. Diagrama de secuencia: Generar esquema de polarización Alice

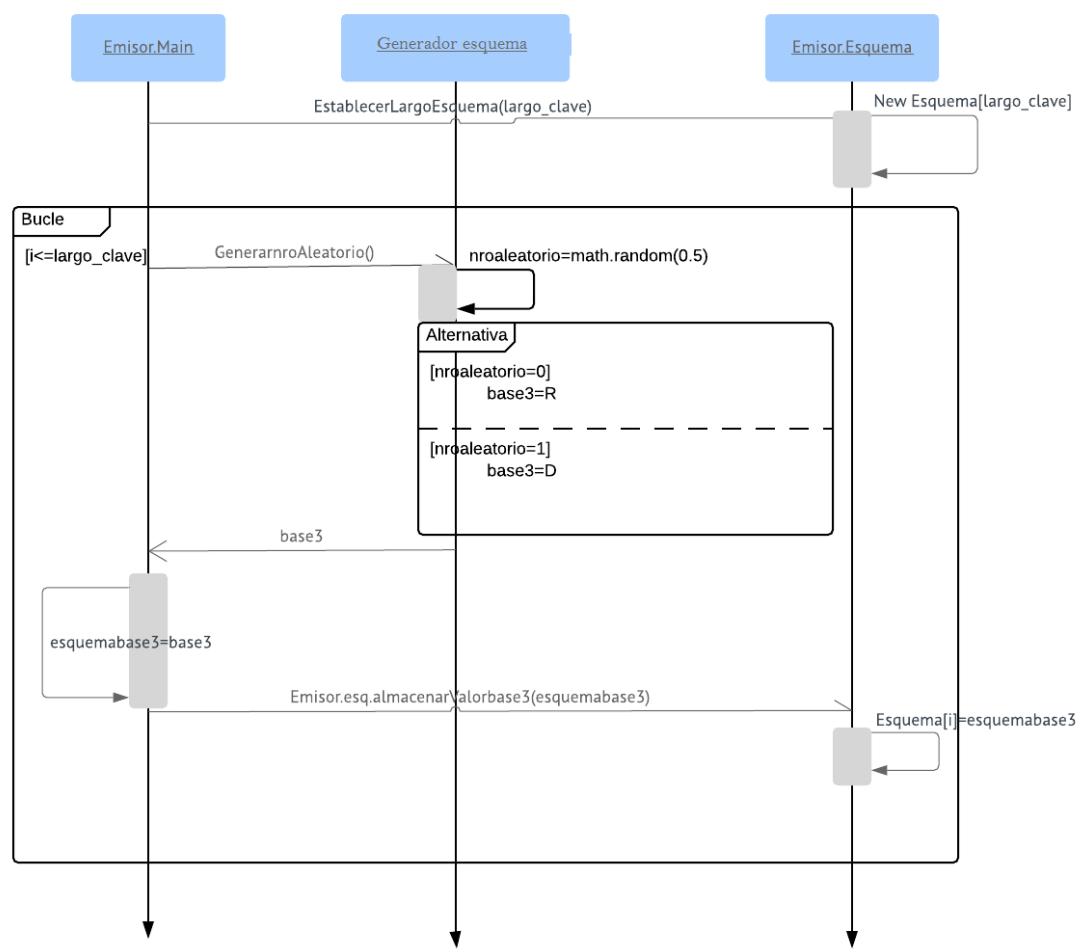


Figura 4.4. Diagrama de secuencia Generar esquema de polarización (Alice)

4.4.4. Diagrama de secuencia: Generar secuencia binaria Bob

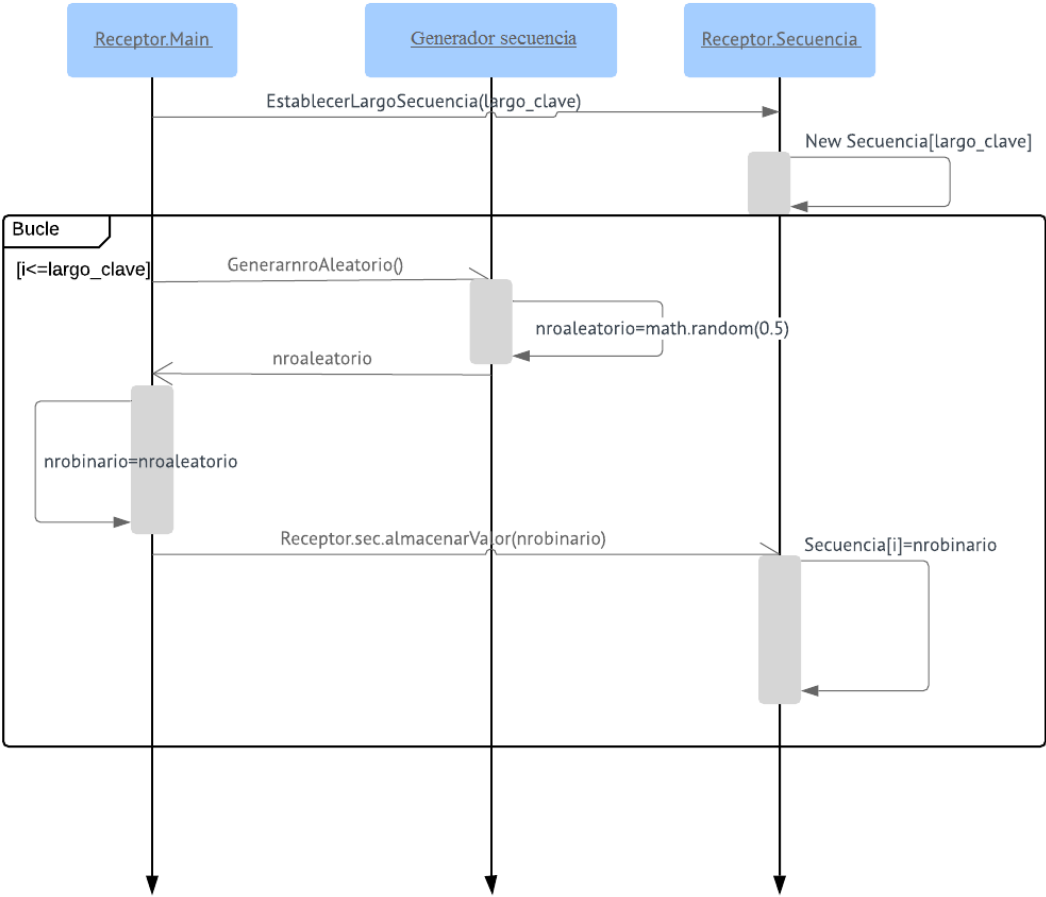


Figura 4.5. Diagrama de secuencia Generar secuencia binaria (Bob)

4.4.5. Diagrama de secuencia: Generar esquema de polarización Bob

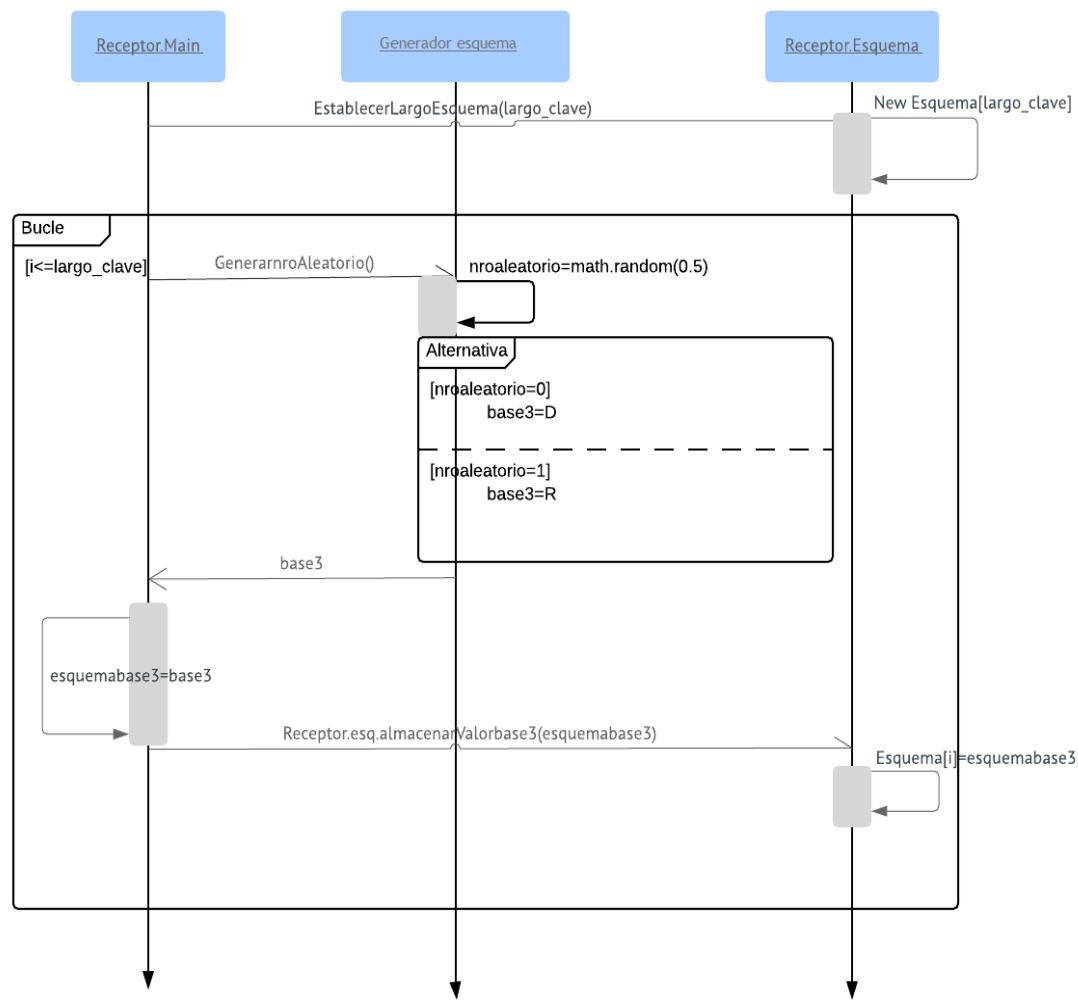


Figura 4.6. Diagrama de secuencia Generar esquema de polarización Bob

4.4.6. Diagrama de secuencia: Transmitir esquema de polarizaciones

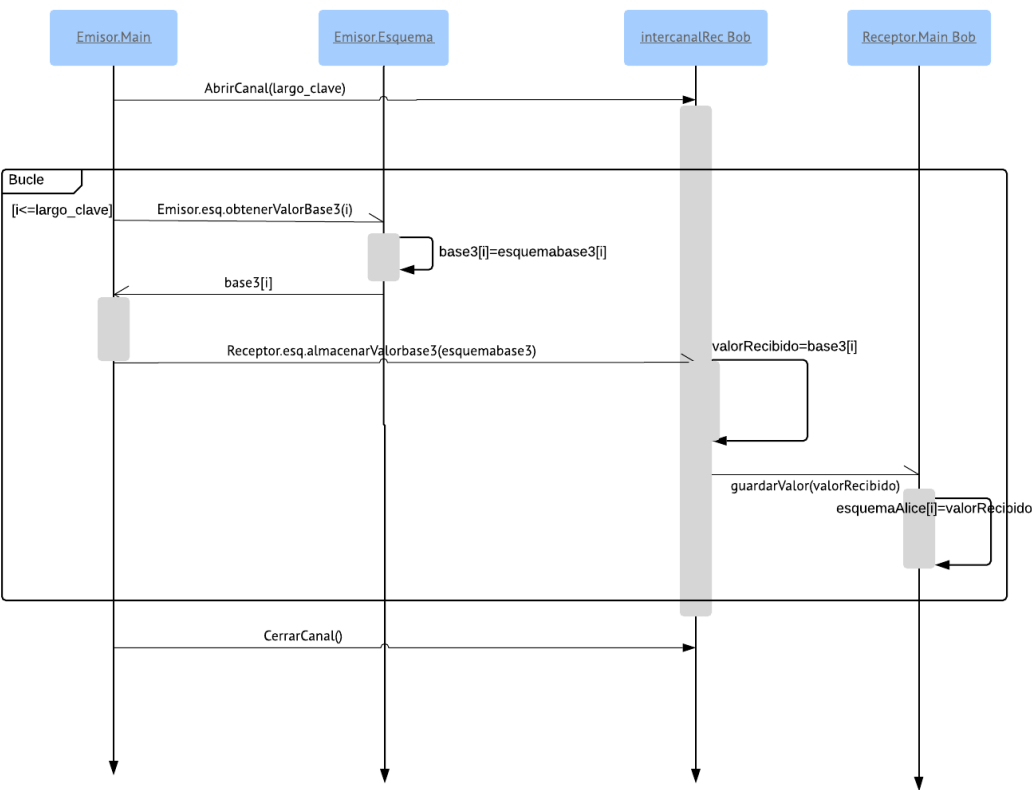


Figura 4.7. Diagrama de secuencia Transmitir esquema de polarizaciones

4.4.7. Diagrama de secuencia: Comparar esquemas de polarización

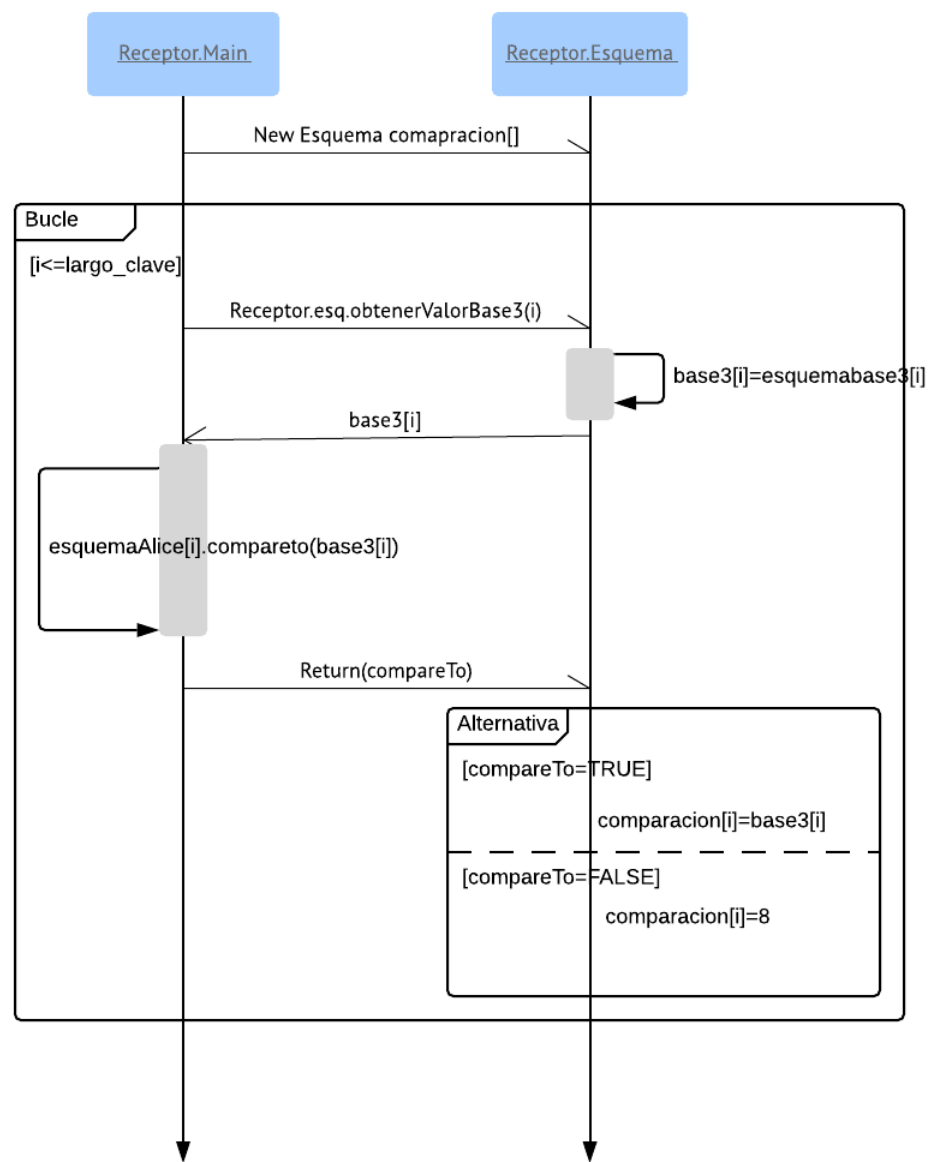


Figura 4.8. Diagrama de secuencia Comparar esquemas de polarización de Alice y Bob

4.4.8. Diagrama de secuencia: Generar la Raw Key

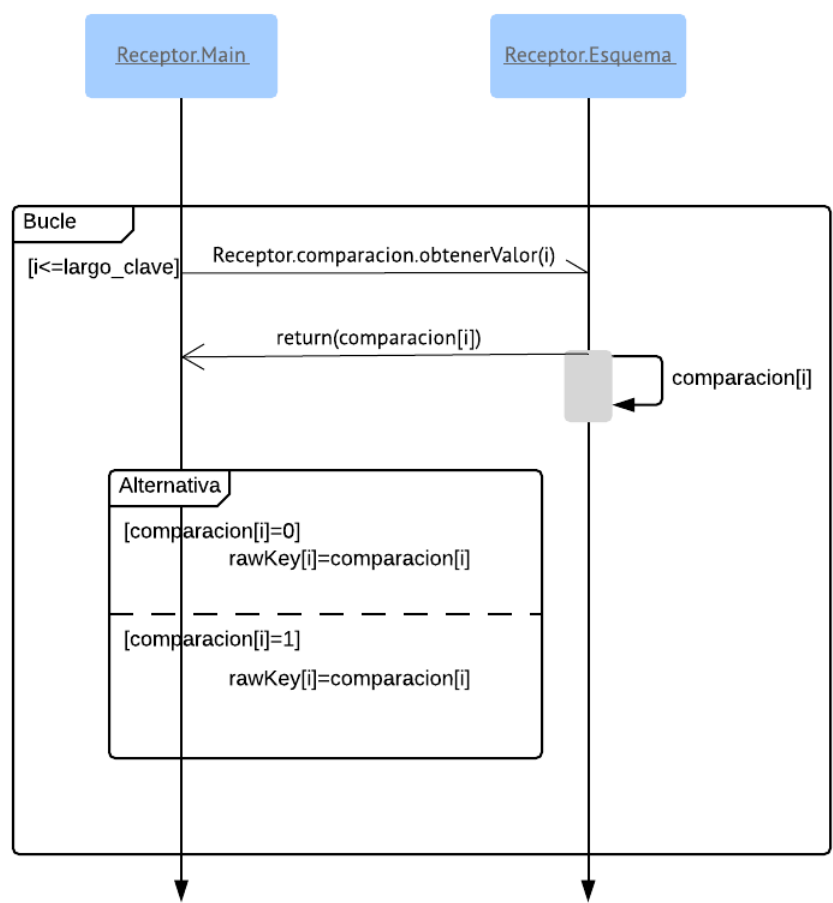


Figura 4.9. Diagrama de secuencia Generar la Raw Key

4.4.9. Diagrama de secuencia: Generar la Sifted Key

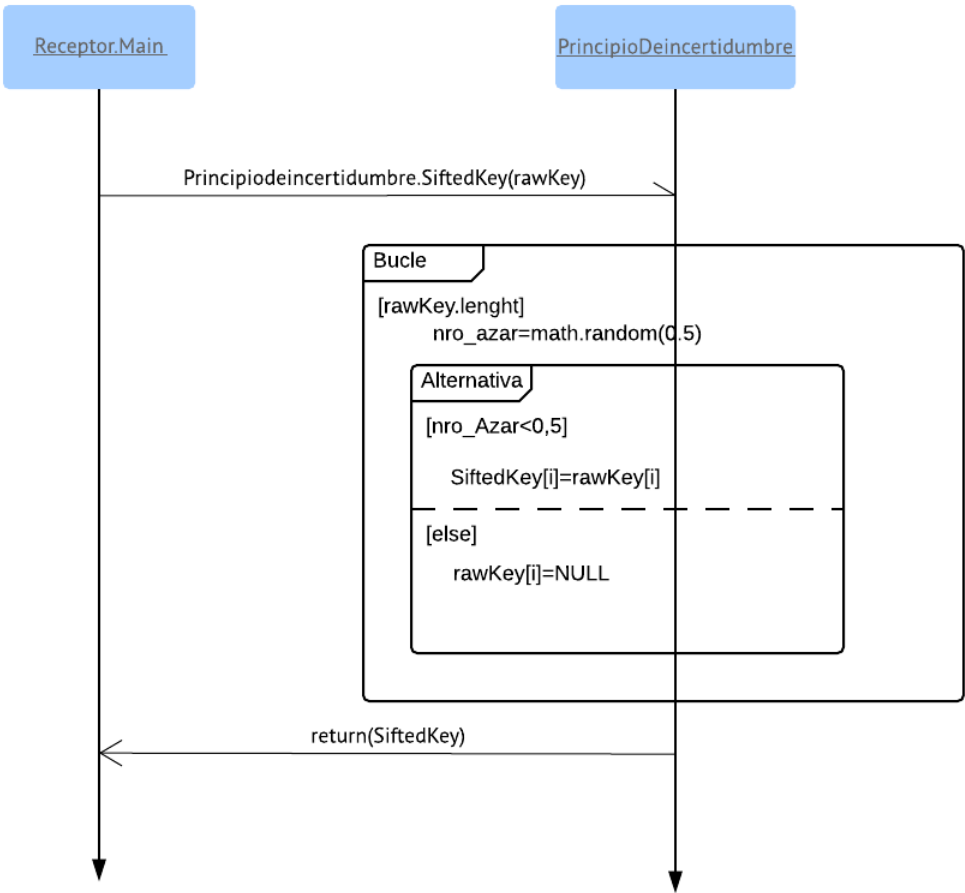


Figura 4.10. Diagrama de secuencia Generar la Sifted Key

4.4.10. Diagrama de secuencia: Transmitir la Sifted Key

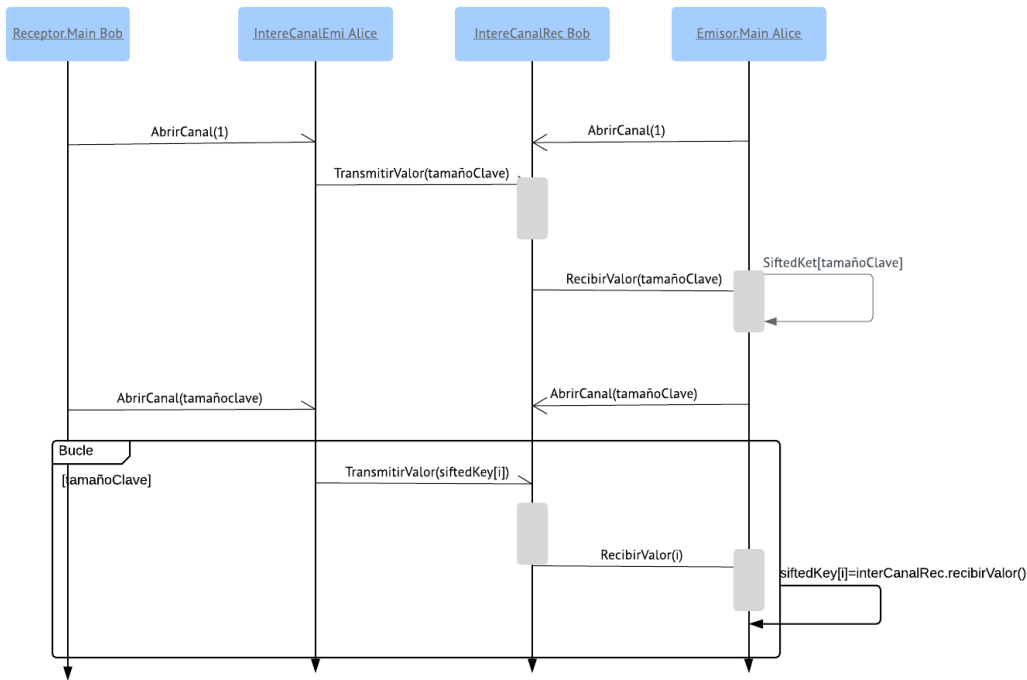


Figura 4.11. Diagrama de secuencia Transmitir la Sifted Key

4.4.11. Diagrama de secuencia: Intercambiar Hashes

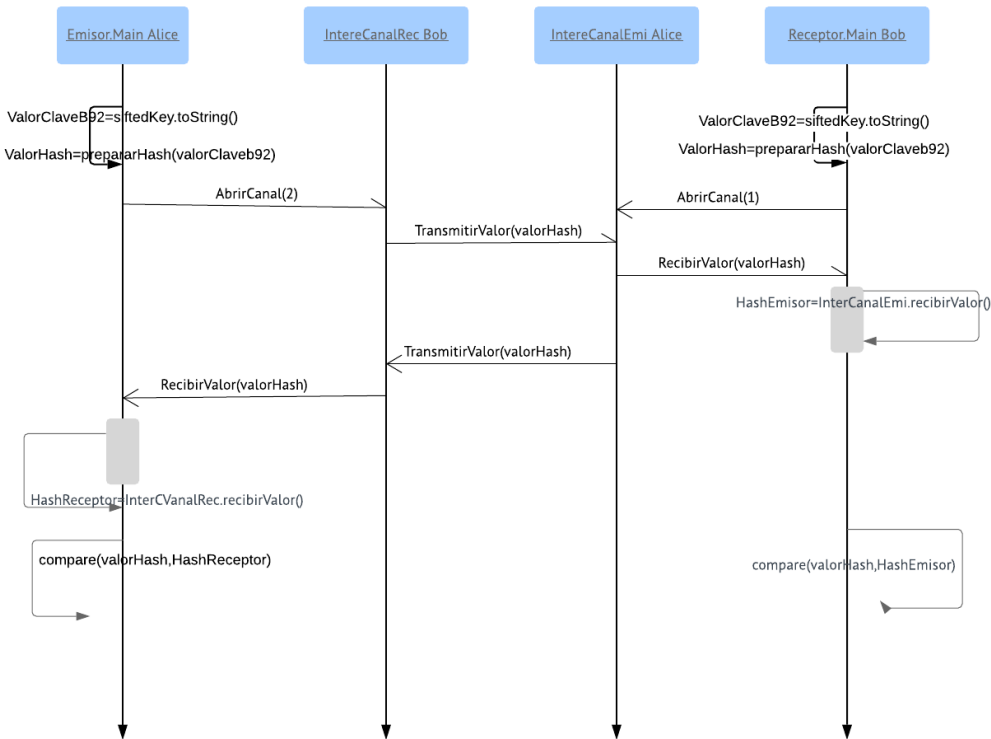


Figura 4.12. Diagrama de secuencia Intercambiar Hashes

4.4.12. Diagrama de secuencia: Enviar mensaje cifrado AES

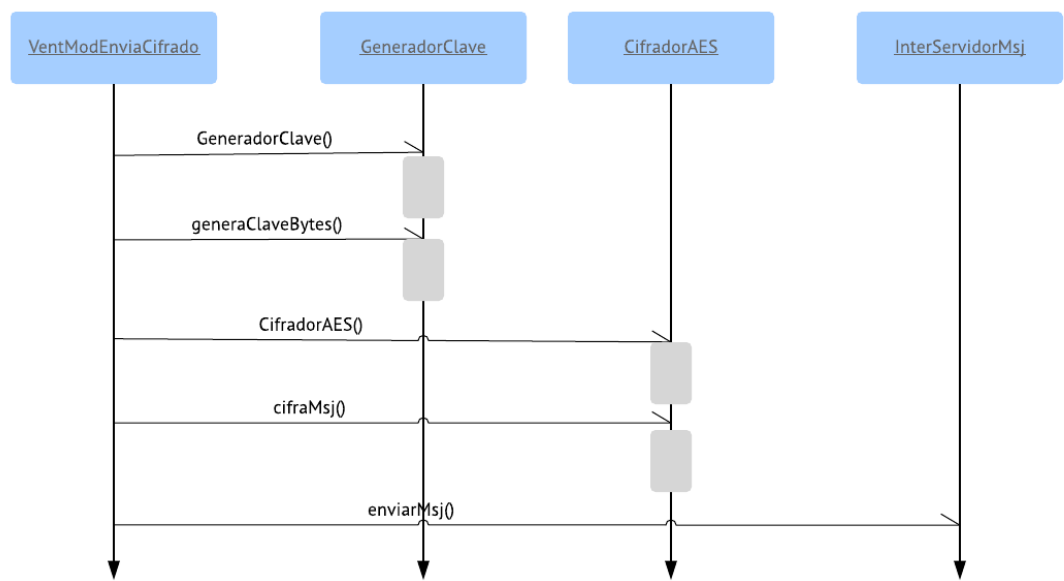


Figura 4.13. Diagrama de secuencia Enviar mensaje cifrado AES

4.4.13. Diagrama de secuencia: Recibir mensaje cifrado AES

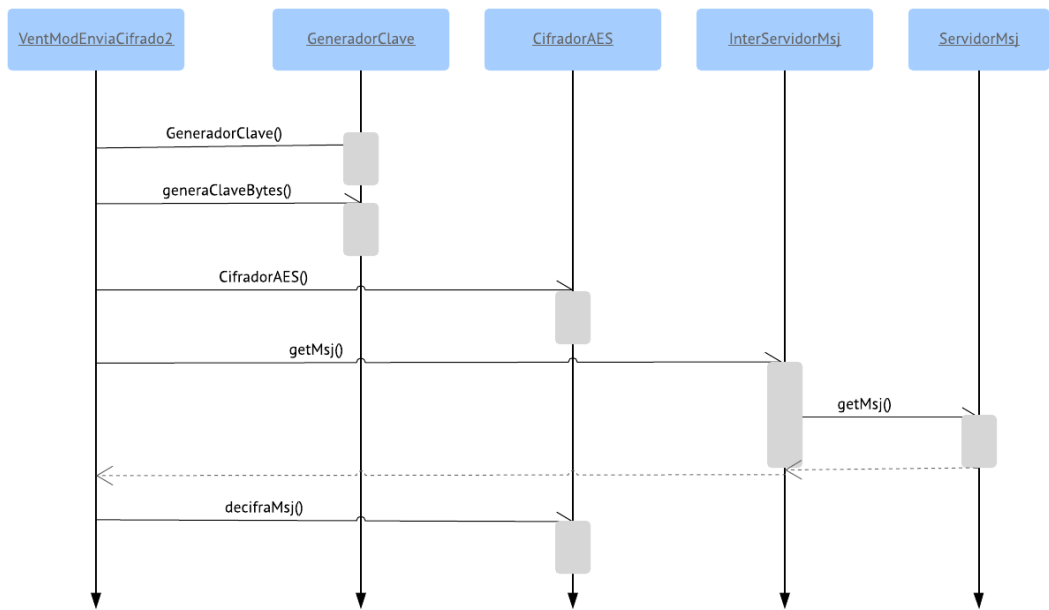


Figura 4.14. Diagrama de secuencia Recibir mensaje cifrado AES

4.5. DIAGRAMA DE CLASES

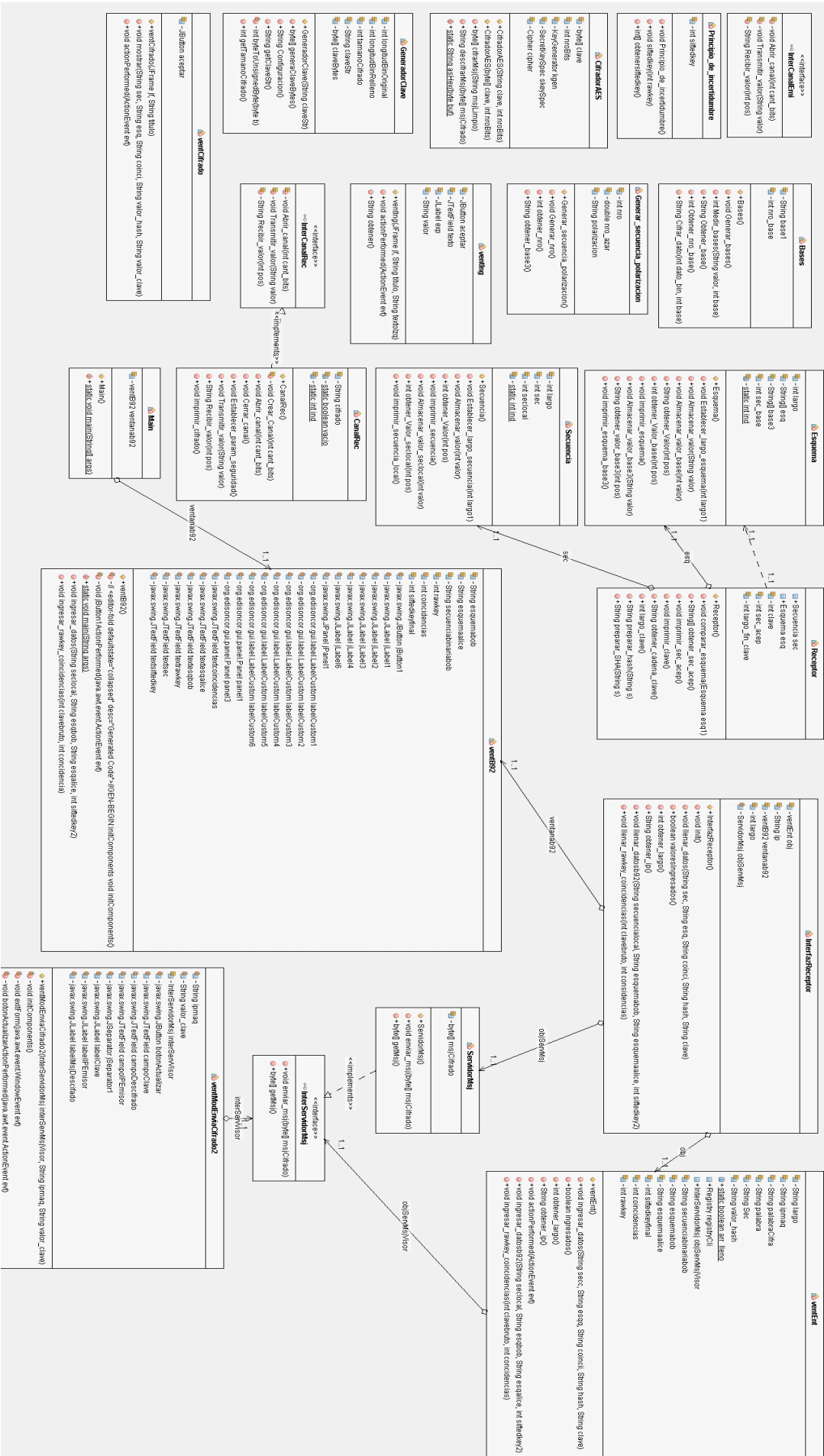


Figura 4.15. Diagrama de clases del sistema

4.6. IMPLEMENTACIÓN

La implementación de los programas Emisor y Receptor, fueron realizadas utilizando el lenguaje de programación Java, tal como se describe en el capítulo “metodología de trabajo y arquitectura del sistema”. El programa en el que está basado este trabajo, es el simulador BB84 desarrollado por Miguel Pinto [3] y optimizado en los trabajos de Roberto Fritis [4] y Patricio Collao [5] . El modulo envía mensaje y recibe mensaje fue desarrollado por Patricio Collao y no se realizó ningún cambio sobre él, solo se cambió la clave de entrada para que funcionara con el protocolo B92 en vez del BB84.

El funcionamiento de la aplicación consta de un modo consola y un modo interfaz gráfica. Todos los resultados aparecen por consola así como al mismo tiempo se despliegan por pantalla mediante el menú “ver cifrado B92”.

A continuación se describirán cada una de las clases que forman el sistema, con el fin de mostrar las interacciones que existen entre ellas, su funcionamiento y como se realizó la simulación del protocolo cuántico. Se utilizó el entorno de desarrollo Netbeans 7.3 y 8.0.

Clases: main Receptor y Emisor

Las clases emisor *main* y receptor *main*, son las clases más importantes de la aplicación. En ellas se llaman a todas las demás clases y es donde se realiza la conexión RMI entre Alice y Bob. En la clase *main* se definen, mediante la interfaz de usuario, los valores para la IP y el largo de la clave, con estos valores se realiza un ciclo *for* en el cual se itera hasta el valor de la clave y en cada iteración se realiza el procedimiento de generación de numero binario y esquema de polarización, guardándose estos valores de manera local en cada uno de los programas emisor y receptor. Luego el programa emisor realiza una conexión RMI para enviar su esquema completo de polarizaciones y ser recibido por el receptor Bob. El receptor luego en su clase *main* llama a las clases encargadas de generar las comparaciones entre los esquemas y la generación de la Raw Key y Sifted Key. Finalmente el receptor realiza una conexión RMI con el emisor para enviarle la Sifted Key final.

```

//Aplicacion B92 ALICE EMISOR
import bb84v10receptor.InterCanalRec;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Main {

    public Main() {}

    public static void main(String[] args) {...}
}

```

Figura 4.16. Código de las clases *main*

Clases: Secuencia

Las clases secuencia emisor y secuencia receptor, son las responsables de almacenar la cadena de bits que se van generando aleatoriamente durante la iteración. En cada programa, tanto emisor como receptor, se genera una cadena de bits completamente distinta y aleatoria, de manera que hay un 50% de probabilidad de que la generación del bit coincida en cada programa, esta información se guarda en la siguiente clase, la clase esquema. La clase consta con métodos para establecer el largo de la cadena, almacenar valor, obtener valor e imprimir la secuencia.

```

class Secuencia{
    int largo;
    int sec[]; // un arreglo de enteros que tendra la secuencia
    static int ind;

    public Secuencia() {...};

    //aqui largo=32
    //sec = new int[32] se crea un arreglo de enteros con 32
    public void Establecer_largo_secuencia(int largo1){...};

    public void Almacenar_valor(int valor) {...};

    public int obtener_valor(int pos) {
        return sec[pos];
    };

    public void imprimir_secuencia(){...};
}

```

Figura 4.17. Código de la clase secuencia

Clases: Esquema

Las clases esquema emisor y esquema receptor, son las encargadas de guardar la cadena de polarización para el fotón simulado. El funcionamiento del protocolo indica que para codificar un bit 1, se debe generar un fotón polarizado en un ángulo de polarización determinado, por ejemplo el bit 1 puede representarse con un fotón polarizado en base diagonal a 45°, por lo tanto esta información se guarda en la clase esquema mediante un arreglo que guarda “D” si la base de polarización es diagonal o una “R” si la base de polarización es rectilínea. Al igual que la clase secuencia, se tienen métodos para definir el tamaño del esquema, almacenar valor, obtener valor e imprimir el esquema.

```
class Esquema{
    int largo;
    String [] esq;//se genera un arreglo de strings que se llama esq
    String [] base3;//arreglo de strings creado por enzo para la base
    int sec_base[];
    static int ind, ind_base, ind2;

    public Esquema() {...};

    public void Establecer_largo_esquema(int largo1) {...};

    public void Almacenar_valor_base3(String valor){...}

    public String obtener_valor_base3(int pos){...}

    public void imprimir_esquema_base3(){...};
}
```

Figura 4.18. Código de las clases esquema

Clase: Generar_secuencia_polarizacion

Esta es la clase encargada de simular el procedimiento de generación cuántico de bits. Aquí es donde se genera el bit y el esquema de polarización para luego ser guardados en las clases secuencia y esquema respectivamente. En el método Generar_nro(), se genera un numero entre 0 y 1, es decir un decimal y existe un 50% de probabilidades de que el numero generado sea mayor a 0,5 o menor a 0,5. Si el numero aleatorio generado es menor a 0,5 entonces la variable nro_aleatorio = 0, si el numero aleatorio generado es mayor a 0,5 entonces la variable nro_aleatorio = 1 y al mismo tiempo si nro_aleatorio = 0 entonces la polarización = R, si nro_aleatorio = 1 entonces la polarización = D. Esto es de vital importancia ya que la única diferencia entre esta clase en el emisor y el receptor es que las polarizaciones son opuestas. En el receptor, si nro_aleatorio = 0 entonces polarización = D, si nro_aleatorio = 1 entonces polarización = R.


```

class Generar_secuencia_polarizacion{
    int nro;
    double nro_azar; // double acepta dos cifras significativas de decimal
    String polarizacion; // creado por enzo para recuperar un caracter R si nro=0 y D si nro=1
    //constructor inicializa la variable nro=-1
    public Generar_secuencia_polarizacion() {nro=-1;};

    //generar_nro() setea el valor de nro dependiendo de nro_azar que dara una probabilidad
    public void Generar_nro() {nro_azar=Math.random();};

    //obtener_nro() retorna el numero guardado en la variable nro y que fue generado
    //probabilidad de que sea un 0 o un 1
    public int obtener_nro() {return nro;};

    //METODO CREADO POR ENZO
    //metodo creado por enzo para obtener la base en el mismo instante en que se crea
    //si nro= 0 entonces base3= R si el nro= 1 entonces base3 = D
    public String obtener_base3() {return nro==0?"R":"D";};
}

```

Figura 4.19. Código clase Generar_secuencia_polarizacion

<pre> public void Generar_nro() { nro_azar = Math.random(); if (nro_azar<0.5){ nro=0; //si nro_azar es menor a 0.5 polarizacion="R"; //crea R } else{ nro=1; // si nro_azar es mayor a 0.5 polarizacion="D"; //crea D } }; </pre>	<pre> public void Generar_nro() { nro_azar = Math.random(); if (nro_azar<0.5){ nro=0; polarizacion="D"; //BOB crea D } else{ nro=1; polarizacion="R"; //BOB crea R } }; </pre>
--	---

Figura 4.20. Método Generar_nro(), programa Emisor y Receptor

Clases: InterCanalEmi, InterCanalRec, CanalEmi, CanalRec

Para la comunicación RMI se deben definir las interfaces tanto en el emisor como en el receptor. En el programa emisor, se tiene la interfaz InterCanalEmi y la clase CanalEmi, en la interfaz se declaran los métodos que pueden ser accedidos de manera remota mediante RMI, luego en CanalEmi están la implementación de dichos métodos. De igual forma en la aplicación receptor, se tienen la interfaz InterCanalRec y CanalRec, en las cuales el funcionamiento es el mismo, en la interfaz se definen unos métodos que son accesibles y visibles para la comunicación RMI y en la clase CanalRec se implementan dichos métodos.

De esta forma cuando el emisor Alice quiere enviar un mensaje al receptor Bob solo debe llamar a la interfaz InterCanalRec que se encuentra definida de manera local como un esqueleto y mediante la conexión RMI se envía la información al receptor remoto y se hace llamada a la clase real CanalRec con sus métodos en la aplicación receptor.

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface InterCanalEmi extends Remote{
    void Abrir_canal(int cant_bits) throws RemoteException;
    void Transmitir_valor(String valor) throws RemoteException;
    String Recibir_valor(int pos) throws RemoteException;
}

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface InterCanalRec extends Remote{
    void Abrir_canal(int cant_bits) throws RemoteException;
    void Transmitir_valor(String valor) throws RemoteException;
    String Recibir_valor(int pos) throws RemoteException;
}

class CanalEmi implements InterCanalEmi{

    String cifrado[];
    static boolean vacio;
    static int ind;

    public CanalEmi() {...};

    void Crear_Canal(int cant_bits) {...};

    //metodo heredado de intercanalemi
    public void Abrir_canal(int cant_bits) {...};

    public void Cerrar_canal() {...};

    public void Establecer_param_seguridad() {...};

    //metodo heredado de intercanalemi
    public synchronized void Transmitir_valor(String valor) {...};

    //metodo heredado de intercanalemi
    public synchronized String Recibir_valor(int pos) {...};

    public void imprimir_cifrado() {...};
}
```

Figura 4.21. Código de las clases InterCanalEmi InterCanalRec y CanalEmi

Clases: VentEnt VentIng VentB92

La ventana principal con la que se abre la aplicación es la clase VentEnt, en la cual se tiene un menú superior con las opciones: Datos, Ver, Cifrado, Palabra, Extensiones, Créditos. En la opción Datos, se ingresa el tamaño de la clave en bits y la dirección IP

de la máquina. En la opción Ver, se muestran por pantalla el largo de la clave en bits ingresado previamente y la dirección IP de la máquina. La opción Cifrado abre un menú en el cual aparecen las opciones: Iniciar, ver cifrado BB84 y ver cifrado B92. La opción Palabra, permite ingresar una palabra y cifrarla. La opción extensiones permite enviar un mensaje cifrado desde Alice a Bob utilizando la clave generada por el protocolo cuántico y haciendo uso del protocolo de criptografía simétrico AES.

```
public class ventEnt extends JFrame implements ActionListener{

    String largo;
    String ipmaq;
    String palabraCifra;
    String palabra;
    String Sec[], Esq[], Cifr[], Coinci[];
    String valor_hash, valor_clave, clavefinal;
    public static boolean arr_lleno, cifrar;
    Registry registryCli;
    InterServidorMsj elModReceptor;
    String secuenciaalice[];
    String esquemaalice[];
    int siftedkeyfinal[];

    public void ingresar_datos(String secc[], String esqq[], String cifra[])

    public boolean ingresados() {...}

    public boolean cifrado_listo() {...}

    public int obtener_largo() {...}

    public String obtener_ip() {...}

    public ventEnt() {...}

    @Override
    public void actionPerformed(ActionEvent evt) {...}
```

Figura 4.22. Código de la clase VentEnt.

```
class ventIng extends JDialog implements ActionListener{

    JButton aceptar;
    JTextField texto;
    JLabel exp;
    String valor;

    public ventIng(JFrame jf, String titulo, String textoIzq) {...}

    public void actionPerformed(ActionEvent evt) {...}

    public String obtener() {...}
```

Figura 4.23. Código de la clase VentIng.

La clase VentB92 es la que muestra los resultados de manera amigable al usuario. Se abre al presionar el botón “ver cifrado B92” en el menú principal y permite imprimir los datos por consola.

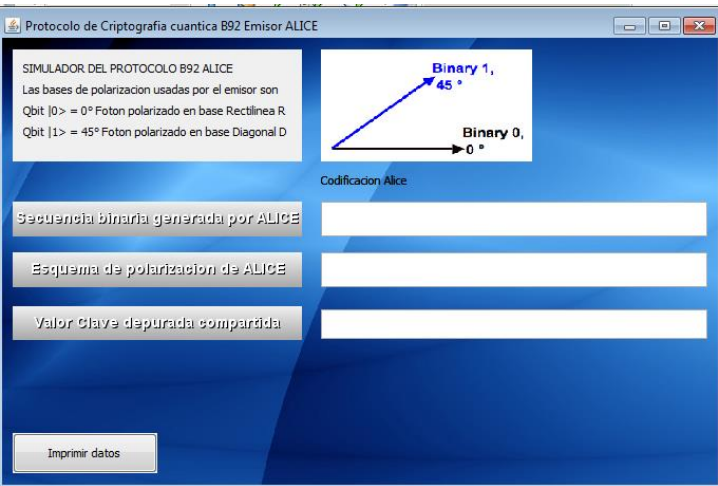


Figura 4.24. Clase VentB92, muestra los resultados por pantalla.

Clase receptor: Principio_de_incertidumbre

La clase Principio_de_incertidumbre es una clase que solo se implementa en el programa receptor, debido a que es el quien debe generar la Sifted Key. En la clase main del receptor, se genera la comparación entre los esquemas de Alice y Bob y de acuerdo a esta comparación se genera una clave en bruto “Raw Key”, la cual es el parámetro de entrada para la clase Principio_de_incertidumbre. El objetivo es, una vez recibida la clave en bruto, calcular con un 50% de probabilidad la eliminación o conservación del bit. Mediante el método SiftedKey(), la clase va eliminando bits de la Raw Key de manera que la clave final será de una tamaño significativamente menor que la original. La clase cuenta con un método para devolver la Sifted Key al programa *main*.

```
public class Principio_de_incertidumbre {
    int siftedkey[];

    public void Principio_de_incertidumbre() {...};

    // ESTA CLASE POSEE 2 METODOS , UNO RECIBE UN ARREGLO QU
    //Y CALCULA CON UN 50% DE PROBABILIDAD SI EL BIT ES DETE
    //LUEGO RETORNA EL ARREGLO RESULTANTE EN LA VARIABLE SIF

    public void siftedkey( int rawkey[]) {...};

    public int[] obtenersiftedkey() {...};
}
```

Figura 4.25. Código de la clase Principio_de_incertidumbre

4.7. EJECUCIÓN DE LA APLICACIÓN

Para ejecutar la aplicación primero se debe iniciar el registro RMI, para guardar nombres de servidores. Esto se ve en la **Figura 4.26**, donde se ejecuta el comando *rmiregistry* en la consola de Windows.

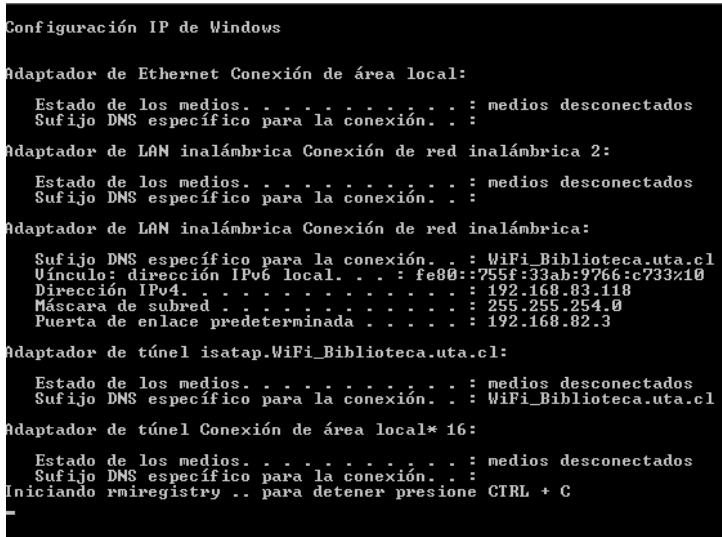


Figura 4.26. Ejecución del comando *rmiregistry* para iniciar el proceso RMI.

Luego se ejecutan los dos programas Emisor y Receptor. Para cada uno se ingresan los datos Largo Clave y Dirección IP del Cliente, para ambas aplicaciones el largo clave debe ser el mismo y la IP es la correspondiente a la máquina.

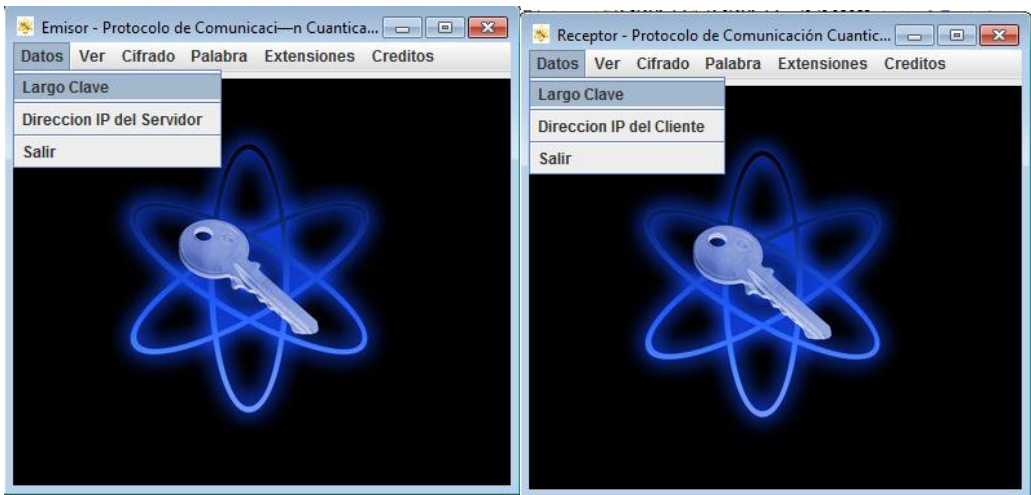


Figura 4.27. Ventana principal para los programas Emisor y receptor.

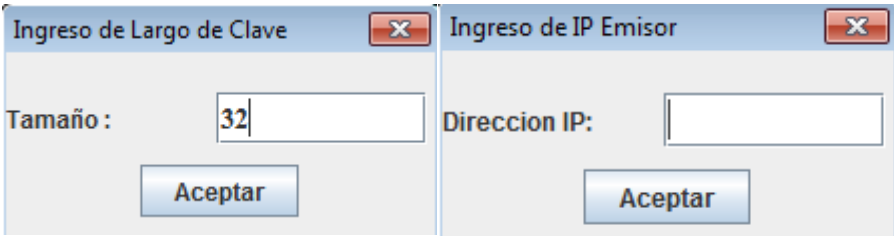


Figura 4.28. Ingreso de datos en el menú principal.

Luego de definir las variables iniciales, se presiona en el menú principal la opción Cifrado->iniciar. La aplicación emisor es la que inicia el cifrado y al terminar se despliega un mensaje por pantalla indicando el término del cifrado como se ven en la **Figura 4.29**.

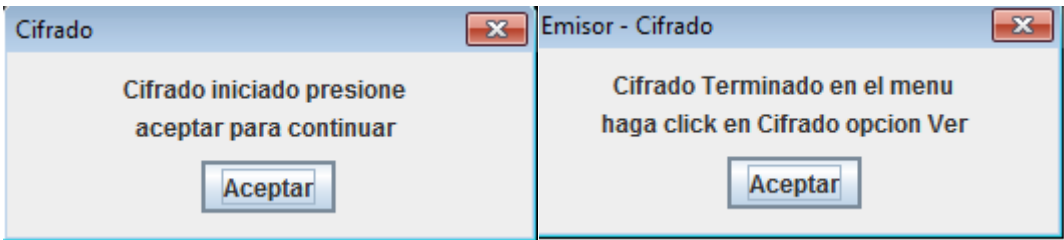


Figura 4.29. Diálogos de inicio de cifrado y termino de cifrado en la aplicación Emisor.

Luego de completar el cifrado, las dos aplicaciones ya tienen compartida la clave depurada Sifted Key, por lo tanto se pueden ver los resultados en la **Figura 4.30**, modo consola y en la **Figura 4.31**, modo interfaz gráfica.

```
Esquema de polarizaciones generado por ALICE:D D D D R R D D R R R R R R D R R D R D D R D D R R R R R
La secuencia binaria de unos y ceros de ALICE:1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0

el tamaño de la clave recibida de BOB es 14
La sifted key compartida entre ALICE y BOB es:110000000010101
```

Figura 4.30. Salida del programa emisor Alice en modo consola

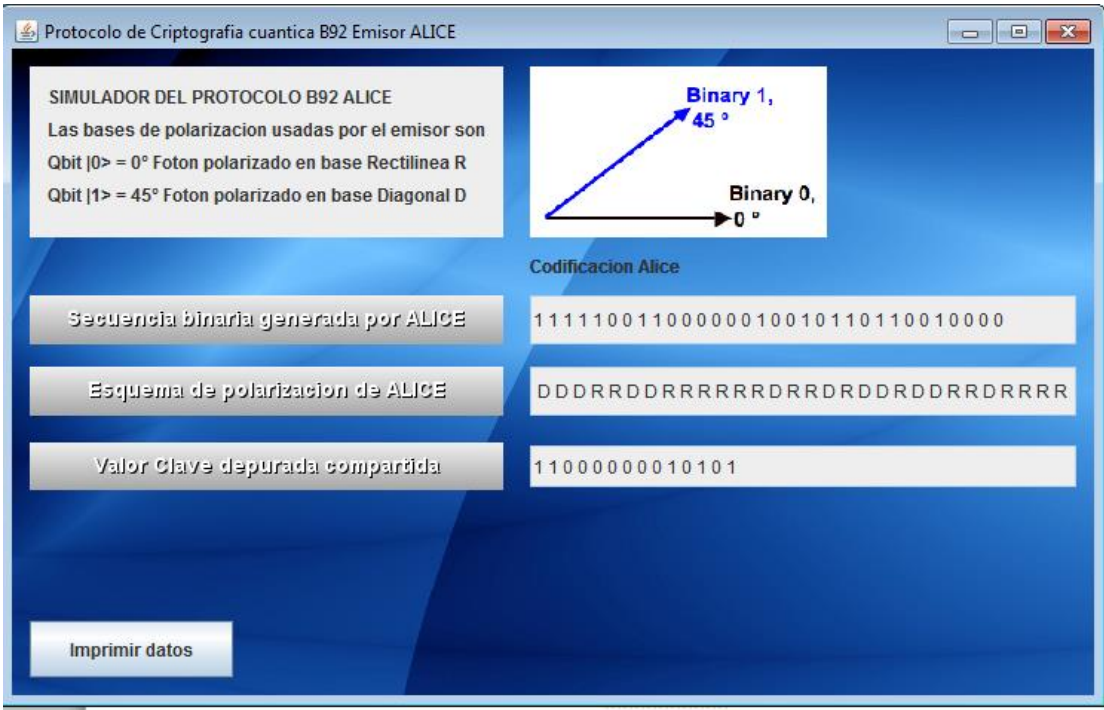


Figura 4.31. Salida del programa emisor Alice en modo interfaz grafica

```
Esquema de polarizaciones generado por BOB:R D R D R D D D D D D D D D D D D R D R R D R D D R D D
La secuencia binaria de unos y ceros de BOB:1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 0

Comienza la comparacion de polarizacioens
el esquema generado por BOB es R D R D R D D D D D D D D D D D D R D R R D R R D D R D R D D
el esquema recibido por ALICE es D D D D R R D D R R R R R R D R R D R D D R D D R R D R R R R
18181008800000080010110110010800

La raw key es
11100000000001011011001000
La sifted key compartida entre ALICE y BOB es 11000000010101
Se esta enviando el tamaño de la clave depurada 14
```

Figura 4.32. Salida del programa Receptor Bob en modo consola

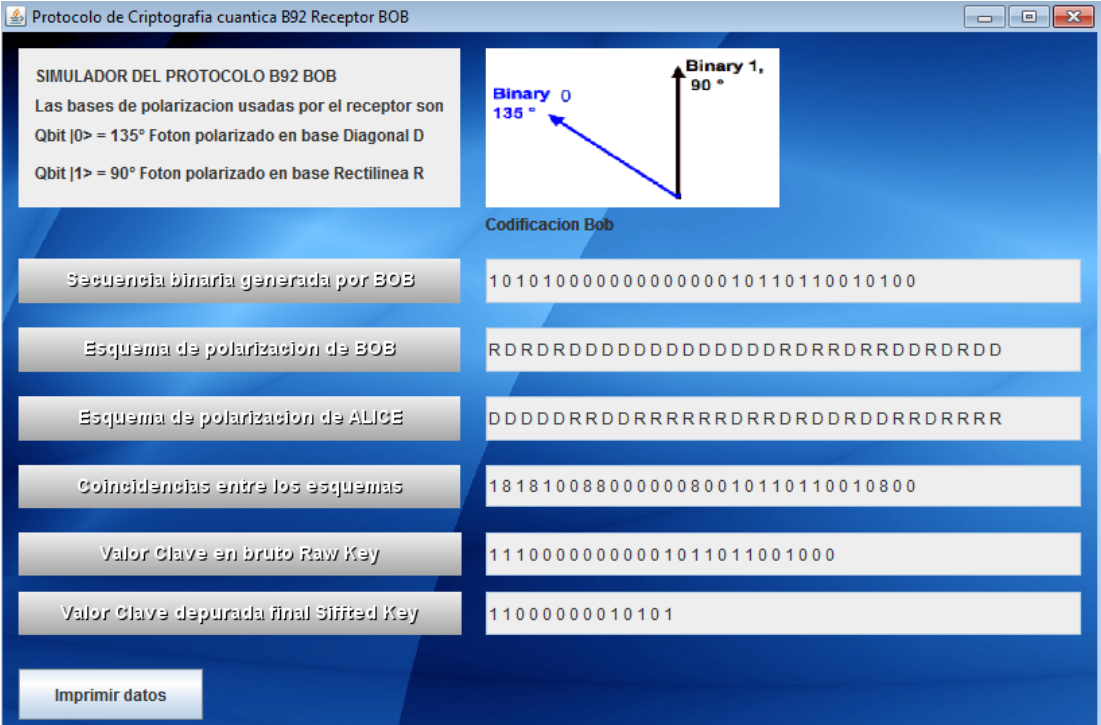


Figura 4.33. Salida de la aplicación Receptor Bob en modo interfaz grafica

4.8. CONCLUSIONES DEL CAPITULO

En este capítulo, se han desarrollado los procedimientos de ingeniería de software necesarios para modelar y construir la aplicación que simula el funcionamiento del protocolo de criptografía cuántica B92. Siguiendo el modelo de trabajo definido en el los objetivos y en el capítulo 3, se procedió al Análisis Diseño implementación y Pruebas del sistema.

En el análisis se investigó y encontró una solución al problema planteado, estudiando los fundamentos matemáticos y físicos del entorno. También se definieron los requerimientos funcionales y no funcionales. En la etapa de diseño y modelamiento se realizaron los trabajos UML, para los cuales se definieron una serie de casos de uso del sistema y para cada uno de ellos se definió su respectivo diagrama de secuencia. En la etapa de implementación se diseñó el diagrama de clases y las distintas clases que conforman los programas. Se realizaron las pruebas preliminares de funcionamiento de manera local, en consola y en interfaz gráfica.

En el siguiente capítulo se avanza en la metodología de trabajo correspondiente a la etapa de pruebas, para lo cual se realizara la conexión en distribuido de los programas y se probaran con distintos tamaños de claves.

CAPÍTULO 5: RESULTADOS

Las pruebas realizadas de la aplicación, constan en una serie de escenarios, en los cuales se midió el funcionamiento de la aplicación. Para cada escenario, se preparó una serie de pruebas con distintos largos de clave en bits. También, para cada escenario se utilizó una topología de red distinta, para mostrar el funcionamiento distribuido del sistema. Se realizaron dos escenarios para los cuales en cada uno se realizaron 10 ejecuciones completas del algoritmo para cada tamaño de clave inicial, de las cuales se calculó un promedio en cantidad de bits perdidos, tiempo empleado en la ejecución y distintos tamaños de bits. Considerando 9 tamaños de bits distintos como muestra la **Tabla 5.4**, se realizó un total de 90 pruebas para cada escenario, considerando dos escenarios, el programa se ejecutó 180 veces en las maquinas correspondientes.

En siguiente capítulo está dividido en secciones donde se definen los requerimientos de hardware y software necesarios para el funcionamiento de las maquinas utilizadas en las pruebas. Luego se evaluarán los requerimientos funcionales y no funcionales en una tabla de comprobación. Luego se ejecutan las pruebas en los distintos ambientes de caso de prueba y finalmente se concluye con los resultados en comparativa y la introducción al capítulo siguiente.

5.1. REQUERIMIENTOS DE LAS PRUEBAS

Los requerimientos de software y hardware para los equipos utilizados en las pruebas se detallan a continuación:

- Dos computadores portátiles y uno de escritorio
- Un Router con velocidad mínima de 100mbps
- Cables UTP categoría 5
- Sistema operativo Windows 7
- Instalar Java Runtime machine y Java SDK JDK, en todas las maquinas
- Deshabilitar los firewall o habilitar el puerto 1099 en todas las maquinas, ya que este es el puerto que usa RMI

5.2. ESPECIFICACION DE LOS EQUIPOS UTILIZADOS

Equipo 1 Emisor Alice	
Marca modelo	Netbook Packard Bell DOT-S
Procesador	Intel Atom dual core 1.6 GHZ
Memoria	2GB DDR3
Gráficos	Intel Graphics Media Accelerator 3600
Sistema operativo	Windows 7 Profecional

Tabla 5.1. Especificación del equipo 1 Emisor

Equipo 2 Receptor Bob	
Marca modelo	PC Gear
Procesador	AMD
Memoria	4GB DDR3
Gráficos	Radeon Graphics
Sistema operativo	Windows 7 Profesional

Tabla 5.2. Especificación del equipo 2 Receptor

Router	
Marca modelo	TP-LINK
Ethernet	100mbps
Wifi	Inhabilitado
Puertos	6
Sistema operativo	No aplica

Tabla 5.3. Especificación router

5.3. PRUEBAS DE LA APLICACIÓN

Las pruebas se realizaron en dos escenarios distintos. El caso de prueba 1 consiste en la realización de las pruebas localmente, utilizando el puerto 127.0.0.0 *Localhost*. El caso de prueba 2 es la ejecución de los programas dentro de una red LAN distribuida.

En la **Tabla 5.4** se muestran los tamaños en bits de la clave inicial para las pruebas. Como se puede apreciar las pruebas solo tienen contemplado hasta un máximo de 2048 bits que es lo que puede resistir el hardware con el que se trabaja (un netbook y un pc de sobremesa).

8 bits
16 bits
32 bits
64 bits
128 bits
256 bits
512 bits
1024 bits
2048 bits

Tabla 5.4. Largo de las claves

5.3.1. Caso de prueba 1

El primer caso de prueba consiste en la ejecución del programa de manera local mediante el puerto *Localhost* 127.0.0.1. Tanto el programa emisor como el programa receptor, se ejecutan simultáneamente en una sola máquina.

Para los 9 tamaños de la clave inicial, se realizaron 10 pruebas y se calculó el promedio de estas. De manera que el valor obtenido puede variar debido a la aleatoriedad de la eliminación de bits. Es posible que en una clave de 32 bits se conserven 20 bits del total y en otra ejecución se conserven 8 bits del total. De esta forma se realizaron un total de 90 ejecuciones del programa para el caso de prueba 1.

Ejecución de la simulación

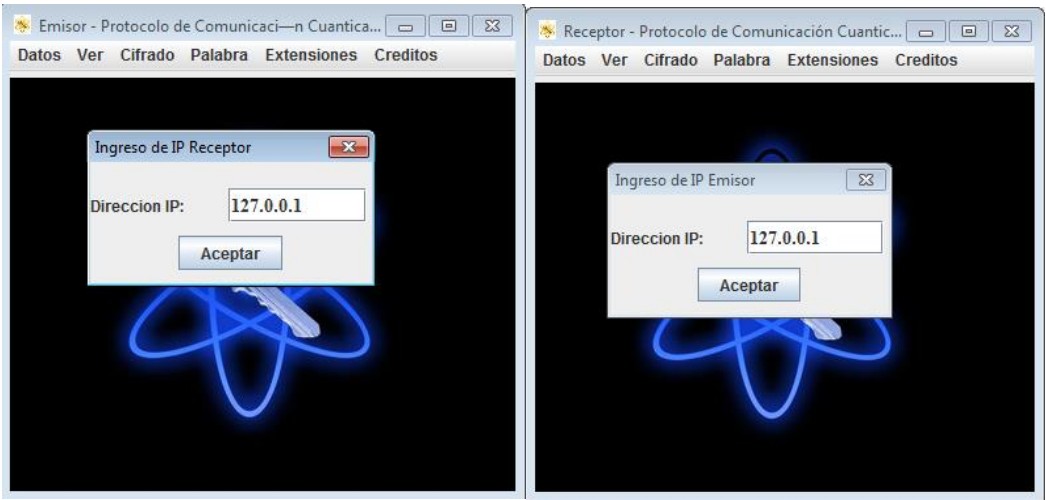


Figura 5.1. Ejecución de los programas en *Localhost*.

Resultados Obtenidos

Resultados Caso de prueba 1				
Largo inicial bits	Largo final bits	Bits perdidos	Porcentaje de bit clave final	Tiempo segundos
8	2	6	25%	8
16	4	12	25%	12
32	7	21	21%	18
64	17	55	26%	33
128	35	92	27%	62
256	65	190	25%	119
512	125	394	24%	232
1024	249	780	24%	444
2048	510	1568	25%	877

Tabla 5.5. Resultados del caso de prueba 1

Análisis de resultados caso de prueba 1

Los resultados de la **Tabla 5.5** muestran las variables: Largo inicial, Largo final, Bits perdidos, Porcentaje de bits clave final y Tiempo en segundos. Las variables largo inicial y final, representan el tamaño en bits de la clave inicial ingresada por el usuario y el tamaño en bits de la clave final generada por el algoritmo. Los bits perdidos y el % de largo representan cuanto de la clave inicial fue eliminado y que porción de bits será utilizada en la clave final. La variable tiempo representa el tiempo en segundos que demora el algoritmo para cada largo de clave de entrada.

Lo primero que se puede apreciar es el orden en el que van incrementando los valores, para una clave de 8 bits su tamaño final es 2, para una clave inicial de 16 bits, su tamaño final es de 8, lo que refleja que al aumentar el doble el tamaño de la clave inicial, también aumenta el doble el tamaño de la clave final, de manera que al aumentar el tamaño inicial, el tamaño final aumenta en la misma proporción. Lo mismo ocurre para la variable “tiempo de ejecución”, la cual para una clave de 1024 bits demora aproximadamente 7 minutos y para la clave de 2048 bits demora 14 minutos y medio.

La variable % de bits finales, representa el porcentaje de bits que logra sobrevivir todas las eliminaciones de bits que dan la seguridad al algoritmo, además este recorte de bits da seguridad a la clave, debido a que se eliminan bits de manera aleatoria y un intruso no podría saber que bits serán eliminados y cuáles no. Recordar que del 100% de los bits iniciales se elimina la mitad debido a la elección del filtro polarizador de detección, luego de este 50% restante se vuelve a eliminar la mitad debido al principio de incertidumbre, quedando un total de solo el 25% de bits para la clave final. EL % promedio de todas las ejecuciones ronda el 24,5%, lo que se aproxima al 25% teórico esperado.

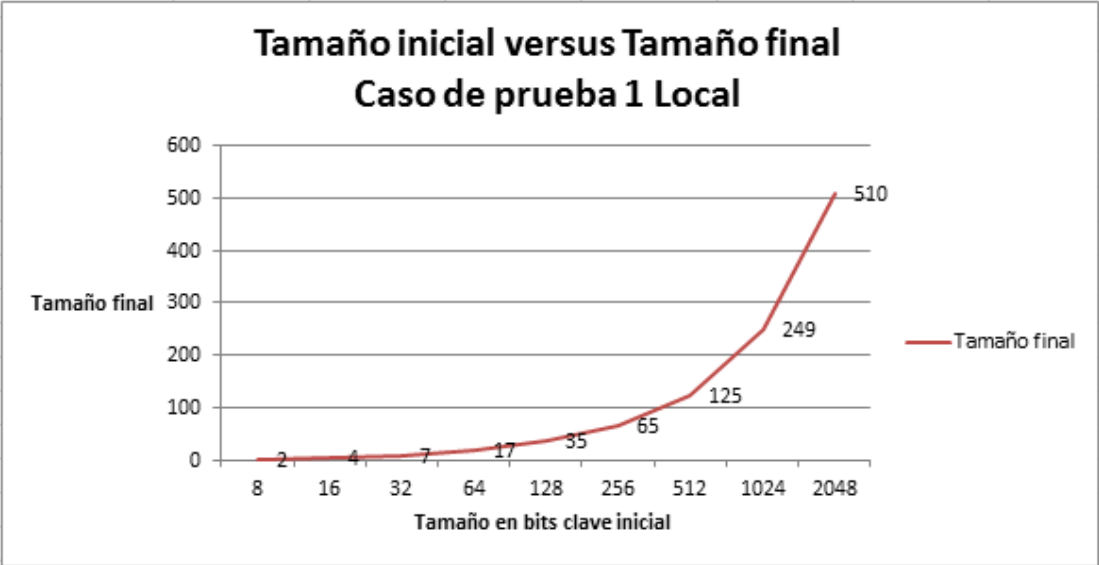


Figura 5.2. Grafico Tamaño final de la clave

El grafico de la **Figura 5.2**, muestra el tamaño inicial de la clave versus el tamaño final obtenido por el algoritmo B92 en un ambiente local. El resultado es el promedio de 10 ejecuciones del algoritmo.

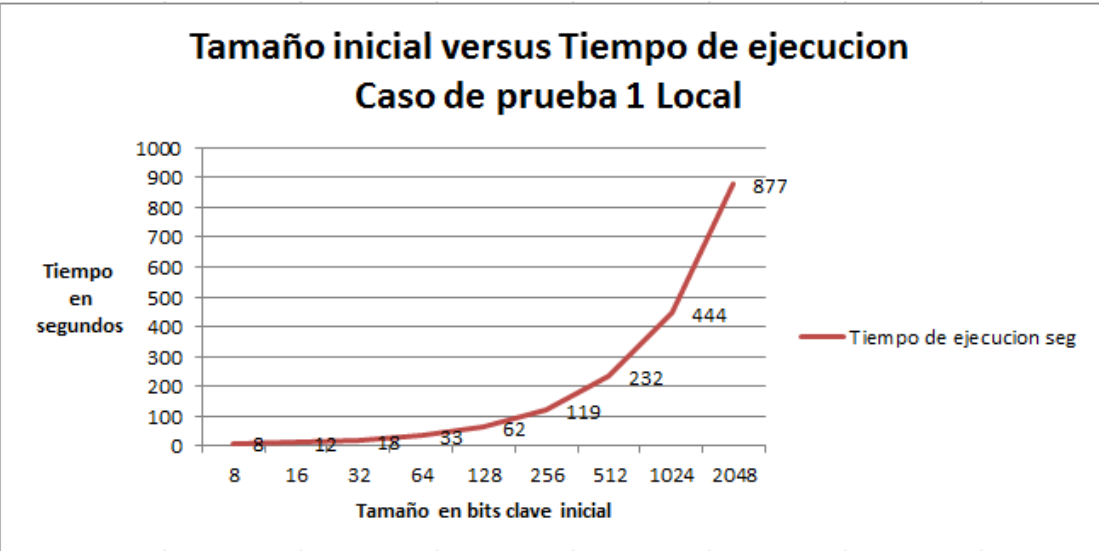


Figura 5.3. Grafito tiempo de ejecución

El grafico de la **Figura 5.3**, muestra el tiempo de ejecución en segundos, de un promedio entre 10 ejecuciones del algoritmo. Al igual que el grafico de tamaño final, el tiempo de ejecución aumenta progresivamente de manera exponencial.

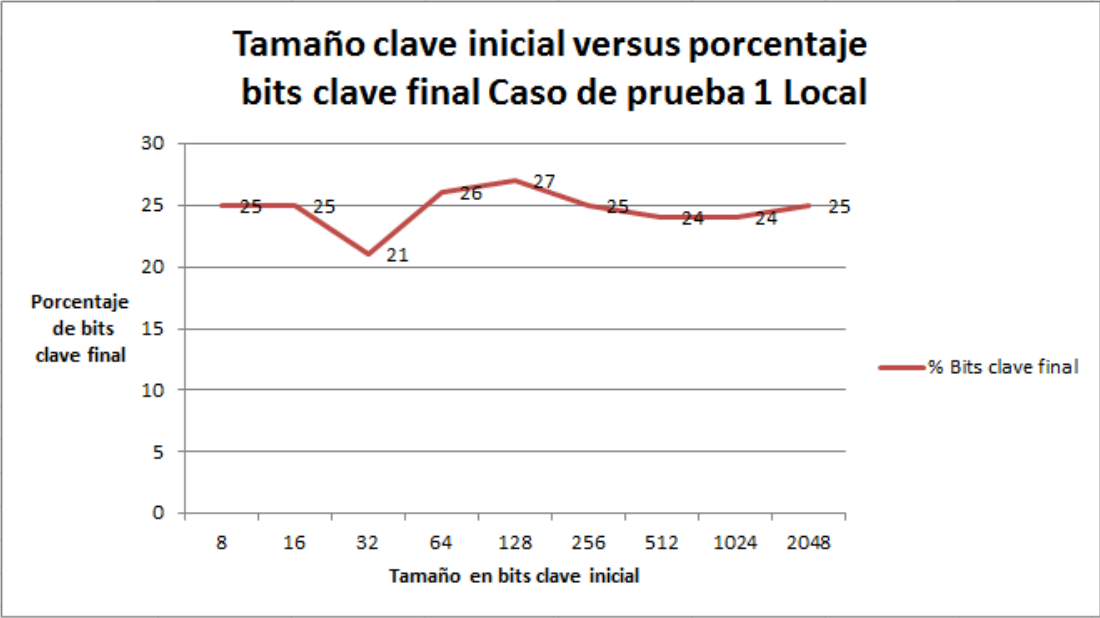


Figura 5.4. Porcentaje de bits que conforman la clave final

El grafico de la **Figura 5.4**, muestra el promedio de bits que logran conformar la clave final. Este valor es un porcentaje que ronda el 20% y 27% con un promedio del 24,5%.

5.3.2. Caso de prueba 2

Para ejecutar los programas en una red distribuida, primero se deben copiar todos los archivos en las maquinas correspondientes, compilar y ejecutar el código. En el caso del equipo emisor, solo se debe ejecutar el programa Emisor y en el caso del equipo receptor solo se ejecuta el programa Receptor. Las IP se ingresan manualmente, de manera que en el programa Emisor se ingresa la IP del receptor al cual serán enviados los datos y en el programa Receptor se ingresa la IP del Emisor. En este caso de prueba se realizaron 10 ejecuciones del programa para cada una de las 9 claves diferentes, con un total de 90 ejecuciones.

Un error se produjo durante la ejecución de los programas de manera distribuida. La conexión RMI estaba utilizando una IP equivocada, esto se debe a que algunas aplicaciones crean redes virtuales y asignan IPs de manera poco transparente. Para solucionar este problema se aisló todo el sistema, sin conexión a internet ni *wifi*, se deshabilitaron las redes que no fueran de Ethernet y solo se permitió el uso de las IP definidas en el trabajo.

Diagrama de red

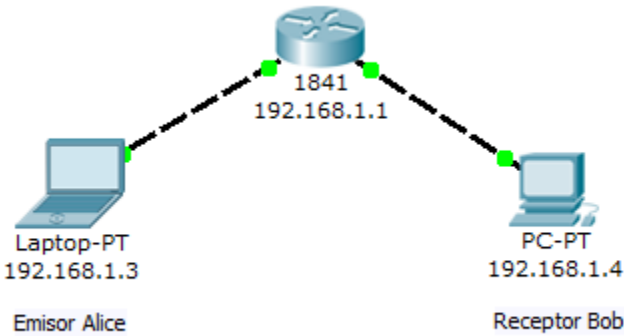


Figura 5.5. Diagrama de red LAN

Caso de prueba 2 Red LAN			
Equipos	Dirección IP	Mascara	Puerta de enlace
Emisor Alice	192.168.1.3	255.255.255.0	192.168.1.1
Receptor Bob	192.168.1.4	255.255.255.0	192.168.1.1
Router	192.168.1.1	255.255.255.0	

Tabla 5.6. Configuración de equipos para el caso de prueba 2

Ejecución de la simulación

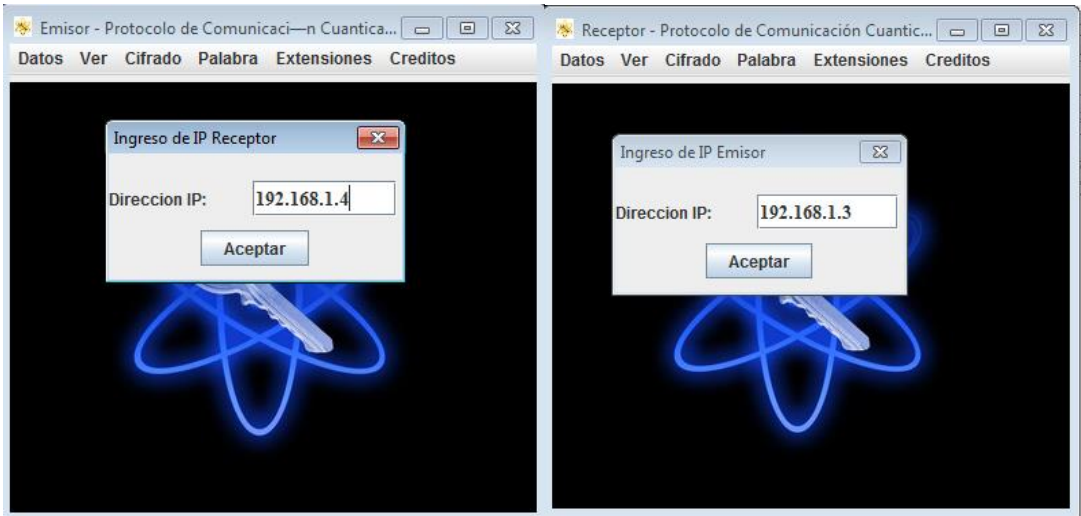


Figura 5.6. Ejecución de los programas en red distribuida

Resultados Obtenidos

Resultados Caso de prueba 2				
Largo inicial bits	Largo final bits	Bits perdidos	Porcentaje de bit clave final	Tiempo en Segundos
8	2	6	25%	8
16	4	12	25%	11
32	8	24	25%	19
64	16	48	25%	32
128	30	98	23%	61
256	62	194	24%	112
512	129	383	25%	229
1024	251	773	24%	456
2048	523	1525	25%	893

Tabla 5.7. Resultados del caso de prueba 2

Análisis de resultados caso de prueba 2

Al ejecutar el algoritmo en una red distribuida, se espera encontrar algún tipo de discrepancia con respecto a la ejecución en forma local. La principal diferencia, seria en los gráficos de tiempo de ejecución, debido a que al estar en una forma local el algoritmo en teoría debería ser más rápido que en un entorno distribuido.

La ejecución del caso de prueba 2 se puede generalizar para cualquier tipo de red distribuida más compleja. Por ejemplo, se podría realizar un tercer caso de prueba donde el emisor y el receptor se encuentren en una red distinta, para este caso, la aproximación de los resultados debería seguir la misma tendencia que en los dos casos de prueba expuestos aquí.

En los gráficos de la **Figura 5.7**, **Figura 5.8** y **Figura 5.9**, se muestran los resultados de 10 ejecuciones del algoritmo en un ambiente de red LAN, de las cuales los valores de la **Tabla 5.7** reflejan un promedio de estas 10 ejecuciones. Todos los gráficos parecen seguir el mismo comportamiento que en la ejecución del caso 1 local, excepto el grafico de porcentaje de bits que conforman la clave final. En este caso los

valores del caso de prueba están menos dispersos que los valores obtenidos en el caso de prueba 1.

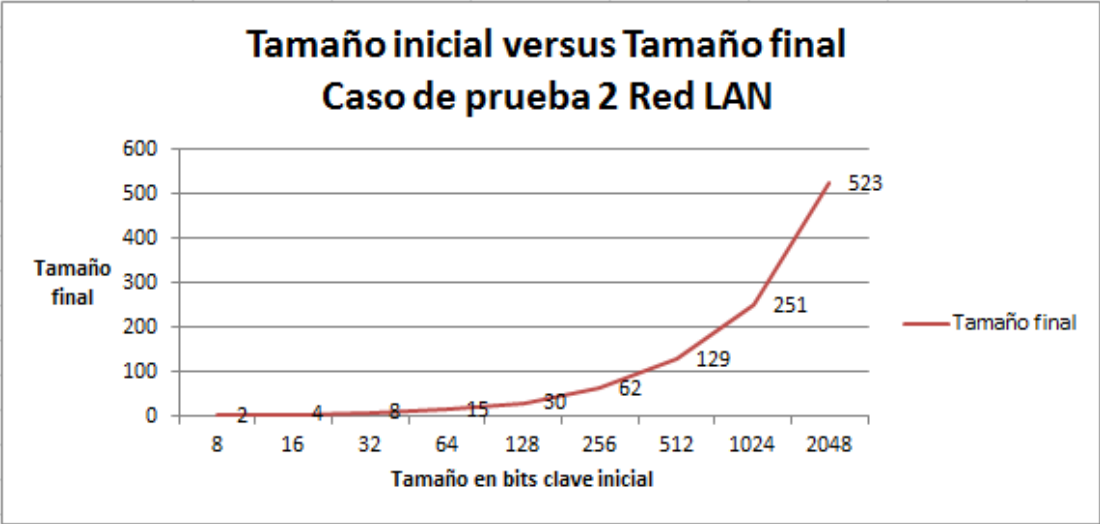


Figura 5.7. Grafica tamaño final de la clave

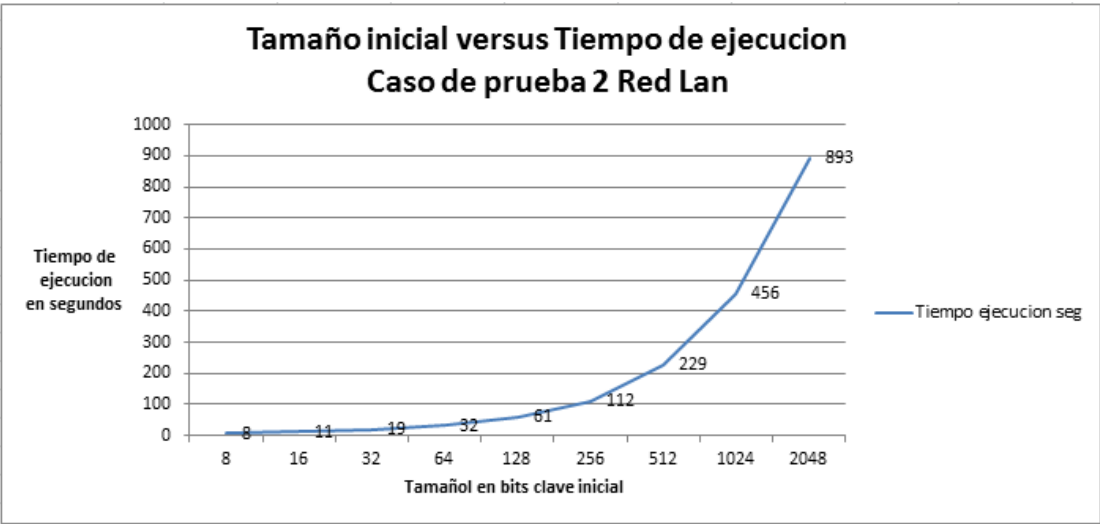


Figura 5.8. Grafico tiempo de ejecución

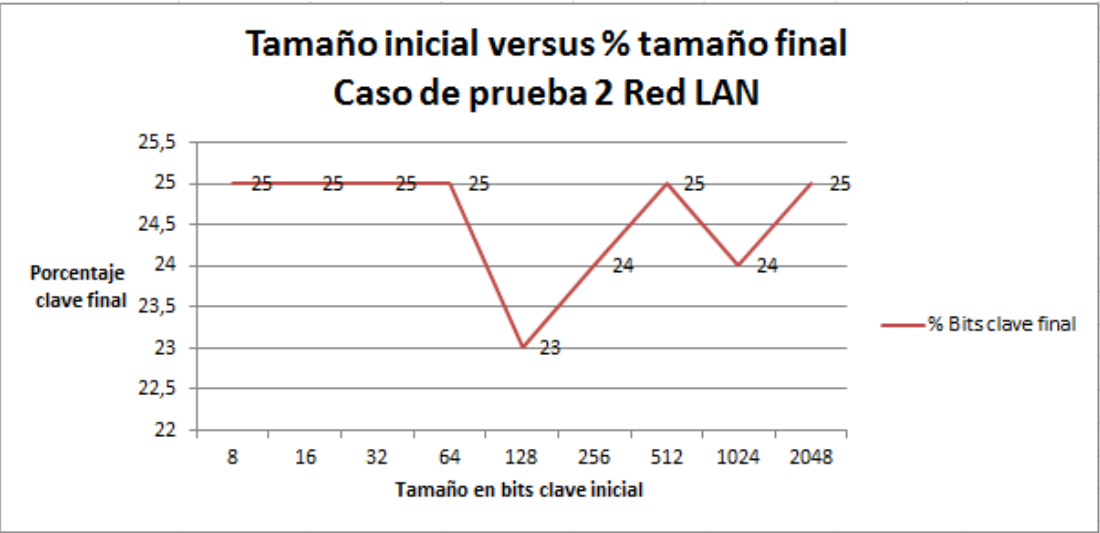


Figura 5.9. Grafico porcentaje de clave final

Análisis de resultados en comparación

En el gráfico de la **Figura 5.10**, se comparan los resultados obtenidos en los dos casos de prueba para la variable tamaño final. Se puede observar que en el caso de prueba 1 para la ejecución local del algoritmo, en promedio se obtiene una clave final más larga. Esto puede ser, debido a que en la comunicación en red se pierdan algunos bits

En el gráfico de la **Figura 5.11**, se comparan los resultados de ambos casos de prueba para la variable tiempo de ejecución. En este caso, la proyección teórica de los resultados indicaría que, en un ambiente local el tiempo de ejecución debería ser menor que en un ambiente distribuido, debido al retraso de la comunicación por los cables. El gráfico corrobora esta predicción ya que el caso de prueba 1 tiene un tiempo de ejecución menor en promedio, pero para algunos largos de clave el caso de prueba 2 tiene menor tiempo de ejecución, esto es posible debido a múltiples factores, como el hardware en el que se realizó la prueba, es posible que el procesador del notebook sea menos potente que el procesador del computador de mesa.

Según los gráficos, esta predicción podría expandirse a una red WAN, de manera que el tiempo de ejecución debería aumentar. Por esta razón no se realizó un tercer caso de prueba donde se ejecutara el algoritmo en una red WAN, porque los resultados obtenidos en local y LAN son suficientes para predecir el comportamiento del algoritmo en cualquier escenario de mayor complejidad.

En el caso de las variables tamaño final y tiempo de ejecución, se observa una tendencia de crecimiento del orden exponencial. Este fenómeno se analizara en profundidad en un análisis asintótico más adelante.

El último gráfico en la **Figura 5.12**, muestra la comparación entre el porcentaje de bits que conforman la clave final. Este gráfico muestra que en el caso de prueba 2 el comportamiento de la variable es regular, acercándose al 25%, mientras que en el caso de prueba 1 se nota una mayor dispersión. A medida que la cantidad de ejecuciones aumenta, se ve que la variable se acerca al 25% teórico.

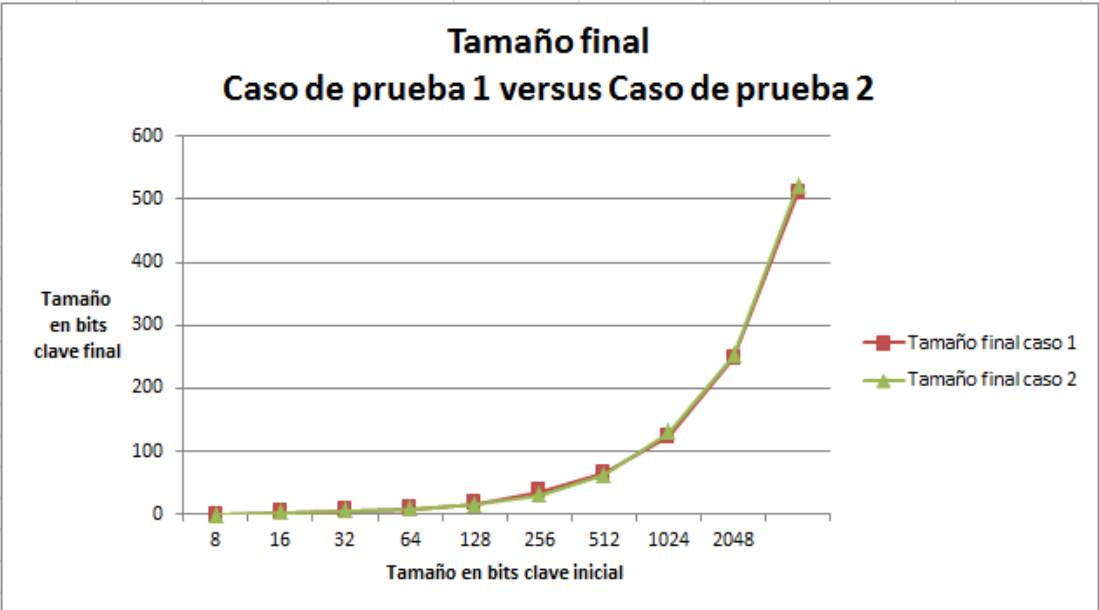


Figura 5.10. Comparación Tamaño final caso de prueba 1 y caso de prueba 2

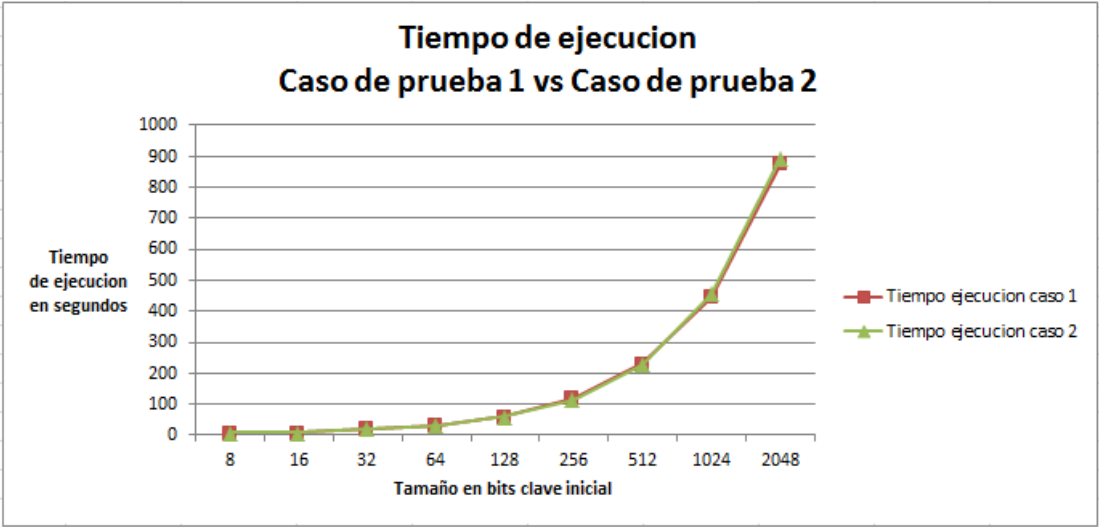


Figura 5.11. Comparación Tiempo de ejecución caso de prueba 1 y caso de prueba 2

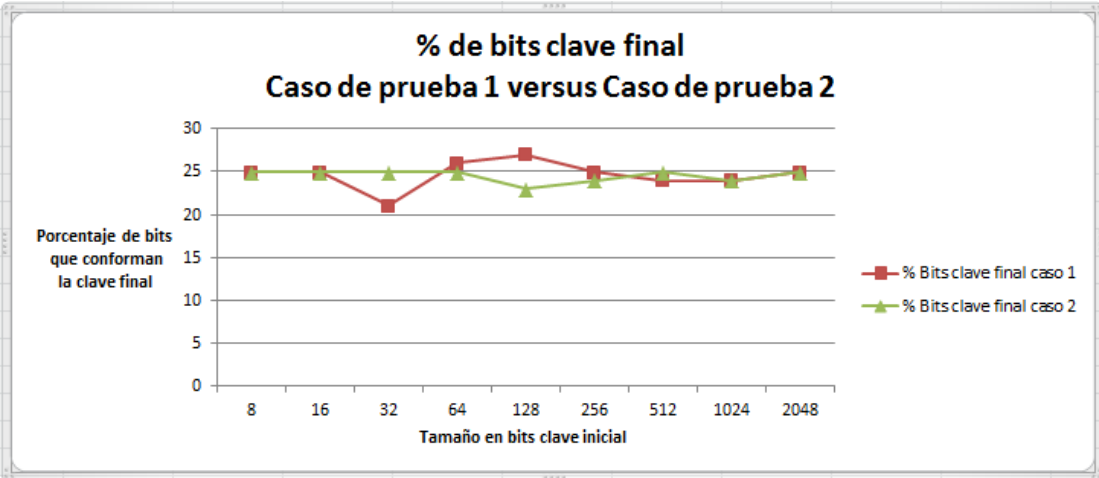


Figura 5.12. Comparación porcentaje de bits clave final

5.3.3. Análisis asintótico y orden de complejidad del algoritmo

La eficiencia de un algoritmo puede definirse como la relación entre los recursos consumidos y los productos conseguidos. El rendimiento de un algoritmo puede ser medido o estimado según el análisis de complejidad de este. El uso eficiente de los recursos, se suele medir en función de dos parámetros: el espacio, es decir la memoria que utiliza, y el tiempo, lo que tarda en ejecutarse. Algunos otros parámetros utilizados para medir el uso de recursos pueden ser:

- Tiempo de ejecución
- Uso de memoria principal
- Accesos a disco
- Tamaño y conteo de datos de entrada
- Cantidad de ciclos en el código o instrucciones *for*, *if-else*, *while*, etc
- Uso de procesadores
- Uso de redes de comunicación
- Uso de operaciones elementales

La variable que nos interesa estudiar es el tiempo de ejecución en segundos, como se aprecia en la **Figura 5.11**. En la figura se ve claramente que independiente del caso de prueba, el algoritmo parece seguir una tendencia en su tiempo de ejecución.

El tiempo de ejecución de un algoritmo va a depender de diversos factores, como los datos de entrada, la calidad del código generado por el compilador y el lenguaje de programación, la naturaleza y la rapidez de las instrucciones del algoritmo y la complejidad intrínseca del algoritmo. Hay dos estudios posibles del tiempo de ejecución:

1. Uno que proporciona una medida teórica, que consiste en encontrar una función que acote el tiempo de ejecución del algoritmo para unos valores de entrada dados.
2. Y otro que ofrece una medida real del tiempo de ejecución, que conociste en medir el tiempo de ejecución del algoritmo para unos valores de entrada dados y en una maquina concreta.

El camino de investigación fue la de la segunda opción, esto debido a que poseemos el experimento en físico y podemos obtener los resultados para realizar la aproximación. El objetivo de este análisis, es encontrar en base a los resultados obtenidos, alguna función la cual permita expresar el comportamiento del algoritmo de manera general. Esto es posible mediante la búsqueda de las cotas asintóticas para la variable tiempo de ejecución.

Para obtener esta función de cota superior, analizaremos los resultados de la gráfica tiempo y lo compararemos con las funciones de orden de complejidad ya existentes. Luego veremos cuál de las gráficas se aproxima al gráfico obtenido por el algoritmo y esa será una candidata a ser la cota superior del algoritmo. Usualmente se utiliza la notación de Landau: $O(g(x))$, Orden de $g(x)$, o Notación O Grande, para referirse a las funciones acotadas superiormente por la función $g(x)$. El argumento en este análisis será el tamaño en bits de la clave inicial, el cual es el parámetro de entrada para el algoritmo. Este argumento toma los valores del siguiente conjunto:

$$A = \{8,16,32,64,128,256,512,1024,2048\}$$

Definiciones

Cota asintótica: una cota asintótica es una función que sirve como cota superior o inferior de otra función, cuando su argumento tiende a infinito

Notación Landau: La notación Landau u “O grande” se utiliza para denotar el orden de complejidad de un algoritmo, se dice que $O(g(n))$ es del orden $g(n)$ para referirse a la funciones acotadas superiormente por la función $g(n)$.

Orden de complejidad: el orden de la función $g(n)$ mide la complejidad temporal del algoritmo, es el que expresa el comportamiento dominante cuando la entrada del algoritmo aumenta. Se dice que la función $f(n)$ pertenece a la clase de complejidad o es del orden de complejidad $g(n)$, si existe un c y un n_0 , tales que para todo $n \geq n_0$ se cumpla que:

$$\forall n \geq n_0, f(n) \leq cg(n).$$

En la **Tabla 5.8** y la **Figura 5.13**, se ven las funciones de complejidad más comunes y sus gráficas. En la gráfica se ve que la función más prometedora para ser una cota superior es la función cuadrática y la exponencial. Sin embargo, cuando tomamos los valores de n como la cantidad de bits de entrada, ocurre que para valores grandes de n , la formula cuadrática y exponencial se disparan casi hasta el infinito, tomando el ejemplo de cuando la entrada de la clave inicial sea de 2048 bits, estas dos funciones adquieren valores de 4.194.304 y la exponencial tiende a infinito. Por lo tanto, a pesar de que en un principio el crecimiento es parecido a la cota cuadrática y exponencial, para valores muy grandes no se adecuan a nuestro algoritmo.

Orden	Nombre
$O(1)$	constante
$O(\log n)$	logarítmica
$O(n)$	lineal
$O(n \log n)$	casi lineal
$O(n^2)$	cuadrática
$O(n^3)$	cúbica
$O(a^n)$	exponencial

Tabla 5.8. Clases de complejidad conocidas

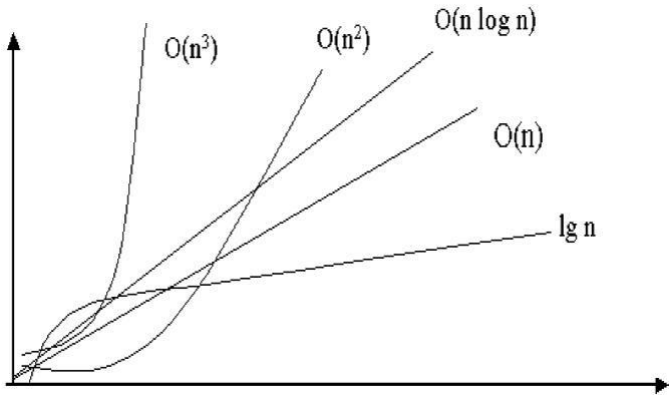


Figura 5.13. Grafica de las clases de complejidad conocidas

Las funciones logarítmicas y logarítmica natural, tienen un crecimiento bastante más lento cercano al comportamiento de nuestro algoritmo. Sin embargo el logaritmo natural crece demasiado lento con respecto a nuestro algoritmo, por lo tanto esta es una candidata a ser cota inferior, mientras el logaritmo aún es muy grande en aumento comparado con nuestra variable. Por lo tanto la otra función que sirve como cota superior será la de orden lineal. Esto se ve en la **Figura 5.14**.



Figura 5.14. Grafica de funciones de complejidad junto a la variable tiempo de ejecución

Podemos concluir que la variable tiempo de ejecución del algoritmo, posee un crecimiento que varía según un orden de complejidad entre $O(\ln n)$ y $O(n)$. [36]

5.4. EVALUACIÓN DE REQUISITOS

En este apartado se realiza una verificación de los requisitos funcionales y no funcionales, definidos en el capítulo 4.2.1. En las tablas se indica el requerimiento y una descripción de cómo fue cumplido durante la ejecución de este trabajo.

Requerimiento funcional	Descripción
Generar las secuencias aleatorias de bits	Para generar las secuencias de bits aleatorias, se utilizaron las clases Generar_secuencia_polarizacion.java y Secuencia.java para guardar el arreglo de bits
Generar los esquemas de polarización	Para generar los esquemas de polarización se usaron las clases Generar_secuencia_polarizacion.java y Esquema.java para guardar el arreglo de polarizaciones
Transmitir los fotones polarizados	Para transmitir los datos se utilizó la tecnología RMI, implementada en las clases InterCanalEmi.java e interCanalRec.java
Comparar los esquemas de polarizaciones de Alice y Bob	Una vez intercambiados los esquemas de polarización, el receptor comienza a comparar el esquema de Alice con el generado localmente, el resultado se guarda en un nuevo arreglo llamado “comparación”
Generar la RAW KEY o clave en bruto	La Raw Key es la clave que se genera directamente al

	comparar los dos esquemas de polarizaciones, esta es generada en el programa Receptor y su valor se guarda en el arreglo “rawkey”
Generar la SIFTED KEY o clave depurada	Para obtener la Sifted Key, se hace uso de la clase Principio_de_incertidumbre.java, el cual permite calcular para cada bit con un 50% de probabilidad el ser detectado o no
Intercambiar la SIFTED KEY entre Alice y Bob	Para intercambiar la clave final se realiza el mismo procedimiento RMI, utilizando las mismas clases de interfaz pero enviando los datos de la clave final
Generar e intercambiar las tablas de Hashes para la seguridad	Una vez que ambas partes de la comunicación tienen la clave final, se genera un hash de comprobación tanto en el emisor como en el receptor, luego se intercambian estos hashes mediante RMI. El objetivo es que si los hashes son iguales entonces la clave es segura y no fue interceptada
Crear la interfaz grafica	La interfaz gráfica fue creada con la librería Swing de Java y las ventanas para el botón ”ver B92”, fue creada con la librería Edisoncor.java
Enviar y recibir mensajes mediante un algoritmo publico AES	Una vez que la clave fue compartida y comprobada de manera segura, esta se utiliza en un algoritmo de criptografía simétrica como lo es el algoritmo AES, este algoritmo utiliza la clave de entrada y permite encriptar un mensaje entre el emisor y el receptor

Tabla 5.9. Requisitos funcionales

Requisito no funcional	Descripción
Utilizar Java como lenguaje de programación	El lenguaje utilizado en su totalidad fue Java, con RMI
El programad debe funcionar en distintas maquinas	Al utilizar Java, nos aseguramos que los códigos sean portables. En el capítulo de pruebas se ejecutaron los programas en distintas maquinas
El sistema debe presentar interfaz gráfica de usuario	Las pruebas y las imágenes muestran que el programa posee una interfaz gráfica de usuario amigable
La red de prueba debe utilizar el modelo Cliente-Servidor	Durante el montado de la red de prueba, se configuro todo el sistema mediante una arquitectura Cliente-Servidor

Tabla 5.10. Requisitos no funcionales

5.5. CONCLUSIONES DEL CAPÍTULO

Las pruebas realizadas en este capítulo permitieron visualizar de manera más fácil el funcionamiento del algoritmo, también se dejó en prueba la eficacia y eficiencia del mismo. Mediante el análisis en gráficos y tablas se pudo obtener una aproximación asintótica de su funcionamiento para la variable tiempo de ejecución. La ejecución del algoritmo en los distintos casos y ambientes de prueba permitió demostrar que el algoritmo funciona eficazmente de manera real.

Algunos inconvenientes ocurridos durante las pruebas en red, fueron que las IP de las máquinas se desconfiguraban debido a aplicaciones de virtualización de escritorio como virtual box, el problema era que el programa RMI hacía uso de una IP distinta a la que debía tener la máquina en la red, esta IP estaba en una red virtual en la máquina y por lo tanto había dos direcciones IP activas, una física y una virtual. Por alguna razón el programa RMI elegía la IP virtual para realizar la comunicación, lo que generaba un error de *time out*. Esto se solucionó al aislar todo el sistema y cerrar cualquier proceso o programa que estuviera abierto en segundo plano.

Respecto a los objetivos presentados en el capítulo 1.2.3, el objetivo específico que se cumple en este capítulo es el sexto (Implementación del protocolo), mientras que los otros objetivos se cumplen en los capítulos cinco, cuatro y dos.

En el análisis asintótico se obtuvo una cota superior y una cota inferior para la variable tiempo de ejecución del algoritmo, esto quiere decir que cualquier ejecución futura debería encontrarse en un tiempo de ejecución mayor al orden logaritmo natural y menor al orden lineal. También se observa en los gráficos que a medida que se duplicaba el tamaño de entrada, también se duplicaba el tiempo de ejecución del algoritmo, lo que es teóricamente correcto.

Los gráficos y el análisis de los resultados, permite obtener como conclusiones que mientras más bits tenga la clave inicial, más bits tendrá la clave final, sin embargo existe un umbral sobre el cual los resultados se estandarizan y eso comienza a ocurrir desde los 256 bits, para claves de menor tamaño los resultados son más dispersos, por lo tanto se recomienda utilizar claves iniciales de 256 bits en adelante, además las gráficas muestran que es en este punto donde el resultado local y en red se asemejan

más. En el caso del porcentaje de bits que conforman la clave final, mientras más bits son eliminados aleatoriamente mayor seguridad obtiene el algoritmo, además, es a partir de los 256 bits que el funcionamiento del experimento se corresponde de manera exacta con el 25% teórico. El análisis profundo de este fenómeno se realizara en el capítulo siguiente.

En el capítulo 6 se realizan las conclusiones finales y el análisis profundo del algoritmo en su modelación teórica y experimental.

CAPÍTULO 6: CONCLUSIONES Y TRABAJO FUTURO

El siguiente capítulo reúne las conclusiones finales del proyecto y el trabajo futuro para expandir esta tesis. El capítulo está dividido en: resumen de todo el trabajo realizado en esta memoria, conclusiones respecto a la ejecución y desarrollo del algoritmo, luego las conclusiones respecto a los resultados obtenidos y el análisis de la hipótesis planteada en el marco teórico.

6.1. RESUMEN

El objetivo principal y general de este trabajo fue desarrollar una aplicación que lograra simular el funcionamiento del protocolo B92, simulando una comunicación de fotones por medio de un canal cuántico, en el cual una clave secreta es generada y compartida por un canal clásico, luego esta clave debe ser capaz de usarse en un algoritmo de criptografía clásico de clave simétrica.

En el primer capítulo se presenta el problema y la solución propuesta, los objetivos generales y específicos del proyecto, así como también la metodología de trabajo, el diagrama de solución, el diagrama de flujo de tareas y la planificación de trabajo.

Los conceptos físicos y matemáticos fundamentales para comprender la criptografía cuántica se encuentran en el capítulo dos, donde se estudiaron temas como: la polarización de fotones, el principio de incertidumbre de Heisenberg, el teorema de no clonación y el principio de superposición. Estos conceptos sientan las bases para la seguridad del algoritmo, así como la seguridad de cualquier algoritmo de distribución de claves cuánticas.

En el capítulo tres se definió la metodología de trabajo, con lo que respecta al ciclo de desarrollo de un proyecto de software, arquitectura del sistema cliente servidor, arquitectura distribuida RMI, definición del lenguaje de programación a utilizar, en este caso Java. También en este capítulo se realiza un estudio de factibilidad para el proyecto, considerando los costos en recursos tecnológicos, hardware y software. El resultado de este estudio de factibilidad fue que faltaban implementos para realizar de manera más satisfactoria las pruebas, debido a que se contaba solo con los implementos del alumno.

El capítulo cuatro consiste en la ejecución del proyecto, desarrollo de la metodología de trabajo definida en el capítulo anterior y por consiguiente el análisis completo de ingeniería de software. Esto consiste en desarrollar todos los requerimientos del sistema y para cada uno diseñar los casos de uso correspondientes, luego para cada caso de uso se diseñan los diagramas de secuencia que muestran el funcionamiento de los paquetes o módulos del sistema, después se genera el diagrama de clases del sistema para luego pasar a la implementación. La implementación fue realizada en el entorno de desarrollo Netbeans utilizando el lenguaje de programación Java con RMI, haciendo uso de licencias libres para todos los programas. Finalmente se realizó una prueba de ejecución del algoritmo en modo local.

El capítulo cinco plantea los casos de prueba para el algoritmo, siendo el primero un escenario local y el segundo un ambiente en red distribuida. En el segundo caso se construyó una red LAN, dejando abierta la posibilidad para ejecutar el algoritmo en una red WAN, pero esta alternativa se dejó para trabajos futuros. Al comparar los resultados y el crecimiento de las variables de prueba en la red LAN versus local, se aprecia un crecimiento que predice el comportamiento del algoritmo en cualquier otro diagrama de red.

6.2. CONCLUSIONES SOBRE LOS RESULTADOS

La premisa de creación del algoritmo B92 y en general todos los protocolos cuánticos, radica en el hecho de que en un futuro no muy lejano, las computadoras cuánticas serán una tecnología accesible y con una potencia exponencialmente superior, debido a esto, la criptografía actual tal y como la conocemos estará en riesgo.

Debido al trabajo realizado por Peter Shor [10], se encontró un algoritmo cuántico capaz de factorizar números primos grandes en un orden de complejidad o tiempo de ejecución logarítmico. Esto supone un peligro para la seguridad del algoritmo RSA, el cual es el algoritmo más usado en criptografía. El problema surge debido a que la seguridad de RSA radica en la factorización de números primos inmensamente grandes. Por lo tanto con la llegada de estas computadoras cuánticas el criptosistema RSA podría ser roto y con ello la seguridad de internet estaría completamente comprometida.

La llegada de estas computadoras cuánticas, genera la necesidad de crear nuevos algoritmos cuánticos para criptografía, que sean seguros en un ambiente de tecnología cuántica o crear mecanismos que hagan que los algoritmos clásicos sean seguros en un ambiente cuántico.

La primicia de investigación de la criptografía cuántica, nace del algoritmo de criptografía simétrica conocido como el cifrado de Vernam “*one time pad*” o cuaderno de un solo uso, el cual fue demostrado como “seguro perfecto” en 1942 por Claude Shannon [37], siempre y cuando la clave que se use sea del mismo tamaño que el texto, aleatoria, y de un solo uso. Teniendo un algoritmo de criptografía seguro como el cifrado de Vernam, surge la necesidad de encontrar mecanismos para distribuir las claves que serán usadas en estos algoritmos. Dentro de estos algoritmos se encuentran los algoritmos BB84, B92 y E91, siendo el B92 el que se estudió en este trabajo.

Bajo estos objetivos se realizó el estudio teórico del protocolo B92, considerando los conceptos físicos de la mecánica cuántica. Buscando la forma de implementar en lenguaje Java la idea central del algoritmo.

Según lo expuesto en el capítulo 2.1.14 y el capítulo 2.1.8, se estudió el concepto de superposición de estados y principio de incertidumbre. Un fotón polarizado por un filtro en un ángulo de inclinación, puede interpretarse como una combinación lineal de dos ángulos de inclinación distintos. Esto es una suma de dos componentes que conforman ese ángulo.

Debido a esta elección de “en que ángulo polarizar el fotón”, se genera un fotón que puede representar información de manera binaria. Esto es, mediante una polarización rectilínea en un ángulo representar el bit 0 y en una polarización diagonal en otro ángulo representar un bit 1.

El teorema de no clonación y el acto de medición en mecánica cuántica, aseguran que cualquier fenómeno cuántico no puede ser observado ni copiado, esto es porque en el momento en que se realiza una medición en un sistema cuántico el estado del sistema se altera y por consiguiente la información cambia, esto permite saber irrefutablemente si un intruso intercepta la comunicación o si intenta clonarla, porque

al intentar clonarla debe observar el fotón y al observarlo cambia su estado, por lo tanto una clonación de un estado arbitrario es imposible.

El principio de incertidumbre indica que cuando se tienen dos magnitudes físicas correlacionadas, (como el caso de la polarización interpretada como la suma de dos polarizaciones distintas) cualquier medición con exactitud en una de las componentes del sistema, vuelve aleatorias las mediciones en la otra componente del sistema. El ejemplo más claro del principio de incertidumbre son la posición y la velocidad, a medida que se obtiene la posición de una partícula con exactitud, se pierde información sobre su velocidad y al conocer con exactitud su velocidad se vuelve incierta la información sobre su posición, por eso es un principio de “incertidumbre”. Este fenómeno ocurre solo a niveles microscópicos, con distancias comparables a las dimensiones del átomo, en el mundo macroscópico cotidiano no existe esta incertidumbre en la medición.

Si consideramos nuevamente nuestro experimento de codificar bits usando fotones polarizados, sabemos que los fotones poseen dimensiones del orden de partículas atómicas, por lo tanto el principio de incertidumbre aplica para ellos. Entonces, si tenemos una fuente que genera un fotón polarizado en una base rectilínea o diagonal hay un 50% de probabilidad de que este fotón este polarizado rectilínea o diagonalmente. Por lo tanto cuando el fotón llega al receptor Bob, él alinea su filtro detector de fotones en un cierto ángulo con una base de polarización que puede ser rectilínea o diagonal (esta elección es aleatoria y tiene un 50% de acertar la base).

La seguridad del protocolo radica en que los ángulos para cada una de las bases se deben invertir, es decir cuando el emisor envía un fotón polarizado rectilíneamente lo hace a un ángulo de 0° , mientras que cuando el receptor mide el fotón usando la misma base rectilínea lo hace en un ángulo de 90° . Ambos ángulos son rectilíneos por lo tanto no debería haber error. Sin embargo, el detector de Bob no detectara ningún fotón debido a que 0° y 90° son ángulos perpendiculares y ningún fotón polarizado a 0° puede pasar un filtro a 90° , por lo tanto, hay 0% de probabilidad de detectar el fotón cuando se miden en la misma base.

El truco es, que las bases de generación y detección del fotón deben ser distintas, esto es, que el fotón debe generarse con una polarización rectilínea y debe ser medido por Bob con un filtro en base diagonal. Es aquí donde entra el principio de incertidumbre,

ya que un fotón polarizado en un ángulo de 0° puede considerarse como la suma de un ángulo de -45° y 45° . Precisamente esto es lo que ocurre en los experimentos y el fotón enviado en polarización rectilínea a 0° tiene un 50% de probabilidades de ser detectado por un filtro diagonal a 45° , este 50% extra de probabilidad agrega una seguridad impenetrable para los espías, ya que la generación es aleatoria y la detección igual es aleatoria. En el caso de que el fotón colapse a la componente de -45° entonces no será detectado porque el filtro de Bob mide a 45° y ambos ángulos son perpendiculares por lo que el fotón se bloqueara.

Si un intruso ajusta su detector de manera diagonal igual como lo hace Bob, no sabrá si debe alinearlos a 45° o a -45° , esta decisión tiene un 50 % de probabilidad de éxito y aun así aunque lograra acertar el 45° existe otra posibilidad del 50% de que el fotón simplemente no colapse a 45° sino que colapse a -45° , por lo tanto no hay forma de saber cuándo está detectando realmente el fotón en la base correcta.

La seguridad del protocolo radica en tres capas, la primera es acertar la base de polarización correcta, para la cual existe un 50% de probabilidad de acierto, luego si se logra acertar la base de polarización se debe acertar otro 50% de probabilidad añadido por el P.I.H, el cual puede hacer colapsar el fotón en una componente incorrecta de la superposición de estados, finalmente una vez terminada la transmisión de fotones, se realiza un chequeo mediante una función hash para ver si las claves generada por Bob y Alice son iguales. Es en este punto donde se plantea la hipótesis teórica crucial:

Si la clave de entrada del algoritmo se divide dos veces en dos fenómenos físicos con un 50% de probabilidad cada uno, entonces solo el 25% de la clave inicial formara parte de la clave final.

En los algoritmos BB84 y E91 también existía esta premisa, para los cuales la teoría indicaba que un 50% de la clave inicial formaba parte de la clave final en el algoritmo BB84 y un 30% de la clave inicial formaba parte de la clave final en el E91. Estos claramente son las predicciones teóricas en dichos algoritmos, los cuales fueron probados y corroborados en los respectivos trabajos de tesis.

En el capítulo cinco sobre resultados de las pruebas, se pudo comprobar de manera exitosa esta hipótesis y corroborar la seguridad del algoritmo, indicando en las 90 pruebas realizadas una convergencia en la variable “% clave final” hacia el 25%, siendo exacto en algunos casos. Tanto en el caso de prueba local como en el de red, la convergencia y la tendencia fue la misma, siempre se eliminaba cerca del 75% de los bits de entrada, quedando solo el 25% restante para la clave final.

Obtener este valor fue un triunfo de la aplicación rigurosa del método científico, un triunfo de los métodos matemáticos y físicos de experimentación. Durante esta experimentación se realizaron pruebas completamente aisladas y con una precisión exacta en los instrumentos. La obtención del 25% de clave final fue crucial para el cumplimiento del objetivo general y los objetivos específicos.

Algunos puntos fuertes del trabajo fueron que se contaba con el algoritmo bb84 desarrollado en tesis pasadas por los Alumnos y algunos puntos débiles del trabajo fueron el bajo poder monetario y recursos de hardware escasos por parte del alumno.

6.3. TRABAJO FUTURO

En primer lugar se sugiere a futuro realizar más pruebas del algoritmo mejorando los equipos de hardware y la potencia de los computadores, también se puede ampliar la red de pruebas mediante una topología más compleja en WAN. Se puede aumentar la cantidad de ejecuciones del algoritmo también.

Realizar una comparativa entre los tres algoritmos generados en la escuela de ingeniería Informática de la universidad de Tarapacá. Actualmente se cuentan con los algoritmos en Java de los protocolos BB84, E91 y B92, por lo tanto es posible realizar una comparativa entre los tres.

Se puede aplicar el algoritmo en un cifrado de Vernam de manera real, ya que este trabajo solo contempla la creación de la clave secreta y su distribución, sería adecuado implementar el protocolo de Vernam usando la clave generada cuánticamente.

En esta sección deben presentarse las tareas que por diversos motivos no se cumplieron durante la memoria, posibles mejoras al trabajo realizado (ya sea en metodologías, uso de otras tecnologías o aproximaciones diferentes a las usadas) y otras líneas de trabajo donde el proyecto de memoria pudiera aplicarse.

A diferencia de los protocolos BB84 y E91, en el algoritmo B92 no se envían los bits generados por Alice, lo que se envía es solo las bases de polarización que representan estos bits, esta información solo es conocida por el emisor y receptor, por lo que si un intruso intercepta esa información (además de ser detectado debido a que el canal es cuántico) no podrá obtener información útil de ella. Sería interesante realizar una prueba con la intromisión de un espía, para corroborar la dificultad de obtener información de la clave.

Finalmente para un trabajo muy avanzado de doctorado, sería posible implementar el protocolo y realizar las pruebas sin simulación. Esto es mediante el uso de fotones reales en un canal cuántico, con detectores y filtros de luz en un laboratorio físico real.

La importancia del trabajo realizado en esta memoria es crucial para dar cierre a toda la familia de protocolos de distribución de claves cuánticas de la familia del principio de incertidumbre. Es importante el hecho de haber corroborado la hipótesis teórica satisfactoriamente.

REFERENCIAS

- [1] G. E. Moore, «Cramming more Components onto integrated circuits,» *Electronics Retrieved*, 1996.
- [2] N. S. Agency, «Commercial National Security Algorithm suite and Quantum Computing,» *FAQ*, Enero 2016.
- [3] M. Pinto, Simulacion de un protocolo de comunicacion cuantica entre procesos en un ambiente distribuido, Arica, Chile: Universidad de Tarapaca, 2010.
- [4] L. A. Caceres , R. P. Fritis y P. C. Collao, «Desarrollo de un simulador para el protocolo de criptografia cuantica E91 en un ambiente distribuido,» *Revista Chilena de ingenieria*, vol. 23, Abril 2015.
- [5] P. C. Collao, Comparacion y analisis de las fortalezas de seguridad de los protocolos de criptografia cuantica BB84 y E91, Arica, Chile: Universidad de Tarapaca, 2014.
- [6] C. H. Bennet y G. Bassards, «Quantum Cryptography: public key distribution and coin tossing,» de *International Conference on computers systems & signal processing* , Bangalore India, 1984.
- [7] A. K. Ekert, «Quantum Cryptography based on the Bell's Theorem,» *Am.Psy.Soc*, vol. 67, n° 6, pp. 661-663, 1991.
- [8] C. H. Bennett, «Quantum Cryptography using any two nonorthogonal states,» *Physical Review Letters*, vol. 68, pp. 3121-3124, 1992.
- [9] W. Heisenberg, «Uber den auschaulichen inhalt der quantentheoretischen Kinematic und Mechanik,» *Zeitschrift fur Physik(Diario de la fisica)*, Marzo 1925.
- [10] P. Shor, «Polynomial-Time algorithms for prime factorization and discrete logarithms on a quantum computer,» *SIAM Journal on Computing* , vol. 26, n° 5, Noviembre 1996.
- [11] R. L. Rivest, A. Shamir y L. Adleman, «A method for Obtaining Digital Signature and Public-Key Cryptosystems,» *Communication of the ACM*, vol. 21, n° 2, pp. 120-126, 1978.

- [12] C. Gidney y M. Eker, «How to factor 2048 bits RSA integers in 8 hours using 20 million noisy qubits,» de *Google INC*, California USA, Mayo 2019.
- [13] H. D. Scolnik, «Nuevos algoritmos de factorizacion de enteros para romper RSA,» de *Reunion española sobre criptografia* , Salamanca España, Septiembre 2008.
- [14] L. Careces, Estudio e implementacion de un algoritmo cuantico para IPv6 en un ambiente distribuido, Marzo 2012.
- [15] E. Lazo , Apuntes de fisica moderna, Arica: Universidad de Tarapaca, 1996.
- [16] R. A. Serway y J. W. Jewett, Fisica para ciencias de la ingenieria con fisica moderna, Cengage Learning, 2008.
- [17] M. Gulis, «blogs.20minutos.es,» 12 Noviembre 2015. [En línea]. Available: <https://blogs.20minutos.es/ciencia-para-llevar-csic/2015/11/12/el-experimento-fisico-mas-hermoso-de-todos-de-los-tiempos-la-doble-rendija/>.
- [18] R. Fernandez, D. Bellver y I. Lloro, «The quantum cryptograpy: Communication and computation,» de *Acta astronautica* 57, Barcelona, 2005.
- [19] C. cuanticos, «cuentos-cuanticos.com,» 2014. [En línea]. Available: <https://cuentos-cuanticos.com/2014/12/15/la-polarizacion-y-la-cuantica-en-orbita-laika/>.
- [20] M. Nielsen y I. Chuang, Quantum computation and Quantum information, EEUU: Cambridge University, 2000.
- [21] W. Wootters y w. Zurek, «A single Quantum Cannot Be cloned,» *Nature*, vol. 92, nº 6, pp. 271-272, 1982.
- [22] O. Ortiz, «¿Computacion cuantica?,» Publicaciones del IPN, Mexico, 2007.
- [23] G. S. Vernam, «Cipher Printing Telegraph Systems for secret wire and radio telegraphic communications,» *Transactions of the American Institute of Electrical Engineers*, vol. 55, pp. 109-115, 1926.
- [24] C. Shannon, «Communication Theory of Secrecy Systems,» *Bell System Technical Journal*, vol. 28, nº 4, pp. 656-715, 1949.
- [25] D. M. Bressoud, «Factorization and primality testing,» UTM Springer-Verlag, 1989.
- [26] H. C. Williams, «A modification of the RSA public-key encryption procedure,»

- IEEE Transaction on information Theory*, nº 26, pp. 726-729, 1980.
- [27] F. 186, *Digital signature standars*, 1994.
- [28] J. Gruska, *Quantum computing*, Mc Graw-Hill, 1999.
- [29] M. Haitjema, «A survey of the prominent quantum key distribution protocols,» 2018.
- [30] H. R. Ortiz, «Fundamentos de criptografia cuantica,» Medellin, 2007.
- [31] I. D. Ivanovic, «How to differentiate between non-ortogonal states,» *physics letters*, vol. 123, nº 6, pp. 257-259, 1987.
- [32] M. Dusek, N. Lutkenhaus y M. Hendrych, «Quantum cryptography,» *Progress in optics*, vol. 49, pp. 381-454, 2006.
- [33] M. Villalobos, «Apuntes de ingenieria de software,» Universidad de Tarapaca, Arica, 2015.
- [34] G. Couloris, «Distributed Systems Concepts and Design,» de *Distributed Systems Concepts and Design*, 2012, p. 716.
- [35] G. Couloris, J. Dollimore, T. Kindberg y G. Blair, *Distributed Systems: concepts and design*, Addison Wesley, 2011.
- [36] H. Beck, «Apuntes de analisis y diseño de algoritmos,» Universidad de Tarapaca, Arica, 2014.
- [37] C. E. Shannon, «Bell labs Advances intelligent Networks,» 2012.
- [38] A. Einstein, B. Podolsky y N. Rosen, «Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?,» *Physical Review*, vol. 47, pp. 777-780, 1935.
- [39] G. Abal, «Paradoja ERP y desigualdades de Bell,» Universidad de la republica, Montevideo, 2007.
- [40] A. Menezes, P. Oorschot y S. Vanstone, «Handbook of applied cryptography,» de *CRC Press*, Canada, 1997.
- [41] J. Ramio, «Libro electronico de seguridad informatica y criptografica,» www.criptored.upm.es/guiateoria/gt_m001a.html.
- [42] A. AL-Kindi, «The origins of cryptology: The Arab contributions,» *Cryptologia*, vol. 16, nº 2, 1992.

- [43] C. J. Mendelsohn, «Blaise de Vigenere and the Chiffre Carre,» *Proceedings of the American Philosophical Society*, vol. 82, nº 2, pp. 103-129, 1940.
- [44] E. Rieffel, «An introduction to quantum computing for non-physicists,» *ACM computing surveys*, vol. 32, nº 3, pp. 300-335, 2000.
- [45] A. Alonso-Arroyo, J. G. de Dios, A. Vidal-Infer, C. Navarro-Molina y R. Aleixander-Benavent, «Fuentes de información bibliográfica (XIII). Gestores de referencias bibliográficas: particularidades sobre RefWorks y Zotero/Sources of bibliographic information (XIII). Bibliographic reference managers: particularities about RefWorks y Zotero.,» *Acta Pediatrica Espanola*, 70(6), p. 265, 2012.
- [46] J. Demasi, «Formato IEEE. Estilo y Referencias Bibliográficas,» *Instituto de Ingenieria Eléctrica (IIE), Facultad de Ingenieria, Universidad de la República.*, vol. 1, nº 1, p. 1, 2011.

APENDICES

Apéndice 1: El microscopio de Heisenberg

El problema de la medición desde el punto de vista cuántico tiene su consecuencia en el principio de incertidumbre y su origen en un experimento mental que Heisenberg planteó. La cuestión es que para medir la posición y la velocidad de un objeto debemos observar ese objeto y a partir de las observaciones hacemos las mediciones.

La palabra “observar” es aquí muy importante, pues «observar» no es algo que no es totalmente objetivo y que requiere de un «medio de observación». El modo de observación normal es por medio de fotones. Los fotones de luz impactan en un objeto que se mueve y nuestros ojos perciben la reflexión de esos fotones en dicho objeto.

Así, la precisión de una medida en una observación viene influida por los fotones usados en esa observación. Así Heisenberg plantea un microscopio de rayos gamma para la observación de un electrón en movimiento, visualizándose la reflexión de dichos rayos gama en una pantalla. Usa rayos gamma pues necesita algo que tenga una longitud de onda pequeña pues el electrón es muy pequeño. Usar fotones de luz visible daría una gran imprecisión en la posición dado que la luz visible tiene una longitud de onda muy grande comparada con el tamaño del electrón.

El problema es que a menor longitud de onda los fotones tienen más energía, energía que modificará la trayectoria y velocidad del electrón observado. Así a menor longitud de onda mayor precisión en la medida de la posición pero mayor error en la medida del momento de la partícula.

Podemos aproximar que el error en la medida de la posición vendrá dado por la longitud de onda de los fotones usados en la observación $\Delta x = \alpha$ y que el error en la medición del momento de la partícula vendrá dado por el momento del fotón $\Delta p = \frac{h}{\alpha}$

Así tenemos que el producto de ambos errores será

$$\Delta x \Delta p = \frac{h}{\alpha} * \alpha = h$$

Lo que da como resultado la ecuación del principio de incertidumbre:

$$\Delta x \Delta p = h$$

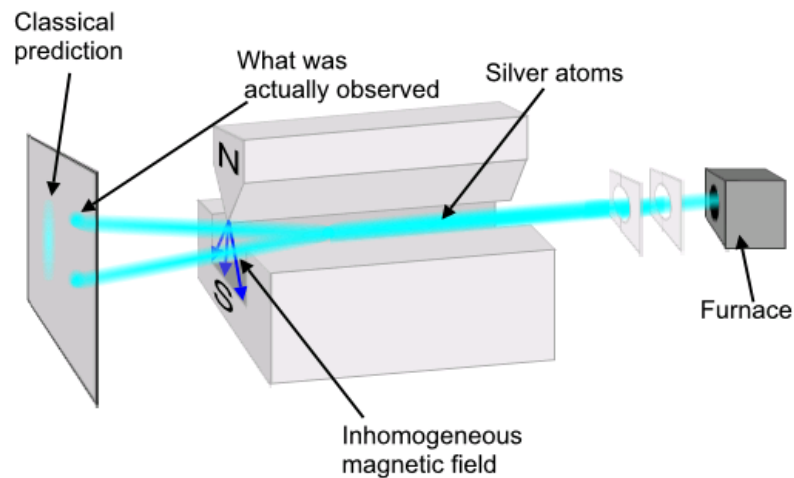
Lo que indica que el producto de los errores de medida de posición y momento de una partícula siempre tiene un valor constante que es la constante de Planck.

La principal consecuencia de este principio de incertidumbre es la conclusión de que jamás podremos conocer la posición y el momento de una partícula con precisión. A mayor precisión para la posición (menor longitud de onda de la luz para observar) menor precisión en su momento, y viceversa.

Apéndice 2: El espín

El nombre espín, proviene de la rotación de las partículas, el descubrimiento del espín se llevó a cabo en 1922 en el experimento de Stern y Garlach, se trata de una propiedad intrínseca de las partículas subatómicas así como la masa o la carga eléctrica. El espín proporciona una medida del momento angular intrínseco de cada partícula.

El experimento Stern y Garlach consiste en arrojar partículas (átomos de plata) a través de un campo magnético. Si las partículas poseen un momento magnético, entonces al pasar por el área del campo magnético, se desviarán de la trayectoria inicial. Esta desviación debería ser continua desde una posición máxima a una mínima.



Los átomos de plata son disparados y pasan a través del campo magnético desviándose hasta impactar en una lámina receptora

Los átomos al pasar por el campo magnético eran desviados, pero solo se alojaban en dos puntos de la lámina receptora, no dejaba un rastro entre ellas como la teoría “continua” predecía.

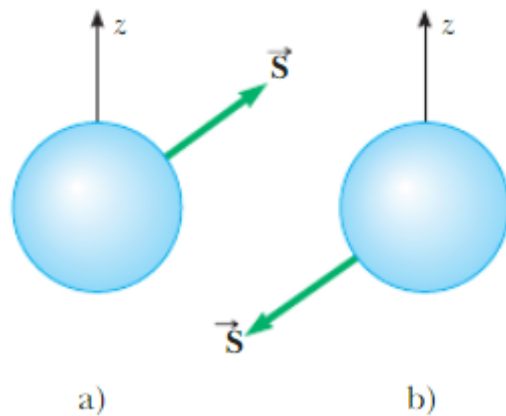
Este experimento pone en evidencia que los electrones tenían una característica que tomaba únicamente dos valores $\pm \frac{1}{2}\hbar$.

Los electrones que giran alrededor del núcleo de un átomo, generan un campo magnético que lleva asociado un determinado **momento angular**. Así mismo existe un momento angular asociado a cada partícula electrón, protón o neutrón y se describe mediante el **número cuántico de espín**

Para obtener este número cuántico, es conveniente (pero incorrecto) pensar en el electrón como si tuviera un movimiento de rotación sobre su propio eje, este movimiento es útil pero no correcto debido a que el movimiento de rotación está asociado a elementos macroscópicos “clásicos”, sin embargo el aparato matemático que describe el momento angular clásico es el mismo al que describe el espín del electrón.

El “giro” del electrón es solamente un efecto cuántico que le da al electrón una cantidad de movimiento angular como si estuviera girando

Solo existen dos direcciones aceptadas para el espín del electrón



El espín de un electrón puede ser a) hacia arriba o b) hacia abajo en relación con el eje z especificado. El espín jamás podrá estar alineado con el eje

Si la dirección respecto al eje x es como la figura a) entonces se dice que el electrón **gira hacia arriba**, si la dirección respecto al eje z es como la figura b) entonces se dice que el electrón **gira hacia abajo**.

La teoría de que el espín proviene del giro es incorrecta pero útil. Debido a que el electrón no posee coordenadas espaciales no puede situarse girando, además el electrón se considera una partícula sin amplitud espacial.

En 1928 Paul Dirac demostró que el número cuántico de espín proviene de las propiedades relativistas del electrón, además los experimentos demuestran que el electrón si posee un momento angular intrínseco, la pregunta es ¿Por qué vale $\pm \frac{1}{2}$?

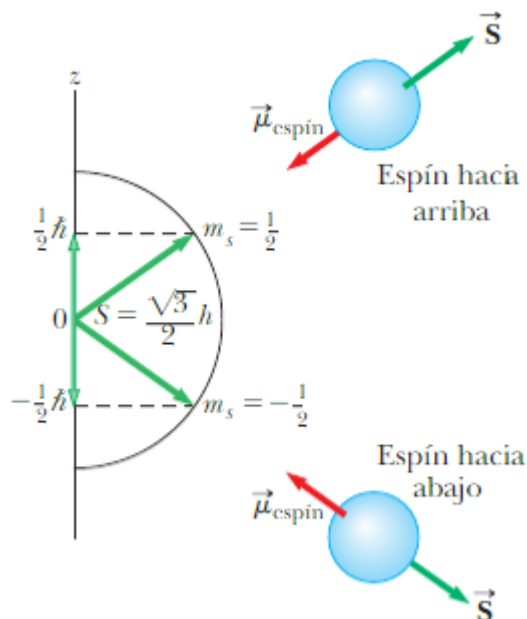
Resolución rigurosa del espín

Debido a la representación del espín como un giro hacia arriba o hacia abajo, la componente z del vector espín \vec{s} se encuentra cuantizado. Al estar cuantizado el número de orientaciones posibles del vector momento angular respecto a un eje z es $2n + 1$ esto es debido a que al n ser un entero arbitrario sus valores van de $-n$ a n pero contando el cero dando $2n + 1$ valores, siempre impar.

Debido a que el espín solo tiene dos orientaciones hacia arriba o hacia abajo, la ecuación queda:

$$2n + 1 \Rightarrow n = \frac{1}{2}$$

Luego n será el valor de espín llamado $s = \pm \frac{1}{2}$



La cantidad de movimiento angular del espín \vec{s} muestra cuantización espacial. Esta figura representa las dos orientaciones permitidas del vector cantidad de movimiento angular del espín \vec{s} y el momento magnético de espín \vec{u} para una partícula de espín $+1/2$ como lo es el electrón.

Números cuánticos

Los números cuánticos de un átomo están asociados a las características de sus electrones y a la resolución de la ecuación de Schrödinger para los mismos, para comprender el espín se deben entender primero los números cuánticos que le dan origen

El desarrollo de la ecuación de Schrödinger para átomos consiste en encontrar soluciones a la ecuación y aplicar condiciones de frontera para cada elemento. Un átomo es un elemento tridimensional, por lo tanto la energía potencial $U(r)$ depende de la coordenada radial r , la ecuación de Schrödinger tridimensional queda:

$$-\frac{\hbar^2}{2m} \left(\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} \right) + V\varphi = E\varphi$$

Esta ecuación es muy compleja de resolver en coordenadas cartesianas, por lo que se transforma a coordenadas esféricas con tres variables (r, θ, ϕ) , con $r = \sqrt{x^2 + y^2 + z^2}$ es la distancia radial desde el origen, θ es la posición angular respecto al eje z y ϕ es la proyección de la posición angular respecto al eje x, de esta manera la ecuación se transforma en:

$$\varphi(x, y, z) \rightarrow \varphi(r, \theta, \phi)$$

La ecuación general se resuelve como el producto de tres ecuaciones diferenciales particulares.

$$\varphi(r, \theta, \phi) = R(r)f(\theta)g(\phi)$$

Cuando se aplican un conjunto de condiciones de frontera a las tres ecuaciones simultáneamente, se obtienen 3 números cuánticos, más un cuarto número cuántico intrínseco.

Numero cuántico principal (n)

Es el primer número cuántico y está asociado a la función radial $R(r)$, las soluciones de esta ecuación diferencial ordinaria, dan la probabilidad de encontrar el electron a una cierta distancia radial del nucleo, asi como definir uno de los estados de energía permitidos.

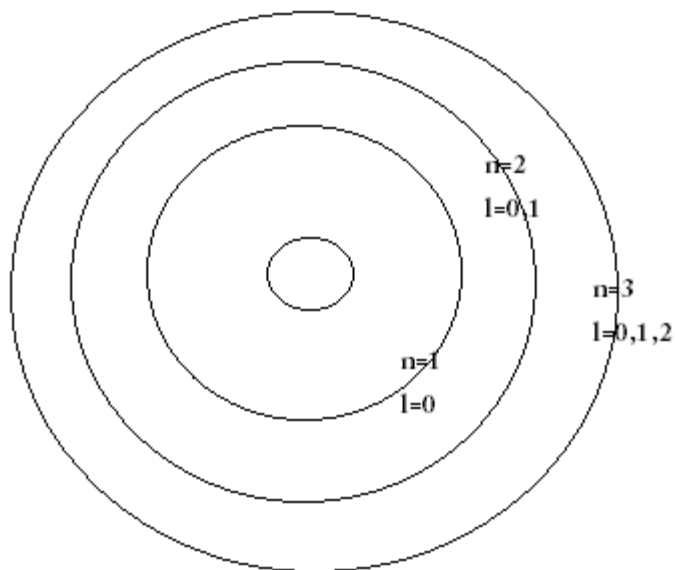
- n toma valores enteros positivos $n = 1, 2, 3, 4, \dots$
- La energía para cada nivel esta dada por la solución a la ecuación de Schrödinger

$$E_n = - \left(\frac{k_e e^2}{a_0} \right) \frac{1}{n^2} = \frac{-13,606 \text{ ev}}{n^2}$$

- Los valores de n son enteros que van desde 1 a ∞

$$E_1 = \frac{-13,606 \text{ ev}}{1^2} = -13,606 \text{ ev}$$

$$E_2 = \frac{-13,606 \text{ ev}}{2^2} = -3,4015 \text{ ev}$$



El estado fundamental de más baja energía ocurre cuando n es igual a 1

Número cuántico orbital (l)

Resulta de la ecuación diferencial de Schrödinger para $\rho(\theta)$ se asocia con el momento angular orbital del electron, las condiciones de frontera asociadas a este número son

-los valores de l son enteros que van desde 1 a $n - 1$, puede tener n valores

$$\text{Si } n = 1 \quad l = n - 1 = 0$$

$$\text{Si } n = 2 \quad l = 1, 0$$

$$\text{Si } n = 3 \quad l = 0, 1, 2$$

Este numero cuantico orbital define los valores que puede tomar el momento angular orbital del átomo completo.

Un átomo en un estado cuyo número cuántico principal es n , puede tomar los siguientes valores discretos de la magnitud de su cantidad de movimiento angular orbital

$$L = \sqrt{l(l + 1)} \hbar$$

$$l = 0, 1, 2, \dots, n - 1$$

L = magnitud del momento angular orbital del átomo

Clásicamente el vector \vec{L} se representa por

$$\vec{L} = m_e v \vec{r}$$

$$m_e = \text{masa del electron}$$

$$\vec{r} = \text{radio}$$

$$v = \text{velocidad}$$

Al estar cuantizado la magnitud \vec{L} debe ser un múltiplo de \hbar , así $L = N \hbar$, con

$$L = \sqrt{l(l+1)} \hbar$$

$$N = \sqrt{l(l+1)}$$

Número cuántico magnético (m_l)

Debido a que el momento angular orbital es un vector, debe especificar una dirección. Al ser el electrón una partícula con carga y estar girando, se produce un campo magnético \vec{B} . La dirección del vector momento magnético $\vec{\mu}$ respecto al vector campo magnético \vec{B} esta cuantizada y solo puede tomar valores discretos.

Esto conlleva a que la dirección de \vec{L} también este cuantizada de manera que la componente de \vec{L} sobre el eje Z es L_z , solo toma valores discretos.

El número cuántico magnético m_l define los valores discretos permitidos para la componente Z del a cantidad de movimiento angular orbital, m_l resulta de resolver la ecuación diferencial $g(\phi)$ considerando las siguientes condiciones de frontera:

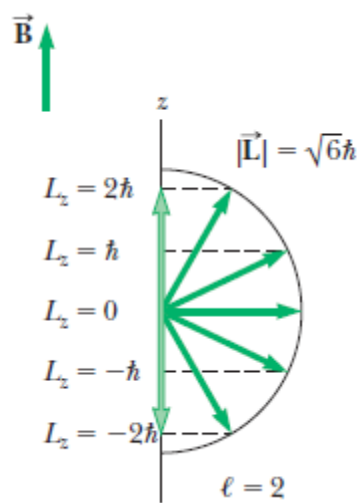
- Los valores de m_l son enteros que pueden ir de $-l$ a l
- La cantidad de números permitidos es $2l + 1$
- La componente L_z solo puede tomar valores cuantizados según la expresión

$$L_z = m_l \hbar$$

- Las orientaciones posibles de \vec{L} para un valor l van desde $-l$ hasta l

$l = 0$	$m_l = 0$	$L_z = 0$
$l = 1$	$m_l = -1,0,1$	$L_z = -\hbar, 0, \hbar$
$l = 2$	$m_l = -2,-1,0,1,2$	$L_z = -2\hbar, -\hbar, 0, \hbar, 2\hbar$

La grafica para l=2



La figura muestra la cuantizacion de las orientaciones en el eje Z, la magnitud $|\vec{L}|$ es

$$L = \sqrt{l(l + 1)} \hbar$$

$$L = \sqrt{6} \hbar$$

Numero cuántico magnético de espín (m_s)

Un cuarto número cuántico fue introducido para poder explicar el experimento Stern Garlach de manera correcta, este número cuántico no proviene de la ecuación de Schrödinger sino de un tratamiento relativista del electrón.

Los electrones al girar alrededor del núcleo generan un momento angular orbital \vec{L} , $L = |\vec{L}|$.

Al estar cargados también generan un momento magnético que cuantiza las orientaciones del momento angular orbital \vec{L} .

L cuantiza la cantidad de movimiento angular según $L = \sqrt{l(l + 1)} \hbar$ y m_l cuantiza la orientación en el eje Z del vector \vec{L} según $L_z = m_l \hbar$, teniendo en claro estos

conceptos se presenta un experimento que trataba de dar evidencias de estas cuantizaciones.

Experimento Stern y Garlach

En 1922 Otto y Walter realizaron un experimento con el propósito de demostrar la cuantización espacial. El experimento consistía en arrojar un haz de átomos de plata a través de un campo magnético no uniforme. Si las partículas se desvían en su trayectoria inicial, se esperaba que esta desviación se reflejara de manera continua en la lámina receptora, desde una posición máxima a una mínima.

Los resultados mostraban que los átomos al pasar por el campo magnético se desviaban, pero lo hacían solo en dos o más componentes discretas, sin dejar un rastro continuo entre ellas como la teoría clásica predecía.

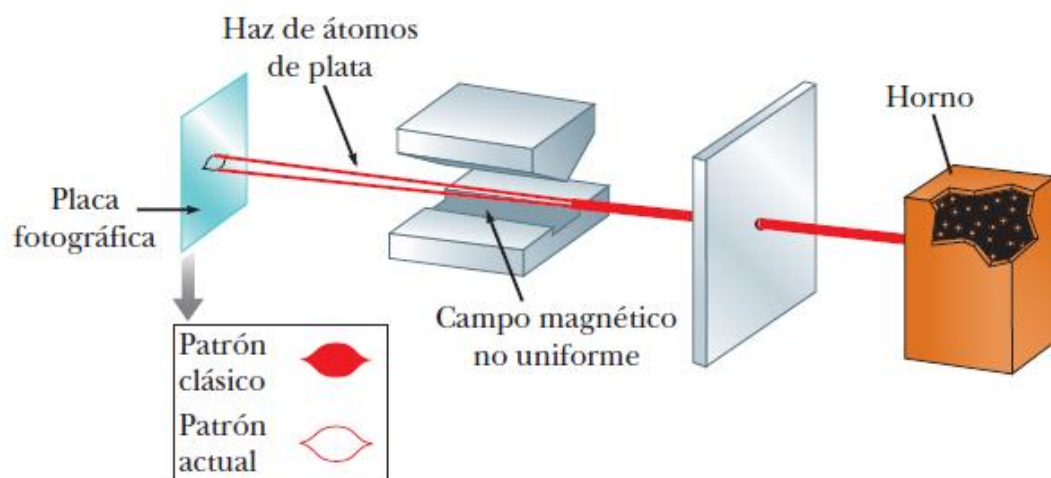


Figura. Técnica utilizada por Stern y Garlach para verificar la cuantización espacial. Un campo magnético no uniforme divide en dos un haz de átomos de plata.

El experimento demostró la cuantización de la componente Z del momento magnético pero surgía un nuevo problema.

Como se estudió en el número cuántico magnético m_l , la cantidad de valores para un número orbital l dado es siempre impar según $2l + 1$, esto es válido para cualquier valor entero de l

Esto es inconsistente con lo visto en el experimento ya que se aprecian dos zonas de desvío, con un número par de componentes en los desvíos.

En 1927 T.E.Phillips y B.Taylor repitieron el experimento Stern y Garlach pero utilizando un haz de átomos de hidrogeno. Se sabe que para el estado fundamental del átomo de hidrogeno se tiene:

$$n = 0$$

$$l = 0$$

$$m_l = 0$$

Por lo tanto el campo magnético del instrumento no debería desviar el haz de átomos debido a que no hay cantidad de movimiento angular orbital.

$$L = \sqrt{l(l+1)} \hbar = \sqrt{0(0+1)} \hbar = 0$$

Así mismo el momento magnético del átomo es igual a cero. Sin embargo en el experimento Phillip-Taylor el haz se dividió de nuevo en dos componentes, lo que significa que hay algo además del movimiento orbital del electrón, que contribuye al momento magnético del átomo

La única solución fue introducir una nueva forma de fuente de momento angular y un cuarto número cuántico EL ESPIN

El espín

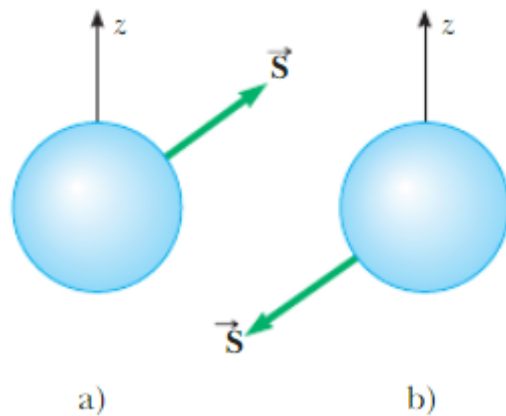
El espín o spin es una propiedad intrínseca de las partículas así como la masa o la carga y no puede derivarse del entorno. El espín proporciona una medida del momento angular intrínseco de la partícula.

Para obtener este nuevo número cuántico es conveniente (pero incorrecto) pensar en el electrón como si tuviera un movimiento de rotación sobre su propio eje, este movimiento es útil pero no correcto debido a que el movimiento de rotación está asociado a cuerpos macroscópicos “clásicos”, sin embargo el aparato matemático que describe el momento angular clásico es el mismo que describe el espín del electrón.

El “giro” del electrón es solamente un efecto cuántico que le da al electrón una cantidad de movimiento angular como si estuviera girando físicamente.

En 1928 Paul Dirac demostró que el numero cuántico de espín proviene de las propiedades relativistas del electrón, además los experimentos demuestran que el electrón si posee un omento angular intrínseco.

Solo existen dos direcciones aceptadas para el espín del electrón.



Si la dirección respecto al eje Z es como la figura a) entonces se dice que el electrón **gira hacia arriba** con espín up \uparrow

Si la dirección respecto al eje Z es como la figura b) entonces se dice que el electrón **gira hacia abajo** con espín \downarrow

Como la componente Z del momento angular orbital esta cuantizada, el espín nunca podrá estar alineado con el eje Z.

La teoría de que el espín proviene del giro es incorrecta debido a que el electrón no posee coordenadas espaciales, no puede situarse girando en el espacio, además el electrón se considera una partícula puntual sin amplitud en espacial

Con esta nueva variable el momento angular total del electrón contiene una contribución de momento angular orbital \vec{L} y una contribución de momento angular intrínseco \vec{S} de espín

Al igual que la cantidad de movimiento angular orbital \vec{L} la cantidad de movimiento angular de espín \vec{S} esta cuantizada en el eje Z recordando las orientaciones posibles para L_z son $2l + 1$, esto da un valor siempre impar de orientaciones (orientaciones positivas y negativas más el cero) para un número cuántico orbital l dado.

Pero el experimento Stern Gerlach mostraba que las partículas se alojaban solo en dos lugares de la lámina receptora.

Para encontrar el valor o los valores posibles del espín se debe añadir esta restricción a la ecuación $2l + 1$.

Sustituimos el número cuántico orbital 1 por el número cuántico de espín que queremos hallar m_s . Se utiliza la misma ecuación debido a que \vec{S} también está cuantizado en el eje Z como \vec{L}

$$2m_s + 1 = 2$$

$$2m_s = 1$$

$$m_s = \frac{1}{2}$$

La ecuación es igual a 2 por las dos orientaciones que se le permiten al espín $\downarrow\uparrow$ y los resultados del experimento

Despejando los posibles valores de $m_s = \pm\frac{1}{2}$. El espín debía ser semientero para poder dar lugar a solo dos tipos de giro y que no hubiese ausencia de este.

Luego la cantidad de movimiento angular del espín \vec{S} es:

$$|\vec{S}| = S = \sqrt{s(s + 1)} \hbar = \frac{\sqrt{3}}{2} \hbar$$

La componente Z del vector \vec{S} queda

$$S_z = m_s \hbar = \pm\frac{1}{2} \hbar$$

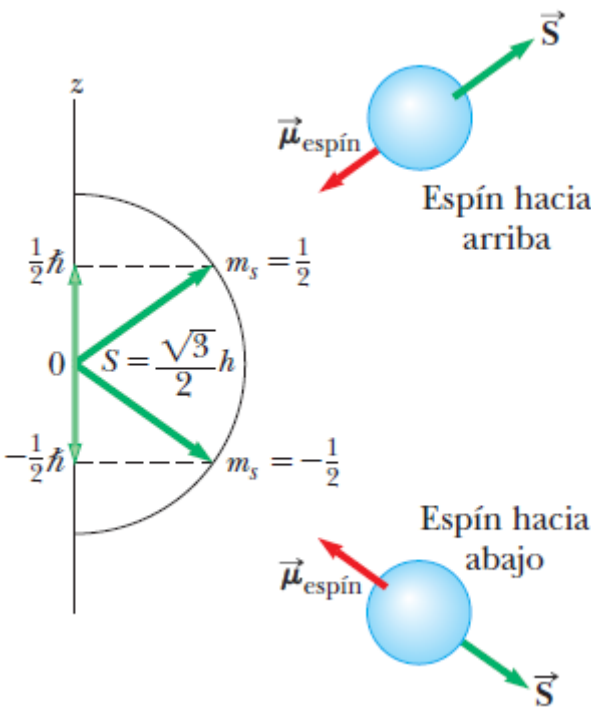


Figura. La cantidad de movimiento angular del espín \vec{S} muestra cuantización espacial. Esta figura representa las dos orientaciones permitidas del vector de la cantidad de

movimiento angular del espín \vec{S} y el momento magnético de espín $u_{espín}$ para una partícula de espín $\pm \frac{1}{2}$ como lo es el electrón.

La figura muestra los dos valores $\pm \frac{\hbar}{2}$ para S_z , corresponden a las dos orientaciones posibles para \vec{S} el valor $m_s = +\frac{1}{2}$ corresponde al espín hacia arriba y $m_s = -\frac{1}{2}$ corresponde al espín hacia abajo.

Con estos nuevos antecedentes se puede replantear la explicación para el experimento Stern y Garlach.

Experimento Stern y Garlach revisado

Al atravesar el campo magnético, los momentos magnéticos observados para los átomos de plata como para átomos de hidrogeno se deben solo a la cantidad de movimiento angular de espín \vec{S} , sin contribución alguna de la cantidad de movimiento orbital \vec{L}

Un átomo de un solo electrón como el hidrogeno cuantiza el espín en el campo magnético de tal forma que la componente Z de la cantidad de movimiento angular de espín ya sea $+\frac{\hbar}{2}$ o $-\frac{\hbar}{2}$, lo que corresponde a las dos posibilidades del número cuántico de espín $m_s = \pm \frac{1}{2}$

Los electrones con espín $+\frac{1}{2}$ son desviados hacia abajo y los electrones con espín $-\frac{1}{2}$ son desviados hacia arriba

Numero cuántico principal	n	1	2				3								
Numero cuántico orbital	1	0	0	1			0			1			2		
Numero	ml	0	0	1	0	-1	0	1	0	-1	2	1	0	-1	-2

cuántico magnético															
Numero cuántico de espín	m_s	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$	$\uparrow\downarrow$

La cantidad de electrones permitidos por capa viene dado por $2n^2$ que equivale a electrón con espín arriba y otro con espín abajo, en el caso de la última capa para $n=3$ tenemos $2 * 3^2 = 18$

Conclusiones

El estudio del espín es fundamental para comprender el entrelazamiento de partículas (el que es el fenómeno en que se basa el protocolo E92), cada vez que se haga alusión al espín, se considerara su flecha de giro hacia arriba o hacia abajo. El protón también posee un espín y su valor es $m_s = \pm \frac{1}{2}$ $\downarrow\uparrow$ de manera que independiente e la partícula que se utilice para interpretar información siempre puede usarse como un espín $up\uparrow$ y un espín down \downarrow

Apéndice 3: Ecuación de Schrödinger

La obtención de la ecuación de Schrödinger surge del tratamiento de los fenómenos cuánticos descubiertos hasta la fecha, aplicados al concepto de conservación de la energía. Al intentar aplicar los conceptos cuánticos al principio de conservación, surge la necesidad de transformar los principios clásicos en operadores cuánticos.

La ecuación de onda asociada a una partícula material es distinta a la ecuación electromagnética de Maxwell para fotones, porque las partículas materiales tienen energía en reposo diferente de cero. La ecuación correcta fue creada por Schrödinger en 1926. Al analizar el comportamiento de un sistema cuántico, el planteamiento es determinar una solución a la ecuación de onda electromagnética y luego aplicarle condiciones frontera apropiadas. La solución proporciona las funciones de onda permitidas u los niveles de energía para el sistema. La adecuada manipulación de la

función de onda hace posible, por tanto calcular todas las características medibles del sistema.

La ecuación de Schrödinger, como se aplica a una partícula de masa m confinada a moverse a lo largo del eje x e interactuar con su ambiente por medio de una función de energía potencial $U(x)$ es:

$$-\frac{\hbar^2}{2m} \left(\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} \right) + U\varphi = E\varphi$$

Donde E es una constante igual a la energía total del sistema (la partícula y su ambiente) se le conoce como ecuación de Schrödinger independiente del tiempo.

La ecuación de Schrödinger es consistente con el principio de conservación de la energía mecánica de un sistema. El primer término de la ecuación de Schrödinger se reduce a la energía cinética de la partícula multiplicada por la función de onda. Por lo tanto la ecuación indica que la energía total es la suma de la energía cinética y la energía potencial y que la energía total es constante $K + U = E = \text{constante}$

En principio si se conoce la función de la energía potencial U para el sistema, es posible resolver la ecuación y obtener las funciones de onda y energías para los estados permitidos del sistema. Porque U puede ser discontinuo con la posición, puede ser necesario obtener soluciones para diferentes regiones del eje x . las soluciones a la ecuación de Schrödinger en diferentes regiones deben unirse fácilmente en las fronteras, es necesario que sea $\varphi(x)$ continua. Además $\frac{d\varphi}{dx}$ debe ser continua para valores finitos de la energía potencial.

Apéndice 4: Espacios vectoriales complejos, espacio de Hilbert

Espacio vectorial

Un espacio vectorial lineal consiste en dos conjuntos de elementos y dos reglas algebraicas

Un conjunto de vectores φ, ϕ, x, \dots y un conjunto de escalares a, b, c, d, \dots

Una regla para la suma de vectores y una regla para la multiplicación de un vector por un escalar que satisfacen las siguientes reglas:

Reglas de suma:

- Si φ y ϕ son vectores (elementos) del espacio su suma $\varphi + \phi$ es también un vector del mismo espacio
- Conmutatividad $\varphi + \phi = \phi + \varphi$
- Asociatividad $(\varphi + \phi) + x = \varphi + (\phi + x)$
- Elemento neutro: debe existir un vector cero, perteneciente al espacio tal que $\varphi + 0 = 0 + \varphi = \varphi$ para todo vector φ
- Elemento simétrico: debe existir un vector simétrico $(-\varphi)$, perteneciente al espacio tal que $\varphi + (-\varphi) = 0$ para todo vector φ

Reglas de multiplicación:

- El producto de un escalar por otro vector da un vector. cualquier combinación lineal $a\varphi + b\phi$ es también un vector del mismo espacio
- Propiedad distributiva: $a(\varphi + \phi) = a\varphi + a\phi$ y $\varphi(a + b) = a\varphi + b\varphi$
- Propiedad asociativa: $a(b\varphi) = (ab)\varphi$
- Escalares unidad y cero: debe existir un escalar unidad I tal que $I * \varphi = \varphi * I = \varphi$, así como un escalar cero 0 tal que $0 * \varphi = \varphi * 0 = 0$

El espacio vectorial se denomina complejo o real dependiendo de si los escalares son números complejos o reales. por defecto se consideraran complejos los siguientes espacios vectoriales.

Producto interno o producto escalar

La definición del producto escalar es necesaria para generalizar los conceptos de distancias y ángulos entre vectores usados en el espacio Euclideo tridimensional permitiendo adoptar la visión geométrica tan familiar en dicho espacio.

Propiedades: dicho producto definido sobre un espacio vectorial debe cumplir los siguientes requisitos:

- El producto escalar entre el vector φ y el vector ϕ representado por (φ, ϕ) da un escalar
- El producto escalar entre φ y ϕ es igual al complejo conjugado del producto entre ϕ y φ (por tanto el orden es importante)

$$(\varphi, \phi) = (\phi, \varphi)^*$$

- Linealidad: $(\varphi, a\phi_1 + b\phi_2) = a(\varphi, \phi_1) + b(\varphi, \phi_2)$
- La norma o longitud de un vector $\|\varphi\|^2$ debe ser estrictamente positiva, donde la igualdad se cumple solo si $\varphi = 0$

$$(\varphi, \varphi) = \|\varphi\|^2 \geq 0$$

Dimensión y base de un espacio vectorial

Conjunto linealmente independiente: un conjunto de N vectores $\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_N$ se dice que es linealmente independiente si y solo si la única solución a la ecuación

$$\sum_{n=1}^N a_n \varphi_n = 0$$

Es $a_1 = a_2 = a_3 \dots = a_N = 0$. Un conjunto infinito de vectores es linealmente independiente si todo subconjunto finito lo es también.

Conjunto linealmente dependiente: sin embargo si existen un conjunto de escalares que no son todos cero, uno de los vectores se podrá escribir como combinación lineal del resto:

$$\varphi = \sum_{i=n}^N a_n \varphi_n$$

Entonces se dice que el conjunto $\{\varphi_n\}$ es linealmente independiente.

Dimensión: se define la dimensión de un espacio vectorial como el número máximo de vectores linealmente independientes que el espacio puede tener: será N —

dimensional si tiene N pero no N+1, siendo de dimensión infinita si tiene N vectores linealmente independientes para todo entero positivo N.

Base y ortogonalidad: se dice que el conjunto de vectores $\{\varphi_n\}$ es una base del sistema si es linealmente independiente y expande el espacio, de modo que todo vector del mismo puede escribirse como una combinación lineal del conjunto $\{\varphi_n\}$

$$\varphi = \sum_{n=1} a_n \varphi_n$$

La dimensión del espacio coincide con la dimensión de una base del mismo

Los coeficientes de la expansión se denominan componentes del vector φ en la base $\{\varphi_n\}$

Dos vectores son ortogonales si su producto escalar vale cero. Una base es ortogonal si $(\phi_i, \phi_j) = \delta_{ij}$

Espacio de Hilbert

Las funciones de onda en mecánica cuántica forman un espacio vectorial de dimensión infinita lo que complica su tratamiento desde el punto de vista matemático

De todos los espacios vectoriales infinito-dimensionales los espacios de Hilbert son, gracias a las propiedades que poseen, los mas sencillos desde el punto de vista matemático y los más cercanos a los espacios de dimensión finita.

Definición: un espacio de Hilbert consiste en un conjunto de vectores $\varphi, \phi, \chi, \dots$ Y un conjunto de escalares a, b, c,.....que satisfacen las tres siguientes propiedades:

- H es un espacio vectorial
- H tiene definido un producto escalar
- H es completo

Consideraciones adicionales

La idea intuitiva de espacio completo es que no hay nada pegado a X que no esté en X . Así por ejemplo, la recta real es un espacio completo, pero si eliminamos un punto, deja de serlo

Un espacio de Hilbert es separable si y solo si admite una base ortogonal numerable.

En todo espacio de Hilbert de dimensión finita, N , y también en espacios de Hilbert de dimensión infinita separables, todo vector puede expresarse mediante:

$$\varphi = \sum_{n=1} a_n \varphi_n = \sum_{n=1} (\varphi_n, \varphi) \varphi_n$$

Donde $\{\varphi_n\}$ es una base ortogonal y el sumatorio se extiende hasta N en el primer caso, siendo infinito en el segundo.

Ejemplos de espacios de Hilbert

Dimensión finita, espacio Euclideo N-dimensional: se denomina así a la generalización de los espacios de 2 y 3 dimensiones estudiados por Euclides y que forman la base de la geometría.

Es un espacio de Hilbert de dimensión finita en el que el producto interno es la generalización a N dimensiones del producto escalar ordinario.

Dimensión infinita espacios vectoriales de funciones: son espacios cuyos miembros son funciones que cumplen una serie de requisitos matemáticos.

En mecánica cuántica, un sistema es descrito por un espacio complejo de Hilbert que contiene las **funciones de onda** para los estados posibles del sistema

Es Apéndice 5: Cifrado Vigenere

Es un ejemplo típico de cifrado polialfabético cuya invención fue imputada erróneamente a Blaise de Vigenère, y que data del siglo XVI. La clave está constituida por una secuencia de símbolos del alfabeto $K = \{k_0, k_1, \dots, k_{d-1}\}$, de longitud d , y que emplea la siguiente transformación congruente lineal de cifrado:

$$E_k(m_i) = m_i + k_{i \bmod d} \pmod n$$

Siendo m_i el i -ésimo símbolo del texto claro y n el cardinal (longitud) del alfabeto de entrada. Como clave se puede utilizar cualquier palabra de una longitud por ejemplo entre 6 y 8 caracteres, que no tenga letras repetidas.

Para ver mejor esto, supongamos que con nuestro alfabeto español de 27 símbolos, queremos cifrar el texto en claro "PLAN", y que para el cifrado utilizamos como clave la palabra "SOL". La primera letra del mensaje, la P se cifrará con la primera letra de la clave, S, lo que indica que tenemos que hacer la sustitución monoalfabetica $E("P") = E(16) = (16 + 19) \bmod 27 = 8 = "I"$, ya que si A ocupa la posición 0, S ocupa la posición 19 en nuestro alfabeto. La letra L del mensaje se cifrará usando la letra O de la clave, y la letra A del mensaje se cifrará usando la letra L de la clave. Para la última letra del mensaje (N) volveremos a usar a primera letra de la clave (S).

Por lo tanto tenemos como resultado:

Mensaje	P	L	A	N
Clave	S	O	L	S
Cifrado	I	Z	L	F

Para facilitar las operaciones con este criptosistema, se dispone el llamado cuadro de Vigenère, que está formado por una matriz cuadrada de 27x27 en el caso de un alfabeto de 27 letras como el español. La primera fila de la matriz está formada por el alfabeto empezando por la letra A y acabando en la letra Z, la segunda por el alfabeto que empieza por la B y acaba en A, y así hasta la última fila, la 27ª, que empieza por las letras ZAB... y acaba con la letra Y.

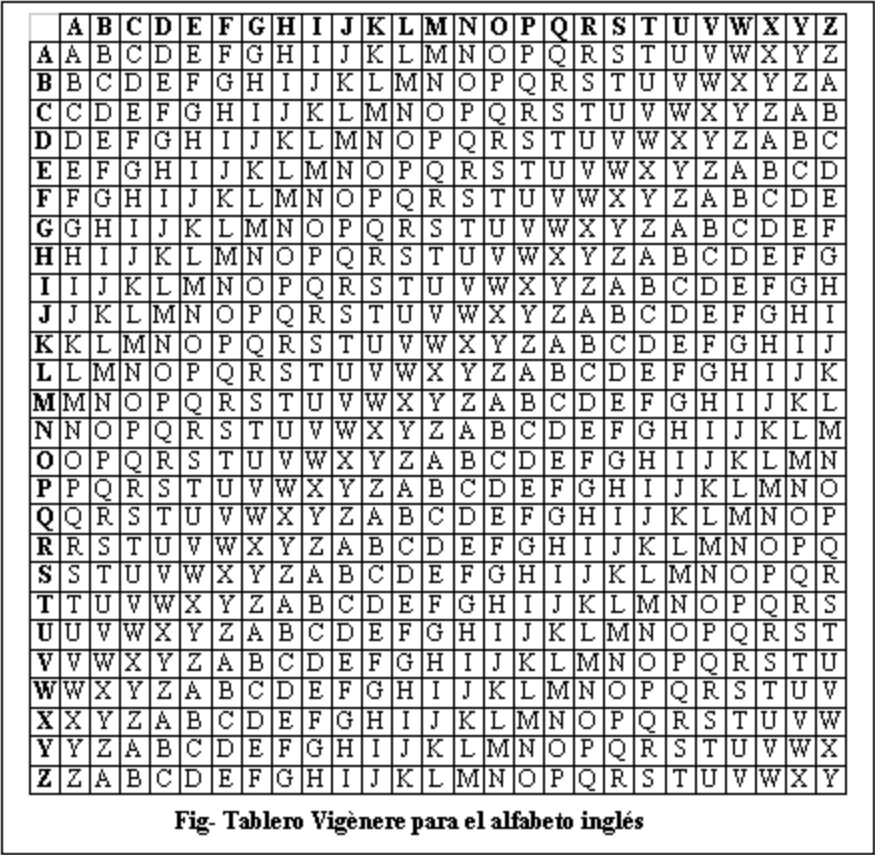


Figura 5.1. Tabla de alfabeto para el cifrado Vigenere.

Apéndice 6: Funcionamiento BB84

La comunicación e intercambio de la clave secreta ocurre en dos fases: Fase de comunicación cuántica y Fase de discusión pública.

Bases de polarización utilizadas:

Bit o Qubit	Angulo de polarización	Base
0>	0°	Rectilínea
1>	90°	Rectilínea
0’>	45°	Diagonal
1’>	135°	Diagonal

Tabla 6.1. El bit 0 se codifica en 0° y 45° mientras que el bit 1 se codifica a 90° y 45°.

1) Alice se comunica con Bob mediante un canal cuántico en el cual le envía una secuencia de bits cuyos estados están codificados en una secuencia de fotones polarizados en alguna de estas dos bases. Alice va eligiendo de manera aleatoria que base utiliza para polarizar cada fotón.

Bits Alice	0	1	1	0	1	0	0	1
Base de polarización Alice	R	R	D	R	D	D	D	R
Fotón enviado	→	↑	↖	→	↖	↗	↗	↑
Base de polarización Bob	R	D	D	D	R	D	R	R
Fotón detectado	→	↗	↖	↗	↑	↗	→	↑
Bit obtenidos RAW KEY	0	0	1	0	1	0	0	1
Discusión publica	v	x	v	x	x	v	x	v
Bit finales SIFTED KEY	0		1			0		1

Tabla 6.2. En la etapa de discusión pública se descartan los fotones con base errónea y también los fotones para los que el detector de Bob haya fallado. La “v” representa acierto y la “x” un fallo en la comprobación.

2) Bob recibe los fotones polarizados por Alice y va midiendo cada uno usando un filtro polarizador alineado a una de las bases de manera aleatoria. Anota los resultados y este conjunto de fotones intercambiados es lo que se conoce como llave en bruto o “Raw Key”. Así termina la fase de comunicación por canal cuántico.

3) Alice y Bob se comunican por medio de un canal público con mecanismos de seguridad clásica (como cifrado AES por ejemplo) e intercambian la información sobre las bases que usaron, Bob le anuncia que bases utilizo para medir los fotones si diagonal o rectilínea pero no anuncia cual fue la medición en si solo publica con que base midió. Alice por su parte también le informa que bases fueron las que se usaron para la polarización inicial y anotan en cuales han tenido éxito y cuáles no. Alice y Bob se ponen de acuerdo para descartar todos los bits donde Bob realizo una medición errónea o también se descartan todos los bits donde el detector de Bob haya fallado. Como estas mediciones fueron aleatorias ambos coincidirán en un 50% de los casos y la información en la polarización de estos fotones guarda información

secretamente compartida entre Alice y Bob sin posibilidades de que Eve haya podido modificarla, estos bits que concuerdan formaran lo que se conoce como llave depurada o “SIFTED KEY”

4) Una vez Alice y Bob tienen la SIFTED KEY deben verificar si Eve intercepto la comunicación. Para ello Alice y Bob comparten un pequeño grupo de bits que es subconjunto de la SIFTED KEY. Para este subconjunto de fotones se informa mediante el canal público cuales debieron ser los valores de bit que deberían tener, de manera que Bob mediante el canal público le comunica a Alice la polarización que uso para este pequeño conjunto de bits y le dice también que bits obtuvo, luego Alice le responde con la polarización correcta y el valor de bit correctos que debió haber obtenido. Los bits de Bob deben concordar un 100% debido a que son un subconjunto de la SIFTED KEY, luego este subgrupo de bits de prueba se descarta y no forma parte de la clave secreta.

Mientras más bits se prueben más probable será detectar un espía. Para cada bit probado la probabilidad de que esta prueba revele la presencia de un espía es del 25% o $\frac{1}{4}$. Entonces si se utilizan n bits de prueba la probabilidad de detectar un espía que esté presente es:

$$1 - \left(\frac{3}{4}\right)^n$$

Para 1 fotón de prueba $n = 1$

$$1 - \left(\frac{3}{4}\right)^1 = 0,25 = 25\%$$

Esto es lo que se conoce como “tasa de error de bit cuántico” (QBER. Quantum Bit Error Rating) lo que compara la porción de bits erróneos en relación al total de bits recibidos. La fortaleza del protocolo BB84 radica en que Alice y Bob pueden asegurar la presencia de Eve si encuentran que hay un porcentaje de errores igual o mayor a 25% es decir $QBER \geq 25\%$, en ese caso se descarta la llave [41].

Una vez Alice y Bob han acordado la clave, ellos pueden comenzar a transferirse información de manera segura.

5) Mientras Alice transmite sus estados a Bob, Eve puede estar escuchando el canal cuántico interceptando fotones y reenviándolos a Bob. Eve no sabe cuál de los cuatro estados ha enviado Alice así que lo único que puede hacer es escoger una de las dos

bases para realiza sus medidas y ver si detecta un fotón. En caso de que acierte y escoja la misma base de Alice. Eve podrá transmitir el estado correcto a Bob. En el caso de que seleccione una base distinta, medirá un bit incorrecto y debido al P.I.H la información original de ese bit se perderá, luego Eve transmitirá un estado incorrecto a Bob y este podrá detectar la presencia del espía.

Gracias al teorema de no clonación Eve no puede copiar un estado desconocido sin medirlo antes y por lo tanto inevitablemente modificara el estado original de Alice.

Eve está en la misma posición que Bob recibiendo un fotón que no sabe con qué base se polarizo y por tanto lo mide usando una base aleatoria de la misma forma que lo haría Bob, teniendo un 50% de probabilidad de acertar y al igual que Bob se equivocara el 50% de las veces. Cuando Alice y Bob se ponen de acuerdo en las bases que coinciden, esta información no le sirve a Eve porque solo en la mitad de las veces habrá acertado su medición, de manera que malinterpretara sus valores finales.

Los bits que conforman la clave compartida entre Alice y Bob pueden ser usados como una libreta de un solo uso en la cifra de Vernam “One-Time Pad”, cuando se vuelva a usar este One-Time-Pad, el proceso es repetido y una nueva cadena de información aleatoria sobre el canal cuántico será compartida.

Apéndice 7: Análisis B92 en presencia de un espía

El espía Eve puede escuchar y detectar fotones de la misma forma que lo hace Bob, pero debe reenviarlos de la misma forma que los recibió sino su presencia será detectada por Alice y Bob en la fase de comunicación pública.

Por ejemplo si Alice y Bob deciden en la fase de comunicación pública comparar el bit detectado y ven que no concuerdan, entonces pueden saber si hay un espía interceptando la comunicación.

Polarizacion Alice	Bit enviado	Detector Eve	Detecta?	Reenvia a Bob	Detector Bob	Detecta?	Bit recibido	Discusión publica
0°	0>	-45	NO	45°	90°	SI	1>	ERROR

Tabla 7.1. Detección de un espía en la etapa de discusión publica

En la tabla Alice prepara un bit $|0\rangle$ enviando un fotón polarizado a 0° . Eve se encuentra interceptando la comunicación y pone su detector aleatoriamente en base diagonal a -45° igual que lo haría Bob.

Si Eve no detecta el fotón entonces no sabe que polarización uso Alice para enviar, podría ser perpendicular al detector de Eve (bloqueándose) o podría simplemente no haber pasado el 50% de incertidumbre como en este caso. Si Eve usa una base incorrecta para reenviar a Bob, ocurrirá un Error y su presencia será descubierta.

Como en este caso no detecta el fotón de Alice, concluye que el fotón se bloqueó debido a que venía polarizado de manera ortogonal a su detector (lo que es un error). Si Eve intenta detectar a -45° y no logra detectar nada, entonces supone que Alice envió un fotón a 45° , por lo tanto reenvía a Bob un fotón polarizado a 45° en base diagonal. Bob alinea su detector en 90° y detecta el fotón ya que son bases no ortogonales y no se bloquea, luego Bob revisa su tabla para ver qué valor de bit corresponde con la polarización a 90° y ve que es un $|1\rangle$. Luego en la etapa de discusión pública detectan que Alice envió un 0 y Bob detecto un 1, lo que revela la presencia de Eve.

En el siguiente ejemplo Alice y Bob comparan el bit que han obtenido y concuerdan en que no hay error. En este caso no se puede saber que filtro polarizador uso Eve para detectar el fotón o si lo detecto o no.

Polarizacion Alice	Bit enviado	Detector Eve	Detecta?	Reenvia a Bob	Detector Bob	Detecta?	Bit recibido	Discusión publica
45°	$ 1\rangle$?	?	?	90°	SI	$ 1\rangle$	CORRECTO

Tabla 7.2. Presencia de Eve pasa desapercibida.

Alice y Bob solo pueden dar cuenta de Eve cuando hay error en la fase de discusión pública. En este caso Eve pudo haber medido usando un detector a -45° y el fotón se bloquea sin detección en los aparatos de Eve, por lo tanto reenvía el fotón a Bob con una polarización de 45° . Si Eve mide usando 90° entonces detectara el fotón y sabe

que viene polarizado a 45° , entonces reenvía el fotón a Bob con la misma polarización que lo recibió a 45° y Bob podrá detectarlo sin causar error.