UNIVERSITAT JAUME I

Grado en Matemática Computacional

Estancia en Prácticas y Proyecto Final de Grado

Algoritmo de descomposición de circuitos cuánticos en puertas de 2 niveles

Autor:
Tomàs Pérez Bort

Supervisor:
Héctor Gramaje Bodí
Tutor académico:
Vicent Gimeno García

Fecha de lectura: __ de _____ de 2019
Curso académico 2018/2019

Resumen

Este documento engloba mi estancia en prácticas y desarrollo teórico de la asignatura del Trabajo Final del Grado de Matemática Computacional por la Universitat Jaume I.

Durante mi estancia en prácticas en Agut Enginyería, me dediqué a programar en C# movimientos de Robots mediante matrices 4×4 de giro y traslación, y también aplicaciones gráficas en WPF.

Por lo que hace a la parte teórica de este TFG, tiene un primera parte matemática donde se presentan conceptos como las matrices unitarias, los espacios de Hilbert y los espacios duales. A continuación podemos encontrar una segunda parte donde se explican conceptos de física cuántica y se relacionan con la primera parte. Para cerrar la parte teórica, presentamos nuestro principal resultado, un algoritmo que descompone cualquier circuito cuántico en puertas de 2 niveles, haciendo uso de las propiedades vistas anteriormente.

Para concluir, hemos aplicado el algoritmo de descomposición al circuito de teleportación cuántica, para recrearlo usando puertas de 2 niveles exclusivamente.

Palabras clave

Qbit, Matrices unitarias, Espacio de Hilbert, Notación de Dirac, Espacio Dual.

Keywords

Qbit, Unitary matrix, Hilbert space, Bra-ket notation, Dual space.

Índice general

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

Con el objetivo de aprovechar la estancia en el programa estudia e investiga de la UJI, tanto mi tutor como yo, decidimos trabajar un tema que no hubiésemos visto con antelación. De esta manera nos supondría reto añadido.

Escogimos trabajar el tema de la computación cuántica por su gran potencial a medio-largo plazo. Y es que desde el principio de la computación siempre se ha buscado obtener la máxima potencia, ya sea mejorando el procesador o desarrollando la programación paralela. Pero cuando estos campos están llegando a sus limitaciones físicas, entra en juego algo que desde hace años ha estado en un segundo plano. La computación cuántica, con la cual se pueden reinventar por completo las bases de los ordenadores para incrementar su velocidad de computación de manera exponencial. Otro motivo por el cual trabajar sobre esto, fue por encontrar varios manuales que abordaban el tema desde un punto de vista plenamente físico y pasaban por alto, sin ninguna desarrollo, la muy interesante parte matemática.

Por otra parte, en lo referente a las prácticas, ha resultado imposible realizarlas estrictamente ligadas al contenido teórico de este TFG. A pesar de ello cuando salió la posibilidad de hacer las prácticas enfocadas a robótica, no lo dudé ya que es uno de los temas que más me llama la atención. La robótica me resulta un tema muy interesante debido a que desde hace años se ha pretendido automatizar las tareas tediosas o imposibles para el ser humano. También me resulta fascinante a nivel financiero, ya que debes ser más rápido que tus competidores directos, ofrecer algo diferente, mejor y minimizar los costes en la medida de lo posible a la hora de construir un robot funcional. Además, a la hora de programar este tipo de robots, se usa mucha geometría en 3D, con giros y composiciones de movimientos, en las cuales debes tener en cuenta

las limitaciones físicas.

Ahora bien, por lo que hace a este trabajo lo he distribuido en 5 partes. La primera es la introducción, donde se encuentra esta sección. Para presentar el contexto y los marcos del trabajo. El capítulo 2 es el desarrollo de mi estancia en prácticas en Agut Enginyeria SL. En el capítulo 3 reside la parte que hemos ido desarrollando Vicent y yo. Está dividido en 2 partes, en la primera se introducen unos conceptos matemáticos necesarios para entender la segunda parte, la cual se basa en la aplicación de los mismos. A continuación tenemos el capítulo 4, donde mostramos el algoritmo al que hemos llegado, y lo aplicamos en unos ejemplos. La sección culminan con la construcción de un importante circuito cuántico usando únicamente puertas 2×2 obtenidas por el algoritmo. Para cerrar el documento tenemos el capítulo 5 , que contiene las conclusiones de todo este año de trabajo.

Capítulo 2

Estancia en prácticas

2.1. La empresa

La empresa en la cual he realizado las prácticas ha sido Agut Enginyería SL, una empresa internacional de robótica situada en Carrer de L'Històric Regne de Valéncia, 36, 12550 Almassora, Castelló.

En ella se trabaja programando y poniendo a punto robots *Kuka*, junto al software que estos utilizan. La mayoría de clientes compran estos robots y el software para hacer encimeras de cocina o esculturas.

Desde el día que fuimos a visitarla, dejamos bastante claro por las dos partes que queríamos que fuese lo más provechosa posible. Acordamos, que desempeñaría una labor que me permitiera aprender, mientras hacía cosas útiles para la empresa.

2.1.1. Plantilla

En la empresa hay contables, técnicos, ingenieros y también informáticos. Unas 12 personas en toda la empresa. Hay un único jefe, que es quien montó la empresa y toma las decisiones en todos los campos de la empresa.

Las personas con las cuales he trabajado han sido mi tutor, Héctor, ingeniero informático que lleva 3 años en la empresa y Toni, ingeniero mecánico, que lleva allí 5 años.

Por lo que a Héctor respecta, siempre estaba en el ordenador de mi lado, y para cualquier duda le podía preguntar. Era quien me iba mandando trabajos y revisando por encima mi código antes de subirlo.

Toni, por otro lado era quien acabaría usando el software que yo iba a programar. Me iba dando diciendo sus preferencias para que yo las fuera implementando en las aplicaciones. Además me explicaba todo lo que necesitaba saber sobre los brazos robot: funcionamiento, ejes, limitaciones físicas...

2.1.2. Perfil

En Agut buscaban un perfil más matemático para complementar al informático que trabaja allí. Ya había trabajado un matemático computacional de la UJI y estaban contentos con el resultado.

En el caso de los informáticos de Agut se requiere conocimiento de bases de datos (SQL), programación en C#, robótica, WPF, *RhinoCeros*. También inglés fluido, tanto hablado como escrito para atender llamadas o consultas de los clientes.

Además de lo anterior, he acabado usando,unas nociones básicas de dibujo técnico para entender mejor el entorno gráfico de *RhinoCeros*, conocimientos de geometría para tratar movimientos del robot, y estructuras de programación para simplificar el código.

2.2. Objetivos del proyecto formativo

En cuanto a los objetivos se refiere, eran bastante claros desde el principio. Acordamos que intentaría ayudar en todo lo posible a la empresa con el trabajo que me dieran para descargar a mi tutor, mientras iba aprendiendo cosas útiles. En definitiva, debería ir modificando algunos programas de la empresa que por falta de tiempo y de personal llevan tiempo necesitando actualizaciones.

La idea era, que al final de las prácticas tuviera un manejo decente de C#, diseño de aplicaciones en WPF, SQL y *RhinoCeros*. Conseguí acabar los 2 grandes proyectos que se me pidieron y a día de hoy los siguen usando.

2.3. Explicación detallada del proyecto realizado en la empresa

Antes de empezar con los proyectos de la empresa como tal, me tuve que familiarizar con distintos programas que me iban a acompañar durante mi estancia en prácticas. Instalé y aprendí a usar en profundidad el entorno de programación *Visual Studio*: refactorizaciones, atajos de teclado, el modo *debug*... Lo conecté con la cuenta de *GitHub* de la empresa para tener un repositorio de código y compartir lo que hiciera sin dificultad. También instalé y apredí a usar librerías de *RhinoCeros* en C#. Aprendí cosas básicas de Rhino y algo de *SQLserver* 2015.

Una vez familiarizado con estos programas ya podía empezar a programar. En primer lugar me puse con una aplicación para crear o modificar entradas en una base de datos. De aquí aprendí bastante sobre crear aplicaciones gráficas, y como usar bases de datos en programas. Además de ganar la noción de que muchas veces el usuario final no sabe casi nada de informática por tanto hay que dejar interfaces muy sencillas para minimizar la cantidad de errores y confusiones.

A continuación hice algunos retoques en los proyectos que se me iban pidiendo. Por ejemplo, hice un método que tomaba una captura de pantalla de la simulación de la piedra una vez movida y la añadía a la cola de trabajo para poder ver el trabajo que se estaba realizando.

Por último, y como proyecto más importante se me pidió actualizar 2 programas ya existentes que se les estaba dando uso: *Manipulat Manual* y *Manipulat Automàtic*. Los programas representaban las piedras con las cuales se va a trabajar, sobre las bases de trabajo. En ellos, se podía modificar la posición original de la piedra, la base sobre la cual se trabajaba y todo ello era representado en 2D para que el usuario lo pudiera observar y modificar con una sencilla interfaz. Añadí un modelo 3D para visualizar el origen de la pieza que se movía con la piedra, creé una pestaña para cambiar entre los 2 modos de cada piedra, programé los movimientos del nuevo modo, y modifiqué algunos de los antiguos. Encapsulé información para usarla des de los 2 programas evitando repeticiones.

En estos programas empecé a apreciar el modo *debug*, ya que es con diferencia el código más grande con el que he trabajado nunca.

2.3.1. Matrices de giro y traslación

Para programar los movimientos del robot *Kuka*, calculamos coordenadas, las añadimos a un fichero de texto sin formato y esto es lo que se le pasa al robot para que vaya haciendo los movimientos. Para hacer el cálculo de las coordenadas usamos unos objetos ya implementados en las librerías de *RhinoCeros*, las matrices de giro y traslación :

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{p}_{3\times1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix}.$$

En nuestro caso, observamos la tabla en cenital. Por tanto podemos reducir el problema a $2D$, y se nos queda una matriz más sencilla :

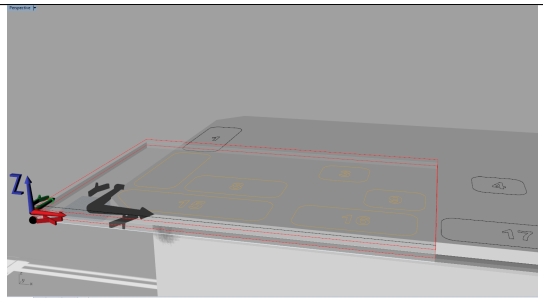
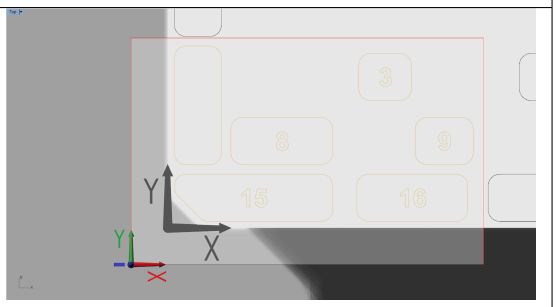
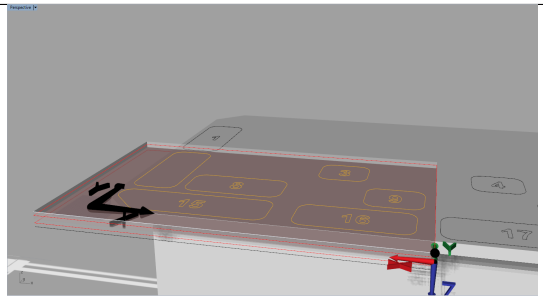
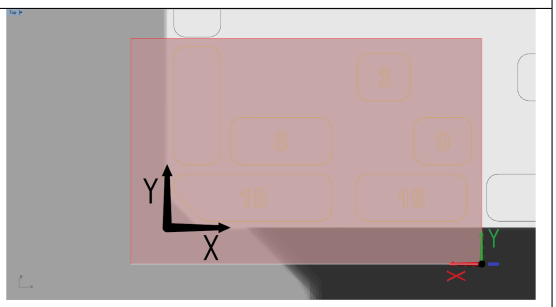
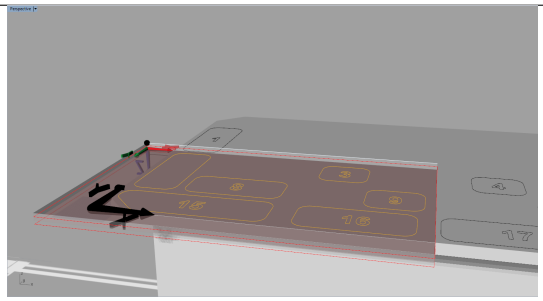
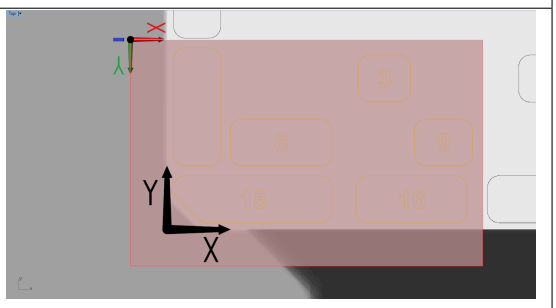
$$\mathbf{T}(\mathbf{v},\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

De esta manera se calcula el giro y el desplazamiento con una única matriz. Así tenemos la información agrupada. Además, son muy interesantes desde el punto de vista matemático ya que el grupo de estas matrices se identifica con el grupo SE(3).

2.3.2. Volteo

El volteo es un giro sobre los ejes X ó Y, pero sabiendo que la piedra debe acabar apoyándose en la bancada por una de sus bases. En este caso era muy sencillo, pues las únicas posibilidades eran giros en 0 ó 180 en los ejes B y C del robot (Y, X).

Estos giros dan la posibilidad de dar la vuelta a la piedra para realizar pulidos, cortes o acabados con CNC por la base inferior de la misma.

Eje	Perspectiva	Zenital
A		
B		
C		

2.3.3. Robots Kuka

Son robots industriales ligeros de 6 ejes. De gran alcance y aptos para cualquier tipo de manipulación. Existen de muchos tipos, para pared, suelo, techo, en ángulo... Y también de varios tamaños. Por el taller de la empresa han habido de todo tipo, mientras se ponían a punto y se preparaba su software.

Por sus 6 ejes tiene gran variedad de movimiento, y al poder cambiar de cabezal, abarca una gran variedad de tareas.

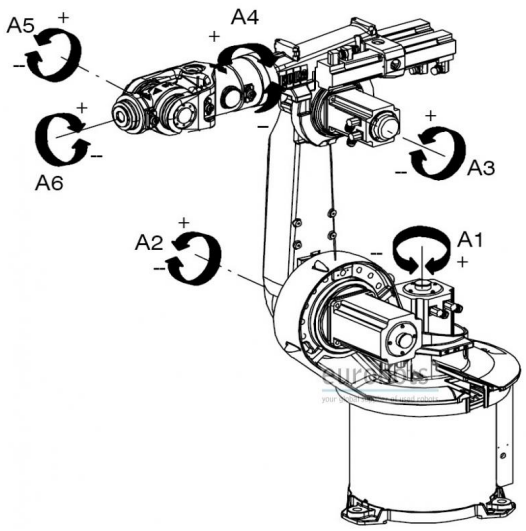


Figura 2.1: Ejes del robot *Kuka*

2.4. Conclusiones

En definitiva, estoy muy contento con la estancia en prácticas en Agut, ya que he aprendido mucho con un trato muy cercano.

De toda la experiencia me quedo con varias lecciones: la utilidad del modo *debug* para proyectos muy grandes, trabajar para gente que no es de informática requiere que te adaptes a ellos, y que internet es una herramienta básica para programar.

Además ha sido una gran experiencia a la hora de coordinarme con gente de otros campos y trabajar en algo aplicado usando los conocimientos que he ido adquiriendo durante el grado.

Capítulo 3

Memoria TFG

3.1. Motivación y Objetivos

Hemos escogido este tema ya que muy interesante y si sigue adelante, dentro de unos años podría suponer una revolución total de los ordenadores tal y como los conocemos. Los objetivos eran claros, reunir en un documento todo lo aprendido durante el programa estudia e investiga, además de crear un manual des de un punto de vista matemático sobre este tema, ya que cabe destacar que todos los que encontramos a penas tocaban el punto de vista matemático o lo obviaban.

Ahora bien, por poner un poco de contexto. ¿Qué es un ordenador cuántico? ¿Por qué es mejor, más potente o más rápido que uno clásico? Pues bien, los dos conceptos relevantes de un ordenador cuántico para ser mejor que uno clásico son la superposición y el entrelazamiento. Por lo que respecta a la superposición, sabemos que se trata de cuando un objeto posee simultáneamente más de un valor. En cambio, el entrelazamiento es cuando los estados cuánticos de varios objetos se deben describir mediante un estado único. Incluso si los objetos están separados físicamente. También necesitamos el concepto de qbit, que en su momento introduciré de una manera más formal. Pero por ahora nos vale con entender que el qbit es el la unidad mínima de información en la computación cuántica. Con estos 3 conceptos, ya podemos ponernos en situación.

Históricamente, se puede decir que la computación cuántica empezó su camino con la presentación del primer ordenador cuántico, en 1998. Funcionaba con tan sólo 2 qbits, y necesitaba una máquina de resonancia magnética nuclear para resolver un problema que para una computadora clásica era sencillo. Este dato nos da que pensar, y es que construir una computadora cuántica es una tarea difícil. En lugar de usar transistores de silicio, se deben crear dispositivos

que atrapen átomos individuales, y materiales superconductores que conduzcan la corriente sin resistencia. Además se debe trabajar a una temperatura cercana al cero absoluto (-273 ºC). Por si esto fuera poco, cada qbit que adicional complica más la máquina. A todo esto hay que añadirle que los qbits no suelen ser estables, es decir, son muy sensibles a las perturbaciones y al ruido. Lo cual lleva a errores en cálculos sencillos y también a que el ordenador deje ser cuántico (debido a que se elimina la superposición). También hay que tener en cuenta que si se entrelazan demasiados qbits, el sistema colapsa. Por si esto aún esto fuera poco, hay que recalcar que no es posible leer el estado de un qbit sin destruirlo. Por tanto, un supuesto sistema de corrección de errores debería funcionar prediciendo y corrigiendo errores que aún no se han producido. Además, como el sistema de corrección de errores también estaría sujeto a las leyes de la física cuántica, también podría fallar. Y este sistema es la única manera de comprobar unos cálculos que no se pueden realizar de ninguna otra manera.

Ahora bien, ¿por qué a pesar de todos estos inconvenientes se sigue investigando este campo? Pues la respuesta es que se está llegando al límite de la computación clásica. En 1965 Gordon Moore enunció la famosa ley de Moore, la cual enunciaba que el número de transistores en un microprocesador se duplicaría aproximadamente cada 24 meses. Ahora mismo, se está muy cerca de los límites físicos de la producción de transistores, pues llega un momento que no se puede reducir más su tamaño. Otra manera de incrementar la potencia de un computador es la computación paralela. Pero de la misma manera se ve limitada por la ley de Moore.

Entonces, además de que la computación clásica esté llegando a sus límites físicos, ¿qué aporta la computación cuántica respecto a las anteriores maneras de mejorar la capacidad de cómputo?

Pues para descubrirlo debemos introducir unos pocos conceptos más. En un primer lugar tenemos el algoritmo de temple cuántico, que sirve para resolver problemas de optimización o muestreo. Está basado, por lo general, en el algoritmo de Montecarlo. Es decir, repite una gran cantidad de muestreos aleatorios sobre espacio de soluciones del problema. Permitiendo de esta manera reducir mucho la complejidad de cómputo a costa de perder algo de precisión estadística. Cabe destacar que el algoritmo de temple cuántico tiene aplicaciones muy interesantes en Machine Learning. Otro concepto muy interesante en computación cuántica es el Algoritmo de Shor (1995). Es un algoritmo cuántico para descomponer en factores un número N en tiempo $O((logN)^3)$ y espacio $O(logN)$. Este algoritmo es muy importante en temas de criptografía ya que con la llegada de la computación cuántica (y del algoritmo cuántico de Shor), los sistemas clave pública dejarían de ser seguros.

De momento, de forma similar a los inicios de la computación clásica, la computación cuántica se va a limitar a empresas grandes que puedan aplicarla a problemas concretos. Ahora mismo están trabajando con dispositivos cuánticos de manera medianamente estable con 20 qbits, y las empresas punteras del sector están investigando con dispositivos de 50, 72 y 160. Y a pesar de que a corto plazo, la computación cuántica no revolucionará la informática, podría promo-

ver grandes avances científicos que veríamos en nuestras vidas diarias. Con todo esto podemos observar que aún queda un largo camino a recorrer.

3.2. Preliminares Matemáticos

En esta primera fase del trabajo teórico, empezaremos introduciendo conceptos matemáticos que usaremos más adelante.

3.2.1. Grupo Unitario

Definición 1 ([5]). *El grupo lineal $GL_{n\times n}(\mathbb{C})$ es el conjunto formado por las matrices complejas $n \times n$, regulares, es decir:*

$$GL_{n\times n}(\mathbb{C}) := \{u \in M_{n\times n}(\mathbb{C}) \,|\, det(u) \neq 0\},$$

siendo $M_{n\times n}(\mathbb{C})$ el conjunto de las matrices $n \times n$ con coeficientes complejos.

Proposición 1 ([5]). *El producto usual de matrices dota a $GL_{n\times n}(\mathbb{C})$ de estructura de grupo.*

A partir del grupo lineal podemos definir el grupo unitario.

Definición 2 ([5]). *El grupo unitario, $U(n)$ es el conjunto formado por las matrices complejas $n \times n$, unitarias, es decir:*

$$U(n) := \left\{u \in M_{n\times n}(\mathbb{C}) \,|\, uu^\dagger = u^\dagger u = \mathbf{1}_{n\times n}\right\},$$

donde $u^\dagger$ denota la matriz conjugada transpuesta de la matriz u, y $\mathbf{1}_{n\times n}$ es la matriz identidad de orden n.

Proposición 2 ([5]). $U(n) \subset GL_{n\times n}(\mathbb{C})$

Demostración.

$$\begin{aligned} uu^\dagger = u^\dagger u = \mathbf{1}_{n\times n} &\Rightarrow u^\dagger = u^{-1} \\ &\Rightarrow \forall\ u \in U(n)\ \exists\ u^{-1}/uu^{-1} = \mathbf{1}_{n\times n} \\ &\Rightarrow \forall\ u \in U(n),\ det(u) \neq 0 \end{aligned}$$

□

Proposición 3 ([5]). *El producto usual de matrices dota a $U(n)$ de estructura de grupo.*

Corolario 1 (Ver también [5]). *$U(n)$ es subgrupo de $GL_{n\times n}(\mathbb{C})$.*

3.2.2. Espacios de Hilbert

Sea X un espacio vectorial sobre $\mathbb{C}$. Sea $\langle\ ,\ \rangle$ un producto hermítico en X. Diremos que $(X, \langle\ ,\ \rangle)$ es un espacio de Hilbert si $\left(X, d_{\langle\ ,\ \rangle}\right)$ es un espacio métrico completo, donde $d(x,y) = \sqrt{\langle x-y, x-y\rangle}$

El espacio formado por los estados cuánticos, matemáticamente se puede describir como un espacio de Hilbert. En concreto, nosotros trabajaremos con el espacio $(\mathbb{C}^n, \langle\ ,\ \rangle)$.

Donde $\langle\ ,\ \rangle : \mathbb{C}^n \times \mathbb{C}^n \longrightarrow \mathbb{C}$

$$v,\ w \ \longmapsto\ \langle v, w\rangle = v^\dagger w = \textstyle\sum_{i=1}^n v_i^* w_i$$

En este capítulo veremos que $\langle\ ,\ \rangle$ es un producto hermítico en $\mathbb{C}^n$, que lo convierte en espacio normado, y por lo tanto un espacio métrico de dimensión finita, es decir, es completo, ergo es espacio de Hilbert.

Empezamos definiendo lo que es un espacio vectorial normado.

Definición 3 ([2]). *Sea X un espacio vectorial sobre un campo de Complejos. Una norma de X , denotada por $\|\ \|$, es una función real con las siguientes propiedades:*

i) $\|x\| \ \geq\ 0\ \forall\ x \in X$

ii) $x \neq 0\ \Rightarrow\ \|x\| \neq 0$

iii) $\|\alpha x\| \ = |\ \alpha\ |\ \ \|x\|$

iv) $\|x + y\| \ \leq\ \|x\| + \|y\|$

Ejemplo 1 ([2]). *N-espacio euclídeo, denotado por E^n, es el espacio lineal normado de n-tuplas de números reales con la norma:*

$$\|(a_1, a_2, ..., a_n)\| = \left(\sum_{i=1}^{n} |\ a_i\ |^2\ \right)^{1/2}$$

Definamos ahora lo que es una métrica.

Definición 4 ([3]). *Una métrica sobre un conjunto X es una función $d : X \times X \to [0, \infty)$ que satisface:*

i) $d(x, y) \geq 0$

ii) $d(x, y) = 0 \Leftrightarrow x = y$

iii) $d(x, y) = d(y, x)$

iv) $d(x, z) \leq d(x, y) + d(y, z)$

Llamaremos al par (X, d) espacio métrico.

Proposición 4 ([2]). *Sea X un espacio lineal normado. $d(x, y) = \|x - y\|$ es métrica.*

Demostración.

i) $\|x - y\| \geq 0$

ii) $\|x - y\| = 0 \Leftrightarrow x = y$

iii) $\|x - y\| = \|y - x\|$

iv) $\|x - z\| \leq \|x - y\| + \|y - z\|$

□

Corolario 2. *Todo espacio vectorial normado es espacio métrico*

La métrica topológica en X , denotada por d será la topología usada de ahora en adelante. En las próximas 2 definiciones recordaremos lo que es un espacio métrico completo.

Definición 5 ([2]). *Sea (X, d) un espacio métrico, una sucesión $\{x_n\}$ se dice sucesión de Cauchy si $\forall\ \epsilon > 0\ \exists\ N > 0$ tal que $d(x_n, x_m) < \epsilon$ si $n > N$ y $m > N$.*

Definición 6 ([2]). *Un espacio métrico (X, d) se dice que es completo si toda sucesión de Cauchy contenida en X converge a un elemento de X.*

Como ya hemos mencionado anteriormente, vamos a trabajar en un espacio de dimensión finita, por tanto enunciamos el siguiente teorema:

Teorema 3.2.1 ([2]). *Todo espacio normado lineal de dimension finita es completo*

Continuemos con la definición del producto hermítico:

Definición 7 ([2]). *Sea X un espacio vectorial sobre los complejos. El producto hermítico en X es una función escalar $\langle\ ,\ \rangle$ definido en el producto Cartesiano $X \times X$ con las siguientes propiedades:*

i) $\langle \alpha x, y\rangle = \alpha\langle x, y\rangle$

ii) $\langle x, y\rangle = \overline{\langle y, x\rangle}$ *, es decir* $\langle x, y\rangle$ *es el complejo conjugado de* $\langle y, x\rangle$

iii) $\langle x + y, z\rangle = \langle x, z\rangle + \langle y, z\rangle$

iv) $\langle x, x\rangle > 0\ \forall\ x \neq 0$

Veamos ahora que todo espacio vectorial con producto hermítico es automáticamente un espacio normado (y por tanto, un espacio métrico).

Proposición 5 ([2]). *Sea V un espacio vectorial complejo de dimensión finita. Sea $\langle\ ,\ \rangle : V \times V \to C$ un producto hermítico en V. Entonces :*

$$\|\ \| : V \to R$$

$$v \longmapsto \|v\| = \sqrt{\langle v, v\rangle}$$

es una norma sobre V.

Demostración.

1. $\langle v, v\rangle = \overline{\langle v, v\rangle}$
2. $\langle \alpha v, \alpha v\rangle = \alpha\langle v, \alpha v\rangle = \alpha\overline{\alpha}\langle v, v\rangle = |\alpha|^2\langle v, v\rangle$
3. $\|\alpha v\| = \sqrt{|\alpha|^2\langle v, v\rangle} = |\alpha|\ \|v\|$

□

Introduzcamos ahora la definición de espacio de Hilbert:

Definición 8 ([2]). *Sea $(X, \langle\ ,\ \rangle)$ un espacio vectorial sobre $\mathbb{C}$. Sea $\left(X, d_{\langle\ ,\ \rangle}\right)$ el espacio métrico asociado, i.e $d(x, y) = \sqrt{\langle x - y, x - y\rangle}$, decimos que $(X, \langle\ ,\ \rangle)$ es un espacio de Hilbert si $\left(X, d_{\langle\ ,\ \rangle}\right)$ es un espacio métrico completo.*

Vamos a estudiar el producto $\langle v, w\rangle = v^\dagger w = \sum_{i=1}^n v_i^* w_i$ en $\mathbb{C}^n$. En la siguiente proposición demostraremos que es un producto hermítico.

Proposición 6 ([2]). *Sea* $\langle\ ,\ \rangle : \mathbb{C}^n \times \mathbb{C}^n \longrightarrow \mathbb{C}$

$$v,\ w \ \longmapsto\ \langle v, w\rangle = v^\dagger w = \textstyle\sum_{i=1}^n v_i^* w_i.$$

Entonces, $\langle\ ,\ \rangle$ *es un producto hermítico en* $\mathbb{C}$

Demostración.

i) $\sum_{i=1}^n \alpha v_i^* w_i = \ \alpha \sum_{i=1}^n v_i^* w_i$

ii) $\sum_{i=1}^n v_i^* w_i = \ \overline{\sum_{i=1}^n w_i^* v_i}$

iii) $\sum_{i=1}^n (v_i + z_i)^* w_i = \ \ \sum_{i=1}^n v_i^* w_i + \ \sum_{i=1}^n z_i^* w_i$

iv) $\sum_{i=1}^n v_i^* v_i > 0\ \forall\ x \neq 0$

□

Como ya sabemos que $\langle\ ,\ \rangle$ es un producto hermítico, entonces por 5 $(\mathbb{C}^n, \langle\ ,\ \rangle)$ es espacio normado, y como tiene dimensión finita, por 3.2.1 es espacio métrico completo. Por lo que podemos enunciar que es espacio de Hilbert.

Corolario 3. $(\mathbb{C}^n, \langle\ ,\ \rangle)$ *es un espacio de Hilbert.*

3.2.3. Operadores y espacio Dual

En este capítulo se pretende definir el espacio dual a $\mathbb{C}^n$ respecto al producto hermítico $\langle\ ,\ \rangle$ visto anteriormente. Estudiar las transformaciones lineales del espacio y ver que las transformaciones unitarias conservan la norma.

Definición 9. *Denotamos por $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$ al conjunto de las aplicaciones $\mathbb{C}$-lineales de $\mathbb{C}^n$ a $\mathbb{C}$. Es decir, T: $\mathbb{C}^n \to \mathbb{C} \in \mathscr{L}(\mathbb{C}^n, \mathbb{C})$ si :*

$$T(az_1 + bz_2) = aT(z_1) + bT(z_2) \forall z_1, z_2 \in \mathbb{C}^n, \forall a, b \in \mathbb{C}.$$

Definición 10. *En $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$ definimos de la forma usual tanto suma como producto por escalares:*

1. $(T_1 + T_2)(z) := T_1(z) + T_2(z)$
2. $(\alpha T_1)(z) := \alpha(T_1(z))$

Observamos que para todo T $\in \mathscr{L}(\mathbb{C}^n, \mathbb{C})$ existe un único $M(T) \in M_{1\times n}(\mathbb{C})$ tal que $T(z) = M(T)z$. Similarmente, para todo M $\in M_{1\times n}(\mathbb{C})$ existe un único $T(M) \in \mathscr{L}(\mathbb{C}^n,\mathbb{C})$ tal que $T(M)(z) = Mz$.

Además,

1. $M(T_1 + T_2) = M(T_1) + M(T_2)$
2. $M(\alpha T_1) = \alpha M(T_1)$

Por tanto, $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$ es isomorfo a $M_{1\times n}(\mathbb{C})$. Y de esta manera, $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$ es $\mathbb{C}$-espacio vectorial de dimensión $dim_{\mathbb{C}}\mathscr{L}(\mathbb{C}^n\mathbb{C}) = dim_{\mathbb{C}}\mathbb{C}^n = n$.

Proposición 7. *Sea $\{e_1, e_2, \cdots, e_n\}$ una base ortonormal de $(\mathbb{C}, \langle\ ,\ \rangle)$*

$$\langle e_j, e_k\rangle = \delta_{jk} = \begin{cases} 1 & si \quad j = k \\ 0 & si \quad j \neq k \end{cases}$$

Sea ahora,

$$f_k : \mathbb{C}^n \to \mathbb{C} \quad , \quad z \in \mathbb{C}^n \to f_k(z) := \langle e_k, z\rangle$$

Entonces, $\{f_1, f_2, \cdots, f_n\}$ es base de $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$.

Demostración. En primer lugar debemos demostrar que $\{f_1, f_2, \cdots, f_n\}$ es sistema generador de $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$, de tal manera que $\langle\{f_1, f_2, \cdots, f_n\}\rangle = \mathscr{L}(\mathbb{C}^n, \mathbb{C})$.

Sea $T \in \mathscr{L}(\mathbb{C}^n, \mathbb{C})$,

$$\begin{aligned} T(z) =& M(T)z = \begin{bmatrix} M_1 & M_2 & \cdots & M_n \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix} \\ =& M_1 Z_1 + M_2 Z_2 + ... + M_n Z_n \\ =& M_1\langle z_1, e_1\rangle + M_2\langle z_2, e_2\rangle + \cdots + M_n\langle z_n, e_n\rangle \\ =& M_1 f_1(z) + M_2 f_2(z) + \cdots + M_n f_n(z) \\ =& \left(\sum_{i=1}^{n} M_i f_i\right)(z) \end{aligned}$$

Por otra parte, debemos ver que $\{f_1, f_2, \cdots, f_n\}$ son linealmente independientes.

Seleccionamos $\lambda_1, \lambda_2, \cdots, \lambda_n \in \mathbb{C}$ tal que :

$$\lambda_1 f_1 + \lambda_2 f_2 + \cdots + \lambda_n f_n = 0$$

Esto ocurre si y solo si $\forall z \in \mathbb{C}^n$

$$\lambda_1 f_1(z) + \lambda_2 f_2(z) + \cdots + \lambda_n f_n(z) = 0$$

$$\lambda_1\langle e_1, z\rangle + \lambda_2\langle e_2, z\rangle + \cdots + \lambda_n\langle e_n, z\rangle = 0$$

Escogemos $z = e_i \Rightarrow \lambda_i = 0$ y todos los productos hermíticos dan 0 □

Definición 11. *Sea* $(\mathbb{C}\ ,\ \langle\ ,\ \rangle)$*, definimos el espacio dual* $(\mathbb{C})^*$ *como el* $\mathbb{C}$*-espacio vectorial* $\mathscr{L}(\mathbb{C}^n, \mathbb{C})$

Observamos que la aplicación

$$\Phi : \mathbb{C}^n \to (\mathbb{C}^n)^*, \quad v \mapsto \Phi(v) := \langle v, \cdot\rangle$$

es un isomorfismo, porque para toda base $\{e_k\}_{k=1}^n$ ortonormal de $\mathbb{C}^n$, $\{\Phi(e_k) = f_k\}_{k=1}^n$ es base de $\mathbb{C}^n$. Esta aplicación nos permite definir un producto hermítico en $(\mathbb{C}^n)^*$,

$$\langle\ ,\ \rangle : (\mathbb{C}^n)^* \times (\mathbb{C}^n)^* \longrightarrow \mathbb{C}, \quad f, g \in (\mathbb{C}^n)^* \mapsto \langle f, g\rangle := \langle \Phi^{-1}(f), \Phi^{-1}(g)\rangle$$

Además para cualquier $v, w \in \mathbb{C}^n$

$$\langle v, w\rangle = \Phi(v)(w).$$

Veamos algunas de las propiedades de Φ.

Proposición 8.

1. $\Phi(\lambda v) = \lambda^* \Phi(v)$
2. $\Phi(u + v) = \Phi(u) + \Phi(v)$
3. *Si* $\{e_k\}_{k=1}^n$ *es base ortonormal de* $\mathbb{C}^n \Rightarrow \{\Phi(e_k) = f_k\}_{k=1}^n$ *es base ortonormal de* $(\mathbb{C}^n)^*$

Proposición 9. $\Phi^{-1}(\lambda v) = \lambda^* \Phi^{-1}(v)$

Demostración. $\Phi\left(\lambda^* \Phi^{-1}(v)\right) = (\lambda^*)^* \Phi\left(\Phi^{-1}(v)\right) = \lambda v$ □

Proposición 10. $T : \mathbb{C}^n \to \mathbb{C}^n$ *es una transformación lineal. Existe una única transformación lineal* $T^* : (\mathbb{C}^n)^* \to (\mathbb{C}^n)^*$ *tal que el siguiente diagrama es conmutativo.*

$$\begin{array}{ccc} \mathbb{C}^n & \xrightarrow{T} & \mathbb{C}^n \\ \downarrow{\scriptstyle \Phi} & & \downarrow{\scriptstyle \Phi} \\ (\mathbb{C}^n)^* & \xrightarrow{T^*} & (\mathbb{C}^n)^* \end{array}$$

Dada para $u \in (\mathbb{C}^n)^*$

$$T^*(u) = \Phi \circ T \circ \Phi^{-1}$$

Sea $\{e_k\}_{k=1}^n$ *BON de* $\mathbb{C}^n$. *Y sea* $\{\Phi(e_k) = f_k\}_{k=1}^n$ *BON de* $(\mathbb{C}^n)^*$.

Si $T(e_k) = \sum_{j=1}^n T_{kj} e_j \Rightarrow T^*(f_k) = \sum_{j=1}^n T_{kj}^* f_j$

Teorema 3.2.2. *Sea* $\{e_k\}_{k=1}^n$ *BON de* $\mathbb{C}^n$. *Sea* $U : \mathbb{C}^n \to \mathbb{C}^n$ *una transformación lineal tal que :* $U(e_k) = \sum_{j=1}^n u_{kj} e_j, u \in U(n)$ *Entonces,* $\|U(v)\| = \|v\|$

Demostración.

$$\begin{aligned}\|U(v)\|^2 =& \langle U(v), U(v)\rangle = \Phi\left(U(v)\right)\left(U(v)\right) = \sum_{k,l=1}^n \Phi\left(U(v_k^* e_k)\right)\left(U(v_l, e_l)\right) \\ =& \sum_{k,l=1}^n \Phi\left(v_k U(e_k)\right)\left(v_l U(e_l)\right) = \sum_{k,l=1}^n v_k^* v_l \Phi\left(U(e_k)\right)\left(U(e_l)\right) \\ =& \sum_{k,l,m=1}^n v_k^* v_l u_{lm} \Phi\left(U(e_k)\right)(e_m).\end{aligned}$$

$$\begin{array}{ccc} \mathbb{C}^n & \xrightarrow{T} & \mathbb{C}^n \\ \downarrow{\scriptstyle \Phi} & & \downarrow{\scriptstyle \Phi} \\ (\mathbb{C}^n)^* & \xrightarrow{T^*} & (\mathbb{C}^n)^* \end{array}$$

Por tanto podemos observar que

$$\begin{aligned}\Phi\left(U(e_k)\right)(e_m) =& U^*\left(\Phi(e_k)\right)(e_m) \\ =& U^*\left(f_k\right)(e_m) = \sum_{p=1}^n u_{kp}^* f_p(e_m) = \sum_{p=1}^n u_{kp}^* \delta_{mp} = U_{km}^*\end{aligned}$$

Por otra parte:

$$\begin{aligned}\|U(v)\|^2 &= \sum_{k,l=1}^n v_k^* v_l \sum_{m=1}^n u_{em} u_{mk}^\dagger \\ &= \sum_{k,l=1}^n v_k^* v_l \delta_{kl} = \sum_{k=1}^n v_k^* v_k\end{aligned}$$

Y también:

$$\begin{aligned}\|v\|^2 =&\langle v, v\rangle = \Phi(v)(v)\\ =&\Phi\left(\sum_{k=1}^{n} v_k e_k\right)\left(\sum_{l=1}^{n} v_l e_l\right) = \sum_{k,l=1}^{n} v_k^* v_l \Phi\left(e_k\right)\left(e_l\right)\\ =&\sum_{k,l=1}^{n} v_k^* v_l \langle e_k, e_l\rangle = \sum_{k,l=1}^{n} v_k^* v_l \delta_{kl} = \sum_{k=1}^{n} v_k^* v_k\end{aligned}$$

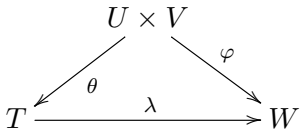
□

3.2.4. Producto tensorial

En este apartado introduciremos el producto tensorial $U \otimes V$ de los espacios vectoriales U y V. También veremos como construir una base del espacio $U \otimes V$ partiendo de las bases de U y V.

Definición 12 ([1]). *Sean U y V espacios vectoriales. Un espacio vectorial T, junto con una aplicación bilineal $\theta : U \times V \to T$, se denotará como producto tensorial de U y V si satisface las siguientes condiciones:*

1. *La imagen del conjunto $\theta(U \times V)$ tiene como envoltura lineal a T.*
2. *Si $\varphi : U \times V \to W$ es cualquier aplicación bilineal, entonces existe una transformación lineal $\lambda : T \to W$ tal que, $\varphi = \lambda \circ \theta$. (Véase la siguiente figura)*



Como la condición 2 requiere que la ecuación $\varphi = \lambda \circ \theta$ tenga solución λ , para toda aplicación bilineal φ, no importa cual sea el espacio W.

El producto tensorial será denotado por $U \otimes V$.

Teorema 3.2.3 ([1]). *Sean U , V espacios vectoriales de dimensión finita. Sean $\{\alpha_1, \alpha_2, \cdots, \alpha_m\}$ base de U, y $\{\beta_1, \beta_2, \cdots, \beta_n\}$ base de V. Entonces:*

1. $\dim(U \otimes V) = mn$.

Se puede construir una base de $U \otimes V$ (de mn elementos) cuyos elementos se denotan por: $\{\alpha_1 \otimes \beta_1, \alpha_1 \otimes \beta_2, \cdots, \alpha_1 \otimes \beta_n, \alpha_2 \otimes \beta_1, \cdots, \alpha_m \otimes \beta_n\}$

Demostración. La aplicación $\left(\sum_{i=1}^{m} a_i\alpha_i \sum_{j=1}^{n} b_j\beta_j\right) \to a_k b_l$

para enteros fijos pero arbitrarios k, l, es bilinial. Por tanto existe un $\lambda_{kl} \in U \otimes V$ tal que:

$$\lambda_{kl}\left[\left(\sum_{i=1}^{m} a_i\alpha_i\right) \otimes \left(\sum_{j=1}^{n} b_j\beta_j\right)\right] = a_k b_l.$$

En particular,

$$\lambda_{kl}\left(\alpha_i \otimes \beta_i\right) = \begin{cases} 0 & si\ i \neq k, j \neq l \\ 1 & si\ i = k, j = l \end{cases} .$$

Suponemos que

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\alpha_i \otimes \beta_j = 0$$

Si aplicamos λ_{kl} a esta relación, encontramos que $a_{kl} = 0$. Mientras esto sea verdad para $1 \leq k\ leq, 1 \leq l \leq n$, podemos observar que los elementos de $\alpha_1 \otimes \beta_1, \cdots\ \alpha_1 \otimes \beta_n$ son linealmente independientes. Por otro lado, todo elemento de $U \otimes V$ de la forma $\alpha \otimes \beta$ son una combinación lineal de estos elementos, por tanto $U \otimes V$ puede esribirse como combinación lineal de éstos, por ende constituyen una base de $U \otimes V$. □

3.3. Conceptos Física Cuántica

En esta segunda parte se pretende introducir los conceptos necesarios para la computación cuántica, así como relacionarlos con las definiciones matemáticas del apartado primero.

3.3.1. Qbit

Un bit es la unidad mínima de información en la informática actual. Un bit puede tomar 2 valores, 0 ó 1.

En computación cuántica será válida cualquier combinación lineal compleja de estos 2 elementos, que forman una base de este espacio de Hilbert (ver definición 8) de referencia.

En notación de Dirac, que veremos en el siguiente apartado, esta base se expresa como $\{|0\rangle, |1\rangle\}$, y un qbit se expresa como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C}$$

Vemos que como espacio de Hilbert, el espacio de los qbits es isomorfo a $\mathbb{C}^2$

Se puede tomar una medida de un bit para ver su valor, pero no se puede hacer eso con los qbits. En este caso $|\alpha|^2$ nos da la probabilidad de que el qbit sea 0 y $|\beta|^2$ nos da la probabilidad de que el qbit sea 1. Por tanto sabemos que $|\alpha|^2 + |\beta|^2 = 1$. Geométricamente deducimos que el qbit está normalizado con longitud 1.

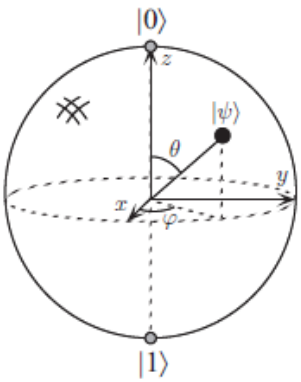


Figure 1.3. Bloch sphere representation of a qubit.

Si continuamos con el paralelismo nos preguntamos, ¿qué ocurre cuando tenemos varios qbits?

Si tenemos 2 bits , tenemos 4 posibilidades 00,01,10,11. Por el contrario, si tenemos 2 qbits, observamos que el espacio es isomorfo a $\mathbb{C}^2 \otimes \mathbb{C}^2$ (ver apartado 3.2.4 para definición y propiedades del producto tensorial).

De hecho, como $\{|0\rangle, |1\rangle\}$ forma una base de $\mathbb{C}^2$, por el teorema 3.2.3, $\{|0\rangle \otimes |0\rangle\}, \{|0\rangle \otimes |1\rangle\}, \{|1\rangle \otimes |0\rangle\}, \{|1\rangle \otimes |1\rangle\}$ forma una base de $\mathbb{C}^2 \otimes \mathbb{C}^2$.

De ahora en adelante, para simplificar la notación, usaremos:

$$
\begin{aligned}
|00\rangle &= |0\rangle \otimes |0\rangle \\
|01\rangle &= |0\rangle \otimes |1\rangle \\
|10\rangle &= |1\rangle \otimes |0\rangle \\
|11\rangle &= |1\rangle \otimes |1\rangle
\end{aligned}
$$

Simultaneamente, si tenemos n-qbits, el espacio de Hilbert de relevancia será isomorfo a

$\underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 \otimes \mathbb{C}^2}_{n}$, y obviamente, la dimensión compleja es 2^n

Retomando lo anterior, cuando tenemos 2 qbits nos queda:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Con $\alpha_i \in \mathbb{C} \quad i \in \{00, 01, 10, 11\} := \{0, 1\}^2$ y $|\alpha_i|^2$ la probabilidad de cada estado. Y como en el caso anterior $\sum_{i \in \{0,1\}^2} |\alpha_i|^2 = 1$

Por último, si tenemos varios qbits, podemos medir un subconjunto.

Si en este caso tenemos 2 y medimos el primer qbit, nos queda que la probabilidad de que el primer qbit sea 0 es $|\alpha_{00}|^2 + |\alpha_{01}|^2$, y si normalizamos nos queda el estado de post-medición:

$$|\psi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

3.3.2. Notación de Dirac

La notación de Dirac es la notación estándar para referirse a los estados en la física cuántica. Consideramos vectores que se encuentran en espacios de Hilbert de dimensión finita, sobre los números complejos. En el apartado anterior, hemos visto que para describir el estado de un qbit nos interesan espacios de Hilbert de dimensión compleja 2^n, que son isomorfos a $\mathbb{C}^{2^n}$

La base canónica está formada por vectores de 2^n elementos:

$$\{\begin{pmatrix}1\\0\\\vdots\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\\vdots\\0\\0\end{pmatrix}\cdots\begin{pmatrix}0\\0\\\vdots\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\\vdots\\0\\1\end{pmatrix}\}$$

De esta manera se asocian vectores columna de 2^n elementos a los vectores de la base de n elementos:

$$\underbrace{|00\ldots00\rangle}_{n}\Longleftrightarrow\left.\begin{pmatrix}1\\0\\\vdots\\0\\0\end{pmatrix}\right\}2^n,\quad |00\ldots01\rangle\Longleftrightarrow\begin{pmatrix}0\\1\\\vdots\\0\\0\end{pmatrix},$$

$$\vdots$$

$$|11\ldots10\rangle\Longleftrightarrow\begin{pmatrix}0\\0\\\vdots\\1\\0\end{pmatrix},\quad |11\ldots11\rangle\Longleftrightarrow\begin{pmatrix}0\\0\\\vdots\\0\\1\end{pmatrix}.$$

Por otro lado, el vector dual para $|a\rangle$ se expresa como $\langle a|$, donde $\langle a| := \Phi\left(|a\rangle\right)$, ver apartado 3.2.3 .

La probabilidad de que el qbit $|\psi\rangle$ pase a ser el qbit $|\alpha\rangle$ viene dada por:

$$Pr\left(|\psi\rangle \to |\alpha\rangle\right) = \|\langle\alpha|\psi\rangle\|^2$$

En notación del apartado 3.2.3 , $\|\langle\alpha|\psi\rangle\|^2 = \|\Phi\left(|\alpha\rangle\right)\left(|\psi\rangle\right)\|^2$.

Los estados $|0\rangle$ y $|1\rangle$ son conocidos como estados computacionales base.

Veamos que dado un estado de 2 qbits, $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$, la probabilidad de transición al estado $|00\rangle$ es justo

$$\begin{aligned} Pr(|\psi\rangle \to |00\rangle) &= \|\langle 00|\psi\rangle\|^2 \\ &= \|\langle 00| \ (\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) \ \rangle\|^2 \\ &= \|\langle 00|00\rangle\alpha_{00} + \langle 00|01\rangle\alpha_{01} + \langle 00|10\rangle\alpha_{10} + \langle 00|11\rangle\alpha_{11}\|^2 \\ &= \|\langle 00|00\rangle\alpha_{00}\|^2 \\ &= \|\alpha_{00}\|^2 \end{aligned}$$

3.3.3. Puertas lógicas

Los circuitos de computación clásicos, o deterministas, están formados por cables y puertas lógicas. Los cables sirven para enviar la información a través del circuito, y las puertas lógicas la manipulan.

Pues bien, en computación cuántica se usan circuitos probabilísticos. Matemáticamente nos interesan transformaciones lineales del espacio de Hilbert que conserven la norma (físicamente, la probabilidad total). Por el teorema 3.2.2 sabemos que estas transformaciones coincides con las transformaciones unitarias.

De esta manera, las puertas para 1 qbit son matrices unitarias 2×2. Además cualquier matriz unitaria 2×2 es una puerta para 1 qbit.

Similarmente las matrices $2^n \times 2^n$ se corresponden con puertas para n qbits.

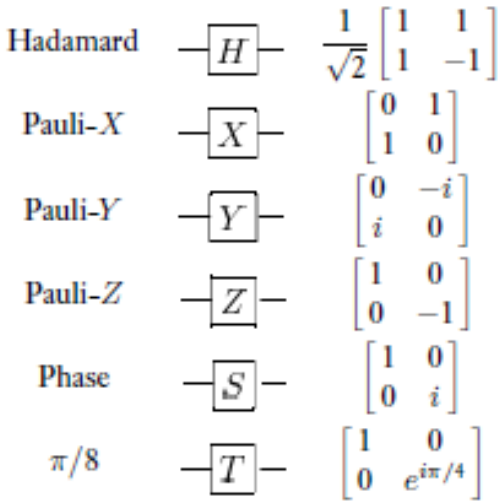


Figura 3.1: Algunas de las puertas más importantes son para circuitos probabilísticos

Una de las puertas más útiles es la Hadamard, que es auto-inversa, es decir $H = H^{-1}$. Por tanto, si observamos su definición y aplicamos esto obtenemos el siguiente resultado.

$$H\left(|0\rangle\right) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow H\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = |0\rangle$$

$$H\left(|1\rangle\right) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow H\left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = |1\rangle$$

Por otro lado la puerta más usada para 2 qbits es la CNOT (controlled NOT):

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

La cual actúa de la siguiente manera:

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |11\rangle \\ |11\rangle &\rightarrow |10\rangle \end{aligned}$$

3.3.4. Circuitos

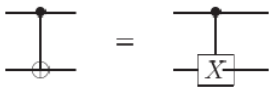
Una vez introducidas las puertas, el siguiente paso es preguntarnos por los circuitos. ¿Cómo son los circuitos cuánticos?

Los circuitos cuánticos se deben leer de izquierda a derecha. Cada linea representa un cable, que no tiene por que ser un medio físico. Por convenio se suele escoger el estado base $|0\rangle$ como *input*.

Las 3 diferencias más importantes con respecto a la computación clásica son:

1. No se permiten bucles de un mismo cable.
2. No se permite dividir cables (FANIN).
3. No se permite unir cables (FANOUT).

Los qbits de control son representados con un punto negro, mientras que los n qbits objetivo van unidos a una puerta, una caja o a puntos blancos.



Otra operación importante es la medición:

3.3.5. Estados Bell

Un resultado muy interesante ocurre cuando aplicamos Hadamard y CNOT :

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad |\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad |\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \quad |\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$

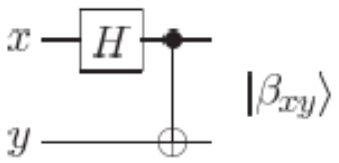


Figura 3.2: Circuito que genera los estados Bell

Estos son los llamados estados EPR, pares EPR o estados Bell. Tiene 2 posibles resultados: 0 con probabilidad 1/2 y dejando la post-medición $|\varphi'\rangle = |00\rangle$ y 1 con probabilidad 1/2 y dejando la post-medición $|\varphi'\rangle = |11\rangle$. De esta manera la medida del segundo qbit siempre da lo mismo que el primero. Es un componente clave de la teleportación cuántica.

Se puede demostrar que el conjunto de salida de Bell State forma una base ortonormal para el espacio de Hilbert que describe los estados cuánticos de 2 qbits.

Escogemos para empezar

$$\begin{aligned}\langle\beta_{00}|\beta_{00}\rangle &= \left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right)\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(\ \langle 00|00\rangle + \langle 00|11\rangle + \langle 11|00\rangle + \langle 11|11\rangle\ \right) \\ &= \frac{1}{2}\left(1 + 0 + 0 + 1\right) = 1\end{aligned}$$

Y ahora:

$$\begin{aligned}\langle\beta_{00}|\beta_{01}\rangle &= \left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right)\left(\frac{|01\rangle + |10\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(\ \langle 00|01\rangle + \langle 00|10\rangle + \langle 11|01\rangle + \langle 11|10\rangle\ \right) \\ &= \frac{1}{2}\left(0 + 0 + 0 + 0\right) = 0\end{aligned}$$

También:

$$\begin{aligned}\langle\beta_{00}|\beta_{10}\rangle &= \left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right)\left(\frac{|00\rangle - |11\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(\ \langle 00|00\rangle - \langle 00|11\rangle + \langle 11|00\rangle - \langle 11|11\rangle\ \right) \\ &= \frac{1}{2}\left(1 - 0 + 0 - 1\rangle\right) = 0\end{aligned}$$

Siguiendo este mecanismo es trivial observar que $\langle\beta_i|\beta_j\rangle = \delta_{i,j}$.

3.3.6. Teleportación Cuántica

La teleportación cuántica es una técnica para enviar información cuántica, incluso con ausencia de comunicación cuántica entre el emisor y el receptor. Es decir un estado cuántico puede ser enviado con absoluta exactitud mediante un canal clásico.

Si Alice y Bob tienen un par de qbits EPR, y cada uno se queda con uno cuando se separan, la teleportación permite a Alice comunicar el estado exacto del qbit a Bob enviando solo 2 bits de información (clásica).

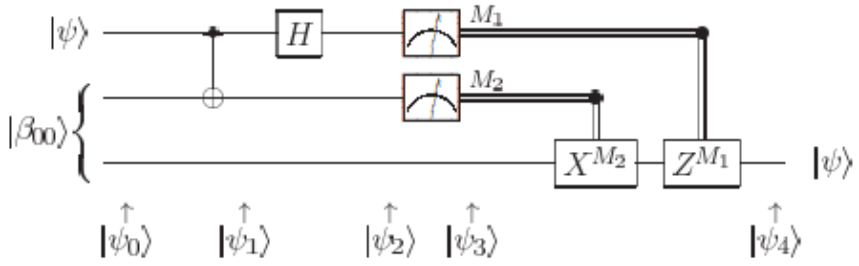


Figura 3.3: Las 2 lineas de arriba son el sistema de Alice, mientras que el de abajo es el de Bob. La doble linea representa que una vez hecha la medición, se envía información clásica

Inicialmente tenemos :

$$|\psi\rangle\,|\beta_{00}\rangle$$

Al pasar a través de la CNOT nos queda:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)]$$

Al pasar a través de la Hadamard:

$$|\psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)]$$

Y ahora podemos ordenar:

$$\begin{aligned}|\psi_2\rangle = \frac{1}{2}[&|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) \\ &+ |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]\end{aligned}$$

Ahora nos queda una expresión con 4 sumandos. El primer término de cada sumando son los qbits de Alice, mientras que lo que se encuentra entre paréntesis es el qbit de Bob. Por tanto si Alice realiza una medición y obtiene 00, el qbit de Bob estará en estado $(\alpha|0\rangle + \beta|1\rangle)$ De esta manera podemos crear una tabla para saber a partir de la medición de Alice, que qbit tiene Bob:

$$\begin{aligned}
00 &\longmapsto |\psi_3(00)\rangle \equiv [\alpha|0\rangle + \beta|1\rangle] \\
01 &\longmapsto |\psi_3(01)\rangle \equiv [\alpha|1\rangle + \beta|0\rangle] \\
10 &\longmapsto |\psi_3(10)\rangle \equiv [\alpha|0\rangle - \beta|1\rangle] \\
11 &\longmapsto |\psi_3(11)\rangle \equiv [\alpha|1\rangle - \beta|0\rangle]
\end{aligned}$$

Ahora Bob puede recuperar el estado original de $|\psi\rangle$, aplicando $Z^{M_1}X^{M_2}$

M1	**M2**	**Puertas**	**Transformación**
0	0	$I:$	$\alpha_0\|0\rangle + \alpha_1\|1\rangle \mapsto \alpha_0\|0\rangle + \alpha_1\|1\rangle = \|\psi\rangle$
0	1	$X:$	$\alpha_0\|1\rangle + \alpha_1\|0\rangle \mapsto \alpha_0\|0\rangle + \alpha_1\|1\rangle = \|\psi\rangle$
1	0	$Z:$	$\alpha_0\|0\rangle - \alpha_1\|1\rangle \mapsto \alpha_0\|0\rangle + \alpha_1\|1\rangle = \|\psi\rangle$
1	1	$Z \cdot X:$	$\alpha_0\|1\rangle - \alpha_1\|0\rangle \mapsto \alpha_0\|0\rangle + \alpha_1\|1\rangle = \|\psi\rangle$

En definitiva, de esta manera, Alice puede enviar estados cuánticos exactos a Bob usando solo 2 bits. Es decir, hemos enviado una combinación compleja de elementos, de manera codificada usando solo 2 bits. Esto es muy interesante para encriptación, ya que es semejante a los algoritmos de clave asimétrica, en el que si se intercepta el mensaje, es imposible sacar nada en claro si no tienes la clave.

3.3.7. Matriz Teleportación

Ahora, si expresamos como una un conjunto de 3 qbits todas las combinaciones posibles de entradas al circuito de la teleportación cuántica, obtenemos la siguiente tabla:

e_i	Entrada	CNOT	Hadamard
e_0	$\|000\rangle$	$\|000\rangle$	$\frac{1}{\sqrt{2}}(\|000\rangle + \|100\rangle)$
e_1	$\|001\rangle$	$\|001\rangle$	$\frac{1}{\sqrt{2}}(\|001\rangle + \|101\rangle)$
e_2	$\|010\rangle$	$\|010\rangle$	$\frac{1}{\sqrt{2}}(\|010\rangle + \|110\rangle)$
e_3	$\|011\rangle$	$\|011\rangle$	$\frac{1}{\sqrt{2}}(\|011\rangle + \|111\rangle)$
e_4	$\|100\rangle$	$\|110\rangle$	$\frac{1}{\sqrt{2}}(\|000\rangle - \|100\rangle)$
e_5	$\|101\rangle$	$\|111\rangle$	$\frac{1}{\sqrt{2}}(\|001\rangle - \|101\rangle)$
e_6	$\|110\rangle$	$\|100\rangle$	$\frac{1}{\sqrt{2}}(\|010\rangle - \|110\rangle)$
e_7	$\|111\rangle$	$\|101\rangle$	$\frac{1}{\sqrt{2}}(\|011\rangle - \|111\rangle)$

Cuadro 3.1: Tabla del circuito de teleportación cuántica

Que si la expresamos en forma de matriz tenemos, una matriz unitaria 8×8 con la información del circuito de la teleportación.

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Capítulo 4

Resultados

4.1. Algoritmo Reducción

En computación clásica hay un resultado muy importante que dice que las puertas NAND son universales, es decir, cualquier circuito se puede construir usando solo puertas NAND.

Pues bien, en computación cuántica hay un resultado que asegura que cualquier circuito puede ser reconstruido usando únicamente puertas de 2 niveles (ver [4]). Estas puertas de 2 niveles representan puertas cuánticas que actúan de manera no trivial en 2 de sus componentes. Matemáticamente, la transformación unitaria asociada a estas puertas dejan invariantes todos los vectores de la base salvo 2.

Se expresan, por lo tanto, como matrices unitarias tales que 4 de sus elementos, distribuidos en forma de cuadrado, son diferentes de lo que correspondería a la matriz identidad.

$$U_{i,j} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{i,i} & \cdots & u_{i,j} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{j,i} & \cdots & u_{j,j} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Definamos ahora $\mathcal{T}(n) := \{u \in U(n) \mid u \text{ es matriz de 2 niveles }\}$

Ya estamos en disposición de enunciar el resultado principal.

Teorema 4.1.1. *El conjunto de matrices de 2 niveles, $\mathcal{T}(n)$, genera el grupo unitario $U(n)$. Es decir:*

$$\langle \mathcal{T}(n) \rangle = U(n).$$

Esto quiere decir, que si tenemos una matriz unitaria genérica:

$$U = \begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

podemos encontrar ρ matrices unitarias de 2 niveles tales que :

$$\prod_{i=\rho}^{1} U_i U = I = U_\rho U_{\rho-1} \cdots U_2 U_1 U.$$

Siendo ρ, como máximo el número de elementos por debajo de la diagonal.

Por tanto

$$U = \prod_{i=1}^{\rho} U_i^\dagger = U_1^\dagger U_2^\dagger \cdots U_\rho^\dagger$$

con $U_i^\dagger$ también unitaria de 2 niveles $\forall i \in \{1 \cdots \rho\}$

La demostración de este teorema es la parte central del trabajo. Se ha realizado de manera constructiva, y de hecho se puede formalizar en forma de algoritmo (ver §A.1).

Demostración del Teorema 4.1.1.

Sea U una matriz unitaria genérica de $U(n+1)$, con $n > 0$.

$$U = \begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

En primer lugar observamos a_{10} , y en caso de que sea $a_{10} = 0$, U_1 será :

$$U_1 \equiv \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

En caso de que a_{10} sea diferente a 0, tomaremos como U_1 :

$$U_1 \equiv \begin{pmatrix} \frac{a_{00}^*}{\sqrt{|a_{00}|^2+|a_{10}|^2}} & \frac{a_{10}^*}{\sqrt{|a_{00}|^2+|a_{10}|^2}} & 0 & \cdots & 0 \\ \frac{a_{10}}{\sqrt{|a_{00}|^2+|a_{10}|^2}} & \frac{-a_{00}}{\sqrt{|a_{00}|^2+|a_{10}|^2}} & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Por tanto, obtenemos :

$$U_1 U = \begin{pmatrix} a_{00}^{'} & a_{01}^{'} & a_{02}^{'} & \cdots & a_{0n}^{'} \\ 0 & a_{11}^{'} & a_{12}^{'} & \cdots & a_{1n}^{'} \\ a_{20}^{'} & a_{21}^{'} & a_{22}^{''} & \cdots & a_{2n}^{'} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0}^{'} & a_{n1}^{'} & a_{01}^{'} & \cdots & a_{nn}^{'} \end{pmatrix}$$

Continuamos con el siguiente elemento de la columna, pero en la nueva matriz $(U_1 U)$:

En caso que $a_{20}^{'} = 0$:

$$U_1 \equiv \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Por el contrario, si $a_{20}^{'} \neq 0$

$$U_2 \equiv \begin{pmatrix} \frac{a_{00}^{'*}}{\sqrt{|a_{00}^{'}|^2+|a_{20}^{'}|^2}} & 0 & \frac{a_{10}^{'*}}{\sqrt{|a_{00}^{'}|^2+|a_{20}^{'}|^2}} & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \frac{a_{10}^{'}}{\sqrt{|a_{00}^{'}|^2+|a_{20}^{'}|^2}} & 0 & \frac{-a_{00}^{'}}{\sqrt{|a_{00}^{'}|^2+|a_{20}^{'}|^2}} & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Ahora obtenemos :

$$U_2U_1U = \begin{pmatrix} a_{00}^{''} & a_{01}^{''} & a_{02}^{''} & \cdots & a_{0n}^{''} \\ 0 & a_{11}^{''} & a_{12}^{''} & \cdots & a_{1n}^{''} \\ 0 & a_{21}^{''} & a_{22}^{''} & \cdots & a_{2n}^{''} \\ a_{30}^{''} & a_{31}^{''} & a_{32}^{''} & \cdots & a_{3n}^{''} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0}^{''} & a_{n1}^{''} & a_{01}^{''} & \cdots & a_{nn}^{''} \end{pmatrix}$$

Se hace esto para todos los elementos de la columna, hasta obtener la primera columna de la matriz identidad. Y ahora, al ser matrices unitarias, sabemos que:

$$U_nU_{n-1}\cdots U_2U_1 = \begin{pmatrix} 1 & b_{01} & b_{02} & \cdots & b_{0n} \\ 0 & b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & b_{n1} & b_{01} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$$

Ahora se procede de igual manera en las siguientes filas, hasta que nos quede la última submatriz 2×2 en la parte inferior derecha.

$$U_{\rho-1}U_{\rho-2}U_{\rho-3}\cdots U_nU_{n-1}\cdots U_2U_1\begin{pmatrix}1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 1 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & c_{n-1n-1} & c_{n-1n}\\ 0 & 0 & 0 & \cdots & c_{nn-1} & c_{nn}\end{pmatrix}$$

Donde

$$U_\rho = \begin{pmatrix}1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 1 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & c^*_{n-1n-1} & c^*_{nn-1}\\ 0 & 0 & 0 & \cdots & c^*_{n-1n} & c^*_{nn}\end{pmatrix}$$

Y de esta manera obtenemos las ρ matrices de dos niveles necesarias, tales que:

$$U_\rho U_{\rho-1}\cdots U_2U_1U = I$$

De donde se deduce

$$U = U_1^\dagger U_2^\dagger U_3^\dagger \cdots U_{\rho-1}^\dagger U_\rho^\dagger$$

□

4.2. Ejemplos

4.2.1. Ejemplo1

Entrada:

$$\begin{pmatrix}0{,}5 & 0{,}5 & 0{,}5 & 0{,}5\\ 0{,}5 & 0{,}5i & -0{,}5 & -0{,}5i\\ 0{,}5 & -0{,}5 & 0{,}5 & -0{,}5\\ 0{,}5 & -0{,}5i & -0{,}5 & 0{,}5i\end{pmatrix}$$

En este caso vamos a descomponer en 6 matrices: para hacer 6 ceros por debajo de la diagonal:

Descomposición :

$$U_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

$$U_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0{,}8165 & 0 & -0{,}5774i \\ 0 & 0 & 1 & 0 \\ 0 & 0{,}5774i & 0 & -0{,}8165 \end{pmatrix}$$

$$U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0{,}433 + 0{,}433i & 0{,}75 - 0{,}25i & 0 \\ 0 & 0{,}75 + 0{,}25i & -0{,}433 + 0{,}433i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_3 = \begin{pmatrix} 0{,}866 & 0 & 0 & 0{,}5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0{,}5 & 0 & 0 & -0{,}866 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} 0{,}8165 & 0 & 0{,}5774 & 0 \\ 0 & 1 & 0 & 0 \\ 0{,}5774 & 0 & -0{,}8165 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6:

$$U_1 = \begin{pmatrix} 0{,}7071 & 0{,}7071 & 0 & 0 \\ 0{,}7071 & -0{,}7071 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

De tal manera que

$$U_6 U_5 U_4 U_3 U_2 U_1 U = I$$

Y se sigue que:

$$U = U_1^\dagger U_2^\dagger U_3^\dagger U_4^\dagger U_5^\dagger U_6^\dagger$$

4.2.2. Ejemplo2

Entrada:

$$\begin{pmatrix} 0 & i & 0 & 0 \\ -i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & -i & 0 \end{pmatrix}$$

En este caso vamos a descomponer en 6 matrices para hacer ceros por debajo de la diagonal:

Descomposición :

$$U_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6:

$$U_1 = \begin{pmatrix} 0 & i & 0 & 0 \\ -i & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

De tal manera que

$$U_6 U_5 U_4 U_3 U_2 U_1 U = I$$

Y se sigue que:

$$U = U_1^\dagger U_2^\dagger U_3^\dagger U_4^\dagger U_5^\dagger U_6^\dagger$$

4.2.3. Ejemplo Teleportación

Entrada:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Descomposición

$$U_{28} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_{27} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_{26} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{25} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_{24} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{23} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{22} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}7071 & 0 & 0 & 0 & -0{,}7071 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -0{,}7071 & 0 & 0 & 0 & -0{,}7071 \end{pmatrix}$$

$$U_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{20} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{19} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{18} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_{17} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0{,}7071 & 0 & 0 & 0 & 0{,}7071 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0{,}7071 & 0 & 0 & 0 & -0{,}7071 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{16} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{15} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{14} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{13} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{11} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}7071 & 0 & 0 & 0 & -0{,}7071 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -0{,}7071 & 0 & 0 & 0 & -0{,}7071 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_9 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_7 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$U_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_4 = \begin{pmatrix} 0{,}7071 & 0 & 0 & 0 & 0{,}7071 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0{,}7071 & 0 & 0 & 0 & -0{,}7071 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

De tal manera que

$$\prod_{i=1}^{28} U_i U = I$$

Y se sigue que:

$$U = \prod_{i=1}^{28} U_i^{\dagger}$$

De esta manera obtenemos el siguiente circuito para simular la teleportación cuántica.

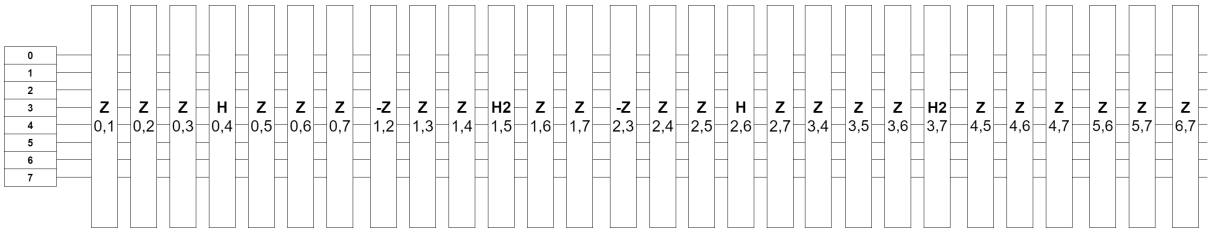


Figura 4.1: Simulación del circutio de teleportación usando puertas 2×2.

Siendo $A_{x,y}$ una puerta en la que solo intervienen los cables x e y.

Y usando únicamente estas 4 puertas:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$-Z = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad H2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$$

Capítulo 5

Conclusiones

Una vez acabado el curso puedo sacar diversas conclusiones habiendo vivido 2 experiencias bien diferenciadas brindadas por la parte teórica y la de prácticas. En ambas he ido usando los conocimientos adquiridos a lo largo del grado, de manera transversal a lo que directamente se me pedía.

Por lo que hace a la parte de prácticas he trabajado programando aplicaciones y movimientos de robots *Kuka*. Además he trabajado usando una gran base de datos y un proyecto enorme que llevaba meses de trabajo. Todo esto encapsulando código y intentando crear una interfaz sencilla para el usuario final. Por otra parte, ha sido muy interesante programar enfocado a hardware ya que se deben tener en cuenta las limitaciones físicas del robot.

En lo referente a la parte del desarrollo teórico, me he tenido que ir formando junto a Vicent para redactarla. En la primera parte íbamos saltando entre varios libros hasta encontrar las definiciones que necesitábamos. Para la segunda parte nos basamos en diversas fuentes para entender los conceptos de física cuántica. Empezando por las matrices unitarias, los espacios de Hilbert y los operadores duales, hemos llegado al resultado del teorema 4.1.1.

Un siguiente paso a este trabajo, sería desarrollar un teorema que dice que cualquier circuito cuántico, puede ser simulado usando únicamente las puertas Hadamard, Phase, $\pi/8$ y CNOT. Con este resultado, la simulación de circuitos cuánticos sería más cómoda, pues siempre se usarían puertas conocidas.

En definitiva, a pesar de no haber podido hacer unas prácticas relacionadas con la parte teórica del TFG, estoy satisfecho de haber estado en Agut Enginyeria y de haber trabajado la computación cuántica, ya que me han permitido experimentar de cerca 2 posibles salidas a los estudios universitarios, que son trabajar en una empresa, o la investigación.

Bibliografía

[1] Louis Auslander and Robert E. MacKenzie. *Introduction to differentiable manifolds*. Dover Publications, Inc., Mineola, NY, 2009. Reprint of the 1977 edition.

[2] Seymour Goldberg. *Unbounded linear operators*. Dover Publications, Inc., Mineola, NY, 2006. Theory and applications, Reprint of the 1985 corrected edition [MR0810617].

[3] John M. Lee. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, New York, second edition, 2013.

[4] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, 2000.

[5] Frank W. Warner. *Foundations of differentiable manifolds and Lie groups*, volume 94 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1983. Corrected reprint of the 1971 edition.

Anexo A

Anexo I

A.1. Código Algoritmo (Python)

```
from inspect import stack
import math
from tkinter.constants import NO


def leeMatriz(filas,columnas):

    matriz = []
    for i in range(filas):
        matriz.append( [0] * columnas )
    print('Ingrese su Matriz')
    for i in range(filas):
        for j in range(columnas):
            matriz[i][j] = complex(int(input('Parte real de (%d,%d):  '
                % (i, j))) , int(input('Parte imaginaria de (%d,%d):  '
                % (i, j))))

    return matriz


def prettyPrint(matriz):
    cadena = ''

    for i in range(len(matriz)):
        cadena += '['
        for j in range(len(matriz[i])):
```

```
25              cadena += '{0:>10s}'.format(str(round(matriz[i][j].real, 4)
                    )) + '+{0}'.format(str(round(matriz[i][j].imag, 4))) +"i
                    "
26          cadena += ']\n'
27      print(cadena)
28
29
30  def multiplica2Matr(matriz1, matriz2):
31
32      filas1=len(matriz1)
33      columnas1=len(matriz1[0])
34      filas2=len(matriz2)
35      columnas2=len(matriz2[0])
36
37      if (columnas1==filas2):
38          matriz3 = []
39          for i in range(filas1):
40              matriz3.append( [0] * columnas2 )
41          for i in range(filas1):
42              for j in range(columnas2):
43                  for k in range(filas2):
44                      matriz3[i][j] += matriz1[i][k] * matriz2[k][j]
45          return(matriz3)
46      else:
47          print ('Dimensiones no adecuadas')
48
49  def creaMatPivote(matriz, col, f1, f2):
50      filas=len(matriz)
51      columnas=len(matriz[0])
52
53      a=matriz[f1][col]
54      b=matriz[f2][col]
55      denom=float(math.sqrt(abs(a)**2+abs(b)**2))
56
57      matrizRes=[]
58      for i in range(filas):
59          matrizRes.append( [0] * columnas )
60          for j in range(columnas):
61
62              if i==f1 and j==f1:
63                  matrizRes[i][j]=a.conjugate()/denom
64              elif i==f2 and j==col:
65                  matrizRes[i][j]=b/denom
66              elif i==f1 and j==f2:
67                  matrizRes[i][j]=b.conjugate()/denom
68              elif i==f2 and j==f2:
69                  matrizRes[i][j]=-a/denom
70              elif i==j :
```

```
71                  matrizRes[i][j]=1
72              else:
73                  matrizRes[i][j]=0
74
75      return matrizRes
76
77  def creaMatriz1Diag(index, nfil,valor):
78      matrix=[]
79
80      for i in range(nfil):
81          matrix.append( [0] * nfil )
82      for i in range(nfil):
83          if i!= index:
84              matrix[i][i]=1
85          else:
86              matrix[i][i]=valor.conjugate()
87      return matrix
88
89
90  def triangulariza(matriz):
91      nfil=len(matriz)
92      ncol=len(matriz[0])
93      pila=[]
94      unext=matriz
95
96      for i in range (nfil-1):
97          for j in range(i+1,ncol):
98              if matriz[i][j]==0 and matriz[i][i]==1:
99                  print("cero encontrado en {0},{1}".format(i,j))
100             elif matriz[i][j]==0 and matriz[i][i]!=1 and j==ncol:
101                 print("cero encontrado en {0},{1}".format(i,j))
102                 MatAux=creaMatriz1Diag(i, nfil, unext[i][i])
103                 pila.append(MatAux)
104                 print("se añade el siguiente pivote a la pila de
                        resultados: ")
105                 prettyPrint(MatAux)
106
107                 unext=multiplica2Matr(MatAux,unext)
108                 print("La matriz unext que paso se queda asi: ")
109                 prettyPrint(unext)
110             else:
111
112                 u=creaMatPivote(unext, i, i, j)
113                 print("se añade el siguiente pivote a la pila de
                        resultados: ")
114                 prettyPrint(u)
115                 pila.append(u)
116
```

```
                    print(i, j)
                    unext=multiplica2Matr(u, unext)
                    print("La matriz unext que paso se queda asi: ")
                    prettyPrint(unext)


    ufinal=[]
    for i in range(nfil):
        ufinal.append( [0] * ncol )
    for i in range(nfil-1):
        ufinal[i][i]=1
    if matriz[nfil-1][ncol-2]!=0:
        ufinal[nfil-1][ncol-1]=unext[nfil-1][ncol-1].conjugate()
        ufinal[nfil-1][ncol-2]=unext[nfil-2][ncol-1].conjugate()
        ufinal[nfil-2][ncol-1]=unext[nfil-1][ncol-2].conjugate()
        ufinal[nfil-2][ncol-2]=unext[nfil-2][ncol-2].conjugate()

        print("se añade el ufinal a la pila de resultados: ")
        prettyPrint(ufinal)
        pila.pop()
        pila.append(ufinal)

        id=multiplica2Matr(ufinal, unext)

        print("final:")
        print("Supuesta identidad")
        prettyPrint(id)
    else:
        print("final:")
        print("Supuesta identidad")
        prettyPrint(unext)
    return pila


def miraCircuitos(pila):
    print("Las puertas necesarias para hacer la matriz son: ")
    i = 0
    pila2=pila[:]
    while len(pila2)>0:
        i+=1
        print(i)
        prettyPrint(pila2.pop())
```

A.2. Código ActualizaEntorno (C#)

```
public override bool ActualizaEntorno(List<int> valores_formulario,
    List<int> ultimos_valores_formulario, ObjetosManipulation destino,
    bool volada, bool modoVolada, List<int> capas, Transform
    baseDestinoMpiedra,bool automatic)
{
        if (volada != modoVolada)
        {
                modoVolada = !modoVolada;
                if (volada)
                {
                        ultimos_valores_formulario[3] = 0;
                        ultimos_valores_formulario[4] = 0;

                        Curve piedra = DevolverCurva(destino.objetos["
                            piedra_id"]);

                        Transform posicionDefectoTipoBase;
                        CalcularPosicionPiedraSobreBaseXY(piedra,
                            DevolverCurva(destino.objetos["base_id"]),
                            out posicionDefectoTipoBase);
                        valores_formulario[3] = Convert.ToInt32(
                            posicionDefectoTipoBase.M03);
                        valores_formulario[4] = Convert.ToInt32(
                            posicionDefectoTipoBase.M13);
                }
        }
        if (volada)
        {
                if (valores_formulario[3] != ultimos_valores_formulario
                    [3] || valores_formulario[4] !=
                    ultimos_valores_formulario[4] || valores_formulario
                    [2] != ultimos_valores_formulario[2])
                {
                        Transform rotation = Transform.Rotation(
                            RhinoFunctions.ConvertirGradosToRadianes(
                            valores_formulario[2] -
                            ultimos_valores_formulario[2]), new Vector3d
                            (0, 0, 1), DevolverCurva(destino.objetos["
                            piedra_id"]).PointAtStart);
                        TransformarGrupo(destino.grupoPiedra, rotation,
                             destino);
                        if (automatic) { TransformarGrupo(destino.
                            grupoVentosa, rotation, destino); }

                        Curve contorno_base = DevolverCurva(destino.
                            objetos["base_id"]);

```

```
29                          Curve piedra = DevolverCurva(destino.objetos["
                               piedra_id"]);
30                          double x, y;
31                          if (valores_formulario[3] > 0)
32                          {
33                                  x = contorno_base.GetBoundingBox(true).
                                       Max.X + destino.punto_origen.M03 -
                                       piedra.GetBoundingBox(true).Max.X +
                                       valores_formulario[3];
34                          }
35                          else
36                          {
37                                  x = contorno_base.GetBoundingBox(true).
                                       Min.X + destino.punto_origen.M03 -
                                       piedra.GetBoundingBox(true).Min.X +
                                       valores_formulario[3];
38                          }
39                          if (valores_formulario[4] > 0)
40                          {
41                                  y = contorno_base.GetBoundingBox(true).
                                       Max.Y + destino.punto_origen.M13 -
                                       piedra.GetBoundingBox(true).Max.Y +
                                       valores_formulario[4];
42                          }
43                          else
44                          {
45                                  y = contorno_base.GetBoundingBox(true).
                                       Min.Y + destino.punto_origen.M13 -
                                       piedra.GetBoundingBox(true).Min.Y +
                                       valores_formulario[4];
46                          }
47                          Transform translation = Transform.Translation(x
                               , y, 0);
48                          TransformarGrupo(destino.grupoPiedra,
                               translation, destino);
49                          if (automatic) { TransformarGrupo(destino.
                               grupoVentosa, translation, destino); }
50                  }
51          }
52          else
53          {
54                  if (valores_formulario[0] != ultimos_valores_formulario
                       [0] || valores_formulario[1] !=
                       ultimos_valores_formulario[1] || valores_formulario
                       [2] != ultimos_valores_formulario[2])
55                  {
56                          Transform rotation = Transform.Rotation(
                               RhinoFunctions.ConvertirGradosToRadianes(
```

```
                                valores_formulario[2] -
                                ultimos_valores_formulario[2]), new Vector3d
                                (0, 0, 1), DevolverCurva(destino.objetos["
                                piedra_id"]).PointAtStart);
57                          TransformarGrupo(destino.grupoPiedra, rotation,
                                destino);
58                          if (automatic) { TransformarGrupo(destino.
                                grupoVentosa, rotation, destino); }
59
60
61                          Curve piedra = DevolverCurva(destino.objetos["
                                piedra_id"]);
62                          double x = valores_formulario[0] - piedra.
                                PointAtStart.X;
63                          double y = valores_formulario[1] - piedra.
                                PointAtStart.Y;
64
65                          Transform translation = Transform.Translation(x
                                , y, 0);
66                          TransformarGrupo(destino.grupoPiedra,
                                translation, destino);
67                          if (automatic) { TransformarGrupo(destino.
                                grupoVentosa, translation, destino); }
68                  }
69          }
70          ultimos_valores_formulario = valores_formulario;
71          //ComprobarLimites();
72          BloquearCapas(true, capas);
73          RhinoDoc.ActiveDoc.Views.Redraw();
74          return modoVolada;
75  }
76  }
```

A.3. Código de la interfaz gráfica (WPF)

```
1   <Window x:Class="AgutHerramientas.MainWindow"
2           xmlns="http://schemas.microsoft.com/winfx/2006/xaml/
                presentation"
3           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5           xmlns:mc="http://schemas.openxmlformats.org/markup-
                compatibility/2006"
6           xmlns:local="clr-namespace:AgutHerramientas"
```

```
7           mc:Ignorable="d" Title="Herramientas" ResizeMode="NoResize"
               Height="600" Width="800" Icon="logo.ico"
               WindowStartupLocation="CenterScreen" Loaded="Window_Loaded">
8
9       <Window.Resources>
10          <ResourceDictionary>
11              <ResourceDictionary.MergedDictionaries>
12                  <ResourceDictionary Source="pack://application:,,,/
                       LogVTP;component/Idiomas/ES/Formularios.xaml"/>
13              </ResourceDictionary.MergedDictionaries>
14          </ResourceDictionary>
15      </Window.Resources>
16
17      <Grid x:Name="grMain">
18
19          <!--Menu superior-->
20          <StackPanel VerticalAlignment="Top">
21              <Menu Width="Auto" Height="20">
22                  <MenuItem Header="{DynamicResource
                       FormularioHerramientasTCP_HMn_Admin}">
23                      <MenuItem x:Name="mnOpenAsAdmin" Header="{
                           DynamicResource
                           FormularioHerramientasTCP_HMn_OpenAsAdmin}"
                           Click="btMnPas_Click" />
24                  </MenuItem>
25                  <MenuItem Header="{DynamicResource
                       FormularioHerramientasTCP_HMn_Añadir}">
26                      <MenuItem x:Name="mnNuevaEntrada" Header="{
                           DynamicResource
                           FormularioHerramientasTCP_HMn_NuevaEntrada}"
                           HorizontalAlignment="Left" Width="170" Click="
                           btMnNuevaEntrada_Click"/>
27                  </MenuItem>
28              </Menu>
29          </StackPanel>
30
31          <!--Pestanyes-->
32          <TabControl x:Name="tcMainTabControl" SelectionChanged="
               TabControl_SelectionChanged" Margin="0,20,0,0">
33              <TabItem x:Name="tiFresa" Header="{DynamicResource
                   FormularioHerramientasTCP_TIName_Fresa}" Width="100">
34                  <Grid  Width="773">
35                      <DataGrid x:Name="dgHerramientasFresa"  Margin="
                           0,0,0,275" SelectionChanged="
                           dgHerramientasFresa_SelectionChanged"
                           SelectionMode="Single" IsReadOnly="True" />
36
37                      <Grid x:Name="gButtonsText" Margin="0,250,0,20">
```

```
                        <Grid.RowDefinitions>
                            <!--275 en total-->

                            <!--0-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--2-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--4-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--6-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--8-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--10-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--12-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />

                            <!--14-->
                            <RowDefinition Height="25" />
                            <RowDefinition Height="5" />
                        </Grid.RowDefinitions>


                        <Grid.ColumnDefinitions>
                            <!--775 en total-->

                            <!--0-->
                            <ColumnDefinition Width="80" />
                            <ColumnDefinition Width="150" />
                            <ColumnDefinition Width="10" />

                            <!--3-->
                            <ColumnDefinition Width="95" />
                            <ColumnDefinition Width="155" />
```

```
86                                  <ColumnDefinition Width="10" />
87
88                                  <!--6-->
89                                  <ColumnDefinition Width="65" />
90                                  <ColumnDefinition Width="155" />
91                                  <ColumnDefinition Width="10" />
92
93                              </Grid.ColumnDefinitions>
94
95                              <Label Grid.Row="0" Grid.Column="0" Content="Id
                                    : " />
96                              <TextBox Grid.Row="0" Grid.Column="1" x:Name="
                                   tbId" PreviewTextInput="
                                   tbInt_PreviewTextInput" AcceptsReturn="True"
                                    IsEnabled="False" TextChanged="
                                   tb_TextChanged" />
97
98                              <Label Grid.Row="2" Grid.Column="0" Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelPosCono}" />
99                              <TextBox Grid.Row="2" Grid.Column="1" x:Name="
                                   tbPosCono" PreviewTextInput="
                                   tbInt_PreviewTextInput" TextChanged="
                                   tb_TextChanged" />
100
101                             <Label Grid.Row="4" Grid.Column="0" Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelIdMaq  }" />
102                             <ComboBox Grid.Row="4" Grid.Column="1" x:Name="
                                   comboBoxIdMaq" SelectionChanged="
                                   comboBoxSelectionChanged" IsEnabled="False"
                                   />
103
104                             <Label Grid.Row="6" Grid.Column="0" Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelFamilia  }"
                                   />
105                             <ComboBox Grid.Row="6" Grid.Column="1"  x:Name=
                                   "comboBoxFamilia" SelectionChanged="
                                   comboBoxSelectionChanged" IsEnabled="False"
                                   />
106
107                             <Label Grid.Row="8" Grid.Column="0"  Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelLongitud  }"
                                   Grid.RowSpan="2" />
108                             <TextBox Grid.Row="8" Grid.Column="1" x:Name="
                                   tbLong" AcceptsReturn="True"
```

```
                PreviewTextInput="tbDouble_previewTextInput"
                 TextChanged="tb_TextChanged"  />
109
110             <Label Grid.Row="10" Grid.Column="0" Content="{
                DynamicResource
                FormularioHerramientasTCP_LabelMinPiv  }" />
111             <TextBox Grid.Row="10" Grid.Column="1" x:Name="
                tbMinPiv" AcceptsReturn="True"
                PreviewTextInput="tbDouble_previewTextInput"
                 TextChanged="tb_TextChanged"  />
112
113             <Label Grid.Row="12" Grid.Column="0" Content="{
                DynamicResource
                FormularioHerramientasTCP_LabelMaxPiv  }" />
114             <TextBox Grid.Row="12" Grid.Column="1" x:Name="
                tbMaxPiv" AcceptsReturn="True"
                PreviewTextInput="tbDouble_previewTextInput"
                 TextChanged="tb_TextChanged"  />
115
116             <Label Grid.Row="14" Grid.Column="0" Content="{
                DynamicResource
                FormularioHerramientasTCP_LabelActiva  }" />
117             <CheckBox Grid.Row="14" Grid.Column="1" x:Name=
                "cbActiva" Content="" HorizontalAlignment="
                Left" VerticalAlignment="Center"  Click="
                ChangeCheckBoxAct" IsEnabled="True"/>
118
119             <Label Grid.Row="0" Grid.Column="3" Content="{
                DynamicResource
                FormularioHerramientasTCP_LabelIdTipoMedi}"
                Grid.ColumnSpan="2" Margin="0,0,147,0" />
120             <ComboBox Grid.Row="0" Grid.Column="4" x:Name="
                comboBoxMedi" HorizontalAlignment="Center"
                VerticalAlignment="Center" Width="150"
                SelectionChanged="comboBoxSelectionChanged"
                IsEnabled="False"  Height="22"/>
121
122             <Label Grid.Row="2" Grid.Column="3" Content="{
                DynamicResource
                FormularioHerramientasTCP_LabelFrecMedi}"
                />
123             <TextBox Grid.Row="2" Grid.Column="4" x:Name="
                tbFrecMedi" AcceptsReturn="True"
                PreviewTextInput="tbInt_PreviewTextInput"
                IsEnabled="False" Margin="3,0" TextChanged="
                tb_TextChanged" />
124
```

```
125                         <Label Grid.Row="4" Grid.Column="3" Content="{
                               DynamicResource
                               FormularioHerramientasTCP_LabelNVeces}"  />
126                         <TextBox Grid.Row="4" Grid.Column="4" x:Name="
                               tbNVUsada" AcceptsReturn="True" IsEnabled="
                               False" Margin="3,0" TextChanged="
                               tb_TextChanged" />
127
128                         <Label Grid.Row="6" Grid.Column="3" Content="{
                               DynamicResource
                               FormularioHerramientasTCP_LabelIdTipoSelec}"
                                Grid.ColumnSpan="2" Margin="0,0,147,0"  />
129                         <ComboBox Grid.Row="6" Grid.Column="4" x:Name="
                               comboBoxSelec" HorizontalAlignment="Center"
                               VerticalAlignment="Center" Width="150"
                               SelectionChanged="comboBoxSelectionChanged"
                               IsEnabled="False" Margin="3,2" />
130
131                         <Label Grid.Row="8" Grid.Column="3" Content="{
                               DynamicResource
                               FormularioHerramientasTCP_LabelFechaUtilizada
                               }" RenderTransformOrigin="0.5,3.06" />
132                         <DatePicker Grid.Row="8" Grid.Column="4" x:Name
                               ="dpFechaUtilizada" HorizontalAlignment="
                               Left" VerticalAlignment="Top"  Width="150"
                               SelectedDateChanged="
                               datePSelectedDateChanged"  Foreground="Black
                               " IsEnabled="False" Margin="3,0,0,0">
133
134                         </DatePicker>
135
136                         <Label Grid.Row="10" Grid.Column="3" Content="{
                               DynamicResource
                               FormularioHerramientasTCP_LabelIdTipoDesactiv
                               }" />
137                         <ComboBox Grid.Row="10" Grid.Column="4" x:Name=
                               "comboBoxDesact" HorizontalAlignment="Center
                               " VerticalAlignment="Center" Width="150"
                               SelectionChanged="comboBoxSelectionChanged"
                               IsEnabled="False" Margin="3,2"/>
138
139                         <Label Grid.Row="12" Grid.Column="3" Content="{
                               DynamicResource
                               FormularioHerramientasTCP_LabelDiámetro  }"
                               />
140                         <TextBox Grid.Row="12" Grid.Column="4" x:Name="
                               tbDiametro" AcceptsReturn="True"
                               PreviewTextInput="tbDouble_previewTextInput"
```

```
                                Margin="3,0" TextChanged="tb_TextChanged"
                               />
141
142                            <Label Grid.Row="14" Grid.Column="3" Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelIdTcpEntero
                                   }" Grid.RowSpan="2"  />
143                            <TextBox Grid.Row="14" Grid.Column="4" x:Name="
                                   tbIdTCP" AcceptsReturn="True"
                                   PreviewTextInput="tbIdTCP_PreviewTextInput"
                                   IsEnabled="False" TextChanged="
                                   tb_TextChanged" />
144
145                            <Label Grid.Row="0" Grid.Column="6" Content="{
                                   DynamicResource
                                   FormularioHerramientasTCP_LabelMaterial  }"
                                   />
146                            <ComboBox Grid.Row="0" Grid.Column="7" x:Name="
                                   comboBoxMaterial" HorizontalAlignment="
                                   Center" VerticalAlignment="Center" Width="
                                   152"  SelectionChanged="
                                   comboBoxSelectionChanged" IsEnabled="False"
                                   Height="22" />
147
148                            <Label Grid.Row="2" Grid.Column="6" Content="X
                                   :"   />
149                            <TextBox Grid.Row="2" Grid.Column="7" x:Name="
                                   tbX" AcceptsReturn="True" PreviewTextInput="
                                   tbDouble_previewTextInput" IsEnabled="False"
                                    Margin="2,0" TextChanged="tb_TextChanged"
                                   />
150
151                            <Label Grid.Row="4" Grid.Column="6" Content="Y
                                   :"   />
152                            <TextBox Grid.Row="4" Grid.Column="7" x:Name="
                                   tbY" AcceptsReturn="True" PreviewTextInput="
                                   tbDouble_previewTextInput" IsEnabled="False"
                                    Margin="2,0" TextChanged="tb_TextChanged"
                                   />
153
154                            <Label Grid.Row="6" Grid.Column="6" Content="Z
                                   :"   />
155                            <TextBox Grid.Row="6" Grid.Column="7" x:Name="
                                   tbZ" AcceptsReturn="True" PreviewTextInput="
                                   tbDouble_previewTextInput" IsEnabled="False"
                                    Margin="2,0" TextChanged="tb_TextChanged"
                                   />
156
```

```
157                             <Label Grid.Row="8" Grid.Column="6" Content="A
                                    :"   />
158                             <TextBox Grid.Row="8" Grid.Column="7" x:Name="
                                    tbA" AcceptsReturn="True" PreviewTextInput="
                                    tbDouble_previewTextInput" IsEnabled="False"
                                     Margin="2,0" TextChanged="tb_TextChanged"
                                    />
159
160                             <Label Grid.Row="10" Grid.Column="6" Content="B
                                     :"    />
161                             <TextBox Grid.Row="10" Grid.Column="7" x:Name="
                                    tbB" AcceptsReturn="True"  PreviewTextInput=
                                    "tbDouble_previewTextInput" IsEnabled="False
                                    " Margin="2,0" TextChanged="tb_TextChanged"
                                    />
162
163                             <Label Grid.Row="12" Grid.Column="6" Content="C
                                     :"    />
164                             <TextBox Grid.Row="12" Grid.Column="7" x:Name="
                                    tbC" AcceptsReturn="True" PreviewTextInput="
                                    tbDouble_previewTextInput" IsEnabled="False"
                                     Margin="2,0" TextChanged="tb_TextChanged"
                                    />
165
166
167                         </Grid>
168
169                         <StackPanel HorizontalAlignment="Right"
                                VerticalAlignment="Bottom" Width="59" Height="42
                                ">
170                             <Button x:Name="btGuardar" Content="{
                                    DynamicResource imBtGuardar}"
                                    HorizontalAlignment="Center"
                                    VerticalAlignment="Center" Click="
                                    btGuardar_Click"  Height="35" Width="44"
                                    Margin="6,0"  />
171                         </StackPanel>
172                     </Grid>
173                 </TabItem>
174                 <!--<TabItem x:Name="tiDisco" Header="Tab 2"/>-->
175             </TabControl>
176         </Grid>
177     </Window>
```