Diseño e implementación de algoritmos cuánticos para problemas de optimización combinatoria

Autor: Stelian Adrian Stanci

Director: Elías Fernández-Combarro Álvarez

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Escuela de Ingeniería Informática

Universidad de Oviedo

2023

Agradecimientos

A los que siempre han estado ahí.

Resumen

La computación cuántica nos abre camino a una serie de posibilidades a la hora de abordar problemas computacionales, a través de métodos que se aprovechan de propiedades únicas de los sistemas cuánticos. A pesar de que actualmente nos encontramos en una fase inicial de experimentación, ya se han desarrollado numerosos métodos para este nuevo paradigma, con los que se pueden realizar experimentos para comprobar cómo se desenvuelven a la hora de resolver problemas. Aunque hoy en día los dispositivos cuánticos con los que contamos no están lo suficientemente dotados como para demostrar sus capacidades en su máxima expresión, sigue siendo posible estudiar el rendimiento de estos métodos, y evaluar la forma en la que se enfrentan a tareas de diferentes dificultades. En este estudio, el enfoque se va a ubicar en un tipo de problemas concreto, los de optimización, para los que ya existen varias alternativas que han demostrado tener potencial. Dos de las más populares, y que se van a tratar en este estudio, son el uso de annealers, y el algoritmo híbrido QAOA, que muestran resultados prometedores en la misión de estos dispositivos de, en un futuro, poder superar a las contrapartes clásicas.

Palabras clave: computación cuántica, optimización combinatoria, quantum annealing, QAOA.

Abstract

Quantum computing offers a plethora of new possibilities for solving computational tasks, by using methods that take advantage of unique properties present in quantum systems. Despite the paradigm currently being in a rather initial state of experimentation, many methods have already been created, which can be tested in experiments to see how they fare when dealing with problems. Even though nowadays the quantum devices at our disposal are limited by technical factors, which prevent them from showing their true capabilities, the performance of these methods can still be evaluated against problems of varying difficulty. In this study, the focus will be placed on optimization problems, for which there are several options that have displayed significant potential. Two of the most popular, which are going to be explored in this study, are the use of annealers, and the hybrid algorithm QAOA, which show promising results in the quest of these devices of, some day in the future, being able to surpass their classical counterparts.

Keywords: quantum computing, combinatorial optimization, quantum annealing, QAOA.

Índice

Contenido

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

La computación cuántica está experimentando un incremento en avances y en exposición en los últimos años. Esto, junto al hecho de que es una tecnología que funciona de manera bastante diferente a los ordenadores clásicos, nos hace preguntarnos cómo se desenvuelve hoy en día a la hora de resolver problemas.

Al intentar responder a esta pregunta, nos encontraremos con que hay diferentes formas de construir ordenadores cuánticos, y también distintos enfoques, o métodos, que se pueden utilizar para hacer las operaciones.

Actualmente, se pueden encontrar estudios realizados sobre distintos problemas de optimización combinatoria, por ejemplo centrando la investigación en torno a un método en concreto (véase Estado del arte), pero el número de estudios que hace este tipo de experimentos, con varios problemas y métodos, en el mismo entorno, es todavía reducido.

Esta escasez de artículos que realicen dichas comparaciones bajo las mismas condiciones genera la duda de qué diferencias se pueden observar entre los resultados de distintos métodos en la resolución de problemas, lo cual se pretende ilustrar en esta investigación.

Una consecuencia de que todavía se trate de una disciplina en una etapa tan latente es que la cantidad de estudios que existen relacionados con este tema no son muy numerosos, por lo que este trabajo podría servir como fuente de información sobre las capacidades y el rendimiento de estos algoritmos en la actualidad.

1.2. Finalidad del proyecto

El objeto de este trabajo es el análisis de la efectividad de distintos métodos de resolución de problemas de optimización combinatoria, mostrando cómo se pueden comportar con diferentes problemas de optimización, probando con diversos tamaños de problemas y con diferentes configuraciones de los parámetros en los métodos a aplicar.

1.3. Marco teórico

La computación cuántica es una disciplina que ha ido cobrando mucha fuerza en los últimos años y que supone una alternativa muy potente a los ordenadores tradicionales a la hora de hacer ciertas tareas, ya que puede llegar a ofrecer una serie de ventajas imposibles de alcanzar con la computación clásica.

La principal diferencia con la computación tradicional es que se aprovechan distintas propiedades de los sistemas cuánticos para resolver algunas tareas de manera más eficiente, como la **superposición**, el **entrelazamiento** y la **interferencia**.

El hecho de poder utilizar estas propiedades resulta muy prometedor, ya que nos abre las puertas a muchas posibilidades nuevas para la computación, potencialmente mejores que las opciones clásicas de las que disponemos, aunque de momento, con los ordenadores cuánticos que contamos, no es posible hacer uso de estos fenómenos de la manera más eficaz.

Hoy en día, los dispositivos cuánticos que están a nuestro alcance, incluso los más modernos, no son suficientemente potentes para llevar a cabo tareas a partir de cierto grado de complejidad, y tampoco tienen la tolerancia a fallos necesaria para ello.

Estos ordenadores actuales, que todavía no pueden mostrar todo el potencial del paradigma cuántico debido a las limitaciones mencionadas, reciben la denominación de **ordenadores cuánticos de escala intermedia y con ruido** (**NISQ**, por las siglas en inglés), y, a pesar de sus imperfecciones, ya se han desarrollado varios algoritmos que se pueden ejecutar en ellos, algunos con resultados muy alentadores.

Uno de los campos en los que más se están investigando algoritmos con dispositivos NISQ es el de la **optimización**, donde una propuesta muy destacada es la del algoritmo QAOA (véase 5.3.2. QAOA: Quantum Approximate Optimization Algorithm).

1.3.1. Introducción a la computación cuántica: modelos computacionales

Esta tecnología funciona de una manera bastante diferente a los ordenadores que estamos acostumbrados.

Para explicar su funcionamiento, y poder visualizar cómo se hace uso de las propiedades mencionadas anteriormente, en las siguientes secciones se va a presentar un sistema con el que expresar de forma matemática las operaciones que vamos a aplicar sobre la información y la representación de la misma, es decir, un **modelo computacional.**

Existen varios modelos de este tipo, pero en este trabajo se van a utilizar principalmente dos: el **modelo de circuitos cuánticos** y el **modelo adiabático** (véase 5.3.1.1. Computación adiabática cuántica).

En el modelo de circuitos cuánticos la información se representa a través de **qubits**, sobre los cuales se aplican una serie de operaciones por medio de **puertas cuánticas**, y cuyo valor se obtiene a través de **mediciones**.

En las siguientes secciones se van a explicar los conceptos necesarios para entender los fundamentos del modelo, ya que tiene mucha relación con los métodos que se han utilizado en el estudio.

1.3.1.1. Modelo de circuitos cuánticos: qubits y mediciones

El qubit, como se ha mencionado en la sección anterior, se utiliza para representar la información en el modelo de circuitos cuánticos. Concretamente, es la unidad mínima de información que puede utilizar un ordenador cuántico.

Para representar el estado de un qubit, vamos a utilizar la **notación de Dirac**, en la que tenemos, por ejemplo, los estados

$$|0\rangle,\ |1\rangle,$$

en los que los números 0 y 1 están contenidos en unos símbolos que se denominan **kets**, y que sirven para indicar que estamos trabajando con vectores columna en lugar de números enteros. Para representar los vectores fila existen otros símbolos, los **bras**,

$$\langle 0|, \langle 1|,$$

que se pueden juntar con los kets para formar bra-kets,

$$\langle 0|0\rangle, \langle 1|1\rangle,$$

y estos se pueden utilizar a la hora de calcular **productos internos**. La notación de Dirac también se conoce como notación bra-ket, en referencia a los símbolos que se utilizan en ella.

Como se ha comentado previamente, una de las propiedades de la que nos vamos a aprovechar es la superposición, y para representar que un qubit está en una superposición de estados se hace de la siguiente manera:

$$a|0\rangle + b|1\rangle,$$

donde tenemos los estados $|0\rangle$ y $|1\rangle$, que están siendo multiplicados respectivamente por a y b, que son números complejos. Estos números reciben la denominación de **amplitudes**, y necesariamente deben cumplir la condición

$$|a|^2 + |b|^2 = 1\,.$$

También cabe mencionar que dependiendo de los valores de a y b podemos determinar si un estado está **normalizado**, calculando la longitud del mismo, de esta forma:

$$\sqrt{|a|^2 + |b|^2},$$

y observando si el resultado es igual a 1, en cuyo caso sería un estado normalizado.

Hemos mencionado previamente que al trabajar con estados de qubits utilizamos vectores, los cuales pertenecen a un espacio vectorial, para el que tenemos que definir una base, que llamaremos la **base computacional**:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

por lo que la superposición se representaría de esta forma

$$a|0\rangle + b|1\rangle = a\begin{pmatrix} 1 \\ 0 \end{pmatrix} + b\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix},$$

donde la probabilidad de obtener como resultado 0 sería $|a|^2$, y la de obtener 1, $|b|^2$.

Una vez hemos hecho una medición y obtenido un resultado, decimos que el estado del qubit **colapsa** al valor obtenido, y pasamos a trabajar con bits clásicos.

1.3.1.2. Modelo de circuitos cuánticos: puertas cuánticas

Las puertas cuánticas se utilizan para manipular el estado de un qubit, es decir, para realizar operaciones sobre él. En el modelo de circuitos cuánticos, las puertas se representan a través de matrices, que se aplican a los estados de los qubits, representados a su vez por vectores.

Estas matrices tienen que ser **unitarias**, es decir tienen que cumplir la condición

$$U^\dagger U = UU^\dagger = I,$$

en la que $U^\dagger$ es la matriz adjunta de U (la cual se obtiene trasponiendo U y sustituyendo cada elemento por su conjugado, que tiene igual parte real pero parte imaginaria con misma magnitud y distinto signo), y la matriz I es la identidad. Estas matrices, que representan operaciones, son las que denominamos puertas cuánticas.

Para ilustrar los ejemplos de puertas cuánticas, primero vamos a considerar que estamos trabajando solamente con un qubit. En este caso, se pueden tomar 2 valores (el vector de estados tiene dimensión 2), por lo que las matrices que vamos a utilizar tendrán tamaño 2x2.

La primera puerta que podríamos poner como ejemplo es la identidad

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

Aunque esta no tiene ningún efecto sobre el estado del qubit, cumple la definición para considerarse una puerta cuántica.

El siguiente ejemplo es la puerta NOT (o la puerta X)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

que actúa como una negación lógica; es decir, si tenemos un estado $|0\rangle$ y le aplicamos la puerta NOT, este pasaría a ser $|1\rangle$, y viceversa.

$$X\,|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \qquad X\,|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

Por tanto, se puede establecer un símil entre esta puerta y la puerta NOT de los circuitos clásicos.

Otro ejemplo sería la puerta Hadamard (o la puerta H)

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

El propósito de esta puerta es el de crear una superposición de estados. Podemos observar su funcionamiento aplicándola sobre el estado $|1\rangle$, que nos daría el resultado

$$H\,|1\rangle = \frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Este estado se conoce como el estado **minus**. Es bastante común, lo que ha hecho que acabe recibiendo su propio nombre. También se puede referir a él como $|-\rangle$. Existe también el estado **plus** (o $|+\rangle$), que es el estado ortogonal al minus, y se obtiene de aplicar una puerta H al estado $|0\rangle$:

$$H\,|0\rangle = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

La siguiente puerta sería la puerta Z, que es equivalente a aplicar sucesivamente una puerta H, una puerta X, y otra puerta H a un qubit

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

Este es el efecto que tiene la puerta Z si la aplicamos sobre los estados que conforman la base computacional

$$Z|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, \qquad Z|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1\rangle$$

Las puertas I, X, Y (que se explica en la siguiente sección) y Z constituyen **las matrices de Pauli**, que son una serie de puertas cuánticas muy conocidas y de gran utilidad en la computación cuántica.

1.3.1.3. La esfera de Bloch y las puertas de rotaciones

El número de matrices de dos dimensiones con las que se pueden hacer puertas es infinito. Debido a esto, no es posible hablar de todas ellas. Sin embargo, existe una forma de visualizar todas las puertas que se pueden aplicar a un qubit como rotaciones, en un cuerpo conocido como la **esfera de Bloch**.

En un principio, puede parecer extraño que estos estados se puedan visualizar en una esfera, ya que se utilizan dos números complejos para describirlo, lo cual parece indicar que necesitaríamos cuatro dimensiones (ya que cada número complejo está compuesto por dos números reales). A pesar de esto, es posible demostrar que podemos representarlos utilizando una esfera, que solamente tiene dos dimensiones.

En primer lugar, tenemos que un número complejo z puede escribirse en coordenadas polares de esta forma

$$z = re^{i\alpha},$$

en la que $r = |z|$ es un número real no negativo y α es un ángulo entre $[0, 2\pi]$.

Teniendo en cuenta un qubit cuyo estado sería $|\psi\rangle = a|0\rangle + b|1\rangle$, donde a y b son, como se ha comentado previamente, números complejos, también podemos escribirlos como coordenadas polares

$$a = r_1 e^{i\alpha_1}, \qquad b = r_2 e^{i\alpha_2},$$

donde se debe cumplir que $r_1^2 + r_2^2 = |a|^2 + |b|^2 = 1$. Sabemos que $0 \leq r_1,\ r_2 \leq 1$, por lo que debe haber un ángulo comprendido entre $[0,\ \pi]$ y para el que $\cos\left(\frac{\theta}{2}\right) = r_1$ y $\sin\left(\frac{\theta}{2}\right) = r_2$. Esto nos deja con la siguiente expresión para el estado $|\psi\rangle$:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) e^{i\alpha_1}|0\rangle + \sin\left(\frac{\theta}{2}\right) e^{i\alpha_2}|1\rangle.$$

Un detalle sobre este resultado es que podríamos multiplicar el estado $|\psi\rangle$ por un número complejo c con valor absoluto 1 y dicho estado no cambiaría (no cambiarían las probabilidades de obtener 0 o 1 en la base computacional). Además, por linealidad, si aplicamos una puerta cuántica U tendríamos que

$$U(c|\psi\rangle) = cU|\psi\rangle,$$

por lo que no hay forma de distinguir $|\psi\rangle$ de $c|\psi\rangle$. El término para ese número c es **fase global**. Existen también fases relativas, las cuales sí que se pueden distinguir,

como pueden ser los estados plus y minus, a los que podemos aplicar la puerta H para diferenciarlos a la hora de medirlos.

Este punto es relevante porque podemos aprovecharnos de ese comportamiento y multiplicar el estado $|\psi\rangle$ por $e^{-i\alpha_1}$, lo que resulta en la siguiente representación, que es equivalente

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

donde $\varphi = \alpha_2 - \alpha_1$.

Gracias a estos cambios, ahora podemos denotar el estado de un qubit utilizando dos números, $\theta \in [0,\ \pi]$ y $\varphi \in [0,\ 2\pi]$, donde θ sería el ángulo polar y φ el ángulo azimutal, ya que estaríamos usando coordenadas esféricas. Con todo esto, a través del punto tridimensional

$$(\sin\theta\cos\varphi\,,\sin\theta\sin\varphi\,,\cos\theta)$$

se puede situar el estado del qubit en la superficie de la esfera de Bloch, que podemos observar en la siguiente imagen (créditos a [1])

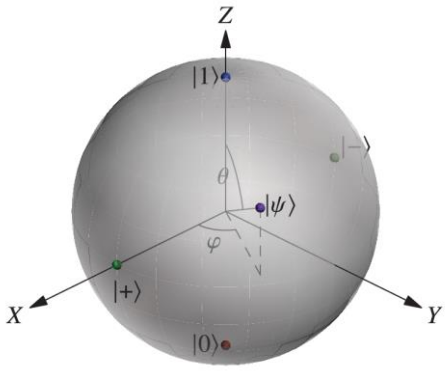


Figura 1.1. Estado $|\psi\rangle$ de un qubit representado en la esfera de Bloch

En la representación se puede ver que θ puede tomar cualquier valor entre 0 y π desde la parte superior hasta el polo sur de la esfera. Esto explica el uso de $\pi/2$ en la demostración anterior porque era el valor máximo que necesitábamos para los ángulos de los senos y los cosenos.

También se puede observar en la imagen que el estado $|0\rangle$ se encuentra en el polo norte, y el estado $|1\rangle$ en el polo sur, y que el estado plus y el minus están ambos situados en el ecuador de la esfera, pero en puntos opuestos. Los estados que son ortogonales con respecto al producto interno son antipodales en la esfera.

Como se ha visto previamente, la puerta X transforma $|0\rangle$ en $|1\rangle$ y $|1\rangle$ en $|0\rangle$, pero no tiene ningún efecto aparente sobre los estados plus y minus. Esto se explica porque la

puerta X lo que hace es aplicar una rotación de π radianes alrededor del eje X de la esfera de Bloch, siendo ese el motivo por el que tiene ese nombre.

Dicho esto, en la sección anterior se mencionó, sin entrar en detalle, la puerta Y, que se representa con esta matriz

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

que también es bastante común (nótese que forma parte del conjunto de las matrices de Pauli), y cuyo efecto es el de aplicar una rotación de π radianes al estado del qubit alrededor del eje Y. La puerta Z tiene un comportamiento similar al de las puertas X e Y, distinguiéndose en que produce la rotación alrededor del eje Z.

Esto se puede extender para construir rotaciones de cualquier ángulo alrededor de cualquier eje de la esfera de Bloch. Para los ejes X, Y y Z tenemos que

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = cos\frac{\theta}{2}I - i\,sin\frac{\theta}{2}X = \begin{pmatrix} cos\frac{\theta}{2} & -i\,sin\frac{\theta}{2} \\ -i\,sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{pmatrix},$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = cos\frac{\theta}{2}I - i\,sin\frac{\theta}{2}Y = \begin{pmatrix} cos\frac{\theta}{2} & -sin\frac{\theta}{2} \\ sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{pmatrix},$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = cos\frac{\theta}{2}I - i\,sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix},$$

donde el símbolo $\equiv$ indica acciones equivalentes salvo una fase global. Se puede observar que $R_X(\pi) \equiv X,\ R_Y(\pi) \equiv Y, R_Z(\pi) \equiv Z, R_Z\left(\frac{\pi}{2}\right) \equiv S, R_Z\left(\frac{\pi}{4}\right) \equiv T$.

Es común ver en algunas arquitecturas de ordenadores cuánticos el uso de una puerta de un qubit universal, que se llama la **puerta U**, que depende de tres ángulos y puede generar cualquier otra puerta de un qubit. La matriz correspondiente a esta puerta es

$$U(\theta, \varphi, \lambda) = \begin{pmatrix} cos\frac{\theta}{2} & -e^{i\lambda}sin\frac{\theta}{2} \\ e^{i\varphi}sin\frac{\theta}{2} & e^{i(\varphi+\lambda)}cos\frac{\theta}{2} \end{pmatrix}.$$

La información de esta sección es relevante porque, posteriormente, las puertas de rotación se utilizan a la hora de explicar algunos conceptos, como por ejemplo el paso a circuitos del algoritmo QAOA (véase 5.3.2.3. Circuitos).

1.3.1.4. Modelo de circuitos cuánticos: ejemplos de circuitos cuánticos

Para visualizar y unificar los conceptos que se han explicado hasta ahora en este marco teórico, así como para presentar un ejemplo de circuito cuántico para familiarizar al lector con la representación de sus componentes, en este apartado se presenta el siguiente circuito:

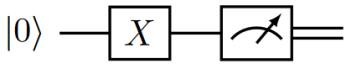


Figura 1.2. Ejemplo de circuito cuántico de un qubit

La lectura del circuito se hace de izquierda a derecha. Contiene una línea horizontal, que representa un qubit. Al inicio (es decir, a la izquierda del todo) podemos observar que el qubit comienza en el estado $|0\rangle$, que se explicó en el apartado 1.3.1.1. Qubits y mediciones.

Navegando a lo largo de la línea hay una caja que contiene la letra X. Esta es la representación de la puerta cuántica X, que se definió en la sección 1.3.1.2. Modelo de circuitos cuánticos: puertas cuánticas, y al estar sobre la línea horizontal quiere decir que la estamos aplicando sobre ese qubit.

A continuación, el siguiente componente que aparece sobre la línea es el que representa que medimos el valor del qubit, es decir, que colapsa a un valor clásico y dejamos de trabajar con información cuántica. Después de la medición la línea horizontal pasa a ser una doble. En este caso, el valor obtenido tras la medición del qubit será 1.

1.3.1.5. Modelo de circuitos cuánticos: trabajar con varios qubits

En las secciones anteriores hemos trabajado limitándonos a operaciones que se aplican sobre un solo qubit. Eso es un desperdicio, ya que no se está aprovechando una de las propiedades más importantes y utilizadas de los sistemas cuánticos, el entrelazamiento.

En esta sección se va a explicar cómo se puede trabajar de manera matemática con un sistema de varios qubits, y cómo es el proceso para entrelazarlos.

Estados de dos qubits

Para explicar cómo funcionan las interacciones entre varios qubits, primero se van a utilizar sistemas formados por dos qubits, donde cada uno puede tener el estado $|0\rangle$ o $|1\rangle$, lo que resulta en estas cuatro combinaciones:

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle,$$

que constituyen la **base computacional** del espacio de cuatro dimensiones con el que vamos a trabajar. El símbolo ⊗ indica un producto tensorial, que para dos vectores columna se calcula de esta manera:

$$\begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} = \begin{pmatrix} a_1 \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \\ a_2 \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \\ \dots \\ a_n \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ \dots \\ a_1 b_m \\ a_2 b_1 \\ a_2 b_2 \\ \dots \\ a_2 b_m \\ \dots \\ a_n b_1 \\ a_n b_2 \\ \dots \\ a_n b_m \end{pmatrix}.$$

Teniendo esto en cuenta, los estados de la base definida previamente pueden escribirse como los siguientes vectores columna

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \qquad |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \qquad |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Existen también otras formas de denotar los estados. Por ejemplo, está también aceptado no incluir el símbolo ⊗,

$$|0\rangle|0\rangle, \qquad |0\rangle|1\rangle, \qquad |1\rangle|0\rangle, \qquad |1\rangle|1\rangle.$$

También puede resumirse algo más, de esta forma:

$$|00\rangle, \qquad |01\rangle, \qquad |10\rangle, \qquad |11\rangle.$$

Y la notación más abreviada de todas, que sería

$$|0\rangle, \qquad |1\rangle, \qquad |2\rangle, \qquad |3\rangle,$$

y para la que tenemos que tener cuidado de que el número de qubits no sea ambiguo, ya que el estado $|0\rangle$ podría ser el de un sistema de un solo qubit, o de cualquier otro número de qubits.

Tras haber hablado de la base computacional que vamos a utilizar para los sistemas de dos qubits, y de distintas maneras que tenemos de escribir los estados, ahora podemos hablar de cómo se representa un estado utilizando la base, que tendría la forma

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

donde a_{00}, a_{01}, a_{10}, y a_{11} son números complejos, las amplitudes, donde se tiene que cumplir la condición $\sum_{x,y=0}^{1} |a_{xy}|^2 = 1$.

Para el estado $|\psi\rangle$ que acabamos de presentar, la probabilidad de medir 00 sería $|a_{00}|^2$, la de medir 01 sería $|a_{01}|^2$, la de 10 vendría dada por $|a_{10}|^2$, y finalmente para el resultado 11 la probabilidad sería $|a_{11}|^2$.

Al medir los qubits, estos colapsarán a un valor clásico, al igual que en los sistemas de un solo qubit, pero en los sistemas con varios qubits, por ejemplo en uno con dos qubits, puede pasar que hayamos medido solamente uno.

Si queremos calcular la probabilidad de que este qubit que hemos medido sea 0 (el caso para 1 seguiría un procedimiento parecido), habría que sumar todas las probabilidades en las que la salida del primer qubit sea 0, lo que en este caso sería $|a_{00}|^2 + |a_{01}|^2$. Si resulta que lo medimos como 0, el sistema todavía no ha colapsado del todo, y se encuentra en el estado

$$\frac{a_{00}|00\rangle + a_{01}|01\rangle}{\sqrt{|a_{00}|^2 + |a_{01}|^2}},$$

en el que tenemos que hacer la división entre $\sqrt{|a_{00}|^2 + |a_{01}|^2}$ para que siga siendo un estado normalizado.

La notación bra-ket, que en la sección 1.3.1.1. Modelo de circuitos cuánticos: qubits y mediciones se mencionó que se podía utilizar para calcular productos internos, también puede cumplir este propósito en sistemas de varios qubits, si tenemos en cuenta la igualdad

$$(\langle\psi_1| \otimes \langle\psi_2|)\,(|\varphi_1\rangle \otimes |\varphi_2\rangle) = \langle\psi_1|\varphi_1\rangle\,\langle\psi_2|\varphi_2\rangle,$$

a la que se puede aplicar la distributividad, y conjugamos los coeficientes complejos a la hora de obtener un bra de un ket.

Esto se puede ver, por ejemplo, a la hora de calcular el producto interno entre $\frac{2}{3}|11\rangle + \frac{5i}{3}|00\rangle$ y $\frac{4}{\sqrt{2}}|11\rangle + \frac{4}{\sqrt{2}}|10\rangle$, que es

$$\left(\frac{2}{3}\langle 11| - \frac{5i}{3}\langle 00|\right)\left(\frac{4}{\sqrt{2}}|11\rangle + \frac{4}{\sqrt{2}}|10\rangle\right) =$$

$$\frac{8}{3\sqrt{2}}\langle 11|11\rangle + \frac{8}{3\sqrt{2}}\langle 11|10\rangle - \frac{20i}{3\sqrt{2}}\langle 00|11\rangle - \frac{20i}{3\sqrt{2}}\langle 00|10\rangle =$$

$$\frac{8}{3\sqrt{2}}\langle 1|1\rangle\langle 1|1\rangle + \frac{8}{3\sqrt{2}}\langle 1|1\rangle\langle 1|0\rangle - \frac{20i}{3\sqrt{2}}\langle 0|1\rangle\langle 0|1\rangle - \frac{20i}{3\sqrt{2}}\langle 0|1\rangle\langle 0|0\rangle = \frac{8}{3\sqrt{2}}.$$

Puertas cuánticas de dos qubits: productos tensoriales

Al igual que para los sistemas de un solo qubit, las operaciones que se pueden realizar sobre un sistema compuesto por dos qubits o más también tienen que ser unitarias. Esto implica que las puertas cuánticas de dos qubits son matrices unitarias 4x4 que se aplican sobre vectores columna de cuatro dimensiones.

Existen varias formas de construir este tipo de matrices, pero una manera sería, por ejemplo, a través del producto tensorial de dos puertas cuánticas de un qubit. Si disponemos de dos de estas puertas de un qubit, U_1 y U_2, y de dos estados de un qubit, $|\psi_1\rangle$ y $|\psi_2\rangle$, es posible construir una puerta $U_1 \otimes U_2$, que se aplica sobre $|\psi_1\rangle \otimes |\psi_2\rangle$, de esta forma

$$(U_1 \otimes U_2)\,(|\psi_1\rangle \otimes |\psi_2\rangle) = (U_1|\psi_1\rangle) \otimes (U_2|\psi_2\rangle).$$

Esto se puede generalizar a una combinación de estados de dos qubits cualquiera.

La matriz de la puerta $U_1 \otimes U_2$ se construye a partir del producto tensorial de las matrices asociadas a U_1 y a U_2. Para dos matrices, por ejemplo A y B, ambas de tamaño $2\,x\,2$, el producto tensorial, $A \otimes B$, se calcularía de esta manera:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{12}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \\ a_{21}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{22}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix}.$$

Al tratarse de una operación unitaria, la matriz resultante cumple las condiciones para poder denominarse puerta cuántica.

1.3.1.6. La puerta CNOT

Al final de la sección anterior, se expuso una manera de construir puertas cuánticas de dos qubits a través del producto tensorial de puertas de un qubit.

Este planteamiento es correcto, pero nos limita el número de puertas de dos qubits que podemos construir, ya que un gran número de puertas no se pueden obtener del producto tensorial de otras más simples. Un ejemplo de estas puertas es la **puerta NOT controlada**, puerta X controlada, o CNOT (controlled-NOT), en inglés, la cual se representa con esta matriz:

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

La puerta NOT controlada tiene el siguiente efecto sobre la base computacional de dos qubits

$$CNOT|00\rangle = |00\rangle, \qquad CNOT|01\rangle = |01\rangle, \qquad CNOT|10\rangle = |11\rangle, \qquad CNOT|11\rangle = |10\rangle.$$

Como se puede observar en los dos últimos casos, en los estados en los que el primer qubit tiene el valor 1, la puerta NOT controlada invierte el valor del segundo qubit. En los primeros dos casos, donde el valor del primer qubit es 0, el valor del segundo qubit se mantiene. Se puede decir que el primer qubit controla el valor del segundo, que es la razón de que la puerta se llame así.

Cada uno de los qubits que participan en la puerta NOT controlada recibe un nombre, el qubit de control (en este caso el primero), y el qubit objetivo (en el ejemplo sería el segundo). En los circuitos cuánticos, la puerta NOT controlada se indica con este símbolo:

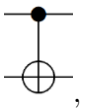


Figura 1.3. Puerta CNOT

donde el qubit de control se señala a través de un círculo negro relleno, y el qubit objetivo se indica a través del símbolo $\oplus$. En el qubit objetivo, también se puede utilizar el símbolo para una puerta X.

Debido a impedimentos técnicos, es posible que en algunas ocasiones no se pueda implementar una puerta NOT controlada en un ordenador cuántico. Puede darse el caso de que en un chip se pueda aplicar la puerta teniendo como objetivo el qubit 1 y como controlador el qubit 0, pero no viceversa.

En un escenario así, es posible utilizar el siguiente circuito, en el cual el qubit objetivo sería el superior, y el de control el inferior

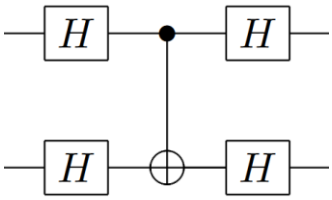


Figura 1.4. Circuito para invertir qubits en puerta NOT controlada

La puerta NOT controlada también se puede utilizar para intercambiar el estado de dos qubits, a través del siguiente circuito

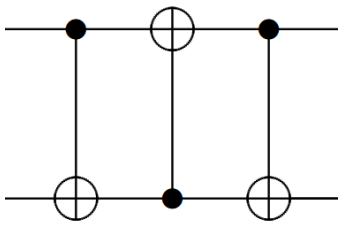


Figura 1.5. Circuito para intercambiar estado de dos qubits

Otro uso, bastante importante, que se le puede dar a la puerta NOT controlada, y que justifica en gran parte su inclusión en el marco teórico de este trabajo, es su capacidad para crear entrelazamiento entre qubits.

1.3.1.7. Entrelazamiento

Un estado $|\psi\rangle$ está entrelazado si no es un estado producto, es decir, cuando no se puede escribir como el producto tensorial de otros dos estados $|\psi_1\rangle$ y $|\psi_2\rangle$, cada uno con por lo menos un qubit, de esta forma

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle.$$

Un ejemplo de estado producto es $|01\rangle$, ya que se puede escribir también como $|0\rangle \otimes |1\rangle$.

Un estado entrelazado, que no se puede escribir como el producto de dos estados de un qubit, sería $\sqrt{1/2}\,(|00\rangle + |11\rangle)$, que se puede crear de la siguiente manera:

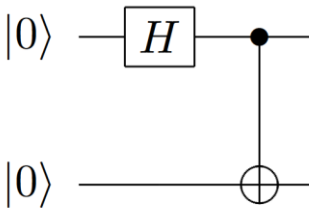


Figura 1.6. Ejemplo de circuito para construir estado entrelazado

Este estado, además, forma parte de un conjunto conocido como **estados de Bell**, de los que hay cuatro, y todos ellos están entrelazados. Los otros tres son $\sqrt{1/2}\,(|00\rangle - |11\rangle)$, $\sqrt{1/2}\,(|10\rangle + |01\rangle)$ y $\sqrt{1/2}\,(|10\rangle - |01\rangle)$.

Los estados entrelazados presentan una particularidad a la hora de medirlos, que se puede visualizar fácilmente a través de un ejemplo, para el que podemos usar el estado entrelazado $\sqrt{1/2}\,(|00\rangle + |11\rangle$, presentado anteriormente.

Inicialmente, sin haber hecho ninguna medición previa, si intentamos obtener el valor del primer qubit, este tendrá una probabilidad de 1/2 de ser un 0 y 1/2 de probabilidad de ser un 1. Lo peculiar ocurre cuando intentemos medir el valor del segundo qubit, el cual estará ya decidido por el valor que obtuvimos en el primero, que en este estado en concreto, será igual.

Si decidimos medir primero el segundo qubit y después el primero, el efecto será el mismo, pero esta vez el valor del segundo qubit será el que determine el resultado del primero.

Este fenómeno, en el que un qubit parece conocer el resultado de la medición del otro, prevalece a cualquier distancia entre los qubits, se ha demostrado en numerosos experimentos a lo largo del tiempo (algunos ejemplos son [2]–[5]), y es una herramienta muy poderosa de la que disponen los ordenadores cuánticos.

1.3.1.8. Sistemas de múltiples qubits

Tras haber explicado como trabajar con un solo qubit, y posteriormente con sistemas de dos qubits, queda por hablar de cómo extender las ideas vistas para sistemas que estén formados por más qubits.

Respecto a la base computacional, en un sistema de n qubits, esta viene dada por los siguientes estados

$$|0\rangle \otimes |0\rangle \otimes \ ... \ \otimes |0\rangle,$$

$$|0\rangle \otimes |0\rangle \otimes \ ... \ \otimes |1\rangle,$$

$$...$$

$$|1\rangle \otimes |1\rangle \otimes \ ... \ \otimes |1\rangle.$$

Donde, al igual que se ha visto en la sección 1.3.1.5. Modelo de circuitos cuánticos: trabajar con varios qubits, se puede omitir el símbolo $\otimes$, y se puede escribir también de otras maneras más abreviadas como

$$|0\rangle\,|0\rangle\ ...\ |0\rangle,$$

$$|0\rangle\,|0\rangle\ ...\ |1\rangle,$$

$$...$$

$$|1\rangle\,|1\rangle\ ...\ |1\rangle,$$

esta otra

$$|00\ldots0\rangle, |00\ldots1\rangle, \ldots\ |11\ldots1\rangle,$$

o también como

$$|0\rangle, |1\rangle, \dots |2^n - 1\rangle,$$

siempre especificando el número de qubits que se está utilizando, para evitar confusiones.

Un estado genérico de dicho sistema tendría esta representación

$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle + \cdots + a_{2^{n-1}}|2^n - 1\rangle,$$

donde se tiene que dar que las amplitudes a_i deberían ser números complejos en los que la condición de normalización se cumpla:

$$\sum_{l=0}^{2^{n-1}} |a_l|^2 = 1.$$

La probabilidad de medir m en la base computacional sería $|a_m|^2$, en cuyo caso el estado colapsaría a $|m\rangle$, pero si sólo midieramos uno de los qubits, el j-ésimo, tendríamos esta probabilidad de obtener 0

$$\sum_{l \in J_0} |a_l|^2 ,$$

donde J_0 es el conjunto de números en los que el j-ésimo bit es 0. El sistema, tras medir el 0, estaría en este estado, en el que la división mantiene la normalización

$$\frac{\sum_{l \in J_0} a_l \, |l\rangle}{\sqrt{\sum_{l \in J_0} |a_i|^2}}.$$

Para calcular el producto interno en estos sistemas utilizando la notación de Dirac, el procedimiento es parecido al de los sistemas de dos qubits

$$(\langle\psi_1| \otimes \dots \otimes \langle\psi_n|)\,(|\varphi_1\rangle \otimes \dots \otimes |\varphi_n\rangle) = \langle\psi_1|\varphi_1\rangle \dots \langle\psi_n|\varphi_n\rangle.$$

Puertas cuánticas de múltiples qubits

El objetivo de este apartado es explicar cómo realizar operaciones cuando tenemos varios qubits. Las puertas de n-qubits se expresan como matrices unitarias de 2^n x 2^n, ya que los estados formados por n-qubits se definen por vectores columna de dimensión 2^n.

Se pueden construir dichas puertas a través del producto tensorial de puertas con un número de qubits menor (como ya se vio anteriormente en la sección Puertas cuánticas de dos qubits: productos tensoriales). Si tenemos, por ejemplo, una puerta U_1 de n_1 qubits y una puerta U_2 de n_2 qubits, y calculamos el producto tensorial entre ellas, $U_1 \otimes U_2$, el resultado será una puerta de $n_1 + n_2$ qubits, cuya matriz será el resultado de dicho producto tensorial.

Este planteamiento tiene el mismo problema que vimos cuando lo explicamos a la hora de trabajar con sistemas de dos qubits, que es que no todas las puertas se pueden construir de esta forma. Un ejemplo de esto es la puerta de Toffoli, que se puede utilizar para construir cualquier operador booleano.

Puertas universales

Los ordenadores cuánticos con los que contamos hoy en día no son capaces de integrar todas las puertas cuánticas que se pueden construir.

Para solucionar esto, se utilizan resultados de universalidad, que demuestran que cualquier operación unitaria se puede descomponer en un circuito que utiliza solamente ciertas puertas primitivas. Por ejemplo, para cualquier operación unitaria, se puede hacer un circuito que la implemente a través de puertas de un qubit y la puerta NOT controlada. Estas puertas se denominan universales.

Capítulo 2

Fijación de objetivos

El objetivo de este trabajo es aumentar la cantidad de información existente sobre las capacidades de los algoritmos cuánticos actuales a la hora de resolver distintos problemas de optimización combinatoria.

Dichos algoritmos se van a poner a prueba utilizando instancias de problemas que variarán en dificultad, hasta llegar al límite de tamaño que permitan los recursos disponibles para ejecutar los métodos, y posteriormente se analizarán los resultados comparando numerosas métricas.

Otro objetivo, quizás algo más indirecto, que se busca con el trabajo, es llevar al lector a desarrollar una mejor perspectiva sobre el estado actual de la computación cuántica, que es una disciplina que todavía se encuentra en una fase inicial, y no está tan extendida como otras en el ámbito de la informática.

2.1. Ámbitos de aplicación

Los resultados de este trabajo pueden utilizarse para varias finalidades. Una de ellas es, por ejemplo, ayudar a alguien que esté buscando información sobre el rendimiento de los métodos para solucionar algún problema en concreto, para poder comparar los resultados obtenidos y tomar una decisión respecto a cuál debería utilizar.

El trabajo también puede servir como fuente para aprender a trabajar con los problemas y utilizar estos métodos, ya que actualmente no existe mucha información respecto a cómo hacerlo en Internet, y en este trabajo se explican varios conceptos importantes, como pueden ser la formulación de los problemas, o las librerías y parámetros que se utilizan para ejecutar los algoritmos. Además del trabajo, también se pone a disposición del lector un enlace al repositorio con el código utilizado para realizar los experimentos (véase 8.2. Difusión de resultados).

Los resultados del experimento también pueden tenerse en cuenta para otras investigaciones, ya que se ha asegurado, en la medida de lo posible, de hacer que los resultados sean reproducibles, a excepción de los momentos en los que esto quedaba fuera de la capacidad del programador (por ejemplo, al utilizar el annealer, que es un dispositivo inherentemente probabilista).

Capítulo 3

Planificación y gestión del TFG

3.1. Planificación del proyecto

La planificación de un proyecto es una de las partes fundamentales para garantizar su realización dentro de unos plazos y costes determinados.

En esta sección se van a presentar distintos aspectos importantes que forman parte de este proceso, como pueden ser las tareas que se van a desarrollar, o el presupuesto, entre otros.

3.1.1. Identificación de interesados (stakeholders)

Una parte importante de la planificación de un proyecto es la identificación de las personas que puedan estar interesadas en él, para asegurarse de que se cumplen con los requisitos o necesidades que estos stakeholders puedan tener.

1. Universidad de Oviedo
    a. Escuela de Ingeniería Informática (EII)
        i. Tutor del TFG
        ii. Director de la EII
        iii. Estudiantes de la EII
    b. Departamento de informática
2. Comunidad científica
    a. Investigadores de computación cuántica
    b. Investigadores de problemas de optimización
3. Investigador y desarrollador: Stelian Adrian Stanci

3.1.2. OBS y PBS

Estos diagramas aportan claridad sobre los roles que se van a desempeñar durante el proyecto y sobre los entregables que se van generando en su realización.

3.1.2.1. OBS

El OBS, que en español se conoce como Estructura de Desglose Organizacional (viniendo las siglas del inglés, *Organizational Breakdown Structure*), se trata de una

representación de las responsabilidades sobre la realización de las actividades del proyecto.

De forma resumida, el OBS describe quién hace el qué dentro del proyecto. Este TFG, al no ser en conjunto entre varias personas, solamente tiene un actor, pero este sigue pudiendo desempeñar varios roles distintos, los cuales figuran en este diagrama, y que varían, por ejemplo, en las tareas a desempeñar y el coste. El diagrama se puede observar en la Figura 3.1. Diagrama OBS del proyecto.

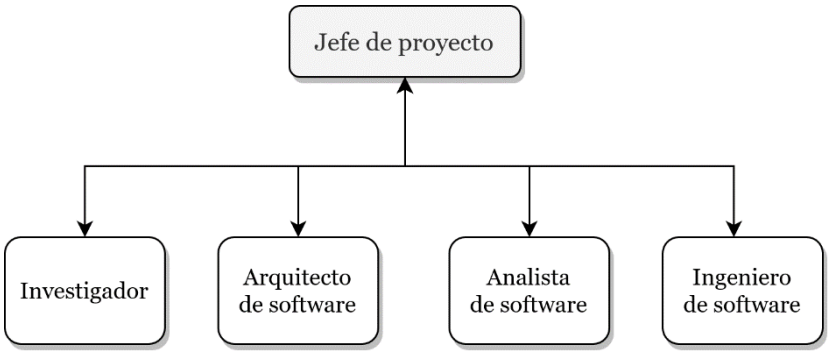


Figura 3.1. Diagrama OBS del proyecto

3.1.2.2. PBS

La Estructura de Desglose de Producto, en inglés conocida como *Product Breakdown Structure*, es una representación, al igual que el OBS en forma de árbol, de los productos que hay que crear para alcanzar el final del proyecto: los entregables. Los entregables están estrechamente relacionados con las actividades, ya que se obtienen a través de ellas.

Para facilitar la visualización de los esquemas, se ha decidido dividir el PBS en varias partes: un PBS general, y otros para la formación, el estudio, el desarrollo, y la documentación.

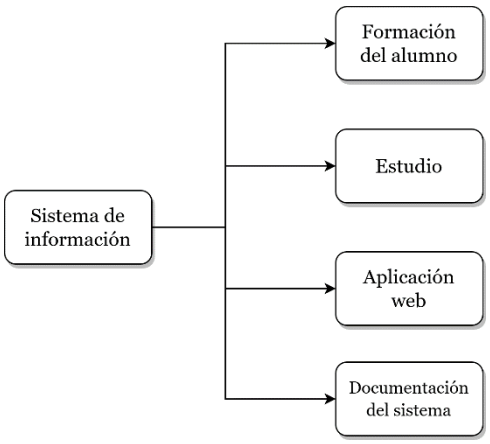


Figura 3.2. PBS General

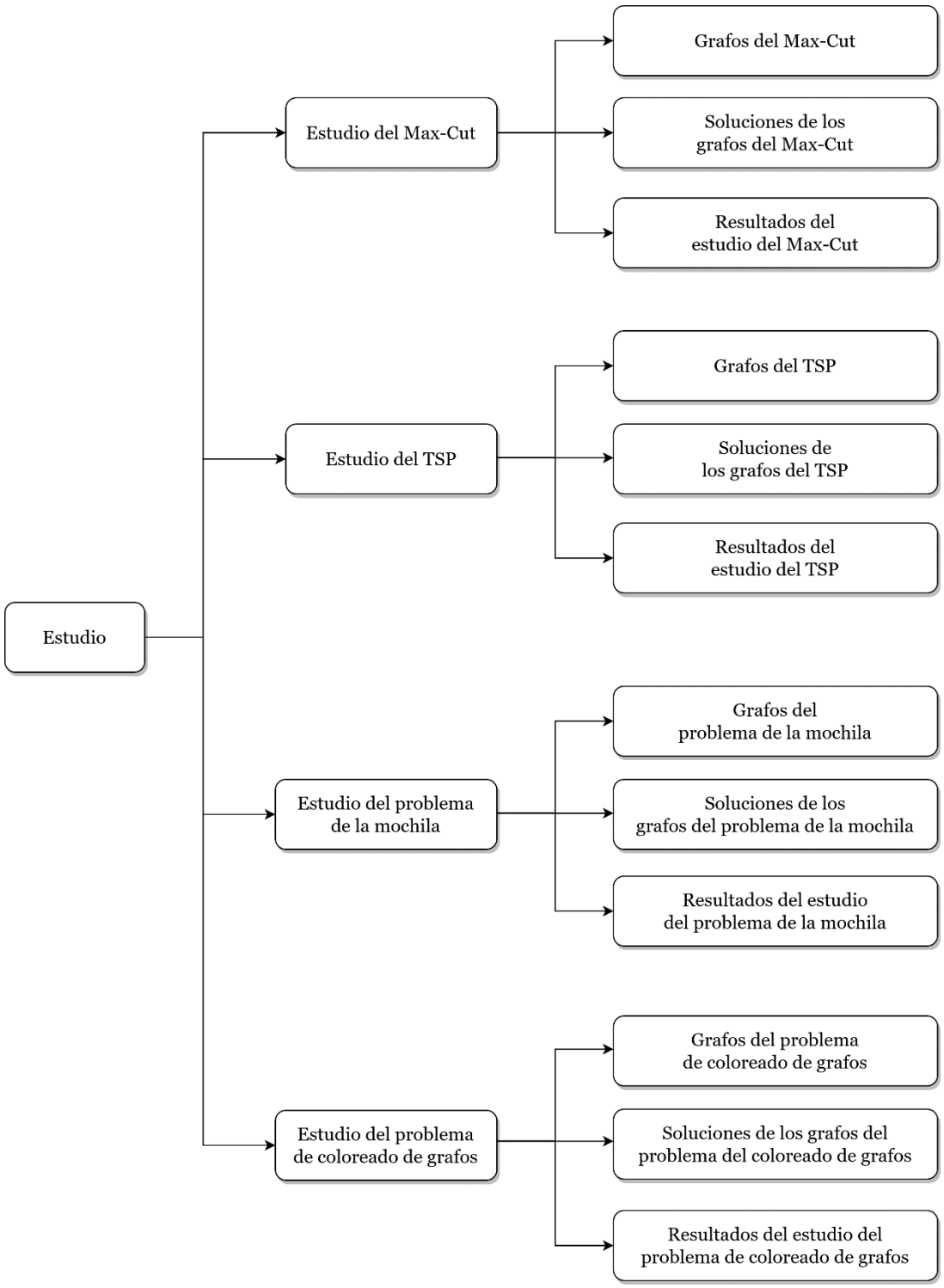


Figura 3.3. PBS del estudio

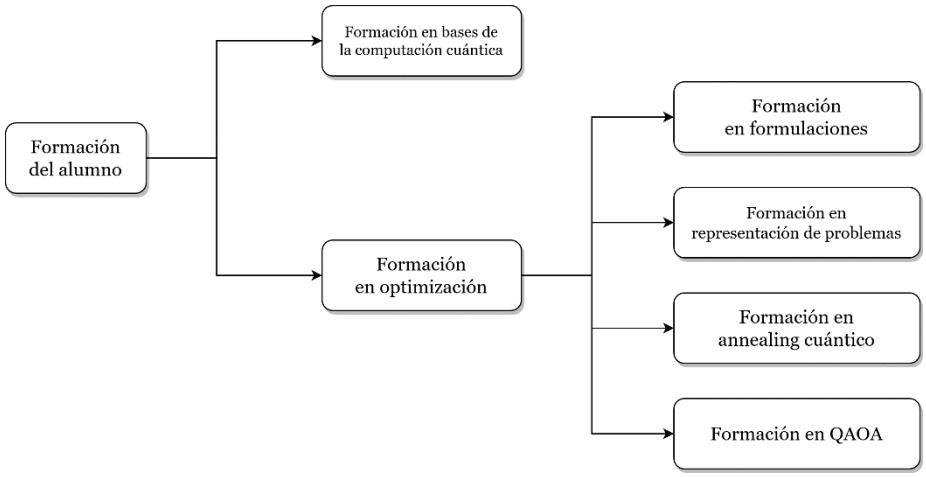


Figura 3.4. PBS de la formación

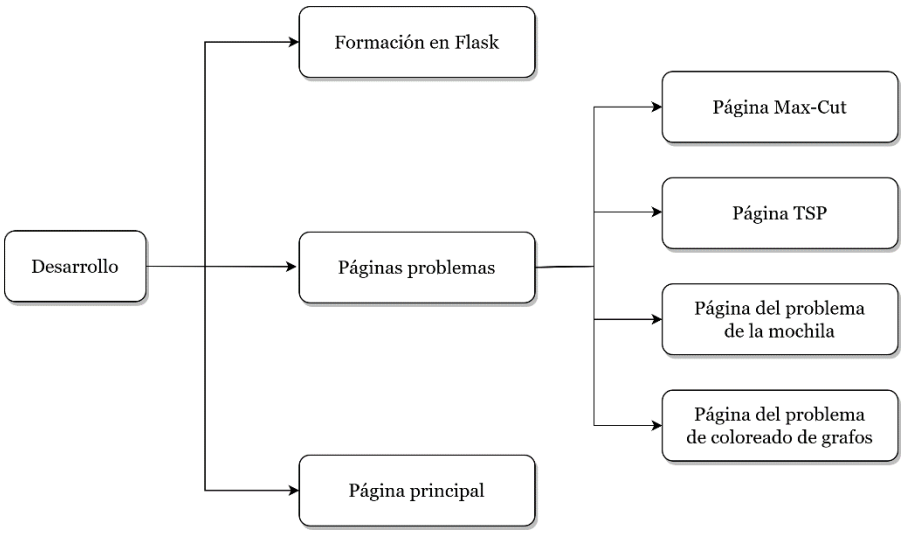


Figura 3.5. PBS del desarrollo

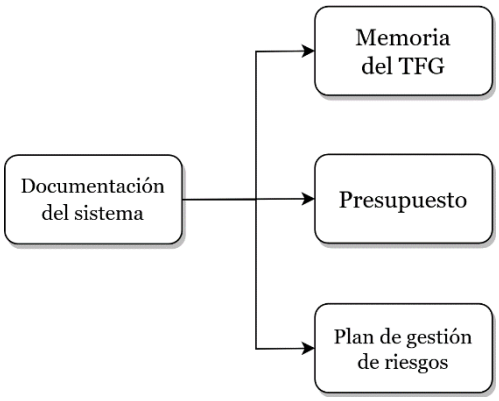


Figura 3.6. PBS de la documentación

3.1.3. Planificación inicial del TFG. WBS.

El objetivo de este apartado es describir las tareas que se han de llevar a cabo en la realización del TFG, así como su extensión en el tiempo (lo cual incluye día de comienzo y de fin de cada tarea y su duración). Esta información aparece de forma resumida en el *Work Breakdown Structure* (en español, Estructura de Descomposición del Trabajo) del proyecto, también conocido por su abreviatura, WBS.

La duración inicial estimada del proyecto es de 75 días, siendo el día de inicio el 10 de febrero de 2023 y el día de fin el 25 de mayo de 2023. El recurso asociado a la realización de las tareas es el propio alumno que realiza el TFG.

A pesar de ser un solo alumno, este desempeña varios roles, lo cual también se ve reflejado en los recursos, y cada rol tiene un coste distinto basado en la complejidad del trabajo a realizar, y también un horario de trabajo distinto.

Al tratarse de un diagrama tan extenso, este ha sido dividido en varias figuras distintas con el fin de que se pueda interpretar mejor.

Primero se presentan los hitos:

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predi
1	**1**	**Diseño e implementación de algoritmos cuánticos para problemas de optimización combinatoria**	**75 días**	**vie 10/02/23**	**jue 25/05/23**	**No**	
2	**1.1**	**Hitos**	**74,69 días**	**vie 10/02/23**	**jue 25/05/23**	**No**	
3	1.1.1	Reunión de arranque	0 días	vie 10/02/23	vie 10/02/23	Sí	113
4	**1.1.2**	**Formación en computación cuántica**	**14,38 días**	**mié 22/02/23**	**mar 14/03/23**	**No**	
5	1.1.2.1	Formación en las bases de la computación cuántica	0 días	mié 22/02/23	mié 22/02/23	Sí	115
6	**1.1.2.2**	**Formación en optimización cuántica**	**13 días**	**jue 23/02/23**	**mar 14/03/23**	**No**	
7	**1.1.2.2.1**	**Formación en formulaciones y problemas**	**5,75 días**	**jue 23/02/23**	**vie 03/03/23**	**No**	
8	1.1.2.2.1.1	Formación en el framework QUBO	0 días	jue 23/02/23	jue 23/02/23	Sí	118
9	1.1.2.2.1.2	Formación en el modelo de Ising	0 días	vie 24/02/23	vie 24/02/23	Sí	119
10	1.1.2.2.1.3	Formación en paso de Ising a QUBO	0 días	lun 27/02/23	lun 27/02/23	Sí	120
11	1.1.2.2.1.4	Formación en representación de problemas de optimización combinatoria	0 días	vie 03/03/23	vie 03/03/23	Sí	121
12	**1.1.2.2.2**	**Formación en annealing cuántico**	**2 días**	**lun 06/03/23**	**mié 08/03/23**	**No**	
13	1.1.2.2.2.1	Formación en computación adiabática	0 días	lun 06/03/23	lun 06/03/23	Sí	123
14	1.1.2.2.2.2	Formación en annealing cuántico	0 días	mar 07/03/23	mar 07/03/23	Sí	124
15	1.1.2.2.2.3	Formación en uso de annealers de D-Wave	0 días	mié 08/03/23	mié 08/03/23	Sí	125
16	**1.1.2.2.3**	**Formación en QAOA**	**3 días**	**jue 09/03/23**	**mar 14/03/23**	**No**	
17	1.1.2.2.3.1	Formación en paso de annealing cuántico a QAOA	0 días	jue 09/03/23	jue 09/03/23	Sí	127
18	1.1.2.2.3.2	Formación en resolución de problemas utilizando el QAOA	0 días	mar 14/03/23	mar 14/03/23	Sí	128
19	**1.1.3**	**Realización del estudio**	**6,25 días**	**mié 05/04/23**	**jue 13/04/23**	**No**	
20	1.1.3.1	Estudio del Max-Cut	0 días	mié 05/04/23	mié 05/04/23	Sí	130
21	1.1.3.2	Estudio del TSP	0 días	vie 07/04/23	vie 07/04/23	Sí	134
22	1.1.3.3	Estudio del problema de la mochila	0 días	mar 11/04/23	mar 11/04/23	Sí	138
23	1.1.3.4	Estudio del problema del coloreado de grafos	0 días	jue 13/04/23	jue 13/04/23	Sí	142
24	**1.1.4**	**Desarrollo de aplicación web**	**12 días**	**jue 30/03/23**	**lun 17/04/23**	**No**	

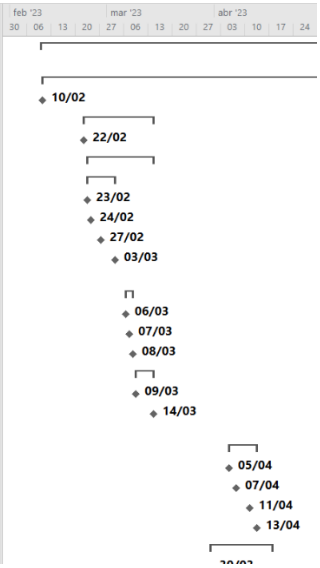


Figura 3.7. Hitos (1)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predec
19	**1.1.3**	**Realización del estudio**	**6,25 días**	**mié 05/04/23**	**jue 13/04/23**	**No**	
20	1.1.3.1	Estudio del Max-Cut	0 días	mié 05/04/23	mié 05/04/23	Sí	130
21	1.1.3.2	Estudio del TSP	0 días	vie 07/04/23	vie 07/04/23	Sí	134
22	1.1.3.3	Estudio del problema de la mochila	0 días	mar 11/04/23	mar 11/04/23	Sí	138
23	1.1.3.4	Estudio del problema del coloreado de grafos	0 días	jue 13/04/23	jue 13/04/23	Sí	142
24	**1.1.4**	**Desarrollo de aplicación web**	**12 días**	**jue 30/03/23**	**lun 17/04/23**	**No**	
25	1.1.4.1	Formación en Flask	0 días	jue 30/03/23	jue 30/03/23	Sí	147
26	1.1.4.2	Implementación página principal	0 días	vie 31/03/23	vie 31/03/23	Sí	148
27	1.1.4.3	Página Max-Cut	0 días	jue 06/04/23	jue 06/04/23	Sí	149
28	1.1.4.4	Página TSP	0 días	mar 11/04/23	mar 11/04/23	Sí	153
29	1.1.4.5	Página del problema de la mochila	0 días	jue 13/04/23	jue 13/04/23	Sí	157
30	1.1.4.6	Página del problema del coloreado de grafos	0 días	lun 17/04/23	lun 17/04/23	Sí	161
31	**1.1.5**	**Documentación del proyecto**	**52,06 días**	**mar 14/03/23**	**jue 25/05/23**	**No**	
32	1.1.5.1	Abstract	0 días	jue 13/04/23	jue 13/04/23	Sí	166
33	**1.1.5.2**	**Introducción**	**6,13 días**	**vie 14/04/23**	**lun 24/04/23**	**No**	
34	1.1.5.2.1	Motivación	0 días	vie 14/04/23	vie 14/04/23	Sí	168
35	1.1.5.2.2	Finalidad del proyecto	0 días	vie 14/04/23	vie 14/04/23	Sí	169
36	1.1.5.2.3	Marco teórico	0 días	lun 24/04/23	lun 24/04/23	Sí	170
37	1.1.5.3	Fijación de objetivos	0 días	mar 25/04/23	mar 25/04/23	Sí	171
38	**1.1.5.4**	**Planificación y gestión del TFG**	**52,06 días**	**mar 14/03/23**	**jue 25/05/23**	**No**	
39	**1.1.5.4.1**	**Planificación del proyecto**	**11,25 días**	**mar 14/03/23**	**mié 29/03/23**	**No**	
40	1.1.5.4.1.1	Identificación de interesados	0 días	mar 14/03/23	mar 14/03/23	Sí	174
41	1.1.5.4.1.2	OBS	0 días	mar 14/03/23	mar 14/03/23	Sí	175
42	1.1.5.4.1.3	PBS	0 días	mar 21/03/23	mar 21/03/23	Sí	176
43	1.1.5.4.1.4	WBS	0 días	lun 27/03/23	lun 27/03/23	Sí	177
44	**1.1.5.4.1.5**	**Riesgos**	**0,88 días**	**lun 27/03/23**	**mar 28/03/23**	**No**	

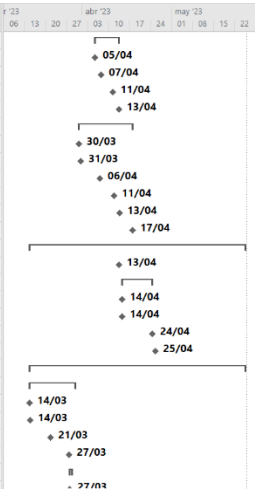


Figura 3.8. Hitos (2)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predec
37	1.1.5.3	Fijación de objetivos	0 días	mar 25/04/23	mar 25/04/23	Sí	171
38	**1.1.5.4**	**Planificación y gestión del TFG**	**52,06 días**	**mar 14/03/23**	**jue 25/05/23**	**No**	
39	**1.1.5.4.1**	**Planificación del proyecto**	**11,25 días**	**mar 14/03/23**	**mié 29/03/23**	**No**	
40	1.1.5.4.1.1	Identificación de interesados	0 días	mar 14/03/23	mar 14/03/23	Sí	174
41	1.1.5.4.1.2	OBS	0 días	mar 14/03/23	mar 14/03/23	Sí	175
42	1.1.5.4.1.3	PBS	0 días	mar 21/03/23	mar 21/03/23	Sí	176
43	1.1.5.4.1.4	WBS	0 días	lun 27/03/23	lun 27/03/23	Sí	177
44	**1.1.5.4.1.5**	**Riesgos**	**0,88 días**	**lun 27/03/23**	**mar 28/03/23**	**No**	
45	1.1.5.4.1.5	Plan de gestión de riesgos	0 días	lun 27/03/23	lun 27/03/23	Sí	179
46	1.1.5.4.1.5	Identificación de riesgos	0 días	mar 28/03/23	mar 28/03/23	Sí	180
47	**1.1.5.4.1.6**	**Presupuesto inicial del TFG**	**0,25 días**	**mié 29/03/23**	**mié 29/03/23**	**No**	
48	1.1.5.4.1.6	Presupuesto de costes	0 días	mié 29/03/23	mié 29/03/23	Sí	182
49	1.1.5.4.1.6	Presupuesto de cliente	0 días	mié 29/03/23	mié 29/03/23	Sí	183
50	**1.1.5.4.2**	**Ejecución del proyecto**	**0,88 días**	**mar 23/05/23**	**mié 24/05/23**	**No**	
51	1.1.5.4.2.1	Plan de seguimiento de planificación	0 días	mar 23/05/23	mar 23/05/23	Sí	185
52	1.1.5.4.2.2	Bitácora de incidencias del proyecto	0 días	mar 23/05/23	mar 23/05/23	Sí	186
53	1.1.5.4.2.3	Riesgos	0 días	mié 24/05/23	mié 24/05/23	Sí	187
54	**1.1.5.4.3**	**Cierre del proyecto**	**0,88 días**	**mié 24/05/23**	**jue 25/05/23**	**No**	
55	1.1.5.4.3.1	Planificación final	0 días	mié 24/05/23	mié 24/05/23	Sí	189
56	1.1.5.4.3.2	Informe final de riesgos	0 días	mié 24/05/23	mié 24/05/23	Sí	190
57	1.1.5.4.3.3	Presupuesto final de costes	0 días	jue 25/05/23	jue 25/05/23	Sí	191
58	1.1.5.4.3.4	Informe de lecciones aprendidas	0 días	jue 25/05/23	jue 25/05/23	Sí	192
59	1.1.5.5	Estado del arte	0 días	lun 01/05/23	lun 01/05/23	Sí	193
60	**1.1.5.6**	**Descripción del sistema**	**6,63 días**	**lun 01/05/23**	**mié 10/05/23**	**No**	
61	**1.1.5.6.1**	**Formulaciones**	**0,75 días**	**lun 01/05/23**	**mar 02/05/23**	**No**	
62	1.1.5.6.1.1	Framework QUBO	0 días	lun 01/05/23	lun 01/05/23	Sí	196

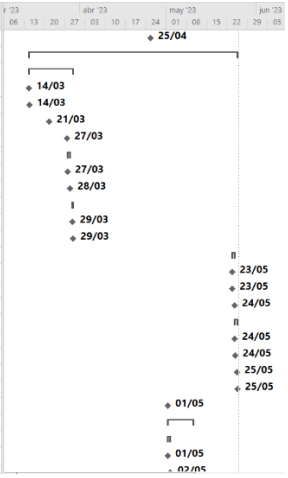
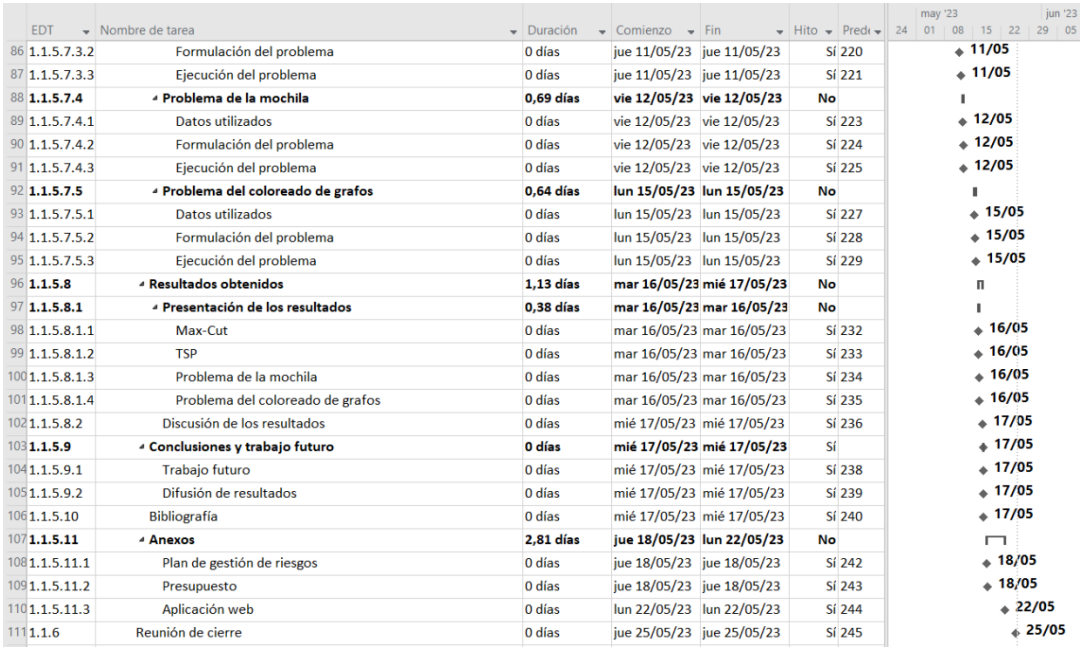


Figura 3.9. Hitos (3)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predec
61	**1.1.5.6.1**	**Formulaciones**	**0,75 días**	**lun 01/05/23**	**mar 02/05/23**	**No**	
62	1.1.5.6.1.1	Framework QUBO	0 días	lun 01/05/23	lun 01/05/23	Sí	196
63	1.1.5.6.1.2	Modelo de Ising	0 días	mar 02/05/23	mar 02/05/23	Sí	197
64	**1.1.5.6.2**	**Problemas**	**2 días**	**mar 02/05/23**	**jue 04/05/23**	**No**	
65	1.1.5.6.2.1	Max-Cut	0 días	mar 02/05/23	mar 02/05/23	Sí	199
66	1.1.5.6.2.2	Programación lineal binaria	0 días	mié 03/05/23	mié 03/05/23	Sí	200
67	1.1.5.6.2.3	TSP	0 días	mié 03/05/23	mié 03/05/23	Sí	201
68	1.1.5.6.2.4	Problema de la mochila	0 días	jue 04/05/23	jue 04/05/23	Sí	202
69	1.1.5.6.2.5	Problema del coloreado de grafos	0 días	jue 04/05/23	jue 04/05/23	Sí	203
70	**1.1.5.6.3**	**Métodos**	**2,75 días**	**vie 05/05/23**	**mié 10/05/23**	**No**	
71	**1.1.5.6.3.1**	**Annealing cuántico**	**0,13 días**	**vie 05/05/23**	**vie 05/05/23**	**No**	
72	1.1.5.6.3.1	Computación adiabática cuántica	0 días	vie 05/05/23	vie 05/05/23	Sí	206
73	1.1.5.6.3.1	Annealing cuántico	0 días	vie 05/05/23	vie 05/05/23	Sí	207
74	**1.1.5.6.3.2**	**QAOA**	**1,88 días**	**lun 08/05/23**	**mié 10/05/23**	**No**	
75	1.1.5.6.3.2	Definición	0 días	lun 08/05/23	lun 08/05/23	Sí	209
76	1.1.5.6.3.2	Algoritmo	0 días	lun 08/05/23	lun 08/05/23	Sí	210
77	1.1.5.6.3.2	Paso a circuitos	0 días	mié 10/05/23	mié 10/05/23	Sí	211
78	**1.1.5.7**	**Metodología de trabajo**	**3,29 días**	**mié 10/05/23**	**lun 15/05/23**	**No**	
79	1.1.5.7.1	Explicación de métodos a utilizar	0 días	mié 10/05/23	mié 10/05/23	Sí	213
80	**1.1.5.7.2**	**Max-Cut**	**0,59 días**	**mié 10/05/23**	**mié 10/05/23**	**No**	
81	1.1.5.7.2.1	Datos utilizados	0 días	mié 10/05/23	mié 10/05/23	Sí	215
82	1.1.5.7.2.2	Formulación del problema	0 días	mié 10/05/23	mié 10/05/23	Sí	216
83	1.1.5.7.2.3	Ejecución del problema	0 días	mié 10/05/23	mié 10/05/23	Sí	217
84	**1.1.5.7.3**	**TSP**	**0,97 días**	**jue 11/05/23**	**jue 11/05/23**	**No**	
85	1.1.5.7.3.1	Datos utilizados	0 días	jue 11/05/23	jue 11/05/23	Sí	219
86	1.1.5.7.3.2	Formulación del problema	0 días	jue 11/05/23	jue 11/05/23	Sí	220

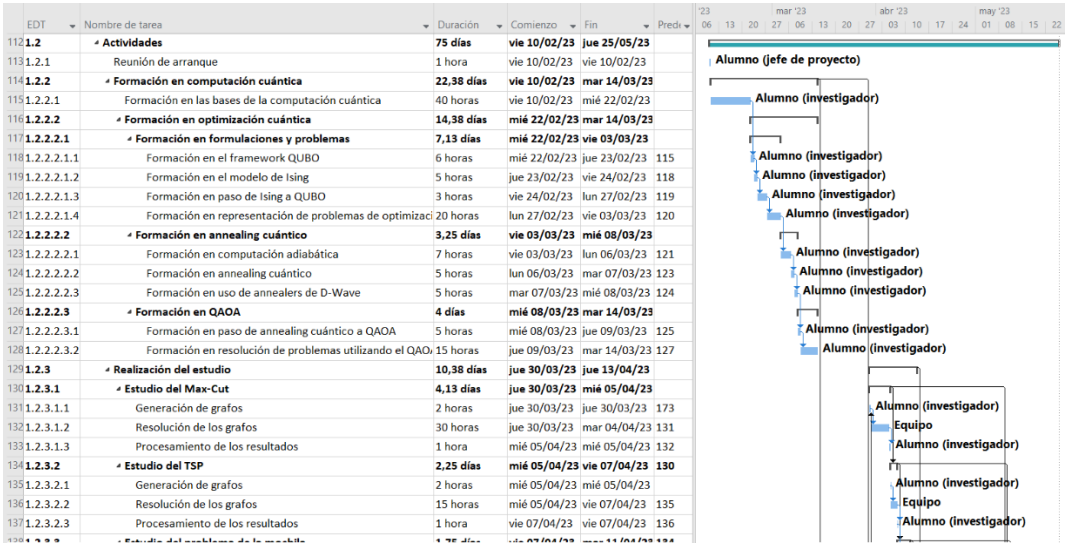
Figura 3.10. Hitos (4)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predec
86	1.1.5.7.3.2	Formulación del problema	0 días	jue 11/05/23	jue 11/05/23	Sí	220
87	1.1.5.7.3.3	Ejecución del problema	0 días	jue 11/05/23	jue 11/05/23	Sí	221
88	**1.1.5.7.4**	**Problema de la mochila**	**0,69 días**	**vie 12/05/23**	**vie 12/05/23**	**No**	
89	1.1.5.7.4.1	Datos utilizados	0 días	vie 12/05/23	vie 12/05/23	Sí	223
90	1.1.5.7.4.2	Formulación del problema	0 días	vie 12/05/23	vie 12/05/23	Sí	224
91	1.1.5.7.4.3	Ejecución del problema	0 días	vie 12/05/23	vie 12/05/23	Sí	225
92	**1.1.5.7.5**	**Problema del coloreado de grafos**	**0,64 días**	**lun 15/05/23**	**lun 15/05/23**	**No**	
93	1.1.5.7.5.1	Datos utilizados	0 días	lun 15/05/23	lun 15/05/23	Sí	227
94	1.1.5.7.5.2	Formulación del problema	0 días	lun 15/05/23	lun 15/05/23	Sí	228
95	1.1.5.7.5.3	Ejecución del problema	0 días	lun 15/05/23	lun 15/05/23	Sí	229
96	**1.1.5.8**	**Resultados obtenidos**	**1,13 días**	**mar 16/05/23**	**mié 17/05/23**	**No**	
97	**1.1.5.8.1**	**Presentación de los resultados**	**0,38 días**	**mar 16/05/23**	**mar 16/05/23**	**No**	
98	1.1.5.8.1.1	Max-Cut	0 días	mar 16/05/23	mar 16/05/23	Sí	232
99	1.1.5.8.1.2	TSP	0 días	mar 16/05/23	mar 16/05/23	Sí	233
100	1.1.5.8.1.3	Problema de la mochila	0 días	mar 16/05/23	mar 16/05/23	Sí	234
101	1.1.5.8.1.4	Problema del coloreado de grafos	0 días	mar 16/05/23	mar 16/05/23	Sí	235
102	1.1.5.8.2	Discusión de los resultados	0 días	mié 17/05/23	mié 17/05/23	Sí	236
103	**1.1.5.9**	**Conclusiones y trabajo futuro**	**0 días**	**mié 17/05/23**	**mié 17/05/23**	**Sí**	
104	1.1.5.9.1	Trabajo futuro	0 días	mié 17/05/23	mié 17/05/23	Sí	238
105	1.1.5.9.2	Difusión de resultados	0 días	mié 17/05/23	mié 17/05/23	Sí	239
106	1.1.5.10	Bibliografía	0 días	mié 17/05/23	mié 17/05/23	Sí	240
107	**1.1.5.11**	**Anexos**	**2,81 días**	**jue 18/05/23**	**lun 22/05/23**	**No**	
108	1.1.5.11.1	Plan de gestión de riesgos	0 días	jue 18/05/23	jue 18/05/23	Sí	242
109	1.1.5.11.2	Presupuesto	0 días	jue 18/05/23	jue 18/05/23	Sí	243
110	1.1.5.11.3	Aplicación web	0 días	lun 22/05/23	lun 22/05/23	Sí	244
111	1.1.6	Reunión de cierre	0 días	jue 25/05/23	jue 25/05/23	Sí	245

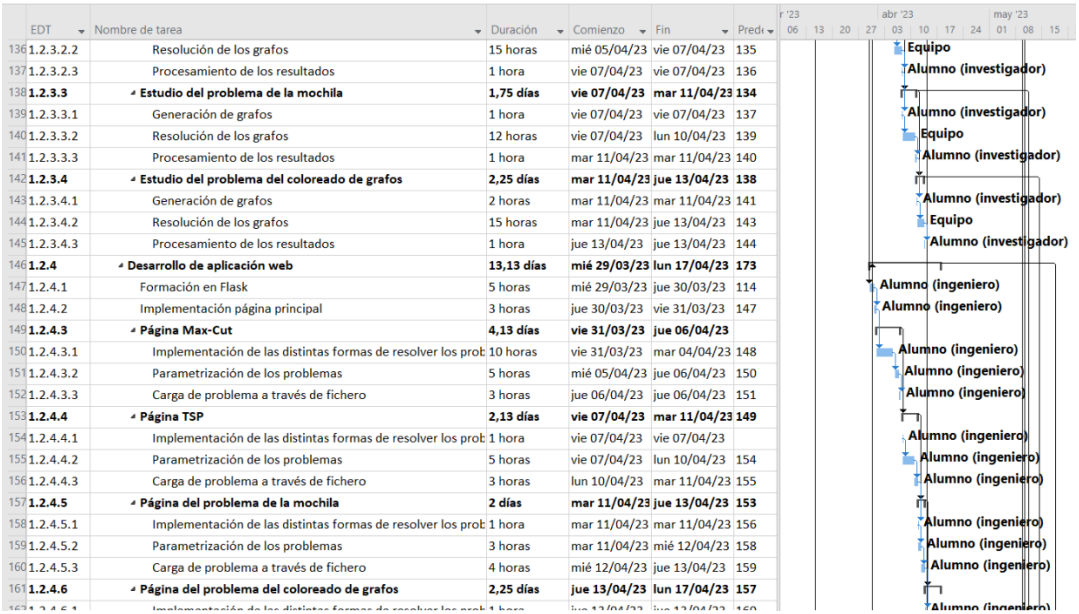
Figura 3.11. Hitos (5)

A continuación se muestran las actividades:



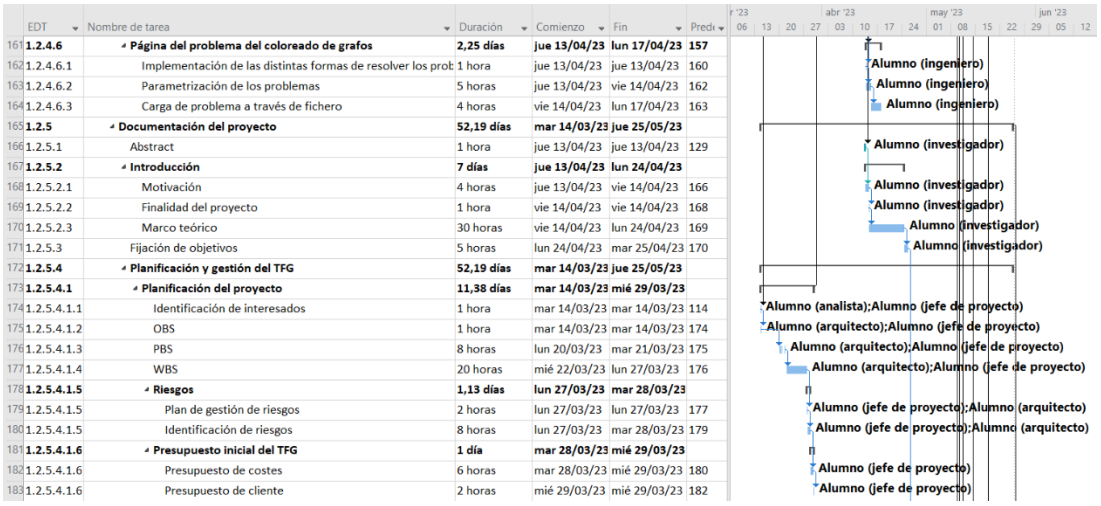
	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
112	**1.2**	**Actividades**	**75 días**	**vie 10/02/23**	**jue 25/05/23**	
113	1.2.1	Reunión de arranque	1 hora	vie 10/02/23	vie 10/02/23	
114	**1.2.2**	**Formación en computación cuántica**	**22,38 días**	**vie 10/02/23**	**mar 14/03/23**	
115	1.2.2.1	Formación en las bases de la computación cuántica	40 horas	vie 10/02/23	mié 22/02/23	
116	**1.2.2.2**	**Formación en optimización cuántica**	**14,38 días**	**mié 22/02/23**	**mar 14/03/23**	
117	**1.2.2.2.1**	**Formación en formulaciones y problemas**	**7,13 días**	**mié 22/02/23**	**vie 03/03/23**	
118	1.2.2.2.1.1	Formación en el framework QUBO	6 horas	mié 22/02/23	jue 23/02/23	115
119	1.2.2.2.1.2	Formación en el modelo de Ising	5 horas	jue 23/02/23	vie 24/02/23	118
120	1.2.2.2.1.3	Formación en paso de Ising a QUBO	3 horas	vie 24/02/23	lun 27/02/23	119
121	1.2.2.2.1.4	Formación en representación de problemas de optimizaci	20 horas	lun 27/02/23	vie 03/03/23	120
122	**1.2.2.2.2**	**Formación en annealing cuántico**	**3,25 días**	**vie 03/03/23**	**mié 08/03/23**	
123	1.2.2.2.2.1	Formación en computación adiabática	7 horas	vie 03/03/23	lun 06/03/23	121
124	1.2.2.2.2.2	Formación en annealing cuántico	5 horas	lun 06/03/23	mar 07/03/23	123
125	1.2.2.2.2.3	Formación en uso de annealers de D-Wave	5 horas	mar 07/03/23	mié 08/03/23	124
126	**1.2.2.2.3**	**Formación en QAOA**	**4 días**	**mié 08/03/23**	**mar 14/03/23**	
127	1.2.2.2.3.1	Formación en paso de annealing cuántico a QAOA	5 horas	mié 08/03/23	jue 09/03/23	125
128	1.2.2.2.3.2	Formación en resolución de problemas utilizando el QAO	15 horas	jue 09/03/23	mar 14/03/23	127
129	**1.2.3**	**Realización del estudio**	**10,38 días**	**jue 30/03/23**	**jue 13/04/23**	
130	**1.2.3.1**	**Estudio del Max-Cut**	**4,13 días**	**jue 30/03/23**	**mié 05/04/23**	
131	1.2.3.1.1	Generación de grafos	2 horas	jue 30/03/23	jue 30/03/23	173
132	1.2.3.1.2	Resolución de los grafos	30 horas	jue 30/03/23	mar 04/04/23	131
133	1.2.3.1.3	Procesamiento de los resultados	1 hora	mié 05/04/23	mié 05/04/23	132
134	**1.2.3.2**	**Estudio del TSP**	**2,25 días**	**mié 05/04/23**	**vie 07/04/23**	**130**
135	1.2.3.2.1	Generación de grafos	2 horas	mié 05/04/23	mié 05/04/23	
136	1.2.3.2.2	Resolución de los grafos	15 horas	mié 05/04/23	vie 07/04/23	135
137	1.2.3.2.3	Procesamiento de los resultados	1 hora	vie 07/04/23	vie 07/04/23	136

Figura 3.12. Actividades (1)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
136	1.2.3.2.2	Resolución de los grafos	15 horas	mié 05/04/23	vie 07/04/23	135
137	1.2.3.2.3	Procesamiento de los resultados	1 hora	vie 07/04/23	vie 07/04/23	136
138	**1.2.3.3**	**Estudio del problema de la mochila**	**1,75 días**	**vie 07/04/23**	**mar 11/04/23**	**134**
139	1.2.3.3.1	Generación de grafos	1 hora	vie 07/04/23	vie 07/04/23	137
140	1.2.3.3.2	Resolución de los grafos	12 horas	vie 07/04/23	lun 10/04/23	139
141	1.2.3.3.3	Procesamiento de los resultados	1 hora	mar 11/04/23	mar 11/04/23	140
142	**1.2.3.4**	**Estudio del problema del coloreado de grafos**	**2,25 días**	**mar 11/04/23**	**jue 13/04/23**	**138**
143	1.2.3.4.1	Generación de grafos	2 horas	mar 11/04/23	mar 11/04/23	141
144	1.2.3.4.2	Resolución de los grafos	15 horas	mar 11/04/23	jue 13/04/23	143
145	1.2.3.4.3	Procesamiento de los resultados	1 hora	jue 13/04/23	jue 13/04/23	144
146	**1.2.4**	**Desarrollo de aplicación web**	**13,13 días**	**mié 29/03/23**	**lun 17/04/23**	**173**
147	1.2.4.1	Formación en Flask	5 horas	mié 29/03/23	jue 30/03/23	114
148	1.2.4.2	Implementación página principal	3 horas	jue 30/03/23	vie 31/03/23	147
149	**1.2.4.3**	**Página Max-Cut**	**4,13 días**	**vie 31/03/23**	**jue 06/04/23**	
150	1.2.4.3.1	Implementación de las distintas formas de resolver los prob	10 horas	vie 31/03/23	mar 04/04/23	148
151	1.2.4.3.2	Parametrización de los problemas	5 horas	mié 05/04/23	jue 06/04/23	150
152	1.2.4.3.3	Carga de problema a través de fichero	3 horas	jue 06/04/23	jue 06/04/23	151
153	**1.2.4.4**	**Página TSP**	**2,13 días**	**vie 07/04/23**	**mar 11/04/23**	**149**
154	1.2.4.4.1	Implementación de las distintas formas de resolver los prob	1 hora	vie 07/04/23	vie 07/04/23	
155	1.2.4.4.2	Parametrización de los problemas	5 horas	vie 07/04/23	lun 10/04/23	154
156	1.2.4.4.3	Carga de problema a través de fichero	3 horas	lun 10/04/23	mar 11/04/23	155
157	**1.2.4.5**	**Página del problema de la mochila**	**2 días**	**mar 11/04/23**	**jue 13/04/23**	**153**
158	1.2.4.5.1	Implementación de las distintas formas de resolver los prob	1 hora	mar 11/04/23	mar 11/04/23	156
159	1.2.4.5.2	Parametrización de los problemas	3 horas	mar 11/04/23	mié 12/04/23	158
160	1.2.4.5.3	Carga de problema a través de fichero	4 horas	mié 12/04/23	jue 13/04/23	159
161	**1.2.4.6**	**Página del problema del coloreado de grafos**	**2,25 días**	**jue 13/04/23**	**lun 17/04/23**	**157**

Figura 3.13. Actividades (2)

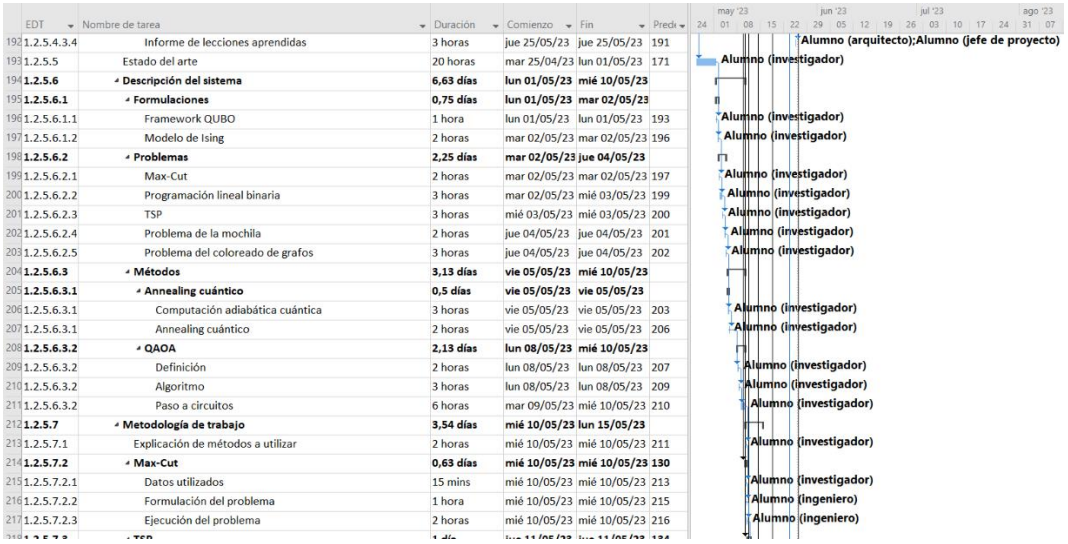


	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
161	**1.2.4.6**	**Página del problema del coloreado de grafos**	**2,25 días**	**jue 13/04/23**	**lun 17/04/23**	**157**
162	1.2.4.6.1	Implementación de las distintas formas de resolver los prob	1 hora	jue 13/04/23	jue 13/04/23	160
163	1.2.4.6.2	Parametrización de los problemas	5 horas	jue 13/04/23	vie 14/04/23	162
164	1.2.4.6.3	Carga de problema a través de fichero	4 horas	vie 14/04/23	lun 17/04/23	163
165	**1.2.5**	**Documentación del proyecto**	**52,19 días**	**mar 14/03/23**	**jue 25/05/23**	
166	1.2.5.1	Abstract	1 hora	jue 13/04/23	jue 13/04/23	129
167	**1.2.5.2**	**Introducción**	**7 días**	**jue 13/04/23**	**lun 24/04/23**	
168	1.2.5.2.1	Motivación	4 horas	jue 13/04/23	vie 14/04/23	166
169	1.2.5.2.2	Finalidad del proyecto	1 hora	vie 14/04/23	vie 14/04/23	168
170	1.2.5.2.3	Marco teórico	30 horas	vie 14/04/23	lun 24/04/23	169
171	1.2.5.3	Fijación de objetivos	5 horas	lun 24/04/23	mar 25/04/23	170
172	**1.2.5.4**	**Planificación y gestión del TFG**	**52,19 días**	**mar 14/03/23**	**jue 25/05/23**	
173	**1.2.5.4.1**	**Planificación del proyecto**	**11,38 días**	**mar 14/03/23**	**mié 29/03/23**	
174	1.2.5.4.1.1	Identificación de interesados	1 hora	mar 14/03/23	mar 14/03/23	114
175	1.2.5.4.1.2	OBS	1 hora	mar 14/03/23	mar 14/03/23	174
176	1.2.5.4.1.3	PBS	8 horas	lun 20/03/23	mar 21/03/23	175
177	1.2.5.4.1.4	WBS	20 horas	mié 22/03/23	lun 27/03/23	176
178	**1.2.5.4.1.5**	**Riesgos**	**1,13 días**	**lun 27/03/23**	**mar 28/03/23**	
179	1.2.5.4.1.5	Plan de gestión de riesgos	2 horas	lun 27/03/23	lun 27/03/23	177
180	1.2.5.4.1.5	Identificación de riesgos	8 horas	lun 27/03/23	mar 28/03/23	179
181	**1.2.5.4.1.6**	**Presupuesto inicial del TFG**	**1 día**	**mar 28/03/23**	**mié 29/03/23**	
182	1.2.5.4.1.6	Presupuesto de costes	6 horas	mar 28/03/23	mié 29/03/23	180
183	1.2.5.4.1.6	Presupuesto de cliente	2 horas	mié 29/03/23	mié 29/03/23	182

Figura 3.14. Actividades (3)

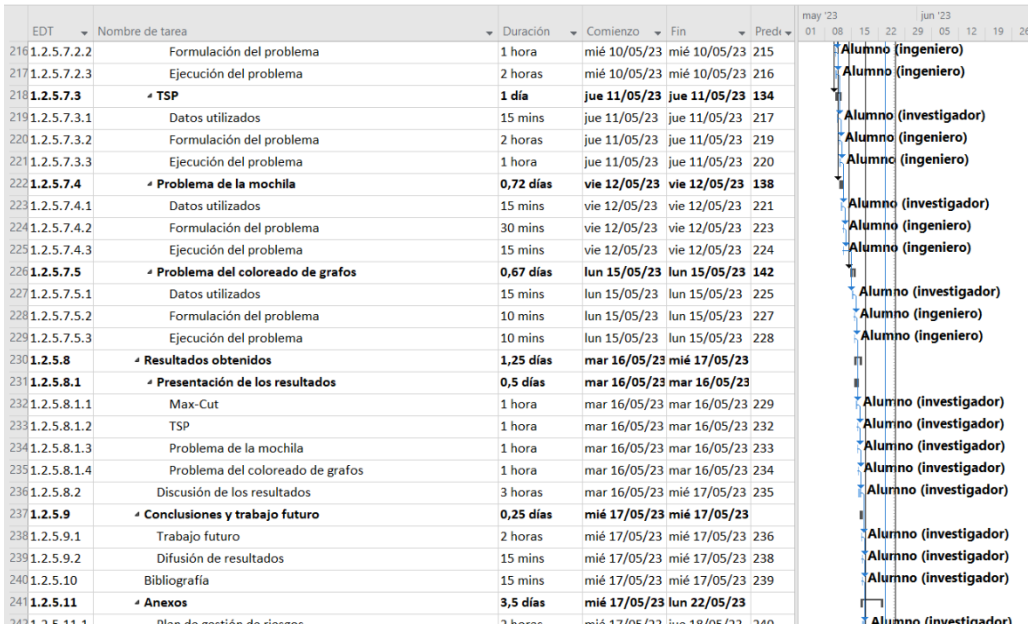
	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
183	1.2.5.4.1.6	Presupuesto de cliente	2 horas	mié 29/03/23	mié 29/03/23	182
184	**1.2.5.4.2**	**Ejecución del proyecto**	**1,69 días**	**lun 22/05/23**	**mié 24/05/23**	
185	1.2.5.4.2.1	Plan de seguimiento de planificación	5 horas	lun 22/05/23	mar 23/05/23	244
186	1.2.5.4.2.2	Bitácora de incidencias del proyecto	1 hora	mar 23/05/23	mar 23/05/23	185
187	1.2.5.4.2.3	Riesgos	3,5 horas	mar 23/05/23	mié 24/05/23	186
188	**1.2.5.4.3**	**Cierre del proyecto**	**1 día**	**mié 24/05/23**	**jue 25/05/23**	
189	1.2.5.4.3.1	Planificación final	1 hora	mié 24/05/23	mié 24/05/23	187
190	1.2.5.4.3.2	Informe final de riesgos	1 hora	mié 24/05/23	mié 24/05/23	189
191	1.2.5.4.3.3	Presupuesto final de costes	3 horas	mié 24/05/23	jue 25/05/23	190
192	1.2.5.4.3.4	Informe de lecciones aprendidas	3 horas	jue 25/05/23	jue 25/05/23	191

Figura 3.15. Actividades (4)



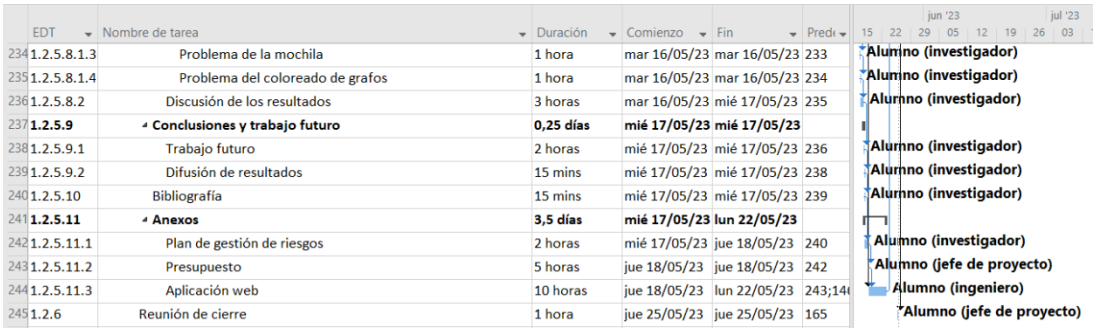
	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
192	1.2.5.4.3.4	Informe de lecciones aprendidas	3 horas	jue 25/05/23	jue 25/05/23	191
193	1.2.5.5	Estado del arte	20 horas	mar 25/04/23	lun 01/05/23	171
194	**1.2.5.6**	**Descripción del sistema**	**6,63 días**	**lun 01/05/23**	**mié 10/05/23**	
195	**1.2.5.6.1**	**Formulaciones**	**0,75 días**	**lun 01/05/23**	**mar 02/05/23**	
196	1.2.5.6.1.1	Framework QUBO	1 hora	lun 01/05/23	lun 01/05/23	193
197	1.2.5.6.1.2	Modelo de Ising	2 horas	mar 02/05/23	mar 02/05/23	196
198	**1.2.5.6.2**	**Problemas**	**2,25 días**	**mar 02/05/23**	**jue 04/05/23**	
199	1.2.5.6.2.1	Max-Cut	2 horas	mar 02/05/23	mar 02/05/23	197
200	1.2.5.6.2.2	Programación lineal binaria	3 horas	mar 02/05/23	mié 03/05/23	199
201	1.2.5.6.2.3	TSP	3 horas	mié 03/05/23	mié 03/05/23	200
202	1.2.5.6.2.4	Problema de la mochila	2 horas	jue 04/05/23	jue 04/05/23	201
203	1.2.5.6.2.5	Problema del coloreado de grafos	3 horas	jue 04/05/23	jue 04/05/23	202
204	**1.2.5.6.3**	**Métodos**	**3,13 días**	**vie 05/05/23**	**mié 10/05/23**	
205	**1.2.5.6.3.1**	**Annealing cuántico**	**0,5 días**	**vie 05/05/23**	**vie 05/05/23**	
206	1.2.5.6.3.1	Computación adiabática cuántica	3 horas	vie 05/05/23	vie 05/05/23	203
207	1.2.5.6.3.1	Annealing cuántico	2 horas	vie 05/05/23	vie 05/05/23	206
208	**1.2.5.6.3.2**	**QAOA**	**2,13 días**	**lun 08/05/23**	**mié 10/05/23**	
209	1.2.5.6.3.2	Definición	2 horas	lun 08/05/23	lun 08/05/23	207
210	1.2.5.6.3.2	Algoritmo	3 horas	lun 08/05/23	lun 08/05/23	209
211	1.2.5.6.3.2	Paso a circuitos	6 horas	mar 09/05/23	mié 10/05/23	210
212	**1.2.5.7**	**Metodología de trabajo**	**3,54 días**	**mié 10/05/23**	**lun 15/05/23**	
213	1.2.5.7.1	Explicación de métodos a utilizar	2 horas	mié 10/05/23	mié 10/05/23	211
214	**1.2.5.7.2**	**Max-Cut**	**0,63 días**	**mié 10/05/23**	**mié 10/05/23**	130
215	1.2.5.7.2.1	Datos utilizados	15 mins	mié 10/05/23	mié 10/05/23	213
216	1.2.5.7.2.2	Formulación del problema	1 hora	mié 10/05/23	mié 10/05/23	215
217	1.2.5.7.2.3	Ejecución del problema	2 horas	mié 10/05/23	mié 10/05/23	216

Figura 3.16. Actividades (5)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
216	1.2.5.7.2.2	Formulación del problema	1 hora	mié 10/05/23	mié 10/05/23	215
217	1.2.5.7.2.3	Ejecución del problema	2 horas	mié 10/05/23	mié 10/05/23	216
218	**1.2.5.7.3**	**TSP**	**1 día**	**jue 11/05/23**	**jue 11/05/23**	**134**
219	1.2.5.7.3.1	Datos utilizados	15 mins	jue 11/05/23	jue 11/05/23	217
220	1.2.5.7.3.2	Formulación del problema	2 horas	jue 11/05/23	jue 11/05/23	219
221	1.2.5.7.3.3	Ejecución del problema	1 hora	jue 11/05/23	jue 11/05/23	220
222	**1.2.5.7.4**	**Problema de la mochila**	**0,72 días**	**vie 12/05/23**	**vie 12/05/23**	**138**
223	1.2.5.7.4.1	Datos utilizados	15 mins	vie 12/05/23	vie 12/05/23	221
224	1.2.5.7.4.2	Formulación del problema	30 mins	vie 12/05/23	vie 12/05/23	223
225	1.2.5.7.4.3	Ejecución del problema	15 mins	vie 12/05/23	vie 12/05/23	224
226	**1.2.5.7.5**	**Problema del coloreado de grafos**	**0,67 días**	**lun 15/05/23**	**lun 15/05/23**	**142**
227	1.2.5.7.5.1	Datos utilizados	15 mins	lun 15/05/23	lun 15/05/23	225
228	1.2.5.7.5.2	Formulación del problema	10 mins	lun 15/05/23	lun 15/05/23	227
229	1.2.5.7.5.3	Ejecución del problema	10 mins	lun 15/05/23	lun 15/05/23	228
230	**1.2.5.8**	**Resultados obtenidos**	**1,25 días**	**mar 16/05/23**	**mié 17/05/23**	
231	**1.2.5.8.1**	**Presentación de los resultados**	**0,5 días**	**mar 16/05/23**	**mar 16/05/23**	
232	1.2.5.8.1.1	Max-Cut	1 hora	mar 16/05/23	mar 16/05/23	229
233	1.2.5.8.1.2	TSP	1 hora	mar 16/05/23	mar 16/05/23	232
234	1.2.5.8.1.3	Problema de la mochila	1 hora	mar 16/05/23	mar 16/05/23	233
235	1.2.5.8.1.4	Problema del coloreado de grafos	1 hora	mar 16/05/23	mar 16/05/23	234
236	1.2.5.8.2	Discusión de los resultados	3 horas	mar 16/05/23	mié 17/05/23	235
237	**1.2.5.9**	**Conclusiones y trabajo futuro**	**0,25 días**	**mié 17/05/23**	**mié 17/05/23**	
238	1.2.5.9.1	Trabajo futuro	2 horas	mié 17/05/23	mié 17/05/23	236
239	1.2.5.9.2	Difusión de resultados	15 mins	mié 17/05/23	mié 17/05/23	238
240	1.2.5.10	Bibliografía	15 mins	mié 17/05/23	mié 17/05/23	239
241	**1.2.5.11**	**Anexos**	**3,5 días**	**mié 17/05/23**	**lun 22/05/23**	

Figura 3.17. Actividades (6)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
234	1.2.5.8.1.3	Problema de la mochila	1 hora	mar 16/05/23	mar 16/05/23	233
235	1.2.5.8.1.4	Problema del coloreado de grafos	1 hora	mar 16/05/23	mar 16/05/23	234
236	1.2.5.8.2	Discusión de los resultados	3 horas	mar 16/05/23	mié 17/05/23	235
237	**1.2.5.9**	**Conclusiones y trabajo futuro**	**0,25 días**	**mié 17/05/23**	**mié 17/05/23**	
238	1.2.5.9.1	Trabajo futuro	2 horas	mié 17/05/23	mié 17/05/23	236
239	1.2.5.9.2	Difusión de resultados	15 mins	mié 17/05/23	mié 17/05/23	238
240	1.2.5.10	Bibliografía	15 mins	mié 17/05/23	mié 17/05/23	239
241	**1.2.5.11**	**Anexos**	**3,5 días**	**mié 17/05/23**	**lun 22/05/23**	
242	1.2.5.11.1	Plan de gestión de riesgos	2 horas	mié 17/05/23	jue 18/05/23	240
243	1.2.5.11.2	Presupuesto	5 horas	jue 18/05/23	jue 18/05/23	242
244	1.2.5.11.3	Aplicación web	10 horas	jue 18/05/23	lun 22/05/23	243;14
245	1.2.6	Reunión de cierre	1 hora	jue 25/05/23	jue 25/05/23	165

Figura 3.18. Actividades (7)

3.1.4. Riesgos

A continuación se van a presentar una serie de riesgos que pueden llegar a tener que afrontarse en la realización del proyecto y que pueden comprometer varios factores del mismo. También se expone el plan de gestión de riesgos, el cual detalla el protocolo que seguir para su manejo y las acciones a tomar para controlarlos.

3.1.4.1. Plan de gestión de riesgos

El plan de gestión de riesgos puede consultarse en el Anexo I. Plan de gestión de riesgos

3.1.4.2. Identificación de riesgos

En el transcurso del proyecto hay una serie de riesgos potenciales a tener en cuenta, los cuales tenemos que identificar de antemano para poder tener una estimación del impacto que estos puedan tener y planificar una respuesta.

A continuación, se listan los riesgos, ordenados por su posible impacto total, donde cada uno tiene una tabla en la que se indica cuánto puede afectar el riesgo en determinados aspectos del proyecto, descripción, una estrategia y la respuesta propuesta.

Cada riesgo puede pertenecer a una de estas categorías: técnico, externo, organizacional, o de gestión de proyecto.

Tiempo de espera para colas de ejecución de ordenadores cuánticos reales demasiado largo

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Externo	Muy alta	Medio	Crítico	Alto	Alto	0,81

Descripción: muchas de las herramientas disponibles para ejecutar tareas en ordenadores cuánticos reales tienen una cola de espera, debido a la escasez de dispositivos de este tipo, y a su dificultad para construirlos, lo que genera una demanda que acaba sobrepasando la capacidad computacional del mismo, por lo que, al enviar un programa a estos ordenadores, hay que esperar a que terminen los que había previamente.

La longitud de la cola depende de varios factores, muchos de los cuales no están en manos de la persona que quiera utilizar la máquina, lo cual puede dificultar mucho su uso o incluso la viabilidad del mismo, en caso de que tengan que hacerse muchas ejecuciones en el dispositivo, por ejemplo, para llevar a cabo un estudio.

Estrategia: mitigar el riesgo en la medida de lo posible.

Respuesta: frente a este problema, una forma de reducir el impacto que pueda tener en el proyecto podría ser buscar el ordenador que menor tiempo de espera debido a cola tenga. En algunas ocasiones se puede consultar el estado de los ordenadores, lo cual incluye la disponibilidad de los mismos.

Sumado a esto, también puede ayudar hacer pruebas localmente, por ejemplo en un simulador, antes de enviar el problema al ordenador de verdad, reduciendo así el número de ejecuciones necesarias.

Cabe mencionar que, a pesar de llevar a cabo estas optimizaciones, aun así no es seguro que las ejecuciones se realicen en un tiempo aceptable. Puede darse el caso de que el mejor ordenador cuántico disponible siga teniendo una cola de un periodo de tiempo demasiado largo.

Limitaciones del hardware cuántico disponible

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Externo	Alta	Bajo	Medio	Crítico	Alto	0,63

Descripción: actualmente, los ordenadores cuánticos de los que disponemos, e incluso los simuladores de los mismos, pueden tener dificultades a la hora de resolver tareas a partir de cierta carga. Esto puede tener como resultado que no sea posible aumentar la complejidad de los problemas que se resuelven con estos dispositivos, debido a que se necesite una cantidad de recursos mayor de la disponible.

Estrategia: asumir el riesgo.

Respuesta: uno de los objetivos de este estudio es el de observar en qué punto se encuentran actualmente los ordenadores cuánticos, por lo que no se busca que los métodos probados tengan unas capacidades mínimas, si no ver hasta dónde llegan esas capacidades, aunque eso pueda significar que no sea posible hacer pruebas con problemas muy complejos.

Este riesgo también puede afectar a la exhaustividad con la que se estudian algunos problemas, ya que, dependiendo de la complejidad del mismo, es posible que no se puedan hacer pruebas con un rango tan variado de valores como en otros problemas de menor dificultad.

Curva de aprendizaje asociada al tema

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Técnico	Alta	Medio	Alto	Medio	Alto	0,39

Descripción: previo al inicio del proyecto, el estudiante no tiene apenas conocimiento sobre el tema tratado en el trabajo debido a que, a pesar de que algunos aspectos relacionados con la materia, como por ejemplo el álgebra lineal, se imparten en asignaturas de la titulación, gran parte de los conceptos todavía no forman parte del plan educativo.

Esto implica que tiene que pasar por un periodo de aprendizaje para familiarizarse con estos conceptos, los cuales son necesarios para llevar a cabo el estudio.

Dicho proceso lleva un riesgo inherente asociado, ya que la falta de soltura en el tema se traduce en posibles contratiempos o malentendidos, lo cual perjudica al desarrollo óptimo del trabajo en cuanto a planificación e incluso calidad, comparado a si se hubiera hecho sobre un tema del que el alumno contase con un mayor conocimiento de manera previa.

Estrategia: mitigar el riesgo.

Respuesta: el estudiante hará uso de bibliografía proporcionada por el tutor con el objetivo de facilitar la introducción al tema y contar con una fuente de información donde los conceptos consten de manera unificada para un mejor entendimiento.

Falta de tiempo

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Organiz.	Alta	Medio	Alto	Medio	Medio	0,39

Descripción: a la hora de planificar las actividades del proyecto, se intenta asignar a cada tarea una duración razonable en la que podría ser completada.

Esto está sujeto a posibles imprevistos, los cuales podrían hacer que la duración estimada de dicha tarea se vea comprometida y acabe siendo mucho mayor de lo esperado, sobre todo en un proyecto de esta índole en el que cuenta con tantas incógnitas y factores externos al control del estudiante.

Hay muchas variables que son difíciles de estimar de manera previa y sobre las que se tiene poco conocimiento de antemano sobre cuánto pueden durar, lo cual puede afectar en un grado considerable a la planificación del proyecto, junto a otros factores.

Estrategia: mitigar el riesgo.

Respuesta: para prevenir en la medida de lo posible que el número de horas de una actividad exceda las horas planificadas, en estas tareas en las que haya componentes que puedan ser problemáticos se va a asignar una cantidad más generosa de tiempo, la cual cubra también posibles imprevistos que puedan ocurrir.

Límite de tiempo de uso en hardware cuántico real

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Externo	Media	Medio	Medio	Alto	Medio	0,28

Descripción: algunos dispositivos que se ponen a disposición de los usuarios para ejecutar tareas en ellos establecen un límite de tiempo de computación que se puede consumir a lo largo de un periodo de tiempo. Un ejemplo de esto es la empresa D-Wave, que ofrece la posibilidad de utilizar annealers a sus usuarios registrados, pero tiene un límite mensual de 1 minuto de tiempo computacional.

Es importante tener en cuenta este tipo de restricciones, ya que en un experimento en el que se tengan que realizar muchos cálculos, este tiempo se puede agotar, y eso obligaría a esperar a que se renueve, o a utilizar una versión de pago del servicio, que puede repercutir en el presupuesto.

Estrategia: mitigar el riesgo.

Respuesta: es importante sacarle el máximo provecho a todo el tiempo del que se dispone para utilizar estos dispositivos.

Para minimizar el consumo, debería esperar a utilizarse el dispositivo para el experimento hasta asegurarse de que la preparación de las instancias a estudiar es correcta, ya que si no podría derivar en un experimento fallido, y tendría que repetirse, lo cual resultaría muy costoso.

Falta de información sobre el uso de las tecnologías

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Externo	Media	Medio	Medio	Alto	Medio	0,28

Descripción: al tratarse de una disciplina tan reciente, las herramientas de las que se dispone para trabajar en ella también están en una fase de crecimiento. Esto conlleva a que todavía no sea posible encontrar mucha información sobre cómo se pueden utilizar.

Hay algunos casos, como por ejemplo Qiskit, en el que existe una gran cantidad de documentación en la que se detalla el uso de la librería, pero incluso teniendo toda esa información pueden surgir problemas, errores, o dudas al utilizar dichas tecnologías cuya solución no se puede encontrar tan fácilmente como otras tecnologías más establecidas, de las que se goza de una gran cantidad de información desarrollada durante mucho más tiempo.

Estrategia: asumir el riesgo.

Respuesta: las herramientas elegidas para este estudio están bastante extendidas y cuentan con una comunidad relativamente grande de personas que las utilizan, además de tener una documentación en gran parte actualizada y completa.

En caso de tener alguna incidencia con estas herramientas, la información disponible debería ser suficiente para dar con una solución sin invertir demasiado tiempo.

Disponibilidad de ordenadores cuánticos reales

Categoría	*Probabilidad*	*Presupuesto*	*Planificación*	*Alcance*	*Calidad*	*Impacto*
Externo	Baja	Bajo	Medio	Alto	Alto	0,17

Descripción: hoy en día construir un ordenador cuántico no es una tarea sencilla. Esto puede dar lugar a que el número de ordenadores disponibles para utilizar sea limitado, y encontrar uno que poder utilizar se vuelva aparatoso.

Estrategia: asumir el riesgo.

Respuesta: es cierto que este riesgo podría ocasionar que el desarrollo del proyecto no fuese tan fluido, pero es una de las consecuencias de trabajar en un campo que se encuentra en una fase todavía inicial.

La poca disponibilidad de estos dispositivos se compensa porque el número de usuarios que buscan utilizar estas máquinas también es reducido, por lo que no debería convertirse en un problema grave dar con un ordenador que utilizar.

3.1.5. Presupuesto inicial del TFG

A continuación se muestran tanto el presupuesto de costes como el presupuesto de cliente. El procedimiento utilizado para realizarlos está explicado en el Anexo II: Presupuesto.

3.1.5.1. Presupuesto de costes

A continuación se muestra el presupuesto de costes agregado, para el que se han tenido en cuenta las tareas de la planificación, su duración y los recursos asociados, cuyo coste se ha calculado en la definición de la empresa que se presenta en el Anexo II. Presupuesto.

También cabe mencionar que el beneficio que se ha buscado obtener es de un 25%, y que para el cálculo de la ponderación se ha excluido el balance de la partida de "Otros gastos".

Presupuesto de costes		
Cod.	**Partida**	**Total**
01	Reuniones y formación	3.971,00 €
02	Estudio	385,00 €
03	Desarrollo	1.855,00 €
04	Documentación	9.571,92 €
05	Otros gastos	60,00 €

TOTAL	15.842,92 €

Tabla 1. Presupuesto de costes inicial

3.1.5.2. Presupuesto de cliente

Para el presupuesto de cliente, tenemos que considerar el beneficio que queremos conseguir, que en este caso será de un 25%, y también tendremos que calcular un factor de ponderación sin tener en cuenta la partida de "Otros gastos", para saber cuánto incrementar el valor de las demás partidas.

También se muestran con mayor detalle los contenidos de las partidas.

Otros gastos	60,00 €
Beneficios (25%)	3.945,73 €
Factor de ponderación	25,10%

TOTAL A COMPENSAR	4.005,73 €

Tabla 2. Beneficios y factor de ponderación inicial

Este es el presupuesto de cliente resultante antes de aplicar el factor de ponderación

Presupuesto de cliente (pre beneficios y ajuste)				
Partida	**Item**	**Detalles partida**	**Importe**	**Total**
01		**Reuniones y formación**		**3.971,00 €**
	001	Reunión de arranque	43,00 €	
	002	Formación en computación cuántica	3.885,00 €	
	003	Reunión de cierre	43,00 €	
02		**Estudio**		**385,00 €**
	001	Estudio del Max-Cut	105,00 €	
	002	Estudio del TSP	105,00 €	
	003	Estudio del problema de la mochila	70,00 €	
	004	Estudio del problema del coloreado de grafos	105,00 €	
03		**Desarrollo**		**1.855,00 €**
	001	Formación en Flask	175,00 €	

	002	Implementación página principal	105,00 €	
	003	Página Max-Cut	630,00 €	
	004	Página TSP	315,00 €	
	005	Página del problema de la mochila	280,00 €	
	006	Página del problema de coloreado de grafos	350,00 €	
04		**Documentación**		**9.571,92 €**
	001	Abstract	35,00 €	
	002	Introducción	1.225,00 €	
	003	Fijación de objetivos	175,00 €	
	004	Planificación y gestión del TFG	4.996,50 €	
	005	Estado del arte	700,00 €	
	006	Descripción del sistema	1.120,00 €	
	007	Metodología de trabajo	352,92 €	
	008	Resultados obtenidos	245,00 €	
	009	Conclusiones y trabajo futuro	78,75 €	
	010	Bibliografía	8,75 €	
	011	Anexos	635,00 €	

TOTAL	15.782,92 €

Tabla 3. Presupuesto de cliente inicial antes de aplicar factor de ponderación

Al aplicar el factor de ponderación, el presupuesto de cliente sería

Presupuesto de cliente (detallado)				
Partida	**Item**	**Detalles partida**	**Importe**	**Total**
01		**Reuniones y formación**		**4.967,54 €**
	001	Reunión de arranque	53,79 €	
	002	Formación en computación cuántica	4.859,96 €	
	003	Reunión de cierre	53,79 €	
02		**Estudio**		**481,62 €**
	001	Estudio del Max-Cut	131,35 €	
	002	Estudio del TSP	131,35 €	
	003	Estudio del problema de la mochila	87,57 €	
	004	Estudio del problema del coloreado de grafos	131,35 €	
03		**Desarrollo**		**2.320,52 €**
	001	Formación en Flask	218,92 €	
	002	Implementación página principal	131,35 €	
	003	Página Max-Cut	788,10 €	
	004	Página TSP	394,05 €	
	005	Página del problema de la mochila	350,27 €	
	006	Página del problema de coloreado de grafos	437,83 €	
04		**Documentación**		**11.974,03 €**

001	Abstract	43,78 €
002	Introducción	1.532,42 €
003	Fijación de objetivos	218,92 €
004	Planificación y gestión del TFG	6.250,39 €
005	Estado del arte	875,67 €
006	Descripción del sistema	1.401,07 €
007	Metodología de trabajo	441,48 €
008	Resultados obtenidos	306,48 €
009	Conclusiones y trabajo futuro	98,51 €
010	Bibliografía	10,95 €
011	Anexos	794,36 €

TOTAL	19.743,70 €

Tabla 4. Presupuesto de cliente inicial teniendo en cuenta factor de ponderación

Y la versión resumida

Presupuesto de cliente (resumido)		
Cod.	**Partida**	**Total**
01	**Reuniones y formación**	**4.967,54 €**
02	**Estudio**	**481,62 €**
03	**Desarrollo**	**2.320,52 €**
04	**Documentación**	**11.974,03 €**

TOTAL	**19.743,70 €**

Tabla 5. Presupuesto de cliente inicial resumido

3.2. Ejecución del proyecto

3.2.1. Plan de seguimiento de planificación

A continuación se describen 4 líneas base que se han establecido a lo largo del desarrollo del proyecto

1. **Línea base inicial:** esta señala el final del periodo de formación, a partir del cual, teniendo ya los conocimientos necesarios, se puede empezar a elaborar el estudio y la aplicación web. Está fechada el 14/03/2023.

2. **Línea base intermedia del estudio:** el objetivo de esta línea es marcar el final de una de las partes principales del trabajo, el estudio, que es un punto clave en el que evaluar el estado del proyecto. Su fecha es el 13/04/2023.

3. **Línea base intermedia de la aplicación:** esta línea base indica que se ha completado el desarrollo de la aplicación web. Fechada para el 20/04/2023.

4. **Línea base final:** habiendo terminado el estudio y la aplicación web, la gran parte del trabajo restante es completar la documentación, punto en el cual se habrá alcanzado esta línea base, que indica el final del proyecto. Marcada para el día 13/10/23.

3.2.2. Bitácora de incidencias del proyecto

En la realización del proyecto, es posible que surjan impedimentos inesperados que pueden comprometer el desarrollo fluido de las actividades el proyecto, de acuerdo a la planificación inicial.

A continuación, se presentan los problemas de este tipo que han surgido en la realización del proyecto.

- *4 de marzo de 2023*, dificultades para cumplir con el horario impuesto en la planificación inicial.

  Los horarios de trabajo establecidos inicialmente no pudieron cumplirse en su totalidad debido a la carga de otros trabajos o quehaceres a los que se vio sometido el alumno. Esto se empezó a ver algo después de la fecha de comienzo del proyecto y su impacto en la planificación fue recurrente.

- *30 de marzo de 2023*, colas de espera para ordenadores cuánticos reales demasiado larga.

  Desde el principio del trabajo había una intención de incluir ordenadores cuánticos reales en el estudio debido a que sería un recurso muy valioso para analizar y completar los resultados.

  A la hora de intentar utilizar estos ordenadores para resolver problemas utilizando el QAOA nos encontramos con que para cada ejecución hay una cola de espera, que juntada al tiempo que tarda en resolverse la instancia del problema y el número de datos que se quieren obtener, resultan en una cantidad de tiempo exagerada, haciendo esta posibilidad poco viable.

- *31 de marzo de 2023*, la resolución de los grafos tarda más de lo esperado.

  Procesar tantos grafos es una tarea computacionalmente muy exigente, y esto se refleja en el tiempo que hay que esperar para que acabe. Esto, sumado a que si hay un error en los datos obtenidos o en la ejecución esta tiene que repetirse, puede repercutir en la cantidad de tiempo necesaria para conseguir datos para el estudio.

- *5 de abril de 2023*, dificultades a la hora de pasar los conocimientos teóricos a la programación.

Incluso después del periodo de formación, en el momento de empezar a programar y utilizar los conceptos aprendidos, la falta de experiencia en el tema resultó en mucha prueba y error hasta que se consiguió trabajar con las ideas nuevas de manera satisfactoria.

La falta de práctica tuvo un impacto mucho mayor del que se esperaba, causando algo de retraso en la planificación esperada.

- *7 de abril de 2023*, limitaciones del hardware cuántico disponible.

Al querer resolver problemas más complejos que el Max-Cut, como el TSP, se observó que algunos métodos empezaban a tener complicaciones para resolver problemas con tamaños incluso algo más reducidos.

Esto limitó el número de datos que se podían obtener, y dio lugar a una discrepancia en el número de nodos a probar bastante notable entre el Max-Cut y otros problemas.

3.2.3. Riesgos

Previamente se han identificado una serie de riesgos que potencialmente podrían afectar al desarrollo del proyecto, pero además de identificarlos también es importante realizar un seguimiento de los mismos para evaluar su impacto.

Tiempo de espera para colas de ejecución de ordenadores cuánticos reales demasiado largo

El día 30 de marzo de 2023, que fue cuando se comenzaron a resolver los grafos del Max-Cut utilizando distintos métodos, se pudo comprobar que las colas que había para utilizar los ordenadores cuánticos reales ofrecidos por IBM eran muy largas, y comparado con los otros métodos no se podían sacar datos de manera tan fluida, por lo que se descartó el método de utilizar el algoritmo QAOA en ordenadores cuánticos reales.

Limitaciones del hardware cuántico disponible

Al principio del estudio, haciendo las pruebas con el problema del corte máximo, fue posible obtener datos para un número considerable de nodos, lo cual resultó en unos gráficos notablemente completos.

Por desgracia, al probar con un problema más complejo (en torno al 9 de abril de 2023) y que hacía uso de más qubits, no pudieron probarse con tantos tamaños ni nodos, por lo que los resultados fueron algo menos elaborados.

Curva de aprendizaje asociada al tema

El alumno era consciente de la falta de conocimiento que tenía sobre la temática del trabajo en el inicio. Por ello, la primera fase del proyecto buscaba dotar al alumno de los conocimientos, soltura, y herramientas necesarias para afrontarlo.

Esta fase demostraría no ser suficiente para evitar problemas en el futuro, ya que, debido a la complejidad del tema y a la continua exposición de conceptos y herramientas nuevas, la necesidad de formación y revisión de los conocimientos adquiridos acabarían estando presentes durante toda la realización del proyecto.

Las dificultades asociadas a ello hicieron aparición desde el primer momento, en el periodo de formación, pero al estar éste dedicado precisamente para resolver este tipo de problemas, no se consideraron una incidencia.

A la hora de programar problemas más complejos, alrededor del mes de abril, las dificultades se acentuarían, y continuarían apareciendo hasta el final del proyecto, aunque cada vez en menor medida.

Dificultades para la creación de modelos representando problemas en QUBO

Las librerías con las que se pensaba trabajar en un principio no permitían modelar los problemas con la libertad que se pretendía, lo cual imposibilitó utilizar las formulaciones de los problemas a través de ellas.

Hubo que buscar una alternativa que utilizar para hacer estos modelos, lo cual tuvo un impacto en la planificación, ya que la cantidad de información disponible de este tema en Internet no es muy extensa.

Falta de tiempo

El manejo del tiempo es uno de los aspectos más importantes en un proyecto, y siempre hay que saber mantener las expectativas del trabajo que se puede hacer en un periodo de tiempo a un nivel razonable.

En este caso, se sobreestimó la cantidad de trabajo que podía hacerse de manera diaria, lo cual tuvo una repercusión importante en varios aspectos de la planificación (véase la incidencia del 4 de marzo).

Otro factor que redujo el tiempo disponible es la resolución de los grafos, cuya duración total era muy difícil de saber de antemano, y en algunos casos sobrepasó el tiempo asignado.

3.3. Cierre del proyecto

3.3.1. Planificación final

La planificación, debido a las incidencias mencionadas previamente, ha sufrido algunos cambios. La duración total del proyecto pasó de 75 días a 194, y el día de fin del 25 de mayo al 8 de noviembre. A continuación se muestra la planificación modificada, que abarca los hitos y las actividades.

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
1	**1**	**Diseño e implementación de algoritmos cuánticos para problemas de optimización combinatoria**	**194 días**	**vie 10/02/23**	**mié 08/11/23**	**No**	
2	**1.1**	**Hitos**	**193,5 días**	**vie 10/02/23**	**mié 08/11/23**	**No**	
3	1.1.1	Reunión de arranque	0 días	vie 10/02/23	vie 10/02/23	Sí	113
4	**1.1.2**	**Formación en computación cuántica**	**24,13 días**	**mié 01/03/23**	**mar 04/04/23**	**No**	
5	1.1.2.1	Formación en las bases de la computación cuántica	0 días	mié 01/03/23	mié 01/03/23	Sí	115
6	**1.1.2.2**	**Formación en optimización cuántica**	**21,25 días**	**lun 06/03/23**	**mar 04/04/23**	**No**	
7	**1.1.2.2.1**	**Formación en formulaciones y problemas**	**9,13 días**	**lun 06/03/23**	**vie 17/03/23**	**No**	
8	1.1.2.2.1.1	Formación en el framework QUBO	0 días	lun 06/03/23	lun 06/03/23	Sí	118
9	1.1.2.2.1.2	Formación en el modelo de Ising	0 días	mar 07/03/23	mar 07/03/23	Sí	119
10	1.1.2.2.1.3	Formación en paso de Ising a QUBO	0 días	mié 08/03/23	mié 08/03/23	Sí	120
11	1.1.2.2.1.4	Formación en representación de problemas de optimización combinatoria	0 días	vie 17/03/23	vie 17/03/23	Sí	121
12	**1.1.2.2.2**	**Formación en annealing cuántico**	**3,75 días**	**mar 21/03/23**	**lun 27/03/23**	**No**	
13	1.1.2.2.2.1	Formación en computación adiabática	0 días	mar 21/03/23	mar 21/03/23	Sí	123
14	1.1.2.2.2.2	Formación en annealing cuántico	0 días	jue 23/03/23	jue 23/03/23	Sí	124
15	1.1.2.2.2.3	Formación en uso de annealers de D-Wave	0 días	lun 27/03/23	lun 27/03/23	Sí	125
16	**1.1.2.2.3**	**Formación en QAOA**	**5 días**	**mar 28/03/23**	**mar 04/04/23**	**No**	
17	1.1.2.2.3.1	Formación en paso de annealing cuántico a QAOA	0 días	mar 28/03/23	mar 28/03/23	Sí	127
18	1.1.2.2.3.2	Formación en resolución de problemas utilizando el QAOA	0 días	mar 04/04/23	mar 04/04/23	Sí	128
19	**1.1.3**	**Realización del estudio**	**16 días**	**lun 24/04/23**	**mar 16/05/23**	**No**	
20	1.1.3.1	Estudio del Max-Cut	0 días	lun 24/04/23	lun 24/04/23	Sí	130
21	1.1.3.2	Estudio del TSP	0 días	mié 03/05/23	mié 03/05/23	Sí	134

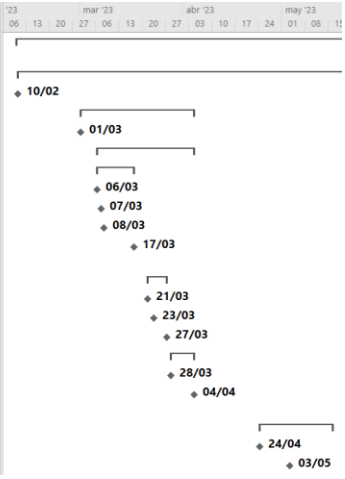


Figura 3.19. Hitos de la planificación final (1)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
19	**1.1.3**	**Realización del estudio**	**16 días**	**lun 24/04/23**	**mar 16/05/23**	**No**	
20	1.1.3.1	Estudio del Max-Cut	0 días	lun 24/04/23	lun 24/04/23	Sí	130
21	1.1.3.2	Estudio del TSP	0 días	mié 03/05/23	mié 03/05/23	Sí	134
22	1.1.3.3	Estudio del problema de la mochila	0 días	mar 09/05/23	mar 09/05/23	Sí	138
23	1.1.3.4	Estudio del problema del coloreado de grafos	0 días	mar 16/05/23	mar 16/05/23	Sí	142
24	**1.1.4**	**Desarrollo de aplicación web**	**24 días**	**mar 18/04/23**	**lun 22/05/23**	**No**	
25	1.1.4.1	Formación en Flask	0 días	mar 18/04/23	mar 18/04/23	Sí	147
26	1.1.4.2	Implementación página principal	0 días	mié 19/04/23	mié 19/04/23	Sí	148
27	1.1.4.3	Página Max-Cut	0 días	mar 02/05/23	mar 02/05/23	Sí	149
28	1.1.4.4	Página TSP	0 días	mar 09/05/23	mar 09/05/23	Sí	153
29	1.1.4.5	Página del problema de la mochila	0 días	lun 15/05/23	lun 15/05/23	Sí	157
30	1.1.4.6	Página del problema del coloreado de grafos	0 días	lun 22/05/23	lun 22/05/23	Sí	161

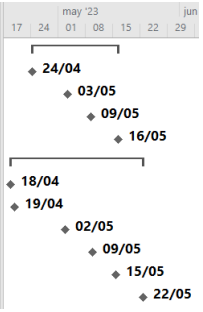


Figura 3.20. Hitos de la planificación final (2)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
31	**1.1.5**	**Documentación del proyecto**	**156 días**	**mar 04/04/23**	**mié 08/11/23**	**No**	
32	1.1.5.1	Abstract	0 días	vie 01/09/23	vie 01/09/23	Sí	166
33	**1.1.5.2**	**Introducción**	**10,13 días**	**lun 04/09/23**	**lun 18/09/23**	**No**	
34	1.1.5.2.1	Motivación	0 días	lun 04/09/23	lun 04/09/23	Sí	168
35	1.1.5.2.2	Finalidad del proyecto	0 días	lun 04/09/23	lun 04/09/23	Sí	169
36	1.1.5.2.3	Marco teórico	0 días	lun 18/09/23	lun 18/09/23	Sí	170
37	1.1.5.3	Fijación de objetivos	0 días	mié 20/09/23	mié 20/09/23	Sí	171

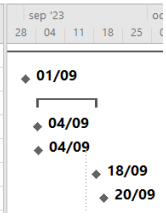


Figura 3.21. Hitos de la planificación final (3)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
38	**1.1.5.4**	**Planificación y gestión del TFG**	**156 días**	**mar 04/04/23**	**mié 08/11/23**	**No**	
39	**1.1.5.4.1**	**Planificación del proyecto**	**7,38 días**	**mar 04/04/23**	**jue 13/04/23**	**No**	
40	1.1.5.4.1.1	Identificación de interesados	0 días	mar 04/04/23	mar 04/04/23	Sí	174
41	1.1.5.4.1.2	OBS	0 días	mar 04/04/23	mar 04/04/23	Sí	175
42	1.1.5.4.1.3	PBS	0 días	jue 06/04/23	jue 06/04/23	Sí	176
43	1.1.5.4.1.4	WBS	0 días	mar 11/04/23	mar 11/04/23	Sí	177
44	**1.1.5.4.1.5**	**Riesgos**	**0,88 días**	**mar 11/04/23**	**mié 12/04/23**	**No**	
45	1.1.5.4.1.5	Plan de gestión de riesgos	0 días	mar 11/04/23	mar 11/04/23	Sí	179
46	1.1.5.4.1.5	Identificación de riesgos	0 días	mié 12/04/23	mié 12/04/23	Sí	180
47	**1.1.5.4.1.6**	**Presupuesto inicial del TFG**	**0,25 días**	**jue 13/04/23**	**jue 13/04/23**	**No**	
48	1.1.5.4.1.6	Presupuesto de costes	0 días	jue 13/04/23	jue 13/04/23	Sí	182
49	1.1.5.4.1.6	Presupuesto de cliente	0 días	jue 13/04/23	jue 13/04/23	Sí	183

Figura 3.22. Hitos de la planificación final (4)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
50	**1.1.5.4.2**	**Ejecución del proyecto**	**0,75 días**	**lun 06/11/23**	**lun 06/11/23**	**No**	
51	1.1.5.4.2.1	Plan de seguimiento de planificación	0 días	lun 06/11/23	lun 06/11/23	Sí	185
52	1.1.5.4.2.2	Bitácora de incidencias del proyecto	0 días	lun 06/11/23	lun 06/11/23	Sí	186
53	1.1.5.4.2.3	Riesgos	0 días	lun 06/11/23	lun 06/11/23	Sí	187
54	**1.1.5.4.3**	**Cierre del proyecto**	**1,38 días**	**mar 07/11/23**	**mié 08/11/23**	**No**	
55	1.1.5.4.3.1	Planificación final	0 días	mar 07/11/23	mar 07/11/23	Sí	189
56	1.1.5.4.3.2	Informe final de riesgos	0 días	mar 07/11/23	mar 07/11/23	Sí	190
57	1.1.5.4.3.3	Presupuesto final de costes	0 días	mié 08/11/23	mié 08/11/23	Sí	191
58	1.1.5.4.3.4	Informe de lecciones aprendidas	0 días	mié 08/11/23	mié 08/11/23	Sí	192
59	1.1.5.5	Estado del arte	0 días	vie 29/09/23	vie 29/09/23	Sí	193
60	**1.1.5.6**	**Descripción del sistema**	**10,13 días**	**vie 29/09/23**	**vie 13/10/23**	**No**	
61	**1.1.5.6.1**	**Formulaciones**	**0,88 días**	**vie 29/09/23**	**lun 02/10/23**	**No**	
62	1.1.5.6.1.1	Framework QUBO	0 días	vie 29/09/23	vie 29/09/23	Sí	196
63	1.1.5.6.1.2	Modelo de Ising	0 días	lun 02/10/23	lun 02/10/23	Sí	197
64	**1.1.5.6.2**	**Problemas**	**3,88 días**	**lun 02/10/23**	**vie 06/10/23**	**No**	
65	1.1.5.6.2.1	Max-Cut	0 días	lun 02/10/23	lun 02/10/23	Sí	199
66	1.1.5.6.2.2	Programación lineal binaria	0 días	mar 03/10/23	mar 03/10/23	Sí	200
67	1.1.5.6.2.3	TSP	0 días	mié 04/10/23	mié 04/10/23	Sí	201
68	1.1.5.6.2.4	Problema de la mochila	0 días	jue 05/10/23	jue 05/10/23	Sí	202
69	1.1.5.6.2.5	Problema del coloreado de grafos	0 días	vie 06/10/23	vie 06/10/23	Sí	203
70	**1.1.5.6.3**	**Métodos**	**4,13 días**	**lun 09/10/23**	**vie 13/10/23**	**No**	
71	**1.1.5.6.3.1**	**Annealing cuántico**	**0,88 días**	**lun 09/10/23**	**mar 10/10/23**	**No**	
72	1.1.5.6.3.1	Computación adiabática cuántica	0 días	lun 09/10/23	lun 09/10/23	Sí	206

Figura 3.23. Hitos de la planificación final (5)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Prede
71	**1.1.5.6.3.1**	**Annealing cuántico**	**0,88 días**	**lun 09/10/23**	**mar 10/10/23**	**No**	
72	1.1.5.6.3.1	Computación adiabática cuántica	0 días	lun 09/10/23	lun 09/10/23	Sí	206
73	1.1.5.6.3.1	Annealing cuántico	0 días	mar 10/10/23	mar 10/10/23	Sí	207
74	**1.1.5.6.3.2**	**QAOA**	**3 días**	**mar 10/10/23**	**vie 13/10/23**	**No**	
75	1.1.5.6.3.2	Definición	0 días	mar 10/10/23	mar 10/10/23	Sí	209
76	1.1.5.6.3.2	Algoritmo	0 días	mié 11/10/23	mié 11/10/23	Sí	210
77	1.1.5.6.3.2	Paso a circuitos	0 días	vie 13/10/23	vie 13/10/23	Sí	211
78	**1.1.5.7**	**Metodología de trabajo**	**5,42 días**	**lun 16/10/23**	**lun 23/10/23**	**No**	
79	1.1.5.7.1	Explicación de métodos a utilizar	0 días	lun 16/10/23	lun 16/10/23	Sí	213
80	**1.1.5.7.2**	**Max-Cut**	**1,47 días**	**lun 16/10/23**	**mar 17/10/23**	**No**	
81	1.1.5.7.2.1	Datos utilizados	0 días	lun 16/10/23	lun 16/10/23	Sí	215
82	1.1.5.7.2.2	Formulación del problema	0 días	lun 16/10/23	lun 16/10/23	Sí	216
83	1.1.5.7.2.3	Ejecución del problema	0 días	mar 17/10/23	mar 17/10/23	Sí	217
84	**1.1.5.7.3**	**TSP**	**1,72 días**	**mié 18/10/23**	**jue 19/10/23**	**No**	
85	1.1.5.7.3.1	Datos utilizados	0 días	mié 18/10/23	mié 18/10/23	Sí	219
86	1.1.5.7.3.2	Formulación del problema	0 días	mié 18/10/23	mié 18/10/23	Sí	220
87	1.1.5.7.3.3	Ejecución del problema	0 días	jue 19/10/23	jue 19/10/23	Sí	221
88	**1.1.5.7.4**	**Problema de la mochila**	**0,69 días**	**vie 20/10/23**	**vie 20/10/23**	**No**	
89	1.1.5.7.4.1	Datos utilizados	0 días	vie 20/10/23	vie 20/10/23	Sí	223
90	1.1.5.7.4.2	Formulación del problema	0 días	vie 20/10/23	vie 20/10/23	Sí	224
91	1.1.5.7.4.3	Ejecución del problema	0 días	vie 20/10/23	vie 20/10/23	Sí	225
92	**1.1.5.7.5**	**Problema del coloreado de grafos**	**0,64 días**	**lun 23/10/23**	**lun 23/10/23**	**No**	
93	1.1.5.7.5.1	Datos utilizados	0 días	lun 23/10/23	lun 23/10/23	Sí	227

Figura 3.24. Hitos de la planificación final (6)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Hito	Predec
92	**1.1.5.7.5**	**Problema del coloreado de grafos**	**0,64 días**	**lun 23/10/23**	**lun 23/10/23**	**No**	
93	1.1.5.7.5.1	Datos utilizados	0 días	lun 23/10/23	lun 23/10/23	Sí	227
94	1.1.5.7.5.2	Formulación del problema	0 días	lun 23/10/23	lun 23/10/23	Sí	228
95	1.1.5.7.5.3	Ejecución del problema	0 días	lun 23/10/23	lun 23/10/23	Sí	229
96	**1.1.5.8**	**Resultados obtenidos**	**2 días**	**mar 24/10/23**	**jue 26/10/23**	**No**	
97	**1.1.5.8.1**	**Presentación de los resultados**	**1 día**	**mar 24/10/23**	**mié 25/10/23**	**No**	
98	1.1.5.8.1.1	Max-Cut	0 días	mar 24/10/23	mar 24/10/23	Sí	232
99	1.1.5.8.1.2	TSP	0 días	mar 24/10/23	mar 24/10/23	Sí	233
100	1.1.5.8.1.3	Problema de la mochila	0 días	mar 24/10/23	mar 24/10/23	Sí	234
101	1.1.5.8.1.4	Problema del coloreado de grafos	0 días	mié 25/10/23	mié 25/10/23	Sí	235
102	1.1.5.8.2	Discusión de los resultados	0 días	jue 26/10/23	jue 26/10/23	Sí	236
103	**1.1.5.9**	**Conclusiones y trabajo futuro**	**0,66 días**	**jue 26/10/23**	**vie 27/10/23**	No	
104	1.1.5.9.1	Trabajo futuro	0 días	jue 26/10/23	jue 26/10/23	Sí	238
105	1.1.5.9.2	Difusión de resultados	0 días	vie 27/10/23	vie 27/10/23	Sí	239
106	1.1.5.10	Bibliografía	0 días	vie 27/10/23	vie 27/10/23	Sí	240
107	**1.1.5.11**	**Anexos**	**5,56 días**	**vie 27/10/23**	**vie 03/11/23**	**No**	
108	1.1.5.11.1	Plan de gestión de riesgos	0 días	vie 27/10/23	vie 27/10/23	Sí	242
109	1.1.5.11.2	Presupuesto	0 días	vie 27/10/23	vie 27/10/23	Sí	243
110	1.1.5.11.3	Aplicación web	0 días	vie 03/11/23	vie 03/11/23	Sí	244
111	1.1.6	Reunión de cierre	0 días	mié 08/11/23	mié 08/11/23	Sí	245

Figura 3.25. Hitos de la planificación final (7)

Y las actividades

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
112	**1.2**	**Actividades**	**194 días**	**vie 10/02/23**	**mié 08/11/23**	
113	1.2.1	Reunión de arranque	1 hora	vie 10/02/23	vie 10/02/23	
114	**1.2.2**	**Formación en computación cuántica**	**37,25 días**	**vie 10/02/23**	**mar 04/04/23**	
115	1.2.2.1	Formación en las bases de la computación cuántica	40 horas	vie 10/02/23	mié 01/03/23	
116	**1.2.2.2**	**Formación en optimización cuántica**	**24,13 días**	**mié 01/03/23**	**mar 04/04/23**	
117	**1.2.2.2.1**	**Formación en formulaciones y problemas**	**12 días**	**mié 01/03/23**	**vie 17/03/23**	
118	1.2.2.2.1.1	Formación en el framework QUBO	6 horas	mié 01/03/23	lun 06/03/23	115
119	1.2.2.2.1.2	Formación en el modelo de Ising	5 horas	lun 06/03/23	mar 07/03/23	118
120	1.2.2.2.1.3	Formación en paso de Ising a QUBO	3 horas	mié 08/03/23	mié 08/03/23	119
121	1.2.2.2.1.4	Formación en representación de problemas de optimizac	20 horas	jue 09/03/23	vie 17/03/23	120
122	**1.2.2.2.2**	**Formación en annealing cuántico**	**5,88 días**	**vie 17/03/23**	**lun 27/03/23**	
123	1.2.2.2.2.1	Formación en computación adiabática	7 horas	vie 17/03/23	mar 21/03/23	121
124	1.2.2.2.2.2	Formación en annealing cuántico	5 horas	mié 22/03/23	jue 23/03/23	123
125	1.2.2.2.2.3	Formación en uso de annealers de D-Wave	5 horas	jue 23/03/23	lun 27/03/23	124
126	**1.2.2.2.3**	**Formación en QAOA**	**6,25 días**	**lun 27/03/23**	**mar 04/04/23**	
127	1.2.2.2.3.1	Formación en paso de annealing cuántico a QAOA	5 horas	lun 27/03/23	mar 28/03/23	125
128	1.2.2.2.3.2	Formación en resolución de problemas utilizando el QAO	15 horas	mié 29/03/23	mar 04/04/23	127
129	**1.2.3**	**Realización del estudio**	**22,13 días**	**vie 14/04/23**	**mar 16/05/23**	**114**
130	**1.2.3.1**	**Estudio del Max-Cut**	**6,13 días**	**vie 14/04/23**	**lun 24/04/23**	
131	1.2.3.1.1	Generación de grafos	2 horas	vie 14/04/23	vie 14/04/23	173
132	1.2.3.1.2	Resolución de los grafos	45 horas	vie 14/04/23	vie 21/04/23	131
133	1.2.3.1.3	Procesamiento de los resultados	1 hora	lun 24/04/23	lun 24/04/23	132
134	**1.2.3.2**	**Estudio del TSP**	**7 días**	**lun 24/04/23**	**mié 03/05/23**	**130**

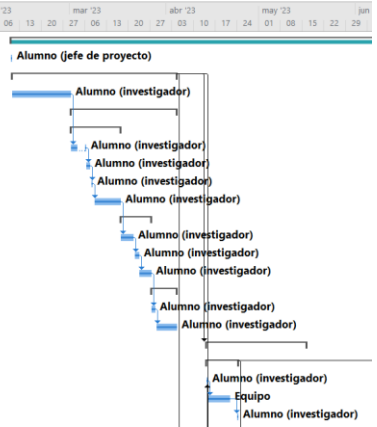


Figura 3.26. Actividades de la planificación final (1)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
133	1.2.3.1.3	Procesamiento de los resultados	1 hora	lun 24/04/23	lun 24/04/23	132
134	**1.2.3.2**	**Estudio del TSP**	**7 días**	**lun 24/04/23**	**mié 03/05/23**	**130**
135	1.2.3.2.1	Generación de grafos	10 horas	lun 24/04/23	jue 27/04/23	
136	1.2.3.2.2	Resolución de los grafos	25 horas	jue 27/04/23	mar 02/05/23	135
137	1.2.3.2.3	Procesamiento de los resultados	1 hora	mié 03/05/23	mié 03/05/23	136
138	**1.2.3.3**	**Estudio del problema de la mochila**	**4 días**	**mié 03/05/23**	**mar 09/05/23**	**134**
139	1.2.3.3.1	Generación de grafos	1 hora	mié 03/05/23	mié 03/05/23	137
140	1.2.3.3.2	Resolución de los grafos	30 horas	mié 03/05/23	lun 08/05/23	139
141	1.2.3.3.3	Procesamiento de los resultados	1 hora	mar 09/05/23	mar 09/05/23	140
142	**1.2.3.4**	**Estudio del problema del coloreado de grafos**	**5 días**	**mar 09/05/23**	**mar 16/05/23**	**138**
143	1.2.3.4.1	Generación de grafos	2 horas	mar 09/05/23	mar 09/05/23	141
144	1.2.3.4.2	Resolución de los grafos	35 horas	mar 09/05/23	lun 15/05/23	143
145	1.2.3.4.3	Procesamiento de los resultados	1 hora	mar 16/05/23	mar 16/05/23	144
146	**1.2.4**	**Desarrollo de aplicación web**	**26,13 días**	**vie 14/04/23**	**lun 22/05/23**	**173**
147	1.2.4.1	Formación en Flask	5 horas	vie 14/04/23	mar 18/04/23	114
148	1.2.4.2	Implementación página principal	3 horas	mar 18/04/23	mié 19/04/23	147
149	**1.2.4.3**	**Página Max-Cut**	**8,25 días**	**jue 20/04/23**	**mar 02/05/23**	
150	1.2.4.3.1	Implementación de las distintas formas de resolver los prob	10 horas	jue 20/04/23	mié 26/04/23	148
151	1.2.4.3.2	Parametrización de los problemas	5 horas	jue 27/04/23	lun 01/05/23	150
152	1.2.4.3.3	Carga de problema a través de fichero	3 horas	lun 01/05/23	mar 02/05/23	151
153	**1.2.4.4**	**Página TSP**	**4,13 días**	**mié 03/05/23**	**mar 09/05/23**	**149**
154	1.2.4.4.1	Implementación de las distintas formas de resolver los prob	1 hora	mié 03/05/23	mié 03/05/23	
155	1.2.4.4.2	Parametrización de los problemas	5 horas	mié 03/05/23	vie 05/05/23	154

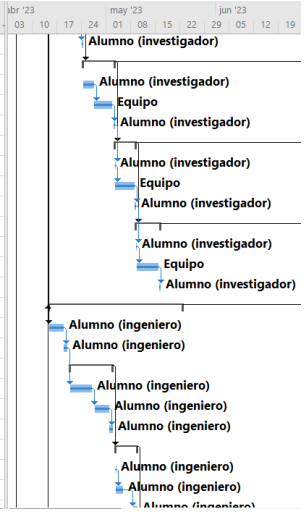


Figura 3.27. Actividades de la planificación final (2)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
154	1.2.4.4.1	Implementación de las distintas formas de resolver los prob	1 hora	mié 03/05/23	mié 03/05/23	
155	1.2.4.4.2	Parametrización de los problemas	5 horas	mié 03/05/23	vie 05/05/23	154
156	1.2.4.4.3	Carga de problema a través de fichero	3 horas	lun 08/05/23	mar 09/05/23	155
157	**1.2.4.5**	**Página del problema de la mochila**	**4 días**	**mar 09/05/23**	**lun 15/05/23**	**153**
158	1.2.4.5.1	Implementación de las distintas formas de resolver los prob	1 hora	mar 09/05/23	mar 09/05/23	156
159	1.2.4.5.2	Parametrización de los problemas	3 horas	mié 10/05/23	jue 11/05/23	158
160	1.2.4.5.3	Carga de problema a través de fichero	4 horas	jue 11/05/23	lun 15/05/23	159
161	**1.2.4.6**	**Página del problema del coloreado de grafos**	**5 días**	**lun 15/05/23**	**lun 22/05/23**	**157**
162	1.2.4.6.1	Implementación de las distintas formas de resolver los prob	1 hora	lun 15/05/23	lun 15/05/23	160
163	1.2.4.6.2	Parametrización de los problemas	5 horas	mar 16/05/23	jue 18/05/23	162
164	1.2.4.6.3	Carga de problema a través de fichero	4 horas	jue 18/05/23	lun 22/05/23	163

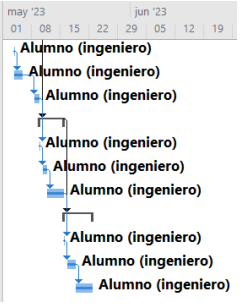


Figura 3.28. Actividades de la planificación final (3)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
165	**1.2.5**	**Documentación del proyecto**	**156,13 días**	**mar 04/04/23**	**mié 08/11/23**	
166	1.2.5.1	Abstract	1 hora	vie 01/09/23	vie 01/09/23	
167	**1.2.5.2**	**Introducción**	**11,25 días**	**vie 01/09/23**	**lun 18/09/23**	
168	1.2.5.2.1	Motivación	4 horas	vie 01/09/23	lun 04/09/23	166
169	1.2.5.2.2	Finalidad del proyecto	1 hora	lun 04/09/23	lun 04/09/23	168
170	1.2.5.2.3	Marco teórico	30 horas	mar 05/09/23	lun 18/09/23	169
171	1.2.5.3	Fijación de objetivos	5 horas	mar 19/09/23	mié 20/09/23	170

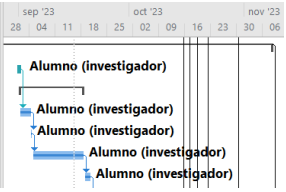


Figura 3.29. Actividades de la planificación final (4)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
172	**1.2.5.4**	**Planificación y gestión del TFG**	**156,13 días**	**mar 04/04/23**	**mié 08/11/23**	
173	**1.2.5.4.1**	**Planificación del proyecto**	**7,5 días**	**mar 04/04/23**	**jue 13/04/23**	
174	1.2.5.4.1.1	Identificación de interesados	1 hora	mar 04/04/23	mar 04/04/23	114
175	1.2.5.4.1.2	OBS	2 horas	mar 04/04/23	mar 04/04/23	174
176	1.2.5.4.1.3	PBS	6 horas	mar 04/04/23	jue 06/04/23	175
177	1.2.5.4.1.4	WBS	20 horas	jue 06/04/23	mar 11/04/23	176
178	**1.2.5.4.1.5**	**Riesgos**	**1,13 días**	**mar 11/04/23**	**mié 12/04/23**	
179	1.2.5.4.1.5	Plan de gestión de riesgos	2 horas	mar 11/04/23	mar 11/04/23	177
180	1.2.5.4.1.5	Identificación de riesgos	8 horas	mar 11/04/23	mié 12/04/23	179
181	**1.2.5.4.1.6**	**Presupuesto inicial del TFG**	**1 día**	**mié 12/04/23**	**jue 13/04/23**	
182	1.2.5.4.1.6	Presupuesto de costes	6 horas	mié 12/04/23	jue 13/04/23	180
183	1.2.5.4.1.6	Presupuesto de cliente	2 horas	jue 13/04/23	jue 13/04/23	182

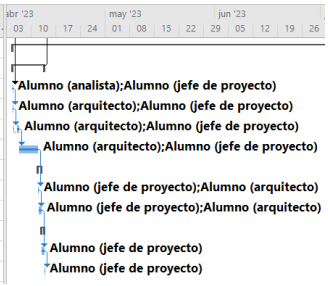


Figura 3.30. Actividades de la planificación final (5)

	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
184	**1.2.5.4.2**	**Ejecución del proyecto**	**1,13 días**	**vie 03/11/23**	**lun 06/11/23**	
185	1.2.5.4.2.1	Plan de seguimiento de planificación	4 horas	vie 03/11/23	lun 06/11/23	244
186	1.2.5.4.2.2	Bitácora de incidencias del proyecto	2 horas	lun 06/11/23	lun 06/11/23	185
187	1.2.5.4.2.3	Riesgos	2 horas	lun 06/11/23	lun 06/11/23	186
188	**1.2.5.4.3**	**Cierre del proyecto**	**1,5 días**	**lun 06/11/23**	**mié 08/11/23**	
189	1.2.5.4.3.1	Planificación final	2 horas	lun 06/11/23	mar 07/11/23	187
190	1.2.5.4.3.2	Informe final de riesgos	2 horas	mar 07/11/23	mar 07/11/23	189
191	1.2.5.4.3.3	Presupuesto final de costes	3 horas	mar 07/11/23	mié 08/11/23	190
192	1.2.5.4.3.4	Informe de lecciones aprendidas	3 horas	mié 08/11/23	mié 08/11/23	191
193	1.2.5.5	Estado del arte	20 horas	mié 20/09/23	vie 29/09/23	171
194	**1.2.5.6**	**Descripción del sistema**	**10,25 días**	**vie 29/09/23**	**vie 13/10/23**	
195	**1.2.5.6.1**	**Formulaciones**	**1 día**	**vie 29/09/23**	**lun 02/10/23**	
196	1.2.5.6.1.1	Framework QUBO	1 hora	vie 29/09/23	vie 29/09/23	193
197	1.2.5.6.1.2	Modelo de Ising	2 horas	vie 29/09/23	lun 02/10/23	196
198	**1.2.5.6.2**	**Problemas**	**4,13 días**	**lun 02/10/23**	**vie 06/10/23**	
199	1.2.5.6.2.1	Max-Cut	2 horas	lun 02/10/23	lun 02/10/23	197
200	1.2.5.6.2.2	Programación lineal binaria	3 horas	mar 03/10/23	mar 03/10/23	199
201	1.2.5.6.2.3	TSP	3 horas	mié 04/10/23	mié 04/10/23	200
202	1.2.5.6.2.4	Problema de la mochila	2 horas	jue 05/10/23	jue 05/10/23	201
203	1.2.5.6.2.5	Problema del coloreado de grafos	3 horas	jue 05/10/23	vie 06/10/23	202
204	**1.2.5.6.3**	**Métodos**	**5,13 días**	**vie 06/10/23**	**vie 13/10/23**	
205	**1.2.5.6.3.1**	**Annealing cuántico**	**1,88 días**	**vie 06/10/23**	**mar 10/10/23**	
206	1.2.5.6.3.1	Computación adiabática cuántica	3 horas	vie 06/10/23	lun 09/10/23	203

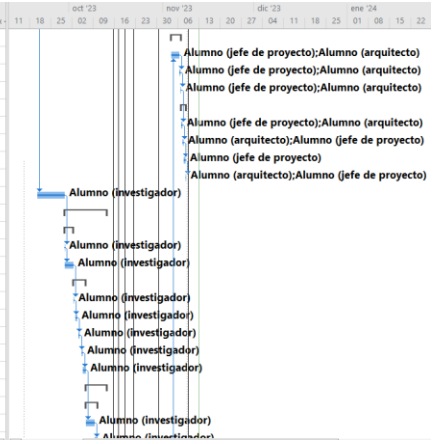
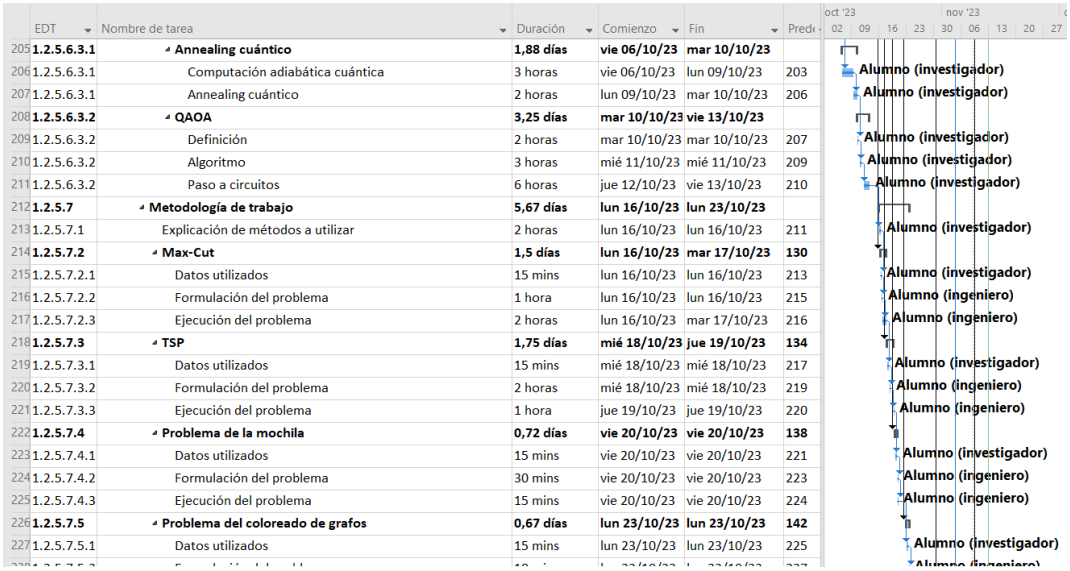
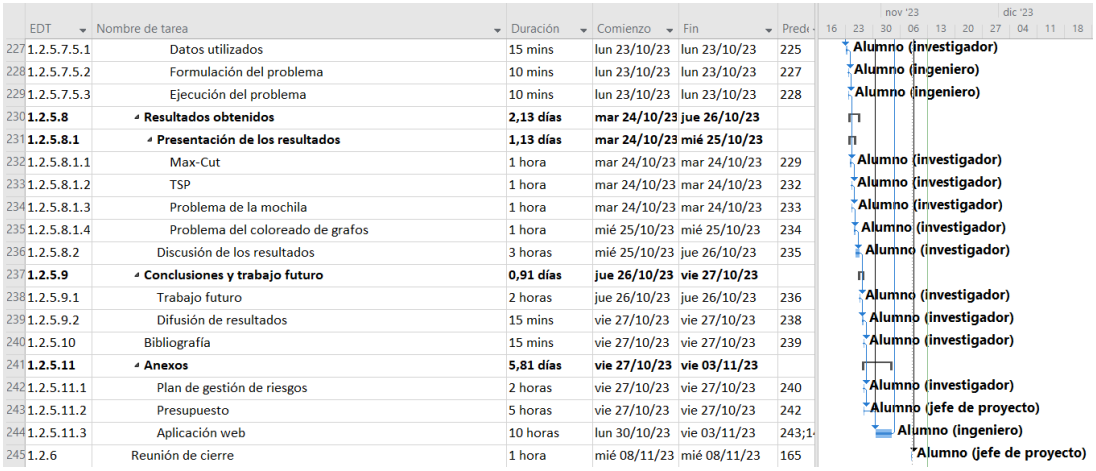


Figura 3.31. Actividades de la planificación final (6)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
205	**1.2.5.6.3.1**	**Annealing cuántico**	**1,88 días**	**vie 06/10/23**	**mar 10/10/23**	
206	1.2.5.6.3.1	Computación adiabática cuántica	3 horas	vie 06/10/23	lun 09/10/23	203
207	1.2.5.6.3.1	Annealing cuántico	2 horas	lun 09/10/23	mar 10/10/23	206
208	**1.2.5.6.3.2**	**QAOA**	**3,25 días**	**mar 10/10/23**	**vie 13/10/23**	
209	1.2.5.6.3.2	Definición	2 horas	mar 10/10/23	mar 10/10/23	207
210	1.2.5.6.3.2	Algoritmo	3 horas	mié 11/10/23	mié 11/10/23	209
211	1.2.5.6.3.2	Paso a circuitos	6 horas	jue 12/10/23	vie 13/10/23	210
212	**1.2.5.7**	**Metodología de trabajo**	**5,67 días**	**lun 16/10/23**	**lun 23/10/23**	
213	1.2.5.7.1	Explicación de métodos a utilizar	2 horas	lun 16/10/23	lun 16/10/23	211
214	**1.2.5.7.2**	**Max-Cut**	**1,5 días**	**lun 16/10/23**	**mar 17/10/23**	**130**
215	1.2.5.7.2.1	Datos utilizados	15 mins	lun 16/10/23	lun 16/10/23	213
216	1.2.5.7.2.2	Formulación del problema	1 hora	lun 16/10/23	lun 16/10/23	215
217	1.2.5.7.2.3	Ejecución del problema	2 horas	lun 16/10/23	mar 17/10/23	216
218	**1.2.5.7.3**	**TSP**	**1,75 días**	**mié 18/10/23**	**jue 19/10/23**	**134**
219	1.2.5.7.3.1	Datos utilizados	15 mins	mié 18/10/23	mié 18/10/23	217
220	1.2.5.7.3.2	Formulación del problema	2 horas	mié 18/10/23	mié 18/10/23	219
221	1.2.5.7.3.3	Ejecución del problema	1 hora	jue 19/10/23	jue 19/10/23	220
222	**1.2.5.7.4**	**Problema de la mochila**	**0,72 días**	**vie 20/10/23**	**vie 20/10/23**	**138**
223	1.2.5.7.4.1	Datos utilizados	15 mins	vie 20/10/23	vie 20/10/23	221
224	1.2.5.7.4.2	Formulación del problema	30 mins	vie 20/10/23	vie 20/10/23	223
225	1.2.5.7.4.3	Ejecución del problema	15 mins	vie 20/10/23	vie 20/10/23	224
226	**1.2.5.7.5**	**Problema del coloreado de grafos**	**0,67 días**	**lun 23/10/23**	**lun 23/10/23**	**142**
227	1.2.5.7.5.1	Datos utilizados	15 mins	lun 23/10/23	lun 23/10/23	225

Figura 3.32. Actividades de la planificación final (7)



	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predec
227	1.2.5.7.5.1	Datos utilizados	15 mins	lun 23/10/23	lun 23/10/23	225
228	1.2.5.7.5.2	Formulación del problema	10 mins	lun 23/10/23	lun 23/10/23	227
229	1.2.5.7.5.3	Ejecución del problema	10 mins	lun 23/10/23	lun 23/10/23	228
230	**1.2.5.8**	**Resultados obtenidos**	**2,13 días**	**mar 24/10/23**	**jue 26/10/23**	
231	**1.2.5.8.1**	**Presentación de los resultados**	**1,13 días**	**mar 24/10/23**	**mié 25/10/23**	
232	1.2.5.8.1.1	Max-Cut	1 hora	mar 24/10/23	mar 24/10/23	229
233	1.2.5.8.1.2	TSP	1 hora	mar 24/10/23	mar 24/10/23	232
234	1.2.5.8.1.3	Problema de la mochila	1 hora	mar 24/10/23	mar 24/10/23	233
235	1.2.5.8.1.4	Problema del coloreado de grafos	1 hora	mié 25/10/23	mié 25/10/23	234
236	1.2.5.8.2	Discusión de los resultados	3 horas	mié 25/10/23	jue 26/10/23	235
237	**1.2.5.9**	**Conclusiones y trabajo futuro**	**0,91 días**	**jue 26/10/23**	**vie 27/10/23**	
238	1.2.5.9.1	Trabajo futuro	2 horas	jue 26/10/23	jue 26/10/23	236
239	1.2.5.9.2	Difusión de resultados	15 mins	vie 27/10/23	vie 27/10/23	238
240	1.2.5.10	Bibliografía	15 mins	vie 27/10/23	vie 27/10/23	239
241	**1.2.5.11**	**Anexos**	**5,81 días**	**vie 27/10/23**	**vie 03/11/23**	
242	1.2.5.11.1	Plan de gestión de riesgos	2 horas	vie 27/10/23	vie 27/10/23	240
243	1.2.5.11.2	Presupuesto	5 horas	vie 27/10/23	vie 27/10/23	242
244	1.2.5.11.3	Aplicación web	10 horas	lun 30/10/23	vie 03/11/23	243;1
245	1.2.6	Reunión de cierre	1 hora	mié 08/11/23	mié 08/11/23	165

Figura 3.33. Actividades de la planificación final (8)

3.3.2. Informe final de riesgos

A continuación se presenta el impacto total que han tenido los riesgos más importantes, que se han monitorizado durante el proyecto.

Riesgo: *Tiempo de espera para colas de ejecución de ordenadores cuánticos reales demasiado largo*

Impacto: debido a la extensa cola de espera asociada a estos ordenadores, finalmente no se pudieron incluir en el estudio todos los que en un principio se quería.

El annealer no presentó este problema, pero los ordenadores cuánticos reales ofrecidos por IBM, en los que se buscaba ejecutar el algoritmo QAOA (ordenadores cuánticos reales de propósito general, basados en circuitos), tenían asociada una cola de espera demasiado larga.

El número de ejecuciones que se tiene que hacer para este estudio es demasiado grande para esperar en cada una de ellas por la cola de ejecución, haciendo esta posibilidad inviable.

Riesgo: Limitaciones del hardware cuántico disponible

Impacto: los ordenadores cuánticos actuales todavía sufren limitaciones por el número y la calidad de los qubits con los que cuentan.

Incluso a la hora de hacer simulaciones, el número máximo de qubits con el que se puede trabajar está solamente en torno a los 30. Estas restricciones acabaron repercutiendo en la cantidad de nodos que se pudieron probar en problemas más complejos.

Riesgo: Curva de aprendizaje asociada al tema

Impacto: la falta de soltura del alumno en el tema de la computación cuántica fue un factor reiterado que perjudicó en numerosas ocasiones la planificación esperada.

Se subestimó el tiempo necesario para aprender a trabajar con los conceptos nuevos y lidiar con contratiempos asociados a la falta de práctica de programar utilizando las librerías nuevas.

Riesgo: Dificultades para la creación de modelos representando problemas en QUBO

Impacto: retraso en la planificación por la búsqueda de herramientas que utilizar que permitiesen construir el modelo de la manera deseada.

Riesgo: Falta de tiempo

Impacto: el alumno realizó la planificación con la idea de dedicarle más horas diarias de las que al final fue posible. Debido a otras tareas no siempre fue posible encontrar el tiempo necesario para avanzar en este trabajo de la manera esperada.

También redujo el tiempo disponible la espera por la resolución de los problemas, que finalmente acabó siendo mayor de lo planificado y causó más retraso.

3.3.3. Presupuesto final de costes

El presupuesto final, tras los cambios producidos en la planificación, es el siguiente:

Presupuesto de costes		
Cod.	**Partida**	**Total**
01	Reuniones y formación	3.971,00 €
02	Estudio	665,00 €
03	Desarrollo	1.855,00 €
04	Documentación	9.530,42 €
05	Otros gastos	60,00 €

TOTAL	16.081,42 €

Tabla 6. Presupuesto de costes final

3.3.4. Presupuesto final de cliente

El presupuesto de cliente también se ha visto afectado tras los cambios. A continuación se muestran los mismos contenidos que en el presupuesto de cliente inicial.

En primer lugar los beneficios, el factor de ponderación y la cantidad a compensar.

Otros gastos	60,00 €
Beneficios (25%)	4.005,35 €
Factor de ponderación	25,09%

TOTAL A COMPENSAR	4.065,35 €

Tabla 7. Beneficios y factor de ponderación final

El presupuesto de cliente antes de tener en cuenta estos factores

Presupuesto de cliente (pre beneficios y ajuste)				
Partida	**Item**	**Detalles partida**	**Importe**	**Total**
01		**Reuniones y formación**		**3.971,00 €**
	001	Reunión de arranque	43,00 €	
	002	Formación en computación cuántica	3.885,00 €	
	003	Reunión de cierre	43,00 €	
02		**Estudio**		**665,00 €**
	001	Estudio del Max-Cut	105,00 €	
	002	Estudio del TSP	385,00 €	
	003	Estudio del problema de la mochila	70,00 €	
	004	Estudio del problema del coloreado de grafos	105,00 €	
03		**Desarrollo**		**1.855,00 €**
	001	Formación en Flask	175,00 €	
	002	Implementación página principal	105,00 €	
	003	Página Max-Cut	630,00 €	
	004	Página TSP	315,00 €	
	005	Página del problema de la mochila	280,00 €	
	006	Página del problema de coloreado de grafos	350,00 €	
04		**Documentación**		**9.530,42 €**
	001	Abstract	35,00 €	
	002	Introducción	1.225,00 €	
	003	Fijación de objetivos	175,00 €	
	004	Planificación y gestión del TFG	4.955,00 €	
	005	Estado del arte	700,00 €	
	006	Descripción del sistema	1.120,00 €	
	007	Metodología de trabajo	352,92 €	
	008	Resultados obtenidos	245,00 €	
	009	Conclusiones y trabajo futuro	78,75 €	
	010	Bibliografía	8,75 €	
	011	Anexos	635,00 €	

TOTAL	16.021,42 €

Tabla 8. Presupuesto de cliente final antes de aplicar factor de ponderación

El presupuesto aplicando el factor

Presupuesto de cliente (detallado)				
Partida	**Item**	**Detalles partida**	**Importe**	**Total**
01		**Reuniones y formación**		**4.967,48 €**
	001	Reunión de arranque	53,79 €	
	002	Formación en computación cuántica	4.859,90 €	
	003	Reunión de cierre	53,79 €	

02		**Estudio**		**831,87 €**
	001	Estudio del Max-Cut	131,35 €	
	002	Estudio del TSP	481,61 €	
	003	Estudio del problema de la mochila	87,57 €	
	004	Estudio del problema del coloreado de grafos	131,35 €	
03		**Desarrollo**		**2.320,49 €**
	001	Formación en Flask	218,91 €	
	002	Implementación página principal	131,35 €	
	003	Página Max-Cut	788,09 €	
	004	Página TSP	394,05 €	
	005	Página del problema de la mochila	350,26 €	
	006	Página del problema de coloreado de grafos	437,83 €	
04		**Documentación**		**11.921,98 €**
	001	Abstract	43,78 €	
	002	Introducción	1.532,40 €	
	003	Fijación de objetivos	218,91 €	
	004	Planificación y gestión del TFG	6.198,41 €	
	005	Estado del arte	875,66 €	
	006	Descripción del sistema	1.401,05 €	
	007	Metodología de trabajo	441,48 €	
	008	Resultados obtenidos	306,48 €	
	009	Conclusiones y trabajo futuro	98,51 €	
	010	Bibliografía	10,95 €	
	011	Anexos	794,35 €	

TOTAL	20.041,83 €

Tabla 9. Presupuesto de cliente final teniendo en cuenta factor de ponderación

Y de forma resumida

Presupuesto de cliente (resumido)		
Cod.	**Partida**	**Total**
01	**Reuniones y formación**	**4.967,48 €**
02	**Estudio**	**831,87 €**
03	**Desarrollo**	**2.320,49 €**
04	**Documentación**	**11.921,98 €**

TOTAL	**20.041,83 €**

Tabla 10. Presupuesto de cliente final resumido

3.3.5. Informe de lecciones aprendidas

Durante la realización del proyecto se han extraído numerosas lecciones:

1. La planificación debe ser realizada de manera realista. Las expectativas de lo que se puede llegar a hacer en un periodo determinado de tiempo deben establecerse desde un punto más bien conservador, para evitar sobrecarga de trabajo y asegurarse de cumplir con los plazos con algo de margen.

2. No descuidar la importancia de la planificación al principio del proyecto. Realizar una buena planificación al comenzar el proyecto puede marcar una diferencia enorme en su evolución futura.

   A pesar de que, en ocasiones, se vuelva un esfuerzo difícil de hacer, y el equipo prefiera ponerse a comenzar con el proyecto directamente, definir desde el primer momento correctamente las tareas a realizar y el plazo para ellas puede resultar muy beneficioso y marcar la diferencia entre un producto satisfactorio y otro de menor calidad.

   Al hacer esto, se despejan muchas posibles dudas y contratiempos que puedan surgir durante el desarrollo, y permite un flujo de trabajo mucho más optimizado.

Capítulo 4

Estado del arte

En la literatura de la computación cuántica podemos encontrar algunos estudios parecidos al realizado en este trabajo, bien que traten con problemas similares o que evalúen diferentes aspectos de los algoritmos y formulaciones que hemos utilizado.

A continuación se ilustran varios ejemplos de dichos estudios. Cabe mencionar previamente que cada estudio se hace con unas condiciones distintas, las cuales pueden variar e influir en la calidad o el alcance del trabajo, como por ejemplo el tamaño del equipo de investigación, o las disciplinas en las que estén formados los miembros del mismo.

4.1. QAOA para resolver el Max-Cut a través de los años

El problema del corte máximo (Max-Cut), uno de los que se trata en este estudio, también ha sido objeto de investigación en otros artículos a lo largo de los años.

Más concretamente, entre el algoritmo QAOA (véase 5.3.2. QAOA: Quantum Approximate Optimization Algorithm) y dicho problema, la relación comienza desde el propio nacimiento del algoritmo, ya que se usa para estudiar su efectividad en el artículo que lo introdujo (escrito por Farhi et al.). En el trabajo se utilizó este problema con grafos regulares, es decir, donde cada vértice tiene el mismo número de conexiones (cabe mencionar que en el estudio de este TFG los grafos no son necesariamente regulares), y un valor de p fijo [6], que es un parámetro que se utiliza para determinar la profundidad a la que llega el algoritmo.

Los resultados indicaron un mejor ratio de aproximación que cualquier otro algoritmo clásico del momento que se ejecutara en tiempo polinomial (aunque un tiempo después surgió otro algoritmo que ofrecía mejores resultados que el QAOA) [1]. En su artículo, Farhi et al. consiguieron cortes que eran por lo menos 0.6924 veces el tamaño del corte máximo (esto utilizando $p = 1$ y grafos regulares de grado 3, es decir, la menor profundidad del algoritmo posible, y donde cada vértice tenía tres conexiones).

Otro estudio del QAOA sobre el problema del Max-Cut es el realizado por Crooks [7] en 2018, que combina el algoritmo con una optimización de los circuitos cuánticos realizada en un ordenador clásico.

Esto resultó en una reducción del tiempo de entrenamiento y también demostró tener mayor efectividad que el algoritmo clásico Goemans-Williamson (que tiene un ratio de aproximación de 0.8785, lo que quiere decir que en el peor de los casos la solución será el 87,5% del valor de la solución óptima) [8], a partir de $p = 8$. El estudio también

menciona la restricción del algoritmo, que está limitado, por el momento, a grafos de tamaño modesto, y también dice que tenemos que esperar a ordenadores cuánticos de mayores capacidades, tanto en número de qubits como tolerancia a fallos, para ver si se puede dar la supremacía cuántica o no (esto implicaría que un ordenador cuántico resolviera una tarea que un ordenador clásico no pudiese o le llevara demasiado tiempo).

Streif et al. demostraron en 2019 que el QAOA era una alternativa muy potente a otras opciones, como el annealing cuántico (véase 5.3.1.2. Annealing cuántico) o el annealing simulado (que, a diferencia del annealing cuántico, se puede ejecutar sobre hardware clásico, y utiliza fluctuaciones térmicas en lugar de cuánticas), puesto que fue capaz de encontrar soluciones a problemas que las otras técnicas no pudieron (a pesar de que el QAOA es una versión trotterizada del annealing, que utiliza variables discretas en lugar de continuas, véase 5.3.2.1. Definición) [9].

Este resultado nos sugiere que el algoritmo QAOA podría demostrar en un futuro la supremacía cuántica, dentro del contexto de la era de dispositivos NISQ [10].

Todo lo expuesto previamente nos hace ver la relevancia del algoritmo QAOA en la computación cuántica y la importancia que tiene seguir investigándolo y trabajando con él.

4.2. Annealing Cuántico

El artículo de Streif et al. nos sirve como transición para hablar de otro de los procedimientos que se ha utilizado en nuestro estudio, el annealing cuántico, ya que el artículo compara estos dos procedimientos resolviendo una serie de problemas.

Como se ha mencionado al final del punto anterior hay algunas instancias de problemas que el annealing tiene dificultades resolviendo. Esto es debido a que es posible que caiga en mínimos locales en lugar del mínimo global [9].

Existen también estudios realizados con los annealers proporcionados por la empresa D-Wave, los cuales se usan en este trabajo también, como el desarrollado por Villar-Rodriguez et al. en 2022 [11], en el que se hacen pruebas con instancias del problema del Travelling Salesman Problem (véase 5.2.3. Problema del viajante de comercio (Travelling Salesman Problem, TSP)). Se utilizan distintas formas de construir hamiltonianos (el uso de hamiltonianos se puede ver en 5.1.2.2. Definición: modelo de Ising) para representar el problema, y se comprueba el efecto de cambiar parámetros, como por ejemplo la fuerza de la cadena (cuánto de acoplados están los qubits adyacentes).

Otro ejemplo de estudio utilizando el annealer para resolver este tipo de problemas es el de Titiloye et al. [12], en el que el problema con el que se trabaja es el de

coloreado de grafos (véase 5.2.5. Problema del coloreado de grafos (Graph coloring)), y donde se demuestra que el annealing cuántico tiene un mejor rendimiento que el annealing simulado.

Capítulo 5

Descripción del sistema

Previo a la descripción de los pasos que se han seguido para realizar el experimento, primero tenemos que definir los conceptos que se van a utilizar para un mejor entendimiento del mismo.

Esto incluye hablar de las formulaciones que se van a usar para definir los problemas y poder trabajar con ellos, explicar los problemas en sí, y hablar de los métodos que se van a utilizar, así como otros conceptos que puedan resultar importantes.

5.1. Formulaciones

La formulación de los problemas es una de las partes más importantes a la hora de resolverlos, ya que tiene un impacto directo en el número de recursos (por ejemplo, qubits) que se van a necesitar, y en lo que va a tardar el algoritmo en encontrar una solución.

Es importante destacar que no hay una única formulación válida para cada problema, y que, actualmente, se están llevando a cabo muchas investigaciones para optimizar estas formulaciones y dar con soluciones mejores.

5.1.1. QUBO (Quadratic Unconstrained Binary Optimization)

La primera formulación de la que vamos a hablar es QUBO, que sirve para escribir problemas de optimización de forma que se pueda mapear en un entorno cuántico, sobre el que luego se pueden ejecutar algoritmos cuánticos y encontrar soluciones a los problemas.

Para hacer uso del framework QUBO, lo que buscamos es asignar **valores binarios** a variables que representen decisiones o elecciones, y luego definimos una **función objetivo** que refleje lo que queremos obtener con el sistema, la cual buscaremos minimizar o maximizar, dependiendo del problema en cuestión.

La formulación general de un problema QUBO es la siguiente:

$$Minimizar \;\; c_0x_0 + c_1x_1 + \ldots + c_jx_j + \sum_{i=0}^{n}\sum_{j=i+1}^{n} q_{ij}x_ix_j$$

$$sujeto\ a\ x_j \in \{0,1\}, \qquad j = 0, \ldots, m,$$

donde los coeficientes c_i y q_{ij} son números reales fijos.

En este tipo de problemas, trabajamos con variables binarias e intentamos optimizar una función cuadrática sin tener en cuenta ninguna restricción (de ahí el nombre, Quadratic Unconstrained Binary Optimization).

A pesar de que QUBO es una opción muy útil para trabajar con problemas, no es la única manera que tenemos de representarlos. En la siguiente sección se va a estudiar otra alternativa muy importante y que también se puede utilizar para la formulación de numerosos problemas: el modelo de Ising.

5.1.2. Modelo de Ising

Para facilitar la explicación y visualización del modelo de Ising, primero vamos a definir un problema que está muy relacionado, y que comparte muchas similitudes a la hora de formularse, el problema del corte máximo, o Max-Cut.

5.1.2.1. Problema del corte máximo (Max-Cut)

El problema del corte máximo (conocido en inglés como Max-Cut) se trata de uno de los problemas más típicos para resolver utilizando algoritmos cuánticos, debido a su facilidad para trabajar con él y a su directa formulación para utilizarlo en dichos algoritmos.

Para definir este problema, consideremos un grafo como el de la Figura 5.1 (que también se usa en [1]), que consta de 5 vértices y una serie de conexiones entre ellos.

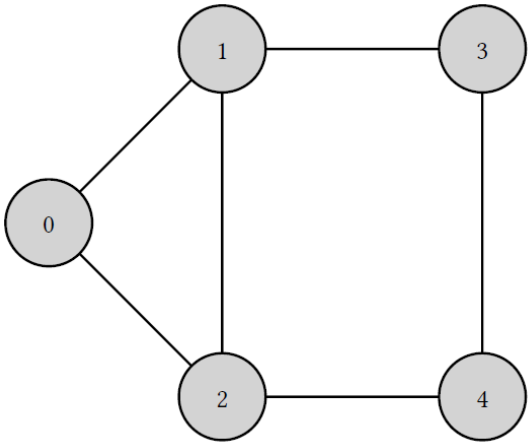


Figura 5.1 - Ejemplo de grafo para el problema del corte máximo

De forma resumida, el problema del corte máximo consiste en, dado un grafo, encontrar una forma de dividirlo en dos conjuntos, de forma que el número de conexiones que tengan en un extremo un vértice en un conjunto y en el otro extremo un vértice en otro conjunto sea el máximo (es decir, **cortamos** el grafo).

El número de vértices que cumple esa condición (tener en cada extremo un vértice en conjuntos distintos), se le denomina el **tamaño del corte**, y esto es lo que buscamos maximizar.

A modo de ejemplo, en el caso de la Figura 5.1, una solución óptima (o un corte óptimo), sería tener los vértices 0, 2 y 3 en un conjunto, y los vértices 1 y 4 en el otro. Esto nos daría un tamaño del corte de 5 (la única conexión que no tendría vértices en conjuntos distintos a cada lado sería la que hay entre los vértices 0 y 2).

Formulación

Para poder trabajar con este problema, debemos establecer una manera de formularlo. Para este experimento se va a utilizar la formulación definida en [1].

El primer paso es asignar a cada vértice ($i = 0 \dots n$) una variable z_i, que puede tomar valores -1 y 1 (así se decide la asignación de un vértice a un conjunto o a otro). En el caso del ejemplo anterior, la asignación para el corte máximo sería $z_0 = z_2 = z_3 = 1$ y $z_1 = z_4 = -1$ (aunque no es la única válida para encontrar la solución óptima)

Para poder formular el corte máximo como un problema de optimización combinatoria, nótese que si dos vértices se encuentran en conjuntos distintos, la multiplicación entre sus valores asociados en las variables siempre va a dar como resultado -1, y en cualquier otro caso el resultado será 1. Es decir, si tenemos dos vértices j y k y estos se encuentran en distintos conjuntos, podemos observar que

$$z_j z_k = -1$$

Por tanto, podemos ver que nuestro objetivo es conseguir el mayor número de multiplicaciones que den ese resultado. Esto se puede resumir como

$$Minimizar \sum_{(j,k)\,\epsilon\,E} z_j z_k$$

donde E es el conjunto de conexiones del grafo, $j = 0 \dots n-1$ y las variables solamente pueden tomar los valores -1 y 1. Esta expresión se puede comprobar con el ejemplo anterior, y nos dará un valor de -4, el mínimo que se puede obtener. Con esto se ha formulado el problema del corte máximo como un problema de minimización.

5.1.2.2. Definición: modelo de Ising

El modelo de Ising se trata de un modelo matemático para la interacción ferromagnética entre partículas con spin. Para codificar este spin se utilizan variables que pueden tomar el valor 1 o -1.

Este modelo recuerda a la formulación del problema del corte máximo, visto en la sección anterior, y esto es debido a que dicho problema es un caso específico de un problema físico: encontrar el estado de menor energía en un modelo de Ising [1].

La energía de este sistema viene dada por la función hamiltoniana, que se calcula de la siguiente manera:

$$-\sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j \, .$$

En esta expresión, J_{jk} indica la interacción entre las partículas j y k, y h_j representa el valor de la influencia de un campo magnético externo sobre la partícula j. Nuestro objetivo es dar con una configuración de spin que haga que el hamiltoniano tenga el menor valor posible. En ese punto, tendríamos el estado de menor energía del sistema.

5.1.2.3. Pasando de la formulación clásica a la cuántica

Todo lo mencionado hasta ahora del modelo de Ising entra dentro de lo clásico. Para pasar a una formulación cuántica, primero tenemos que considerar la matriz Z, que se puede utilizar para calcular el valor de distintos términos de la función que queremos minimizar.

Tenemos que

$$\langle 0|\mathrm{Z}|0\rangle = (1\ 0)\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1$$

y

$$\langle 1|\mathrm{Z}|1\rangle = (0\ 1)\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$$

Teniendo en cuenta el estado $|010\rangle$, y el producto tensorial $Z \otimes \mathrm{Z} \otimes \mathrm{I}$ se cumple que

$$\langle 010|Z \otimes \mathrm{Z} \otimes \mathrm{I}|010\rangle = \langle 010|(Z|0\rangle \otimes Z|1\rangle \otimes \mathrm{I}|0\rangle) = \langle 0|\,\mathrm{Z}\,|0\rangle\,\langle 1|\mathrm{Z}\,|1\rangle\,\langle 0|\,\mathrm{I}\,|0\rangle$$
$$= 1 \cdot (-1) \cdot 1 = -1$$

Si estuviésemos trabajando con el problema del corte máximo, por ejemplo, podríamos entender esto como una asignación en la que los vértices 0 y 2 están en un conjunto, y el vértice 1 en otro, y al ser el resultado del producto -1 significa que la conexión (0,1) tiene vértices en diferentes conjuntos, por la posición de las matrices Z, que actúan sobre los qubits 0 y 1. Una forma más resumida de denotar esto sería Z_0Z_1 (en las posiciones que no aparecen ninguna Z se aplica la identidad).

Teniendo en cuenta esto, podemos reformular el problema como encontrar un estado $|x\rangle$ para el que $\langle x|Z_0Z_1 + Z_0Z_2|x\rangle$ alcance el valor mínimo. Esto se puede generalizar para minimizar sobre todos los estados posibles.

Existen varios problemas que, al igual que el Max-Cut, se comportan de una forma similar al modelo de Ising, por lo que es un concepto bastante importante a la hora de entender mejor las formulaciones que se hacen para estos problemas. Además, hay varios tipos de ordenadores cuánticos, como los annealers (5.3.1. Annealing cuántico), que están hechos con el propósito de encontrar el estado de menor energía en sistemas que siguen este modelo.

5.2. Problemas

A continuación, se presenta la teoría relacionada con los problemas que se van a estudiar en el trabajo. Para cada problema se aporta su definición, la manera en la que se va a formular y una serie de aplicaciones prácticas.

La única excepción es el Max-Cut, el cual ya se definió y formuló previamente en la sección 5.1.2.1. Problema del corte máximo (Max-Cut), para facilitar la explicación del modelo de Ising.

5.2.1. Corte máximo (Max-Cut): aplicaciones

El problema del corte máximo tiene aplicaciones en varias áreas. Por ejemplo, en la construcción de **circuitos eléctricos**, se puede utilizar para dividir los componentes en dos grupos, maximizando la separación y minimizando las interconexiones, lo que puede mejorar la eficiencia y reducir la interferencia.

En el campo del **machine learning**, puede aplicarse para dividir conjuntos de datos en dos grupos, maximizando la separación entre las clases y mejorando la capacidad de clasificación del modelo.

El Max-Cut también se puede utilizar en la **elaboración de circuitos digitales**. En este caso los nodos del grafo podrían representar componentes del circuito, y los ejes conexiones entre ellos. Con esto en mente, el Max-Cut se podría usar para separar estos componentes en dos grupos de forma que se pueda construir un circuito con un diseño más eficiente, reduciendo costes y mejorando el rendimiento.

5.2.2. Programación Lineal Binaria (Binary Linear Program, BLP)

Antes de hablar de otros problemas que se van a estudiar en este trabajo además del Max-Cut, primero es importante entender los problemas de programación lineal binaria, ya que introduce conceptos que posteriormente se van a utilizar también en las formulaciones de los problemas que vamos a ver más tarde.

Las transformaciones que se van a ver en este punto sirven para pasar de cualquier problema linear binario a la forma QUBO, lo cual es relevante para este estudio ya

que varios de los problemas con los que se trabajan se pueden escribir como este tipo de problemas.

5.2.2.1. Definición

En los problemas de programación lineal binaria tenemos una función lineal, y nuestro objetivo es optimizar el valor de las variables binarias con las que cuenta, además de asegurarnos de cumplir una serie de restricciones, también lineales. La representación general de este tipo de problemas sería:

$$Minimizar \ \ c_0x_0 + c_1x_1 + \ldots + c_mx_m$$

$$sujeto\ a\ Ax \leq b,$$

$$x_j \in \{0,1\}, \qquad j = 0, \ldots, m,$$

que consta de unos coeficientes enteros, c_j, una matriz de enteros, A, la traspuesta de $(x_0, \ldots, x_m)$, que es x, y un vector columna de enteros, b.

La programación lineal binaria es NP-dura (es decir, al menos tan difícil como los problemas más difíciles de la categoría de tiempo polinomial no determinista, que se cree que nunca va a ser posible resolver de manera eficiente utilizando ordenadores clásicos), y las soluciones que cumplan las restricciones se llaman factibles.

5.2.2.2. Formulación

Aunque la forma en la que se escriben los BLP sea parecida a QUBO, todavía no se puede considerar como tal, y antes tenemos que hacer una serie de transformaciones, y para ello tendremos que trabajar con dos conceptos nuevos, las **variables de holgura**, y los **términos de penalización**.

El primer cambio que vamos a hacer es cambiar las restricciones de desigualdad a restricciones de igualdad, por medio de las variables de holgura.

Para explicar más claramente el proceso, vamos a ilustrar los cambios que iremos haciendo por medio del siguiente ejemplo [1]

$$Minimizar \ -5x_0 + 3x_1 - 2x_2$$

$$sujeto\ a\ x_0 + x_2 \leq 1,$$

$$3x_0 - x_1 + 3x_2 \leq 4$$

$$x_j \in \{0,1\}, \qquad j = 0,1,2\,.$$

En este caso estamos trabajando con dos restricciones, donde ambas son desigualdades. Para saber qué variables de holgura tenemos que añadir, primero hay que considerar cuándo se consigue el valor mínimo en cada restricción.

En la primera de ellas, esto sucede si x_0 y x_2 son 0, lo cual resulta en un valor mínimo total de 0. Sabiendo esto, podemos añadir una variable de holgura binaria, y_0, y sustituimos el símbolo de la desigualdad $\leq$ por una igualdad $=$, y el resultado sería

$$x_0 + x_2 + y_0 = 1,$$

que solamente se cumple si $x_0 + x_2 \leq 1$ también se satisface. En el caso de que $x_0 = x_2 = 0$, entonces podemos asignar $y_0 = 1$, y si $x_0 = 0$ y $x_2 = 1$, o viceversa, podemos asignar $y_0 = 0$. Lo que, por ejemplo, no sería posible en esta condición es $x_0 = x_2 = 1$, ya que no sería una solución factible debido a que incumple la restricción.

Respecto a la segunda restricción el valor mínimo se obtiene cuando $x_0 = 0, x_1 = 1,$ y $x_2 = 0$, siendo el valor total -1. Para que $3x_0 - x_1 + 3x_2$ llegue a 4, necesitaríamos añadir un número que como máximo valga 5. Para ello podemos utilizar tres nuevas variables binarias (ya que con tres bits se puede representar el 5), $y_1,\ y_2,\ y_3$, de esta forma

$$3x_0 - x_1 + 3x_2 + \ y_1 + 2y_2 + \ 4y_3 = 4$$

que se cumple solo si la condición original, sin las variables añadidas, también se cumple.

Tras haber añadido las nuevas variables, y haber pasado las desigualdades a igualdades, nuestro ejemplo inicial ahora quedaría así

$$Minimizar\ -5x_0 + 3x_1 - \ 2x_2$$

$$sujeto\ a\ x_0 + x_2 + y_0 = 1,$$

$$3x_0 - x_1 + 3x_2 + \ y_1 + 2y_2 + \ 4y_3 = 4$$

$$x_j \in \{0, 1\}, \qquad j = 0, 1, 2$$

$$y_j \in \{0, 1\}, \qquad j = 0, 1, 2, 3$$

La segunda transformación que vamos a hacer es introducir los términos de penalización, para lo cual vamos a usar las restricciones que acabamos de escribir como igualdades. También vamos a hacer uso de un entero B, al que le vamos a asignar un valor de forma que incumplir las restricciones penalice a las soluciones no factibles.

$$Minimizar\ -5x_0 + 3x_1 - \ 2x_2 + B(x_0 + x_2 + y_0 - 1)^2$$

$$+ B(3x_0 - x_1 + 3x_2 + \ y_1 + 2y_2 + \ 4y_3 - 4)^2$$

$$sujeto\ a\ x_j \in \{0,1\}, \qquad j = 0,1,2$$

$$y_j \in \{0,1\}, \qquad j = 0,1,2,3$$

Para asignar el valor de B, tendremos en cuenta que en la expresión original, $-5x_0 + 3x_1 - 2x_2$, el valor mínimo que se puede obtener es -7, y el máximo 3, por lo que podemos asignar $B = 11$, y así cualquier solución no factible o inválida obtendrá como mínimo un valor de 4, por lo que nunca será elegida como óptima (siempre que haya por lo menos una solución factible).

Con esto ya hemos completado la transformación del problema original a una forma QUBO, con la que podríamos resolver el problema utilizando algoritmos cuánticos

$$Minimizar\ -5x_0 + 3x_1 - 2x_2 + 11(x_0 + x_2 + y_0 - 1)^2$$

$$+ 11(3x_0 - x_1 + 3x_2 + y_1 + 2y_2 + 4y_3 - 4)^2$$

$$sujeto\ a\ x_j \in \{0,1\}, \qquad j = 0,1,2$$

$$y_j \in \{0,1\}, \qquad j = 0,1,2,3$$

5.2.3. Problema del viajante de comercio (Travelling Salesman Problem)

5.2.3.1. Definición

El TSP se trata de otro problema de optimización muy conocido. En este problema contamos con una serie de nodos (que pueden representar, por ejemplo, ciudades), y conexiones entre ellos, cada una con un coste asociado.

Tenemos que tratar de encontrar un camino que pase por todos los nodos una única vez, y que tenga el menor coste total posible. En este caso, vamos a considerar una versión algo simplificada del problema, en la que no es necesario que el camino acabe en el mismo nodo en el que empezó, como aparece en algunas ocasiones.

5.2.3.2. Formulación

Para este estudio vamos a formular el TSP por medio de grafos [1]. Consideramos el conjunto de vértices $j = 0, \dots, m$, que representa las ciudades, y para cada par de vértices j y l tenemos el coste w_{jl}, de viajar desde j a l. Un detalle a tener en cuenta es que el coste w_{jl} no tiene que ser siempre el mismo que w_{lj}.

El objetivo, en términos de grafos, sería dar con una serie de conexiones, en las que necesariamente el final de una sea el comienzo de la siguiente, que pase por cada vértice una sola vez, y que resulte en la menor suma de los costes de las conexiones utilizadas.

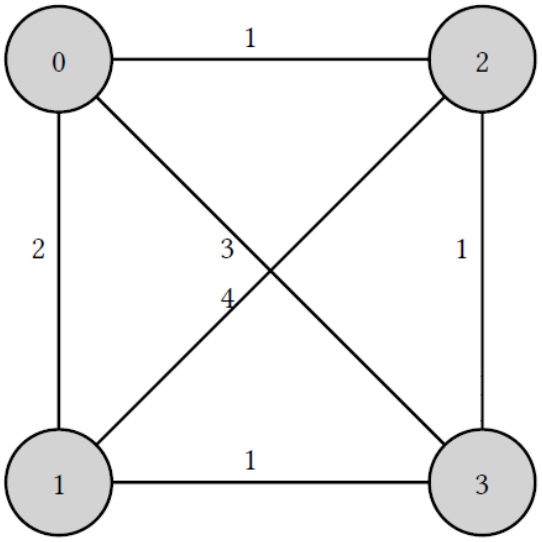


Figura 5.2 – Ejemplo de grafo para el problema del TSP

Consideremos el grafo de la Figura 5.2, que es un ejemplo que también se presenta en [1]. Este representa una instancia de un problema del TSP en el que tenemos 4 ciudades, que están conectadas entre sí, y en el que cada conexión tiene un coste (en este caso w_{jl} y w_{lj} tienen el mismo coste, pero, como se ha comentado previamente, no tiene por qué ser necesariamente siempre así).

El primer paso para construir nuestra formulación del problema utilizando QUBO es considerar variables binarias w_{jl}, que representarán el orden en el que se ha pasado por cada vértice.

Esto quiere decir que si por ejemplo el vértice j es el l-ésimo en ser visitado en el camino, entonces la variable x_{jl} tendrá el valor 1, y para todas las variables x_{jh} en las que $h \neq l$ el valor de la variable será 0.

Respecto a las restricciones, para cumplir que cada vértice se visite solamente una vez tenemos que para cada vértice j

$$\sum_{l=0}^{m} x_{jl} = 1$$

Y también tenemos que imponer la restricción de que sólo se puede visitar una ciudad al mismo tiempo, de esta manera

$$\sum_{j=0}^{m} x_{jl} = 1$$

Además de tener en cuenta esas restricciones, otro de los objetivos que teníamos es el de minimizar el coste total del camino. Para ello necesitaremos una expresión que nos dé el coste en función de las variables x_{jl}.

Una conexión entre los nodos j y k sólo se puede usar si los nodos j y k son consecutivos en el camino, es decir, solo si existe una posición l de manera que j se visite en la posición l y k se visite en la posición $l+1$, en cuyo caso el coste de utilizar dicha conexión sería $w_{jk}x_{jl}x_{kl+1}$, porque $x_{jl}x_{kl+1}=1$ (si j y k no fuesen consecutivos, el resultado sería 0).

Esto resulta en que el coste total del camino se puede calcular utilizando la expresión

$$\sum_{l=0}^{m-1}\sum_{j=0}^{m}\sum_{k=0}^{m} w_{jk}x_{jl}x_{kl+1}\,,$$

en la que el valor de w_{jj} sería 0 para $j=0,\dots,m$, ya que el coste de quedarse en el mismo sitio sería nulo.

El siguiente paso es añadir los términos de penalización a la función que queremos minimizar utilizando las restricciones. El problema quedaría así

$$Minimizar \sum_{l=0}^{m-1}\sum_{j=0}^{m}\sum_{k=0}^{m} w_{jk}x_{jl}x_{kl+1} + B\left(\sum_{l=0}^{m} x_{jl}-1\right)^2 + B\left(\sum_{j=0}^{m} x_{jl}-1\right)^2$$

$$sujeto\ a\ x_{jl} \in \{0,1\}, \qquad j,l=0,\dots,m,$$

en la que tenemos que considerar un valor B lo suficientemente grande como para que las soluciones inválidas no puedan llegar a tener un valor óptimo, por ejemplo

$$B = 1 + \sum_{j,k=0}^{m} w_{jk}.$$

Con ese valor de B las soluciones inválidas siempre tendrán un coste mayor al coste de cualquier camino válido, por lo que no podrá ser seleccionada como óptima, y B estaría cumpliendo su función de manera satisfactoria.

5.2.3.3. Aplicaciones

Además del escenario que se plantea en la definición del problema, del comerciante que tiene que viajar a través de ciudades, el TSP tiene una gran variedad de usos.

Un ejemplo de estos usos, que se puede visualizar bastante bien, es el de la **logística y entrega de paquetes**. Se pueden reemplazar los nodos por los destinos en los que se

tienen que repartir los paquetes, y las conexiones entre los nodos serían las rutas que hay para llegar a estos destinos.

El TSP puede encontrar la ruta más eficiente, en base a la medida que quedamos minimizar, como puede ser el tiempo que se tarde en visitar todos los destinos, o los litros de combustible utilizados, por ejemplo.

Otra circunstancia en la que el TSP puede ser útil es a la hora de **diseñar circuitos electrónicos**, ya que puede ayudar a optimizar la manera en la que se conectan los componentes que forman dichos circuitos, por ejemplo minimizando la distancia entre los mismos, lo cual puede verse reflejado en mejoras en la cantidad de energía o espacio que se usa, y en la velocidad de procesamiento.

5.2.4. Problema de la mochila (Knapsack problem)

5.2.4.1. Definición

En el problema de la mochila tenemos una serie de objetos $j = 0, \dots, m$, cada uno con un peso w_j y un valor c_j, y tenemos que elegir el conjunto de objetos que nos dé el máximo valor sin pasarnos de un peso máximo W, siendo esa la única restricción con la que trabajamos.

5.2.4.2. Formulación

Como se expone en [1], es posible formular el problema de la mochila como un programa lineal binario. Primero tenemos que decidir las variables binarias que vamos a utilizar. Un enfoque válido sería usar una variable para cada objeto, la cual tendría el valor 1 en caso de que lo escojamos, y 0 si no.

De manera matemática, esto se traduce en que trabajamos con las variables binarias x_j, con $j = 0, \dots, m$, en las que j indica el objeto que decidimos si seleccionar $(x_j = 1)$ o no $(x_j = 0)$. Esto resulta en la función

$$Minimizar - c_0x_0 - c_1x_1 - \dots - c_mx_m$$

$$sujeto\ a\ w_0x_0 + w_1x_1 + \dots + w_mx_m \leq W,$$

$$x_j \in \{0, 1\}, \qquad j = 0, \dots, m,$$

en la que c_j es el valor del objeto, w_j su peso, y W es el peso máximo al que podemos llegar. Como se comentó previamente, nuestro objetivo es maximizar el valor de los objetos que escogemos, pero en este caso el problema se presenta de una forma distinta pero equivalente, que es minimizar el valor negativo.

Para convertir esa función en un problema QUBO, tendríamos que añadir variables de holgura y términos de penalización, y ya sería posible resolver el problema utilizando algoritmos cuánticos.

5.2.4.3. Aplicaciones

Al igual que los otros problemas vistos hasta ahora, el problema de la mochila tiene diversas aplicaciones en numerosas áreas.

Un ejemplo de estas aplicaciones se puede ver en la **planificación de proyectos**, donde se puede usar para optimizar la asignación de recursos a tareas específicas. A cada tarea se le asigna una duración o un costo, y una recompensa, y a través del problema de la mochila se puede seleccionar un conjunto de tareas que maximice la recompensa total, teniendo en cuenta las restricciones de tiempo y de dinero.

Otro caso en el que puede ser de utilidad es en la **gestión de activos**, donde los activos tienen unos valores, que puede ser por ejemplo el rendimiento que puede llegar a dar, y unos pesos, como por ejemplo el riesgo asociado.

También se puede utilizar en un escenario que ya ha aparecido en problemas anteriores, el de la **logística**, para maximizar la cantidad de productos que se pueden entregar dentro de unas restricciones.

5.2.5. Problema del coloreado de grafos (Graph coloring)

5.2.5.1. Definición

En el problema del coloreado de grafos tenemos un grafo con numerosos vértices, y nuestro objetivo es asignar un color a cada vértice de forma que todos los vértices adyacentes (es decir, los que están conectados entre sí) tengan colores distintos.

En el estudio vamos a considerar también la restricción de que tengan que usarse como máximo k colores (en cuyo caso se diría que el grafo es k-coloreable). Otro concepto importante en este problema es el **número cromático**, que se trata del número mínimo de colores con el que se puede colorear un grafo.

5.2.5.2. Formulación

Para formular este problema vamos a hacer uso también del framework QUBO [1]. Tenemos un grafo con los vértices $0, \dots, m$. Sabiendo eso, definimos las variables binarias x_{jl} con $j = 0, \dots, m$ y $l = 0, \dots, k-1$, donde $x_{jl} = 1$ si el vértice j recibe el l-ésimo color, y 0 en el caso de que no.

Para asegurarnos de que al vértice j se le asigne solamente un color, tenemos que imponer

$$\sum_{l=0}^{k-1} x_{jl} = 1\,.$$

También tenemos que establecer la restricción de que los vértices adyacentes no pueden tener el mismo color. Para ello debemos tener en cuenta que si dos vértices j y h reciben el mismo color, l, entonces tendríamos que $x_{jl}x_{hl} = 1$, por lo que para todos los vértices adyacentes tenemos que asegurarnos que

$$\sum_{l=0}^{k-1} x_{jl}x_{hl} = 0$$

Ahora que tenemos formuladas las restricciones, podemos incorporarlas en la función a minimizar como términos de penalización para obtener la forma QUBO

$$Minimizar \sum_{j=0}^{m}\left(\sum_{l=0}^{k-1} x_{jl} - 1\right)^2 + \sum_{(j,h)\in E}\sum_{l=0}^{k-1} x_{jl}x_{hl}$$

$$sujeto\ a\ x_{jl} \in \{0,1\}, \qquad j = 0,\dots,m, \qquad l = 0,\dots,k-1$$

donde E es el conjunto de conexiones del grafo. Si la solución óptima es 0, el grafo es k-coloreable.

5.2.5.3. Aplicaciones

El problema del coloreado de grafos también es uno de los que más popularidad tiene entre los problemas de optimización, y puede ser de utilidad en distintas circunstancias.

Un escenario en el que se usa bastante es a la hora de asignar **frecuencias en redes de telecomunicaciones**. Se trata de evitar que torres vecinas tengan frecuencias similares, para evitar interferencias.

También se puede utilizar para hacer **horarios**, como por ejemplo el de la universidad, para evitar conflictos, como por ejemplo actividades que se lleven a cabo en la misma aula al mismo tiempo, o alumnos tengan asignaturas distintas a la misma hora.

5.3. Métodos

Una vez tenemos los problemas formulados, podemos pasar a hablar de qué opciones tenemos para resolverlos.

Como se ha mencionado previamente, existen distintas implementaciones de ordenadores cuánticos, y también distintos métodos derivados de ellos para resolver problemas. A continuación, se van a definir los métodos utilizados y analizados en este estudio.

5.3.1. Annealing cuántico

Los annealers cuánticos son un tipo de ordenadores cuánticos que se pueden utilizar para resolver problemas como los formulados anteriormente, y obtener soluciones aproximadas. Para explicar este concepto antes hay que hablar del paradigma de la **computación adiabática cuántica**.

5.3.1.1. Computación adiabática cuántica

En el marco teórico se habló de que había distintos modelos de computación cuántica. Uno de los más populares es el de circuitos cuánticos, pero otro de ellos es la computación adiabática cuántica [13].

Una de las principales diferencias con el modelo de circuitos cuánticos es que en vez de utilizar operaciones discretas y de manera secuencial, por medio de puertas cuánticas, en este modelo se usan transformaciones continuas.

Para ello utilizaremos un **hamiltoniano** $H(t)$ que va a variar a lo largo del tiempo, y del que depende el estado de los qubits, que va a alterarse siguiendo la ecuación de Schrödinger, que está en función del tiempo (es importante diferenciarla de la que no está en función del tiempo):

$$H(t)|\psi(t)\rangle = \mathrm{i}\,\hbar\,\frac{\partial}{\partial \mathrm{t}}|\psi(t)\rangle$$

El otro componente importante de este modelo es la **evolución adiabática**, que, en resumen, se trata del paso de un sistema cuántico de un estado inicial a uno final de forma gradual y suave. Durante esta evolución, el sistema sigue el principio de la adiabaticidad, lo que significa que se mantiene en un estado de equilibrio en cada etapa y no experimenta cambios abruptos, por lo que obtenemos un mayor control sobre las transiciones en los sistemas cuánticos, y evitamos que se pase de un estado fundamental (el de mínima energía) a uno excitado (que tendría mayor energía y por tanto no sería óptimo).

Para entender cómo intervienen estos conceptos en la resolución de los problemas que hemos formulado, tenemos que entender que estamos formulándolos de manera que la solución óptima se va a corresponder con el estado fundamental del hamiltoniano que estemos utilizando de un modelo de Ising.

La idea fundamental sobre la computación adiabática cuántica que nos interesa de cara al estudio, para entender cómo funcionan las operaciones, es que vamos a comenzar con un hamiltoniano para el que podamos obtener y preparar el estado fundamental, y luego lo vamos a ir evolucionando de manera adiabática, sin que salte a un estado excitado, cambiando el sistema hasta que el estado fundamental se corresponda con la solución del problema, que sería el punto en el que ya se podría hacer la medición para obtener un resultado.

El hamiltoniano que usemos, H_1, podría tratarse, por ejemplo, de un hamiltoniano de Ising, obtenido de haber transformado un problema QUBO. Si consideramos el hamiltoniano inicial, que construimos en estado inicial, como H_0, y ejecutamos el proceso durante una cantidad de tiempo T, consideramos el hamiltoniano que depende del tiempo

$$H(t) = A(t)H_0 + B(t)H_1,$$

donde A y B son funciones con valores reales cuyos valores están en el intervalo [0, T], de modo que $A(0) = B(T) = 1$ y $A(T) = B(0) = 0$.

La computación cuántica adiabática es polinomialmente equivalente a los otros modelos de computación cuántica [14], por lo que todo lo que se puede calcular utilizando los otros modelos también se puede hacer con este, y dependiendo de los problemas (o del tipo de ordenadores a los que tenemos acceso) puede resultar interesante probar con un modelo u otro.

5.3.1.2. Annealing cuántico

Para explicar el annealing cuántico, previamente hemos tenido que hablar de la computación adiabática cuántica. Esto es debido a que el annealing es una implementación práctica, algo limitada, de ese modelo.

El annealing comparte la manera de trabajar con hamiltonianos de la computación adiabática cuántica, pero se diferencia de ella principalmente en dos aspectos [1]:

1. En esta implementación práctica, el hamiltoniano final no se puede elegir libremente, sino que tiene que estar dentro de una clase restringida.

   Esta restricción hace que el número de problemas que el annealer puede resolver comparado con la computación adiabática sea menor, pero tiene la ventaja de que permite construir dispositivos cuánticos físicos más simples, y que se pueden escalar a un gran número de qubits.

2. La otra diferencia es que en el annealer no se garantiza que la evolución sea adiabática, lo cual implica que no siempre vamos a encontrar la solución óptima, pero a cambio se obtiene mayor velocidad de procesamiento (la evolución adiabática puede tardar demasiado tiempo en realizarse).

La empresa D-Wave fue la primera en comercializar un annealer, como el que hemos descrito en este punto [1], y también permite utilizar este tipo de dispositivos de manera online. Esta será la herramienta que se utilizará para resolver problemas con este método en el estudio.

Estos annealers se pueden utilizar para aproximar soluciones a problemas, que se pueden formular por medio del estado fundamental de un modelo de Ising o como QUBO. También permiten controlar muchos parámetros a la hora de hacer las ejecuciones, como por ejemplo la forma de mapear los qubits. Configurando estos parámetros se puede aumentar la calidad de los resultados.

5.3.2. QAOA: Quantum Approximate Optimization Algorithm

Tras haber hablado sobre el annealer y cómo se puede utilizar para resolver problemas, en esta sección se va a presentar otro método con mucho potencial para el que se pueden utilizar los dispositivos NISQ, el algoritmo QAOA.

Este algoritmo está bastante relacionado con el annealing, ya que originalmente se desarrolló a partir de la computación adiabática, aunque tiene algunas diferencias muy importantes, como por ejemplo que utiliza circuitos cuánticos mientras que el annealer no.

5.3.2.1. Definición

El **Algoritmo Cuántico de Optimización Aproximada** (QAOA utilizando las siglas en inglés) se puede entender como un análogo del annealing pero en el modelo de circuitos cuánticos [1], y es posible explicar cómo funciona partiendo de lo que se ha explicado anteriormente sobre el annealing.

Como se ha visto previamente, las operaciones que se realizan en un annealer tienen una naturaleza continua, pero el modelo de circuitos cuánticos funciona con operaciones discretas.

Existe un mecanismo para pasar de variables continuas a discretas, a través de una serie de pasos, que se llama **trotterización**. Esto es relevante, ya que el algoritmo QAOA surgió como una trotterización (o discretización) de la computación adiabática [6].

Aplicando la discretización, podemos pasar de esta expresión, que es la que se utiliza para el annealing cuántico

$$H(t) = A(t)H_0 + B(t)H_1$$

a esta otra

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

que se trata de un producto de operadores aplicado al estado inicial. En esta expresión i es la unidad imaginaria, t_c es un punto temporal fijo entre [0, T] y Δt es una cantidad pequeña de tiempo (cuanto más pequeña, mejor la aproximación).

En el intervalo $[t_c, t_c + \Delta t]$ asumimos que el hamiltoniano es constante, y que tiene el valor

$$H(t_c) = A(t_c)H_0 + B(t_c)H_1$$

Si consideramos $|\psi_0\rangle$ como estado inicial, el estado final se aproximaría como

$$\left(\prod_{m=0}^{p} e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)}\right)|\psi_0\rangle$$

Donde $t_m = m\frac{\Delta t}{T}$ y $p = m\frac{T}{\Delta t}$. Para calcular este estado con un circuito cuántico, tenemos que hacer otra aproximación, que se cumple si Δt es pequeño. Se conoce como la fórmula Lie-Trotter:

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)} = e^{i\Delta tA(t_c)H_0}e^{i\Delta tB(t_c)H_1}$$

Con todo esto, podemos concluir que la aproximación del estado final de la evolución adiabática, que se trata de la inspiración para el QAOA, se puede obtener de esta forma:

$$\prod_{m=0}^{p} e^{i\Delta tA(t_m)H_0}e^{i\Delta tB(t_m)H_1}|\psi_0\rangle$$

5.3.2.2. Algoritmo

Al igual que para el annealing, comenzamos con un hamiltoniano de Ising, H_1, que se usa para codificar el problema. También buscamos que se produzca una evolución en el estado cuántico, pero esta vez a través de un circuito.

Para llevar a cabo esta evolución, comenzaremos con el estado inicial $|\psi_0\rangle$ e iremos alternando aplicaciones de los operadores $e^{i\gamma H1}$ y $e^{i\beta H0}$ para una serie de valores de γ y β un total de p veces (estos operadores se pueden implementar en los circuitos utilizando puertas cuánticas de uno y de dos qubits). Haciendo esto, estamos preparando el siguiente estado, con $p \geq 1$

$$e^{i\beta_p H_0}e^{i\gamma_p H_1} \dots e^{i\beta_2 H_0}e^{i\gamma_2 H_1}e^{i\beta_1 H_0}e^{i\gamma_1 H_1}|\psi_0\rangle$$

Los coeficientes de β y de γ se guardan en tuplas, y el estado se denota como $|\beta,\gamma\rangle$. Tenemos libertad a la hora de escoger los valores de β y de γ, por lo que podemos buscar los que mejores resultados nos den, que en nuestro caso serían los que hagan que el estado $\langle\beta,\gamma|H_1|\beta,\gamma\rangle$ tenga la menor energía, por lo que nuestro problema pasaría a ser obtener los valores de β y de γ que minimicen ese estado:

$$E(\beta,\gamma) = \langle\beta,\gamma|H_1|\beta,\gamma\rangle$$

Una vez tengamos un valor de $E(\beta,\gamma)$, podemos utilizar un algoritmo clásico de minimización de funciones, e ir probando con valores de $E(\beta,\gamma)$ (que se calculan en el ordenador cuántico, debido a que es mucho más eficiente que uno clásico en esta tarea) hasta que se llegue al criterio de parada del algoritmo clásico.

En este caso estamos hablando de un algoritmo híbrido, ya que se utilizan tanto ordenadores clásicos como cuánticos para llevar a cabo algunas tareas del proceso.

Cuando hayamos dado con los valores óptimos β^* y γ^* o, en su defecto, aproximaciones, se utiliza el ordenador cuántico para preparar el estado $|\beta^*,\gamma^*\rangle$ que, al medir, nos debería dar una solución, si no óptima, aproximada, del problema que queremos resolver.

Cuando se presentó el algoritmo QAOA, este tenía una mayor efectividad de aproximación que cualquier algoritmo clásico que se ejecutase en tiempo polinomial, aunque esto cambiaría algo después, con la aparición de un algoritmo que tenía mejores resultados [15].

Para poder trabajar con el QAOA, no es necesario que se implemente manualmente. Tenemos diversas herramientas a nuestra disposición, como por ejemplo las que ofrece Qiskit, que permite hacer ejecuciones tanto en simuladores cuánticos como en ordenadores cuánticos reales. En este estudio se han elegido las herramientas proporcionadas por Qiskit para llevar a cabo los experimentos, las cuales permiten trabajar con problemas formulados con QUBO y hamiltonianos.

5.3.2.3. Circuitos

En la sección anterior se ha hablado de las partes del QAOA en las que se utilizan ordenadores cuánticos, que tienen como objetivo la preparación del estado

$$|\beta,\gamma\rangle = e^{i\beta_p H_0} e^{i\gamma_p H_1} \dots e^{i\beta_2 H_0} e^{i\gamma_2 H_1} e^{i\beta_1 H_0} e^{i\gamma_1 H_1} |\psi_0\rangle$$

donde $|\psi_0\rangle$ es un estado fundamental de H_0. En esta sección se va a explicar la forma en la que se construyen los circuitos para llevar a cabo esa operación.

Empezamos teniendo en cuenta que H_0 suele ser $-\sum_{j=0}^{n-1} X_j$, y H_1 un hamiltoniano de Ising

$$-\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j,$$

donde los coeficientes J_{jk} y h_j son números reales.

El estado fundamental de H_0 es $\bigotimes_{i=0}^{n-1} |+\rangle$, el cual se puede preparar usando una puerta H en cada qubit, teniendo el estado inicial $|0\rangle$.

La siguiente operación a codificar es $e^{i\beta_k H_0}$, en la que β_j es un número real. Sabiendo que $H_0 = -\sum_{j=0}^{n-1} X_j$, y que todas las matrices X_j conmutan entre ellas, se puede cambiar el exponente de la suma por el producto de los exponentes, lo que resulta en:

$$e^{i\beta_k H_0} = e^{-i\beta_k \sum_{j=0}^{n-1} X_j} = \prod_{j=0}^{n-1} e^{-i\beta_k X_j}$$

La expresión $e^{-i\beta_k X_j}$ es la de la puerta de rotación $R_X(2\beta)$, por lo que, al igual que con la puerta H, también tenemos que aplicar esta puerta a todos los qubits.

Otro término a representar es $e^{i\gamma_l H_1}$ para cualquier coeficiente real γ_l. Al ser H_1 una suma de los términos en la forma de $J_{jk} Z_j Z_k$ y $h_j Z_j$, y al conmutar las matrices entre ellas nos queda

$$e^{i\gamma_l H_1} = e^{-i\gamma_l (\sum_{j,k} J_{jk} Z_j Z_k + \sum_j h_j Z_j)} = \prod_{j,k} e^{-i\gamma_l J_{jk} Z_j Z_k} \prod_j e^{-i\gamma_l h_j Z_j}$$

La operación $e^{-i\gamma_l h_j Z_j}$ se puede implementar a través de puertas de rotación R_Z.

Nos quedaría por traducir $e^{-i\gamma_l J_{jk} Z_j Z_k}$. Primero podemos sustituir $\gamma_l J_{jk}$ por a para hacer el cambio más sencillo. La expresión $e^{-iaZ_j Z_k}$ es la exponencial de una matriz diagonal, porque $Z_j Z_k$ es el producto tensorial de matrices diagonales. Si $|x\rangle$ es un estado de la base computacional en el que los qubits j y k tienen el mismo valor, entonces

$$e^{-iaZ_j Z_k} |x\rangle = e^{-ia} |x\rangle$$

Si los qubits tienen valores diferentes, entonces

$$e^{-iaZ_j Z_k} |x\rangle = e^{ia} |x\rangle$$

Este paso se lleva a cabo de la forma mostrada en el siguiente circuito, en el que se representa lo que ocurre sobre los qubits j y k (para los otros se aplicaría la puerta identidad).

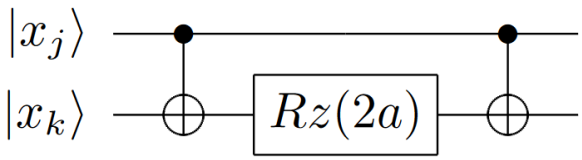


Figura 5.3. Circuito para implementar parte del QAOA

Tras hablar de cómo utilizar puertas para realizar las operaciones, se puede mostrar cómo preparar un estado $|\beta, \gamma\rangle$ a través de un circuito (en este caso tendría $p = 1$). Podemos tener por ejemplo un hamiltoniano de Ising para nuestro problema que sea $3Z_0Z_2 - Z_1Z_2 + 2Z_0$

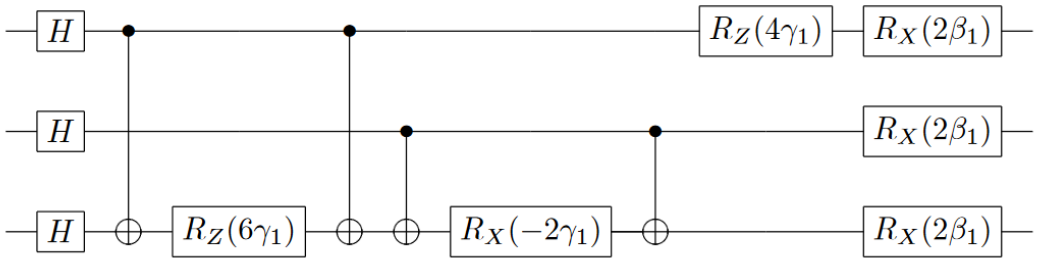


Figura 5.4. Circuito de ejemplo para el QAOA

Primero preparamos el estado fundamental de H_0 utilizando puertas Hadamard en todos los qubits. Para implementar $e^{-i3\gamma_1 Z_0 Z_2}$ utilizamos una puerta NOT controlada entre los qubits 0 y 2, luego una puerta R_Z en el qubit 2, y finalmente otra puerta NOT controlada entre los qubits 0 y 2. Para codificar $e^{-i\gamma_1 Z_1 Z_2}$ se haría lo mismo pero trabajando con los qubits 1 y 2.

A continuación, para representar $e^{-i2\gamma_1 Z_0}$ utilizamos una puerta R_Z en el qubit 0, y después aplicamos una puerta R_X a todos los qubits para implementar $e^{-i\beta_1 \Sigma_j X_j}$.

Si estuviéramos trabajando con un número mayor de p, tendríamos que ampliar el circuito repitiendo $p - 1$ veces el circuito a excepción de las puertas Hadamard iniciales. También tendríamos que sustituir en cada repetición los parámetros γ_1 y β_1 por los correspondientes a cada capa.

Capítulo 6

Metodología de trabajo

A modo general, para llevar a cabo el estudio, se ha seguido un proceso similar para cada problema.

Primero se ha generado un número de grafos para cada número de vértices que se iba a utilizar para obtener las estadísticas. Cada uno de estos grafos representa una instancia del problema a estudiar, y contiene los datos necesarios para construirlo.

A continuación, se recorre cada uno de los grafos y se resuelve utilizando los distintos métodos que se están comparando en el estudio. Una vez se obtiene el resultado, se guardan los siguientes valores sobre cada muestra:

1. Si se ha obtenido la solución óptima
2. Si se ha obtenido una solución válida
3. Energía de la solución obtenida
4. Energía de la solución óptima

Una vez se han obtenido los datos de todas las muestras, estas se utilizan para llevar a cabo una serie de comparaciones entre los resultados obtenidos al variar el número de vértices, el método, o el número de repeticiones (si aplica al método).

Los métodos que se van a utilizar para resolver los problemas son los siguientes:

QAOA en simulador local

Su teoría se encuentra en la sección 5.3.2. QAOA: Quantum Approximate Optimization Algorithm. Una de las formas de ejecutar este algoritmo es simulándolo de forma local, a través de diversas clases proporcionadas por IBM y Qiskit.

La mayor ventaja que ofrece esta opción es que evitamos tener que esperar por posibles colas que pueda haber al querer utilizar dispositivos reales que ofrezcan las empresas, los cuales son escasos y puede resultar muy costoso en cuanto a tiempo utilizarlos.

Al usar el simulador local, tenemos que tener en cuenta que las capacidades del algoritmo, como pueden ser el número máximo de qubits que se pueden utilizar, dependerá de las especificaciones del dispositivo en el que se esté haciendo la simulación.

En este caso, el ordenador en el que se ha realizado la simulación cuenta con los siguientes componentes:

Sistema operativo: Windows 10 Pro

Procesador: Intel Core i7-10750H.

Tarjeta gráfica: NVIDIA GeForce GTX 1660 Ti Max-Q.

Memoria RAM: 16GB.

Con estas características, la simulación se pudo realizar hasta cerca de 30 qubits. La resolución de los problemas que necesitaban alrededor de ese número de qubits comenzó a ser demasiado prolongada (del orden de horas para cada instancia).

Además del hardware, también cabe mencionar otras características del entorno utilizado, como, por ejemplo, las librerías. Estas vienen listadas, junto con sus versiones, en el archivo requirements.txt del repositorio en el que se encuentra el trabajo realizado en este TFG (véase 8.2. Difusión de resultados).

Otros datos relevantes sobre la configuración son

Versión de Python: 3.9.13

IDE: IntelliJ IDEA 2021.3.2

QAOA en simulador remoto

Su teoría también se encuentra en la sección 5.3.2. QAOA: Quantum Approximate Optimization Algorithm, al tratarse del mismo algoritmo. Además de simuladores locales, también tenemos a nuestra disposición simuladores remotos, gracias a organizaciones, como por ejemplo IBM.

El simulador remoto que se va a utilizar para este estudio es el ibmq_qasm_simulator, proporcionado de manera gratuita por IBM, que cuenta con 32 qubits.

Annealer

Su teoría se encuentra en la sección 5.3.1. Annealing cuántico. Como se mencionó también en ese apartado, se va a utilizar un annealer real proporcionado por la empresa D-Wave. En concreto, se ha utilizado el dispositivo Advantage_system4.1, que tiene soporte tanto para problemas formulados en el modelo de Ising como en QUBO, y que cuenta con estas características:

Número total de qubits: 5760.

Número de qubits funcionales: 5627.

Topología: Pegaso.

La topología es la manera en la que los qubits están conectados entre sí, ya que actualmente no se pueden conectar todos entre todos. En concreto, en la topología

Pegaso los qubits se distribuyen de la siguiente manera, donde cada qubit se representa con una elipse elongada:

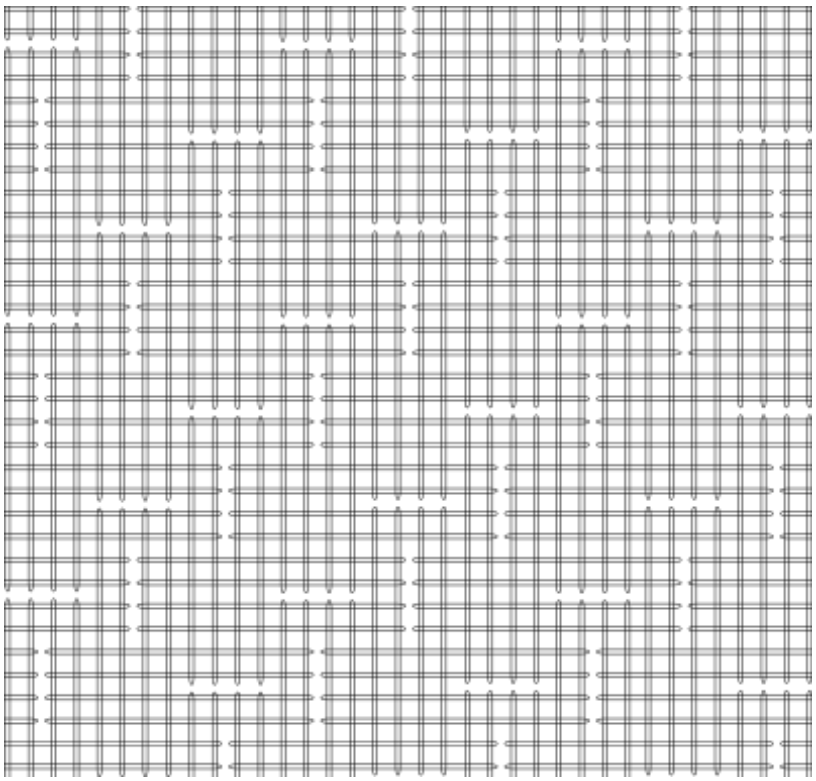


Figura 6.1. Topología Pegaso (créditos a [16]*)*

Además, cada qubit está conectado a otros 15 (esto se puede ver en la Figura 6.2, donde cada qubit acoplado se marca con un punto negro):

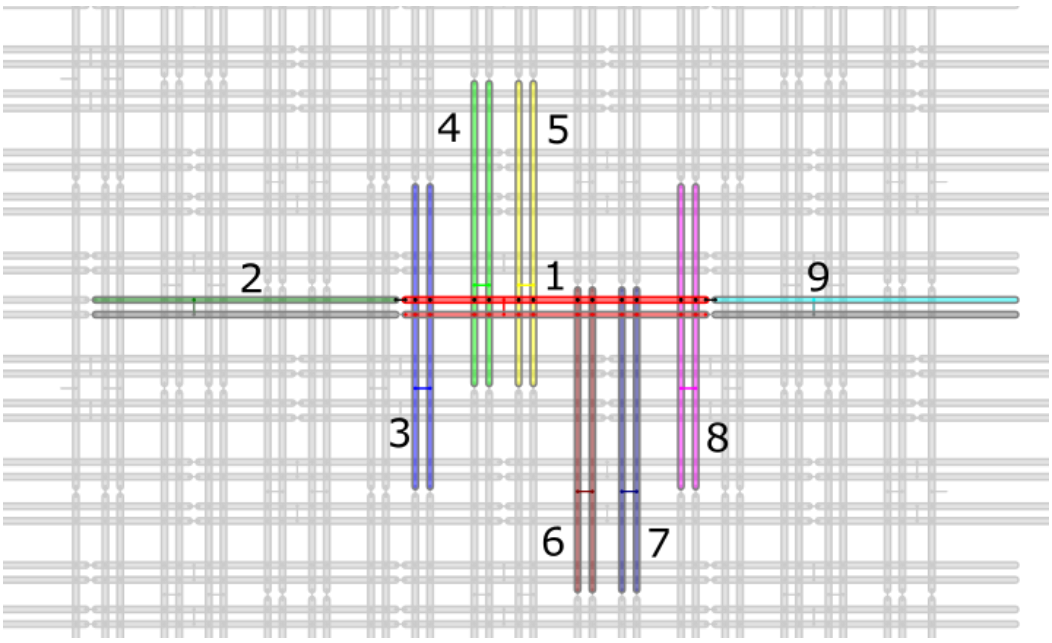


Figura 6.2. Topología Pegaso, qubits acoplados (créditos a [16]*)*

6.1. Corte máximo (Max-Cut)

6.1.1. Datos utilizados

Para el estudio sobre este problema se ha considerado una muestra de 20 grafos generados aleatoriamente para cada número de vértices, que están comprendidos entre 5 y 13.

Estos datos se han proporcionado en el repositorio indicado en el punto 8.2. Difusión de resultados, con el objetivo de permitir la reproducción de los resultados obtenidos en el experimento.

6.1.2. Formulación del problema

6.1.2.1. QAOA – Simulador local y simulador remoto

El problema se ha formulado construyendo directamente el hamiltoniano de Ising, elaborando primero los términos de Pauli (a partir de las conexiones del grafo para las posiciones de las matrices Z y X, y teniendo en cuenta las convenciones de numeraciones de qubits en Qiskit), y posteriormente sumándolos todos, para obtener el hamiltoniano.

Esta formulación es válida para la ejecución en ambos simuladores.

6.1.2.2. Annealer

Para el annealer, la formulación se ha hecho a través de los coeficientes J del hamiltoniano de Ising, a partir de los cual dimod permite construir un BinaryQuadraticModel (que se puede usar para guardar un problema formulado en QUBO), junto con los valores de h, el spin y el offset.

6.1.3. Ejecución del problema

6.1.3.1. QAOA – Simulador local y simulador remoto

Tras formular el problema, se establecen las seed o semillas, que posteriormente se utilizarán para los simuladores y para la generación de números aleatorios de numpy pasándolas como parámetro (gracias a las seed podemos conseguir que los resultados sean reproducibles). Después, se hace uso de la clase QuantumInstance, que nos permite especificar y controlar cómo se ejecutan los algoritmos cuánticos en Qiskit.

En la creación de la instancia especificamos como backend "aer_simulator", que es el simulador local de QAOA que nos ofrece Qiskit, el número de shots o muestras, que es el número de veces que se ejecuta el circuito y se obtienen los resultados (en el caso de este estudio se han utilizado 1024 para cada instancia, que es un valor que permite

obtener bastante precisión), y la seed_simulator y seed_transpiler, cuyo valor se ha definido previamente.

A continuación, creamos una instancia de la clase QAOA(), también proporcionada por Qiskit, a la cual le pasamos la QuantumInstance creada previamente, el número de repeticiones (p), el optimizador a utilizar (en el estudio se ha utilizado COBYLA), y el initial_point, que para el estudio se ha establecido en un valor aleatorio (pero definido por la seed), entre 0 y 2*π, en contrapartida a usar un número fijo para todas las ejecuciones, para explorar un rango mayor de valores.

Con el entorno ya configurado, se procede a ejecutar el algoritmo, con compute_minimum_eigenvalue(). Esto nos devolverá un resultado con bastante información, como por ejemplo el eigenstate (y también $|\beta^*,\gamma^*\rangle$), que podemos usar para calcular las probabilidades con las que cada estado ha sucedido (que sale en base a las muestras ejecutadas), y con ello obtener el porcentaje de soluciones óptimas de cada ejecución.

Una vez tenemos el hamiltoniano y el eigenstate, podemos recorrer los valores de este último y probar a calcular los valores esperados de cada uno de los estados del vector, para ver la energía que se obtiene, y con ello poder valorar la calidad de la solución.

Para la ejecución con el simulador remoto, el proceso es exactamente el mismo, a excepción de que a la hora de seleccionar el backend, en lugar de ser "aer_simulator", indicaremos "ibmq_qasm_simulator".

6.1.3.2. Annealer

Para resolver el problema utilizando el annealer el procedimiento es bastante simple. Primero construimos el muestreador utilizando la clase EmbeddingComposite, y pasándole como parámetro un DWaveSampler, que es el backend que se va a utilizar (la infraestructura en la que se resuelve la tarea, que en este caso es real, pero podría tratarse de un algoritmo clásico también).

La clase EmbeddingComposite se ocupa de realizar el mapeado a los qubits físicos del annealer, es decir, de asignar los qubits que utilizamos en nuestro problema QUBO para representar variables y restricciones a los qubits reales con los que cuenta el dispositivo.

Este es un proceso importante ya que dependiendo de la topología que utilice el ordenador cuántico en sus chips, si no hacemos las asignaciones correctas puede que haya operaciones entre varios qubits que no sea posible realizar.

Para obtener los resultados, usaremos la funcionalidad sample, del muestreador, donde proporcionamos el problema formulado como se ha indicado en 6.1.2.2. Annealer, y el número de muestras que queremos obtener (en este estudio se han utilizado 100 muestras, para evitar excederse del tiempo límite mensual de D-Wave).

Cada muestra contiene el valor de energía asociado, por lo que puede usar para determinar si una solución es óptima o no. Para determinar esto último, se hace uso de la clase ExactSolver, que encuentra la solución óptima a problemas QUBO o Ising a través de una búsqueda exhaustiva, en la que evalúa todas las posibles combinaciones de valores para las variables.

Tras haber obtenido todas las soluciones, se utiliza el valor de energía de las mejores, y posteriormente se compara con la energía obtenida en la solución del annealer.

6.2. Problema del viajante de comercio (TSP)

6.2.1. Datos utilizados

Para este problema se han considerado un total de 10 grafos para cada número de vértices, que en este caso ha sido menor que en el problema del Max-Cut, al tratarse de un problema más complejo, que hace uso de más recursos. Se han realizado ejecuciones con 3 y 4 vértices, siendo 5 demasiados para simular de forma local o remota con el QAOA.

6.2.2. Formulación del problema

6.2.2.1. QAOA – Simulador local y simulador remoto

Para formular este problema, se han seguido una serie de pasos, que se comparten en varios de los demás problemas que se estudian en el trabajo, con el objetivo de hacer el proceso y las condiciones lo más similares posible.

Lo primero que se ha hecho es construir el modelo del problema, para lo cual se ha elegido utilizar DOcplex (que es una librería de Python ofrecida por IBM pensada para la formulación y resolución de problemas de optimización), debido a la comodidad que ofrece para construir este tipo de funciones.

Para construir dicho modelo, primero se han creado las variables binarias que se van a utilizar en la expresión, que representarán el orden en el que se han visitado las ciudades, y posteriormente se han creado y añadido al modelo las dos restricciones con las que trabajamos en el problema, todo esto por medio de los métodos que ofrece la librería de DOcplex.

A continuación, se ha construido la función a minimizar, haciendo uso del número de nodos del grafo, y de las conexiones entre ellos (que también incluyen el peso), y con ello el modelo estaría completo.

Con el modelo de DOcplex completado, lo siguiente que tendríamos que hacer para poder usar este modelo con los métodos de igual manera que con el Max-Cut sería convertirlo en un QuadraticProgram.

Para esto, la librería de Qiskit nos pone a disposición una serie de traductores y convertidores, entre los cuales hay uno que sirve para pasar de un modelo DOcplex a un QuadraticProgram (que es la clase que nos ofrece Qiskit para trabajar con problemas QUBO), que es justo lo que buscamos.

Una vez tenemos la instancia del QuadraticProgram, ya tendríamos la formulación preparada, y podríamos operar con normalidad para resolver el problema utilizando el QAOA, entre otros métodos.

6.2.2.2. Annealer

Para la formulación del annealer, podemos aprovechar el QuadraticProgram, el cual se ha explicado cómo puede ser obtenido en la sección anterior, y utilizar otro convertidor de los que nos proporciona Qiskit para pasar de QuadraticProgram a QUBO, y ya sería posible ejecutarlo en el annealer de D-Wave.

6.2.3. Ejecución del problema

6.2.3.1. QAOA – Simulador local y simulador remoto

Una vez formulado el problema, para ejecutarlo en el simulador local creamos una instancia de la clase QuantumInstance, que ya se utilizó previamente, y al crearla especificamos el backend "aer_simulator" (para ejecutar el problema en el remoto sería, de nuevo, especificando el backend "ibmq_qasm_simulator"). También indicamos las seed para la aleatoriedad.

Posteriormente, creamos una instancia de la clase QAOA(), con los mismos parámetros que usamos a la hora de resolver el Max-Cut.

Para minimizar la función, esta vez en lugar de utilizar compute_minimum_eigenvalue() haremos uso de la clase MinimumEigenOptimizer. La razón de haber seleccionado esta clase en lugar del método compute_minimum_eigenvalue() es que ahora, a diferencia del Max-Cut, no estamos trabajando directamente con el hamiltoniano de Ising. Con el Max-Cut esto era posible porque se trataba de un problema de menor complejidad, pero para problemas cuya formulación no es tan simple la clase MinimumEigenOptimizer puede ayudar a encapsular y facilitar el proceso.

Resolvemos el problema (midiendo el tiempo que tarda en hacerlo), y después guardamos en un diccionario cada uno de los resultados obtenidos (los 1024 shots), con el número de veces que se ha obtenido, el valor, la asignación de valores y el estado de la solución (si es factible o no).

6.2.3.2. Annealer

Al igual que para el Max-Cut, ejecutar la resolución del problema utilizando el annealer es una tarea más sencilla que con el QAOA.

De hecho, la única diferencia es el modelo que se utiliza, ya que, al igual que para el Max-Cut, hacemos uso de la clase EmbeddingComposite, del DWaveSampler y ejecutamos el método sample. El número de muestras también es 100.

6.3. Problema de la mochila (Knapsack problem)

6.3.1. Datos utilizados

En este problema se han utilizado instancias de 3 a 5 vértices, 15 grafos para cada tamaño de vértices. Como en el TSP, se nota el aumento en complejidad del problema, ya que al subir el número de vértices se necesitan también más variables, y por tanto más qubits.

Actualmente, el número de qubits que se pueden simular está bastante limitado. Alrededor de 30 qubits, las ejecuciones comienzan a tomar del orden de horas en completarse. Esto puede variar en función del equipo utilizado para hacer el experimento.

6.3.2. Formulación del problema

6.3.2.1. QAOA – Simulador local y simulador remoto

Para la formulación de este problema se han seguido los mismos pasos que para la formulación del TSP. Primero se ha construido el modelo DOcplex, usando la función y las restricciones descritas en 5.2.4.2. Formulación, y después se ha convertido este modelo DOcplex en un QuadraticProgram usando la funcionalidad proporcionada por Qiskit.

6.3.2.2. Annealer

Usando el QuadraticProgram obtenido previamente, se utiliza la clase QuadraticProgramToQubo para pasarlo a una formulación válida que pueda utilizar el annealer.

6.3.3. Ejecución del problema

6.3.3.1. QAOA – Simulador local y simulador remoto

La ejecución con el QAOA en ambos simuladores es igual que la descrita en 6.2.3.1. QAOA – Simulador local y simulador remoto.

6.3.3.2. Annealer

La ejecución con el annealer es igual que la descrita en 6.2.3.2. Annealer.

6.4. Problema del coloreado de grafos (Graph coloring)

6.4.1. Datos utilizados

En este problema se han utilizado instancias de 3 a 5 vértices, 15 grafos por cada número de vértices. De nuevo, se han probado numerosos números de vértices hasta dar con la cifra hasta la que las ejecuciones comenzaban a tardar un periodo de tiempo demasiado elevado.

6.4.2. Formulación del problema

6.4.2.1. QAOA – Simulador local y simulador remoto

La formulación se ha hecho de la misma manera que la ilustrada en 6.2.2.1. QAOA – Simulador local y simulador remoto, cambiando la función y las restricciones para que se ajusten a la formulación explicada en 5.2.5.2. Formulación.

6.4.2.2. Annealer

La formulación para el annealer se ha hecho de la misma forma que en 6.2.2.2. Annealer.

6.4.3. Ejecución del problema

6.4.3.1. QAOA – Simulador local y simulador remoto

La ejecución con el QAOA para los dos simuladores es igual que la descrita en 6.2.3.1. QAOA – Simulador local y simulador remoto.

6.4.3.2. Annealer

La ejecución con el annealer es igual que la descrita en 6.2.3.2. Annealer.

Capítulo 7

Resultados obtenidos

En este apartado vamos a exponer y analizar los resultados que hemos obtenido en cada problema y para cada método utilizado en el estudio. Los puntos que se van a estudiar de cada resultado son

1. Porcentaje de solución óptima
2. Porcentaje de solución válida (la solución cumple las restricciones impuestas)
3. Energía media
4. Tiempo de ejecución (si aplica al método)

Además de los resultados de cada muestra, también se han guardado los datos relativos a cada grafo con cada método, para así poder incluir valores de desviación típica y hacer el estudio más completo.

En general, para cada problema primero se van a presentar los resultados del QAOA local, con las métricas que correspondan. A continuación se mostrarán los resultados del simulador remoto, comparándolos con los del local, y, por último, los del annealer, también estableciendo comparaciones entre sus resultados y los del QAOA local.

7.1. Resultados obtenidos

7.1.1. Corte máximo (Max-Cut)

En este caso, cabe mencionar que todas las soluciones cumplen las restricciones, por la naturaleza de la formulación usada.

7.1.1.1. QAOA – Simulador local

Este método se ha probado con distintas configuraciones de reps (p), para poder observar también cómo influye este parámetro a la efectividad y al rendimiento.

En total se han obtenido 711772 muestras, y para cada una de ellas se ha guardado si se trata de una solución óptima, si cumple las restricciones, la energía de la solución obtenida, la energía de la solución óptima, y el tiempo de ejecución (para comparar el rendimiento con distintos valores de p).

En la siguiente gráfica se muestra un resumen del porcentaje de soluciones óptimas obtenidas con grafos con distintos números de vértices y usando distintos valores de p (reps). Estos valores cuentan con una desviación estándar que aporta más información sobre la fluctuación de los datos.

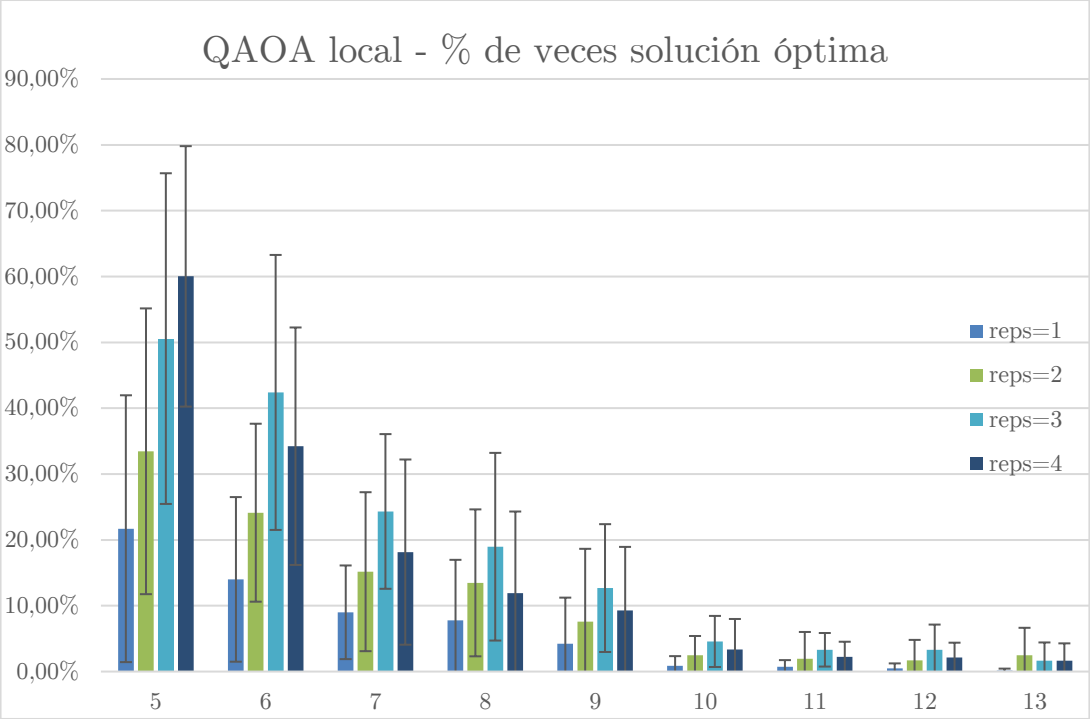


Figura 7.1. QAOA local: porcentaje de soluciones óptimas (Max-Cut)

En esta otra gráfica se muestra la comparativa entre la energía media obtenida y la óptima, lo cual aporta una idea sobre la calidad de las soluciones obtenidas (una energía media más cercana a la óptima indica que las soluciones son de mayor calidad, al tener un valor mejor en la función objetivo).

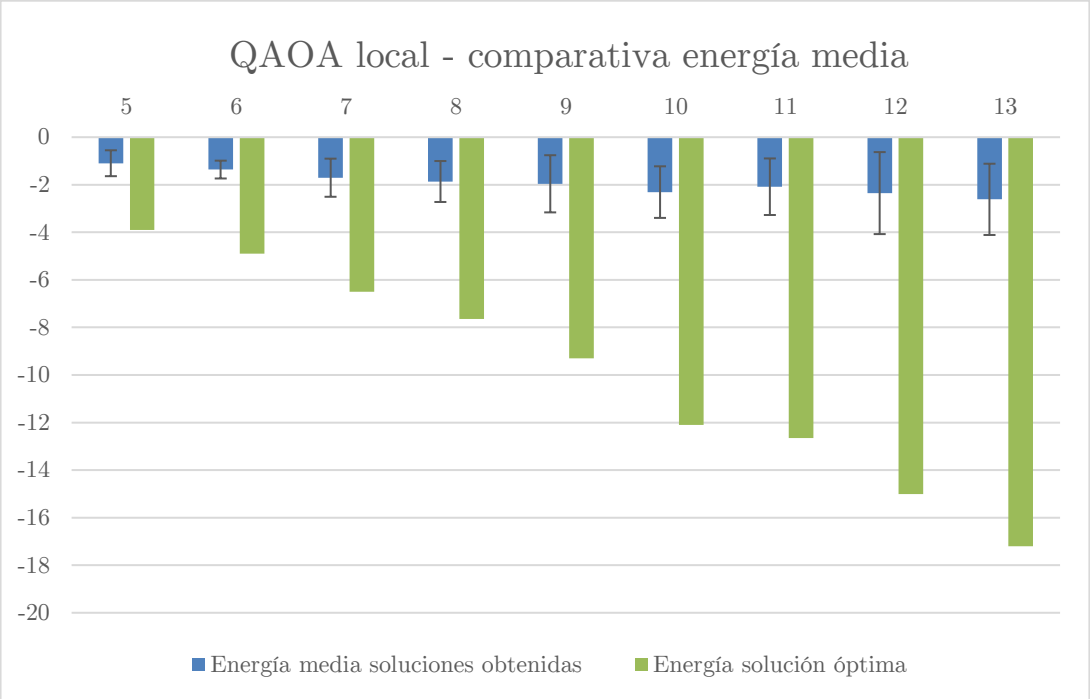


Figura 7.2. QAOA local: comparativa entre energía media y óptima (Max-Cut)

Por último, queda por ver la variación en el tiempo de ejecución para los distintos números de vértices y valores de p.

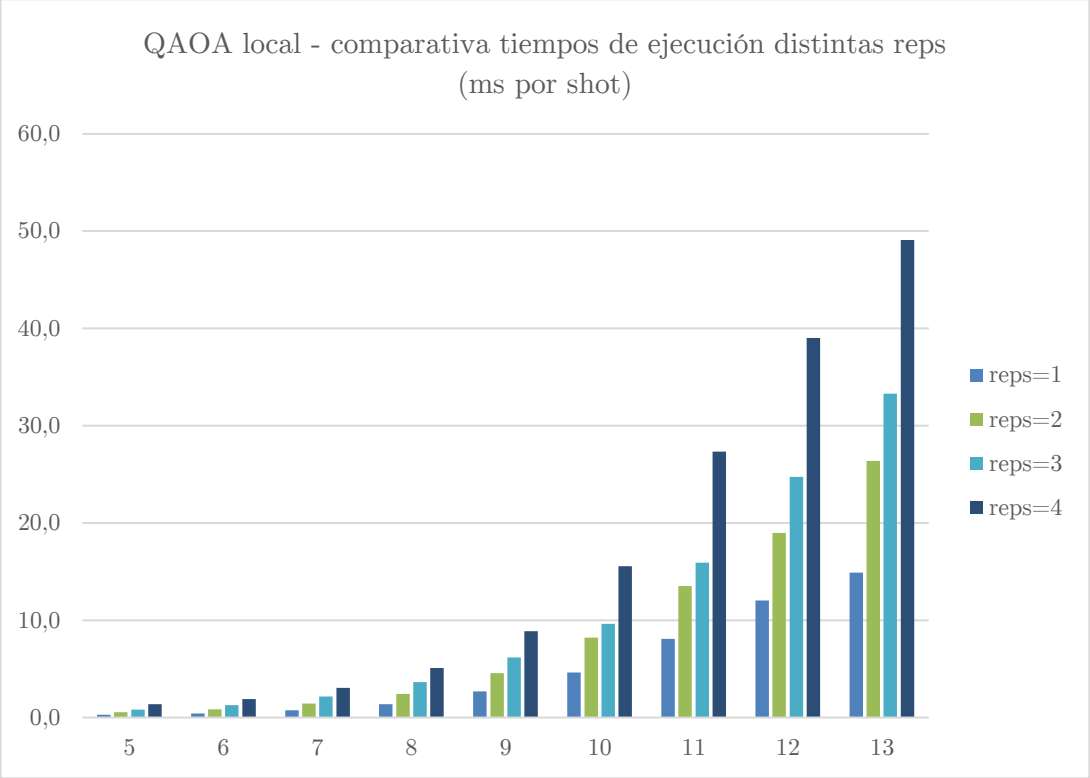


Figura 7.3. QAOA local: comparativa del tiempo de ejecución (Max-Cut)

7.1.1.2. QAOA – Simulador remoto

Tras haber presentado los resultados del simulador local, ahora se van a exponer los resultados obtenidos con el simulador remoto, mostrando también una comparativa entre ellos. En total con este método se han obtenido 711907 muestras.

Comenzamos con el porcentaje de soluciones óptimas obtenidas, donde los resultados del simulador local se muestran con una paleta de colores marrones, y los del simulador remoto con colores azules:

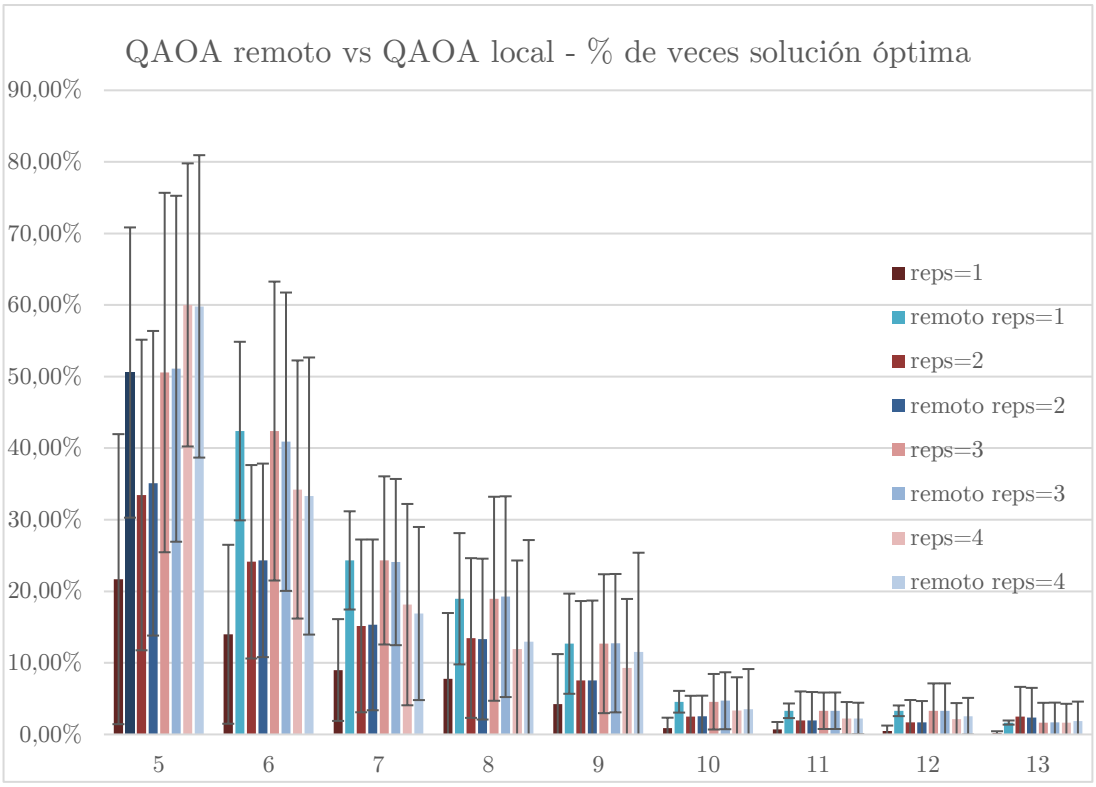


Figura 7.4. QAOA remoto: comparativa soluciones óptimas (Max-Cut)

A continuación se muestra una comparación de las energías medias obtenidas con el simulador local, con el remoto y con la energía media óptima (esta última en verde).

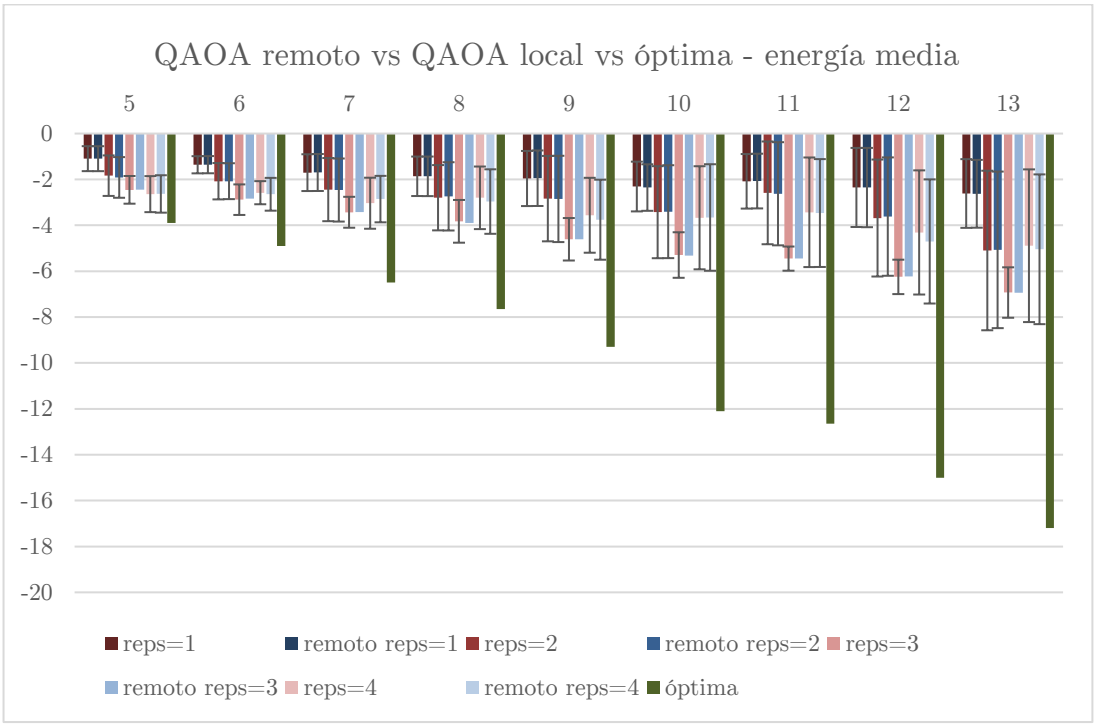


Figura 7.5. QAOA remoto: comparativa energía media (Max-Cut)

En el siguiente gráfico se puede ver la evolución de los tiempos de ejecución del simulador remoto, y la comparación con los del QAOA local.

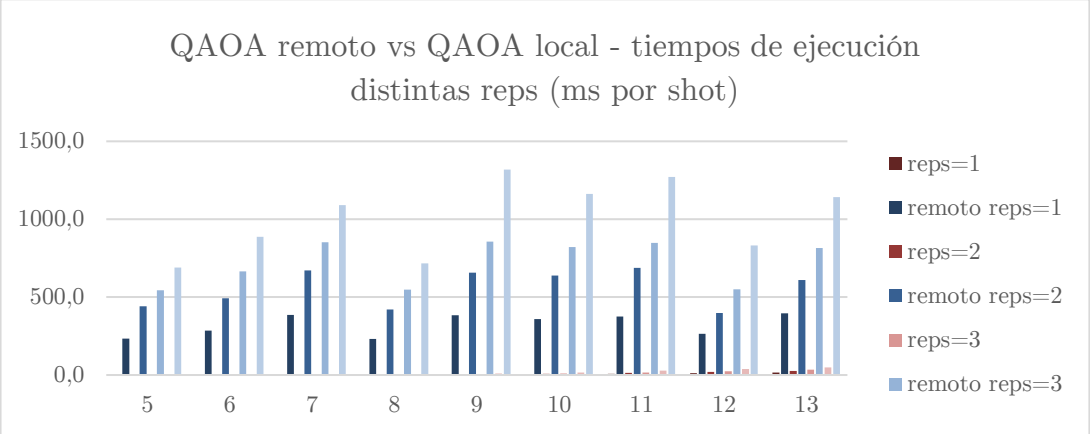


Figura 7.6. QAOA remoto: comparativa del tiempo de ejecución (Max-Cut)

7.1.1.2. Annealer

A continuación, se muestran los resultados obtenidos con el annealer. Primero una gráfica del porcentaje de solución óptima con distintos números de vértices (de 5 a 13), haciendo una comparación con los porcentajes obtenidos con el algoritmo QAOA local ($p = 1$). En total, se han obtenido 18000 muestras.

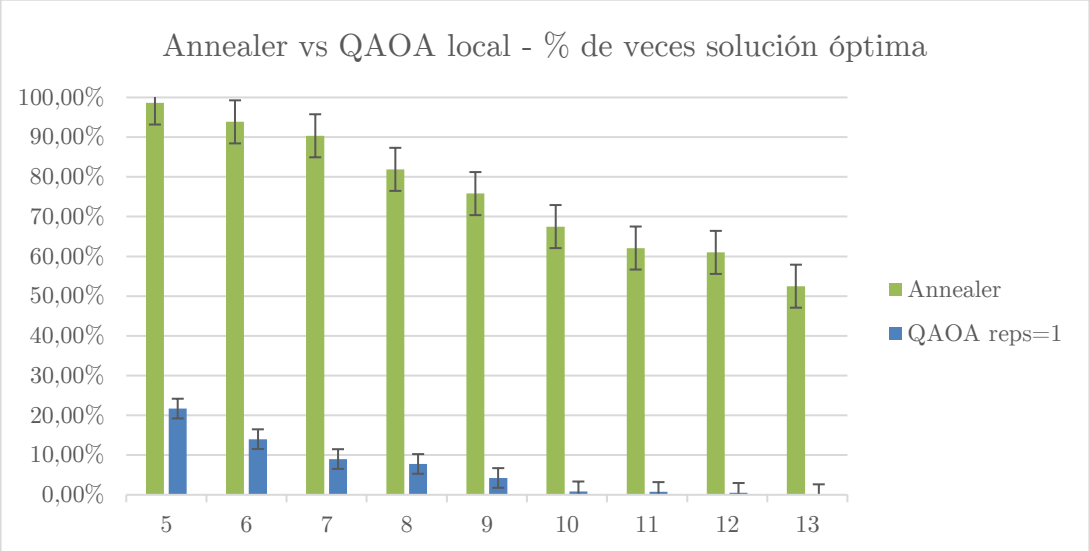


Figura 7.7. Annealer: comparativa de soluciones óptimas (Max-Cut)

Y nos falta por hablar de la energía media obtenida en las soluciones, que se puede observar en esta otra gráfica

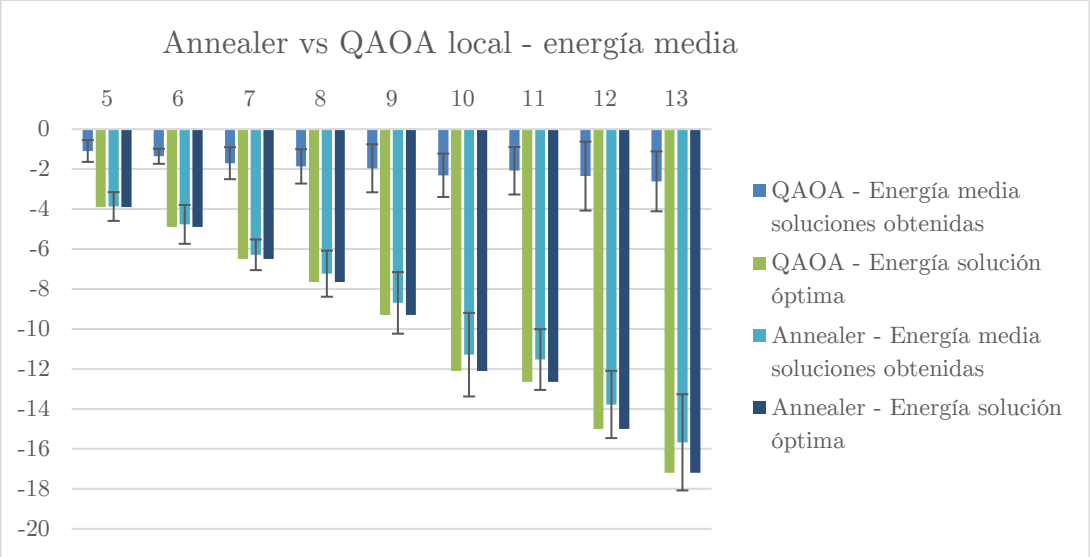


Figura 7.8. Annealer: comparativa de energía media (Max-Cut)

7.1.2. Problema del viajante de comercio (TSP)

A diferencia del Max-Cut, en este problema sí que tenemos que verificar que las soluciones sean válidas, ya que no todas van a cumplir este criterio. Se van a comprobar todos los resultados (cada shot del QAOA y cada muestra del annealer) para determinar si cumplen las restricciones.

7.1.2.1. QAOA – Simulador local

También se ha trabajado con distintos valores de p para evaluar el cambio que puede tener en los resultados (se ha probado hasta $p = 4$ en todos los vértices).

Se han obtenido 46080 muestras, siendo este un número bastante menor que el número de muestras obtenido con el Max-Cut debido a estar trabajando con un número mucho menor de grafos, y al no poder probar con tantos vértices.

A continuación, se muestran las gráficas detallando los resultados obtenidos. Primero una gráfica con el porcentaje de veces que se ha obtenido una solución óptima con cada p y para cada número de vértices (con cuatro vértices y $p = 1$, $p = 2$ y $p = 4$ no se obtuvo la solución óptima en ninguna de las ejecuciones).

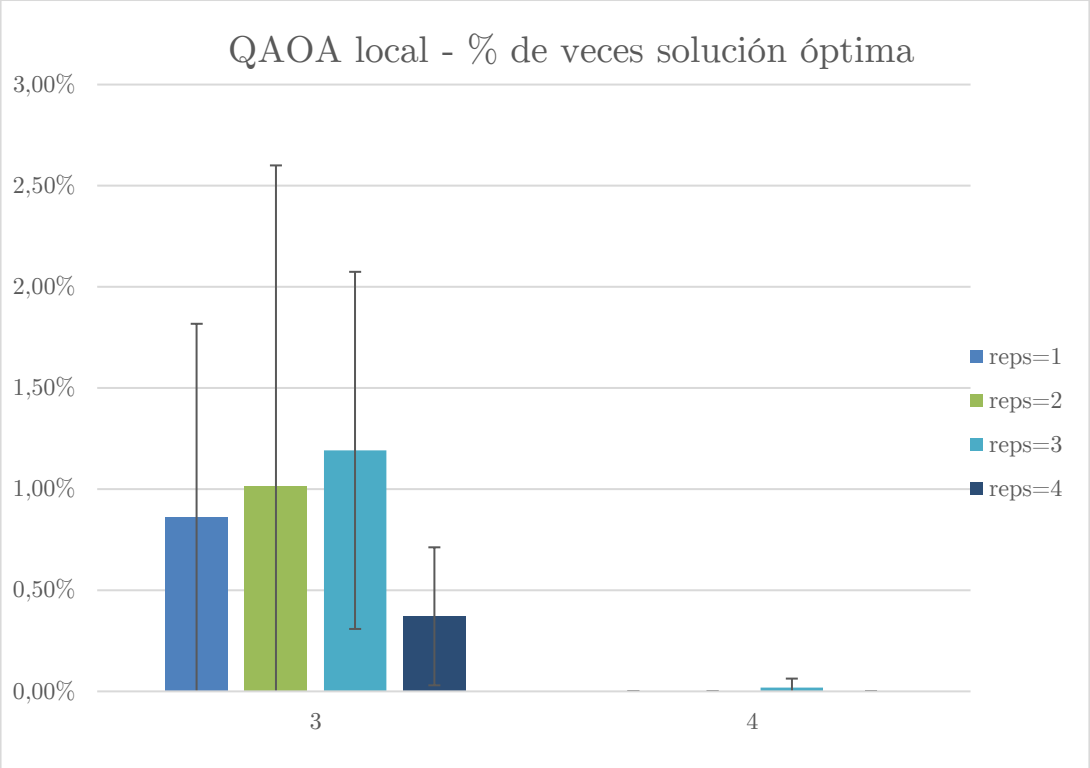


Figura 7.9. QAOA local: porcentaje de soluciones óptimas (TSP)

A continuación el porcentaje de soluciones válidas, es decir, que cumplen las restricciones, para los distintos valores de p

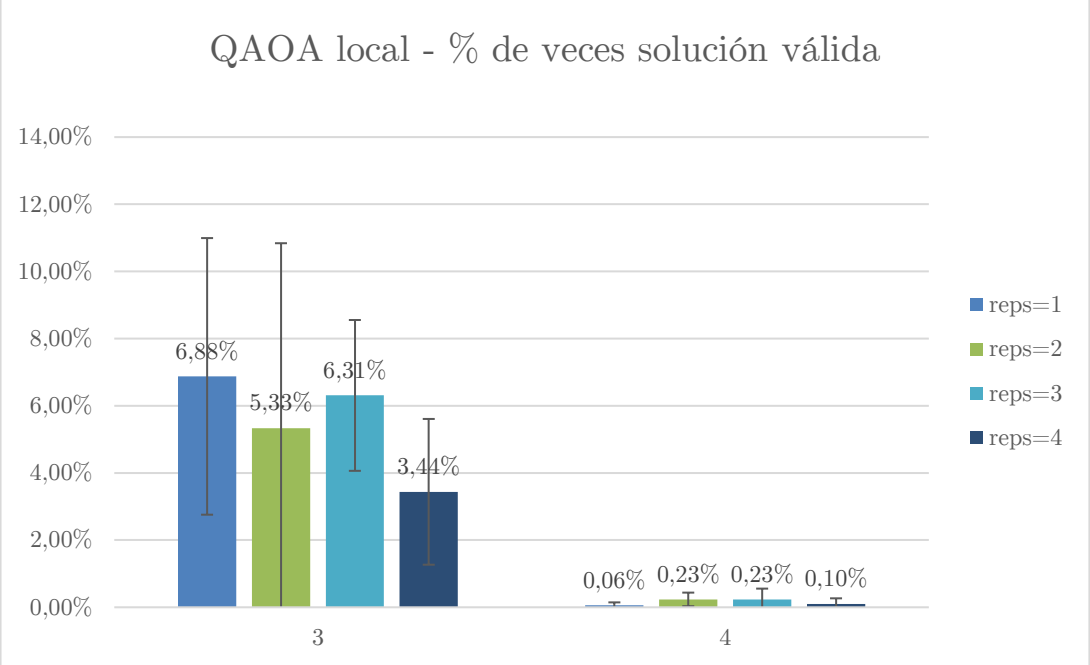


Figura 7.10. QAOA local: porcentaje de soluciones válidas (TSP)

Los distintos valores de energía media obtenidos

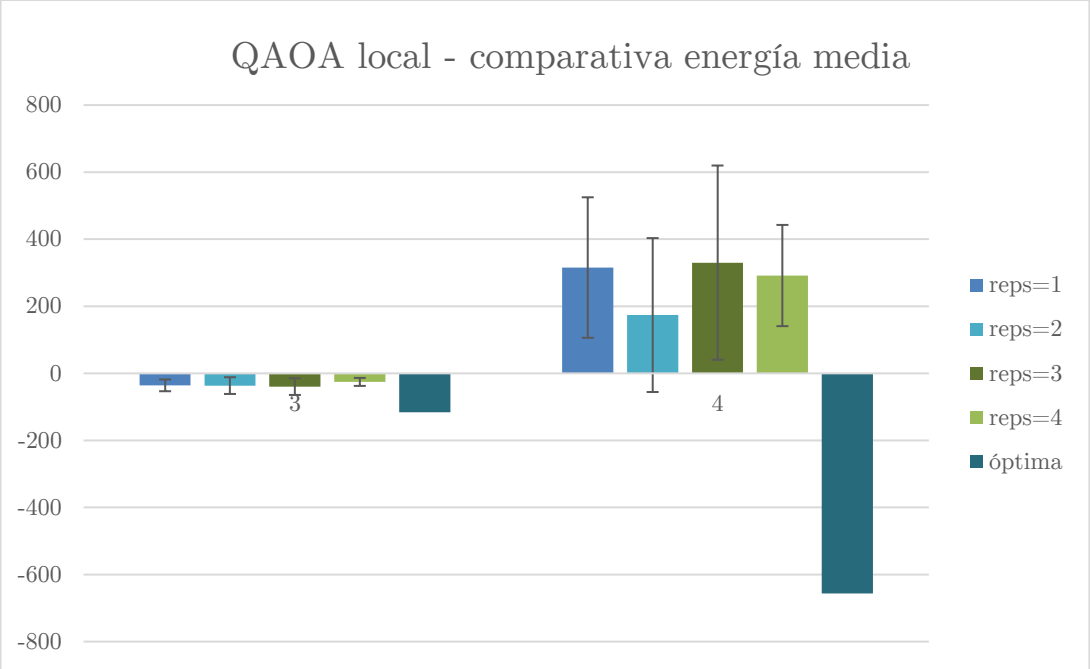


Figura 7.11. QAOA local: comparativa entre energía media y óptima (TSP)

Y los tiempos de ejecución

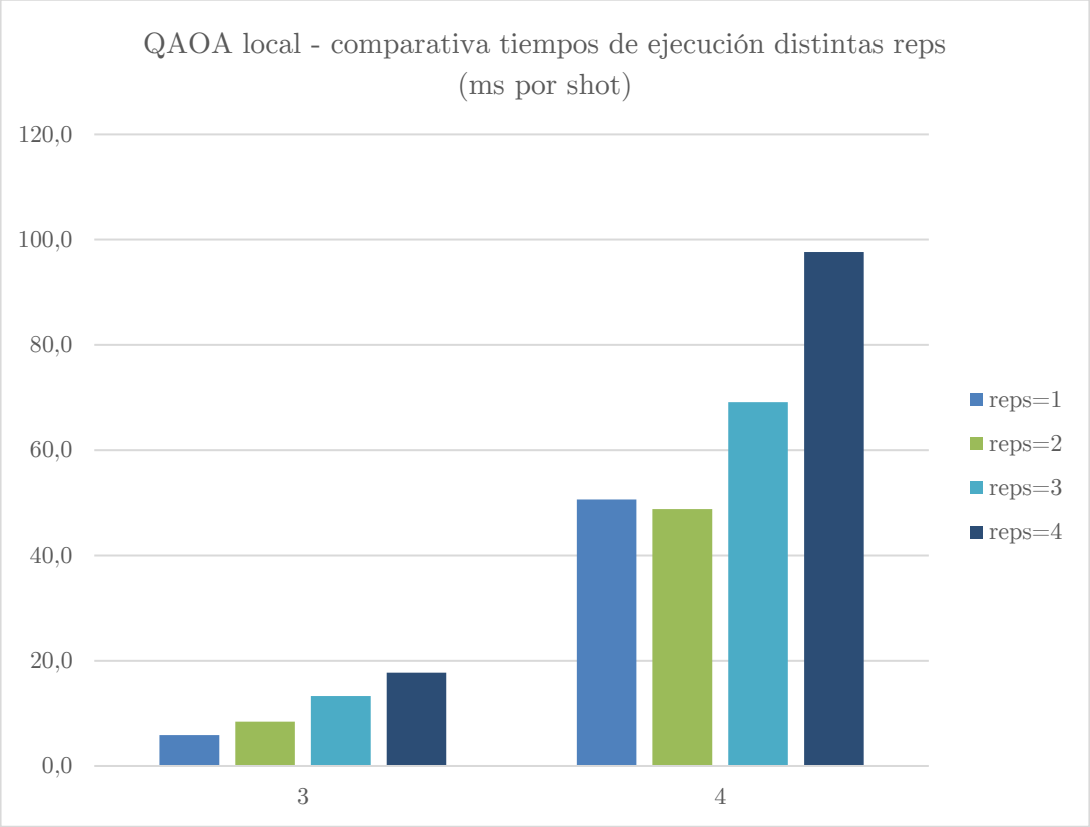


Figura 7.12. QAOA local: comparativa del tiempo de ejecución (TSP)

7.1.2.2. QAOA – Simulador remoto

Con el simulador remoto se han obtenido 40960 muestras. A continuación se muestran los resultados

Primero el porcentaje de soluciones óptimas, comparado con el simulador local.

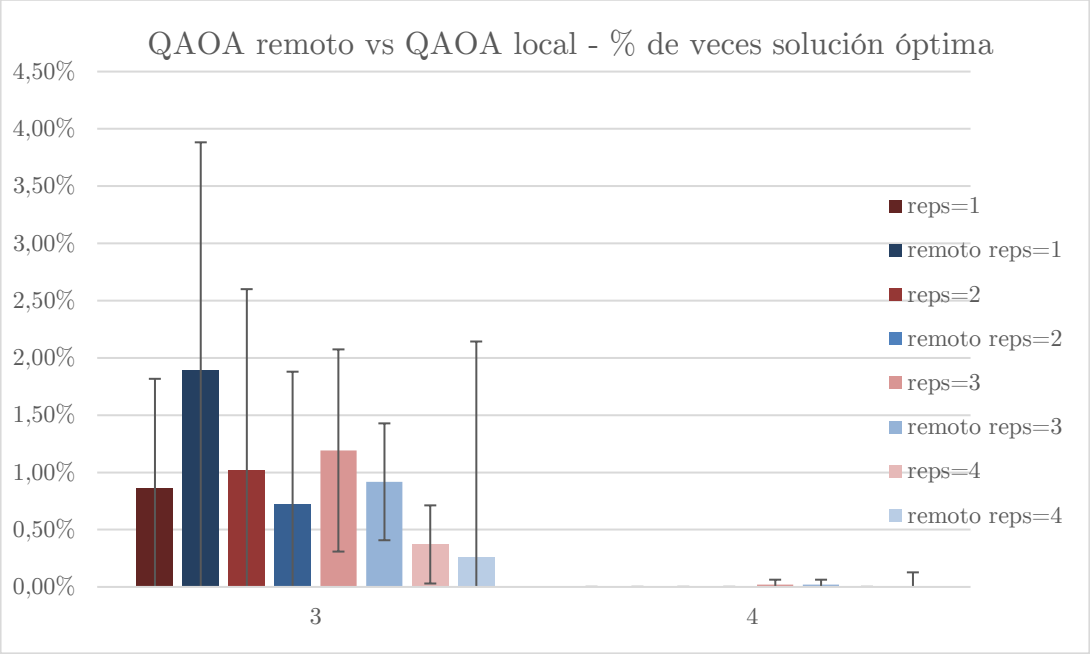


Figura 7.13. QAOA remoto: comparativa soluciones óptimas (TSP)

A continuación el porcentaje de soluciones válidas

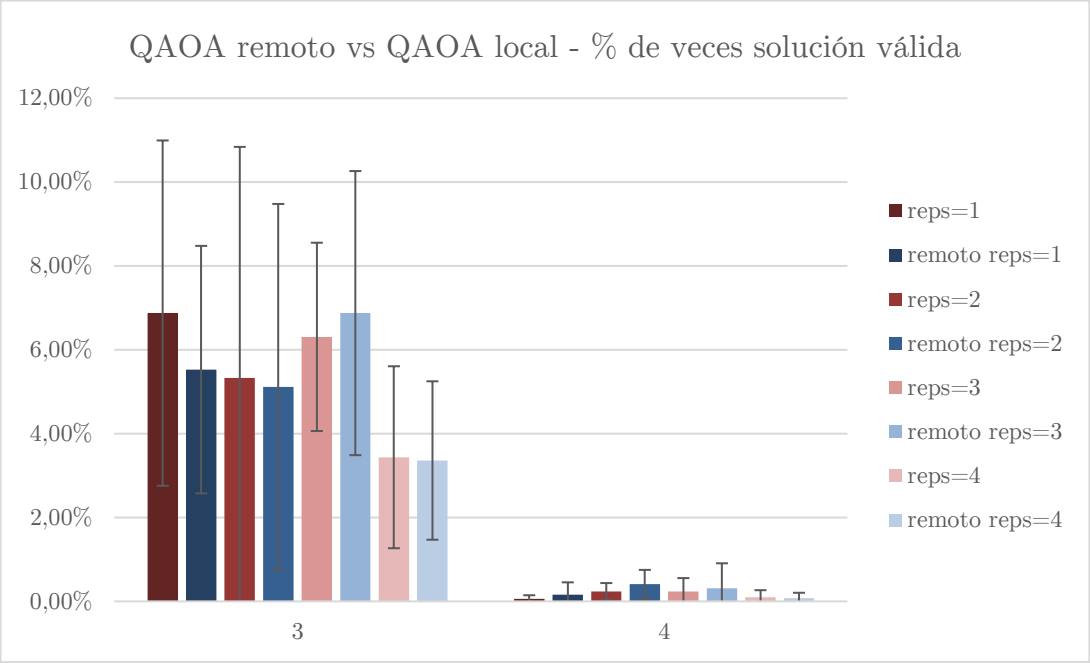


Figura 7.14. QAOA remoto: comparativa soluciones válidas (TSP)

La energía media

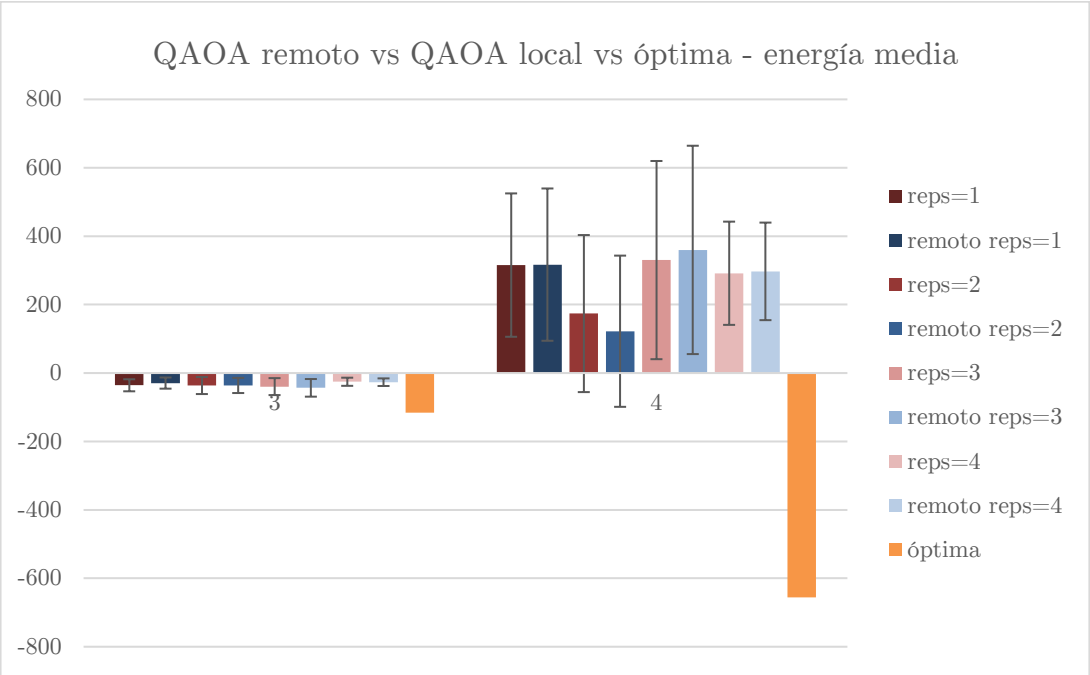


Figura 7.15. QAOA remoto: comparativa energía media (TSP)

Y los tiempos de ejecución

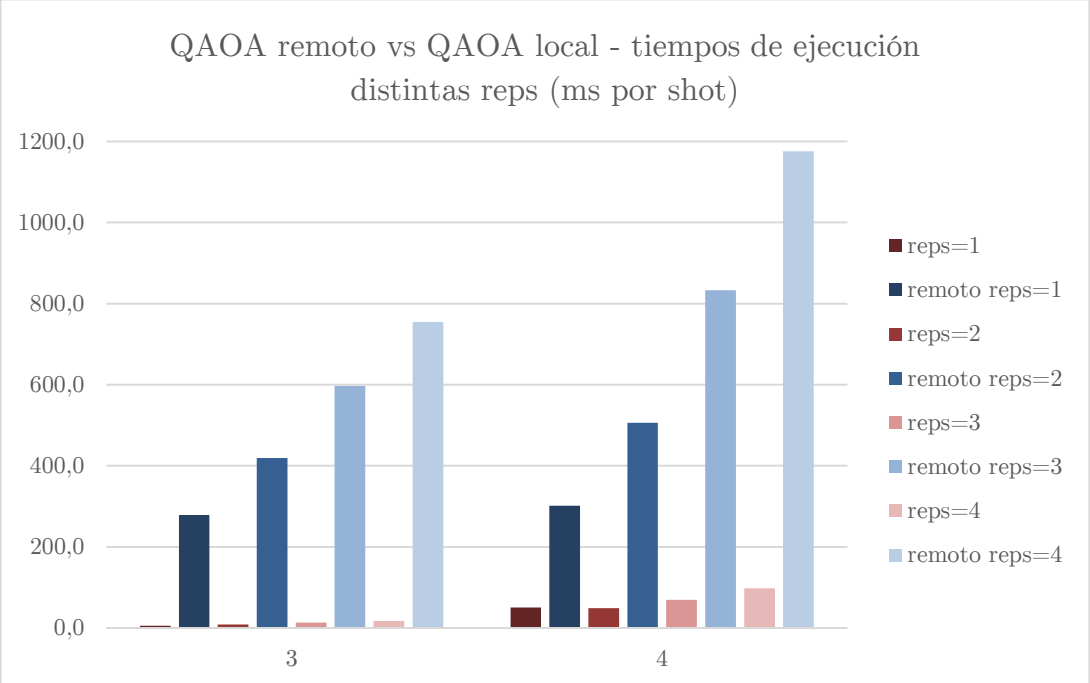


Figura 7.16. QAOA remoto: comparativa del tiempo de ejecución (TSP)

7.1.2.3. Annealer

En total para el annealer se han obtenido 1000 muestras. Se muestran los mismos gráficos que para el Max-Cut, añadiendo una comparativa del total de soluciones que se obtuvieron que cumplieran las restricciones (es decir, que fueran factibles).

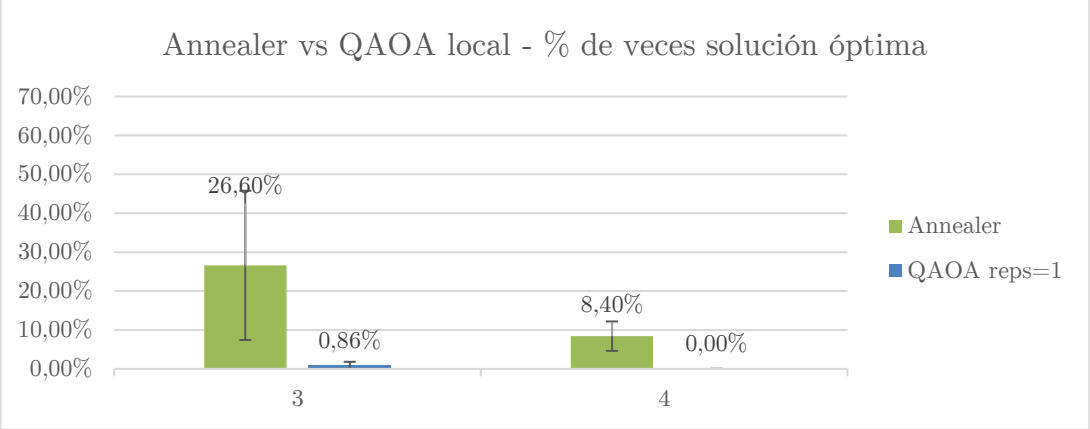


Figura 7.17. Annealer: comparativa de soluciones óptimas (TSP)

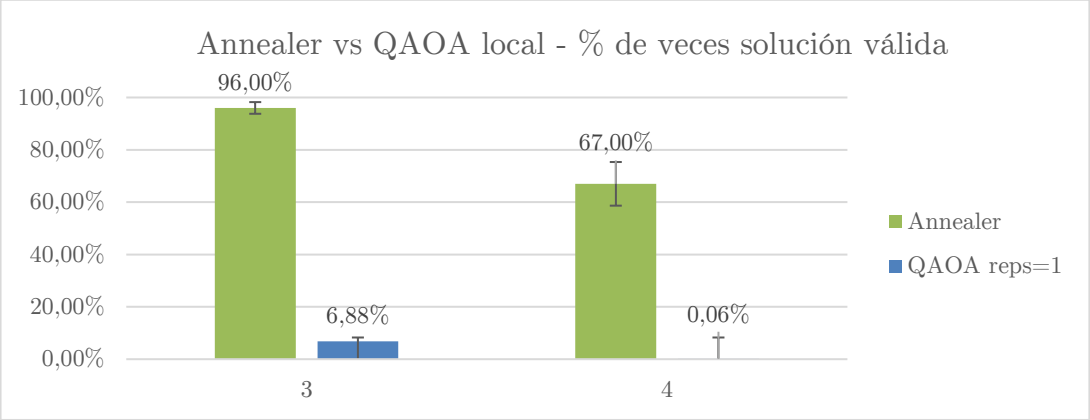


Figura 7.18. Annealer: comparativa de soluciones válidas (TSP)

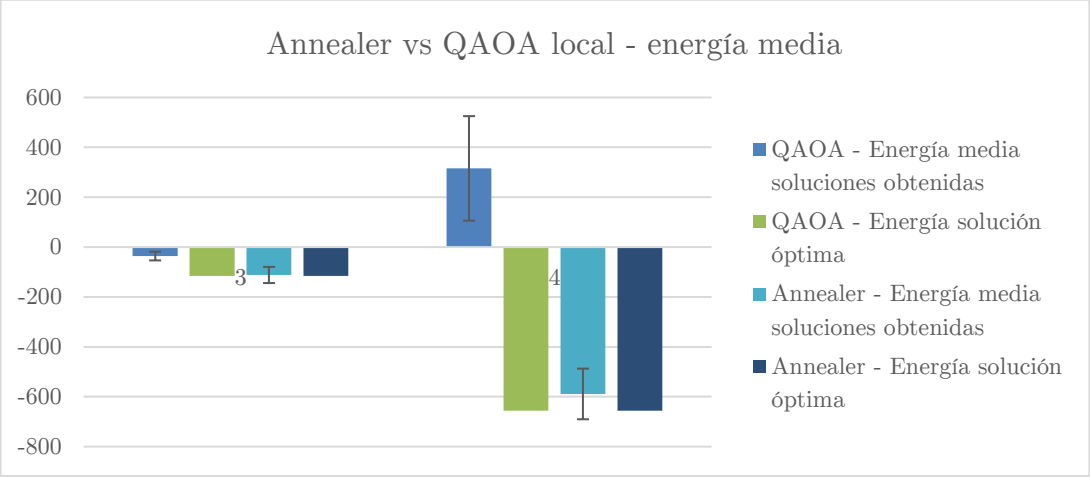


Figura 7.19. Annealer: comparativa de energía media (TSP)

7.1.3. Problema de la mochila (Knapsack problem)

En este problema se muestran los mismos gráficos y se van a analizar las mismas métricas que en 7.1.2. Problema del viajante de comercio (TSP), usando para la validez de la solución las restricciones propias del problema.

Se han probado de 3 a 5 vértices, 15 grafos cada uno, resultando en 45 en total.

7.1.3.1. QAOA – Simulador local

Se han usado valores de p desde uno a cuatro para todos los vértices. En total se han conseguido 184320 muestras del QAOA.

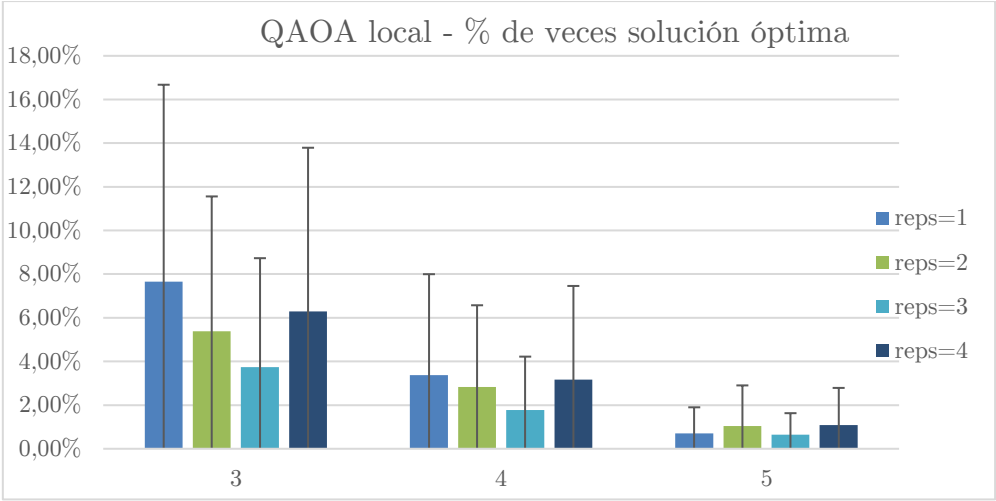


Figura 7.20. QAOA local: porcentaje de soluciones óptimas (Knapsack)

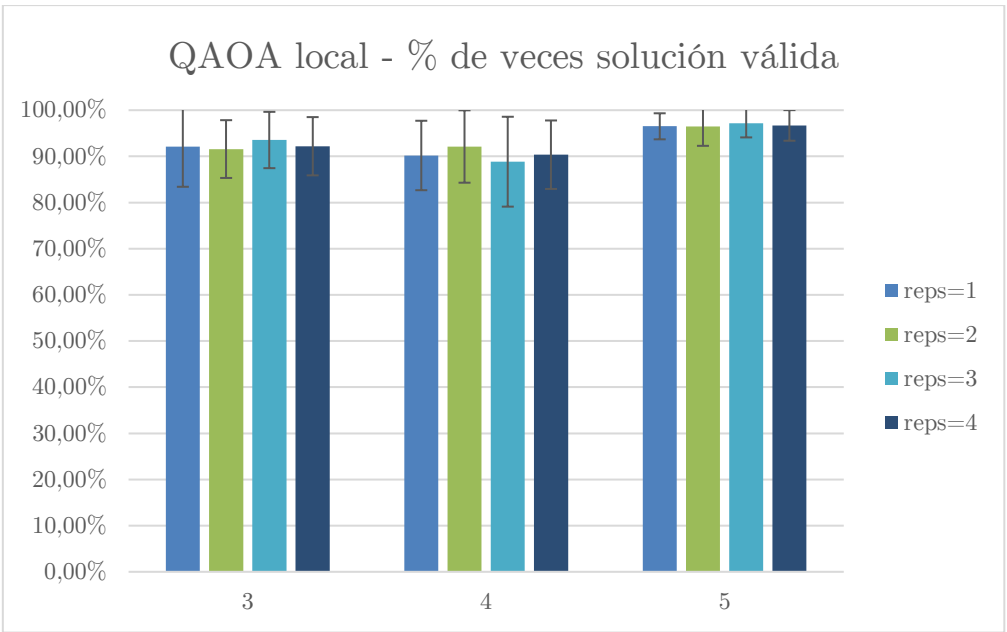


Figura 7.21. QAOA local: porcentaje de soluciones válidas (Knapsack)

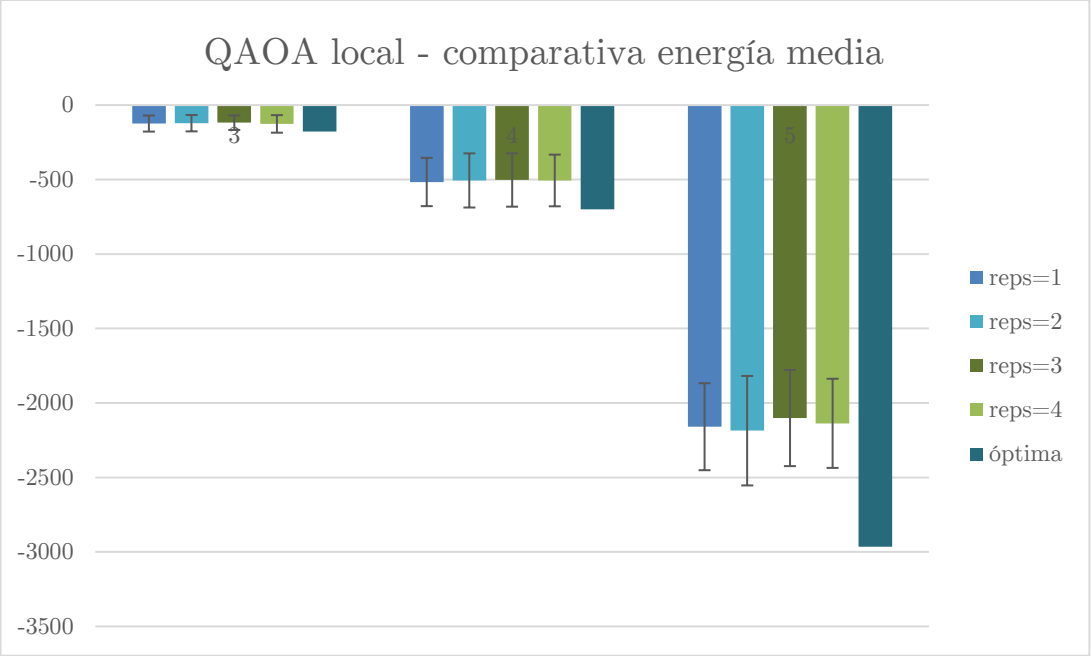


Figura 7.22. QAOA local: comparativa entre energía media y óptima (Knapsack)

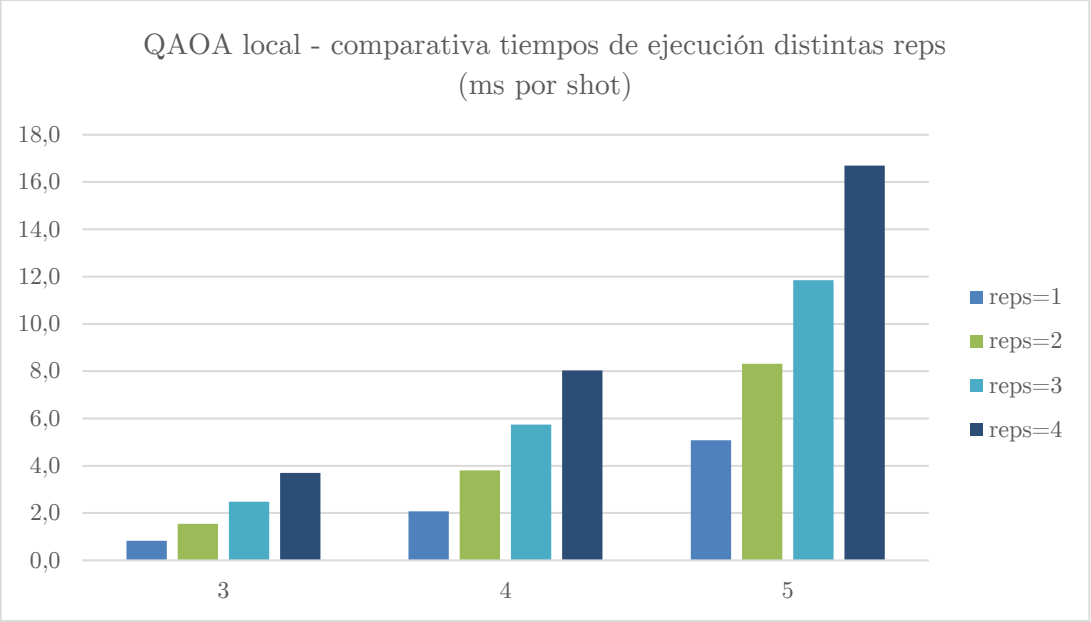


Figura 7.23. QAOA local: comparativa del tiempo de ejecución (Knapsack)

7.1.3.2. QAOA – Simulador remoto

El procesamiento de los problemas ha resultado en 184320 muestras para el simulador remoto.

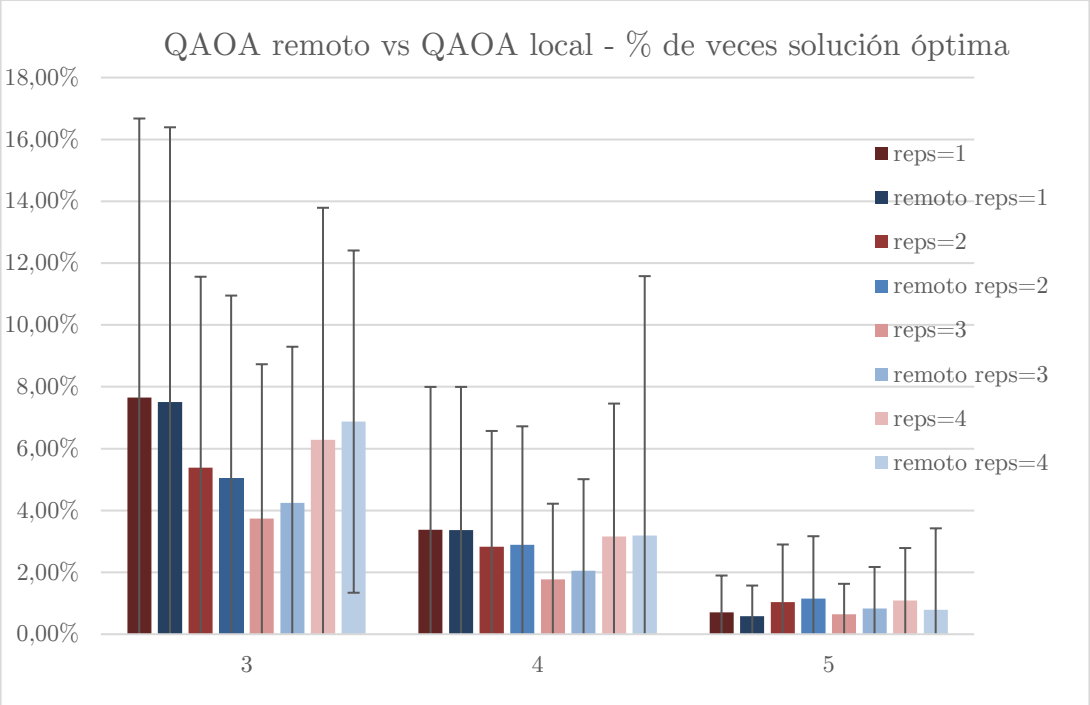


Figura 7.24. QAOA remoto: comparativa soluciones óptimas (Knapsack)

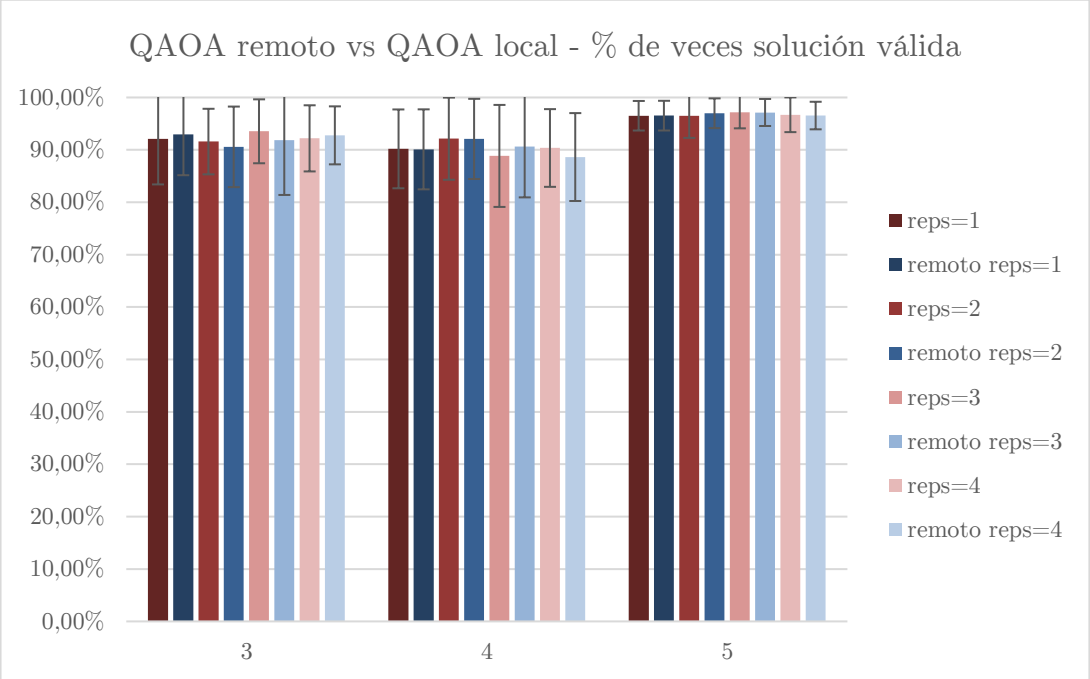


Figura 7.25. QAOA remoto: comparativa soluciones válidas (Knapsack)

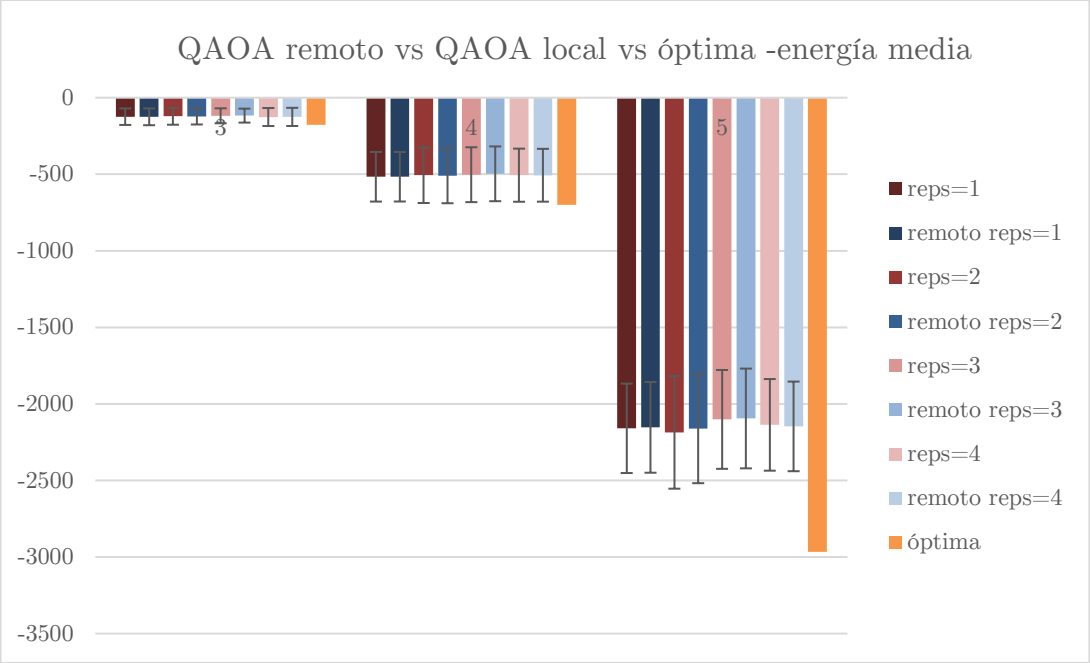


Figura 7.26. QAOA remoto: comparativa energía media (Knapsack)

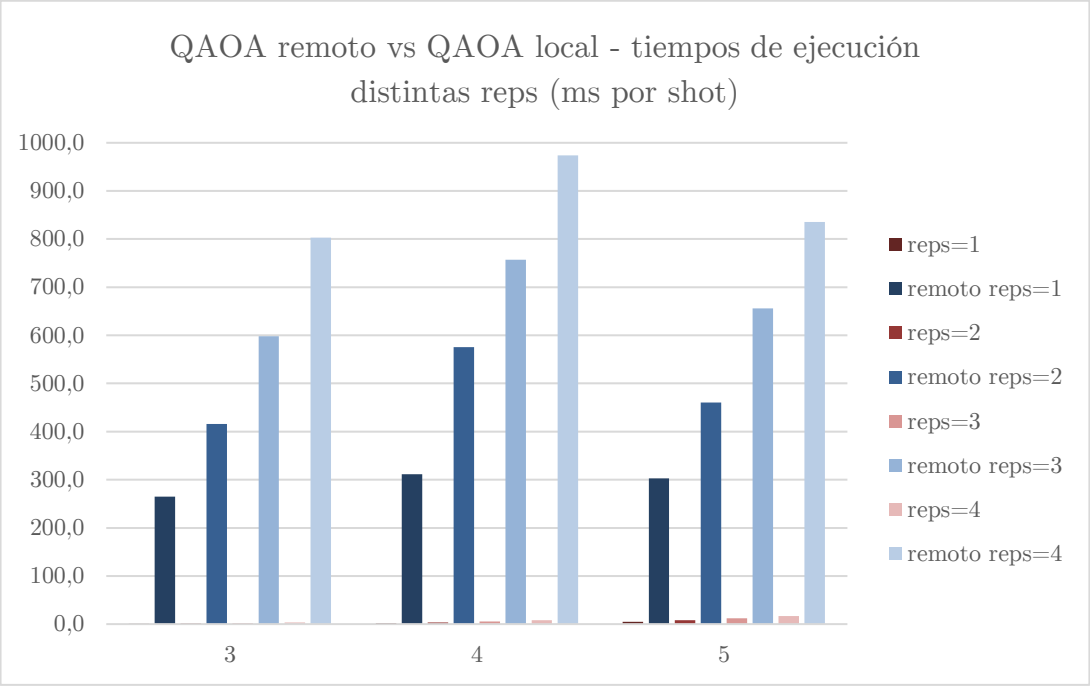


Figura 7.27. QAOA remoto: comparativa del tiempo de ejecución (Knapsack)

7.1.3.2. Annealer

Para este problema se han obtenido 4500 muestras, siendo estos los resultados que muestran los gráficos.

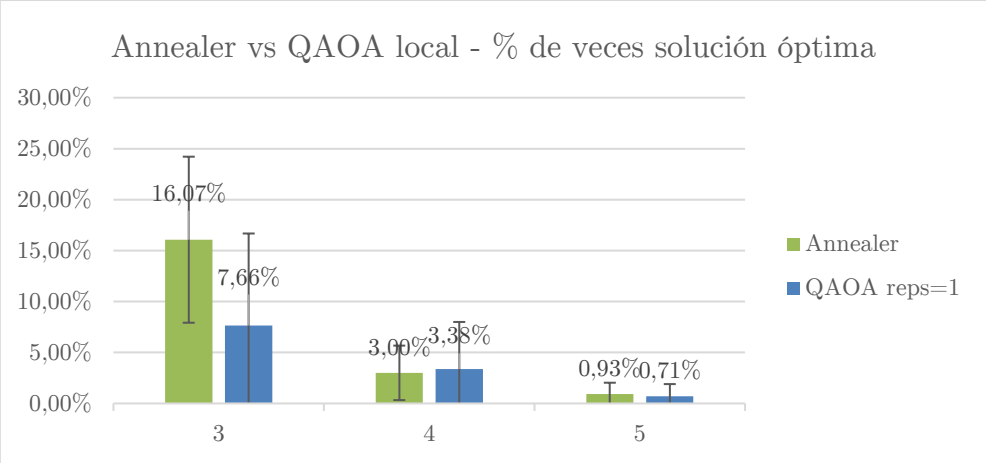


Figura 7.28. Annealer: comparativa de soluciones óptimas (Knapsack)

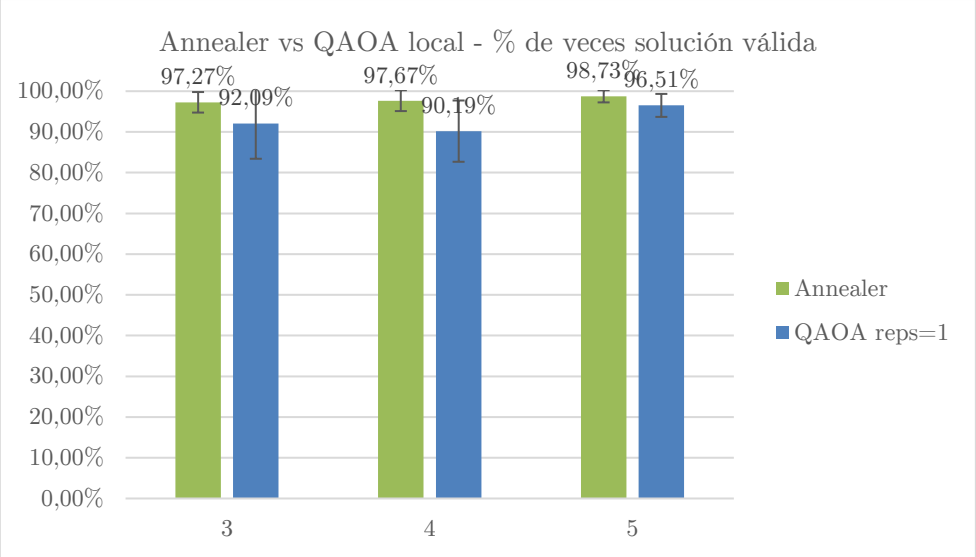


Figura 7.29. Annealer: comparativa de soluciones válidas (Knapsack)

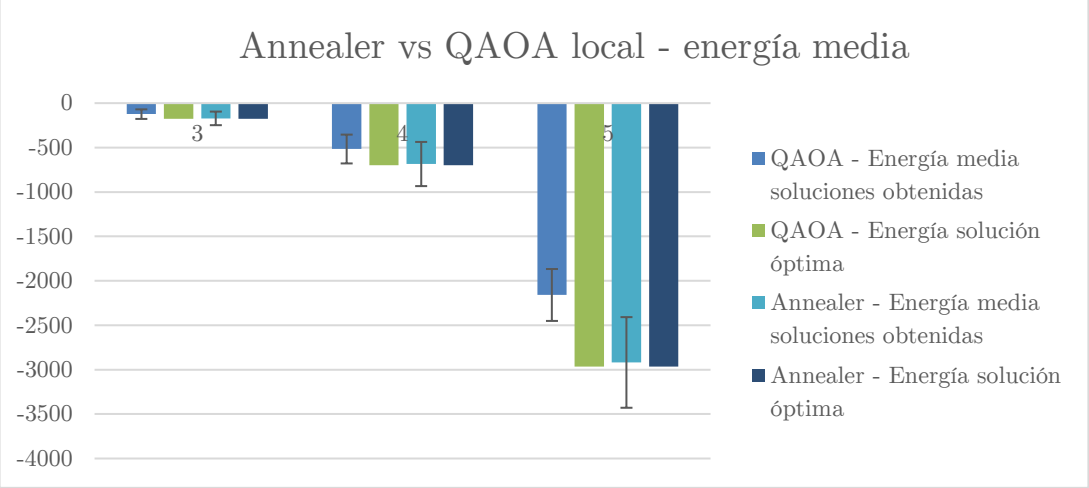


Figura 7.30. Annealer: comparativa de energía media (Knapsack)

7.1.4. Problema del coloreado de grafos (Graph coloring)

Los resultados de este problema se muestran utilizando los mismos gráficos y métricas que en el TSP y el problema de la mochila. Los datos van de 3 a 5 vértices, y en cada vértice se han probado 10 grafos, por lo que son 30 en total.

7.1.4.1. QAOA – Simulador local

Se han usado valores de p desde uno a cuatro para todos los vértices. Para el QAOA hay un total de 122880 muestras.

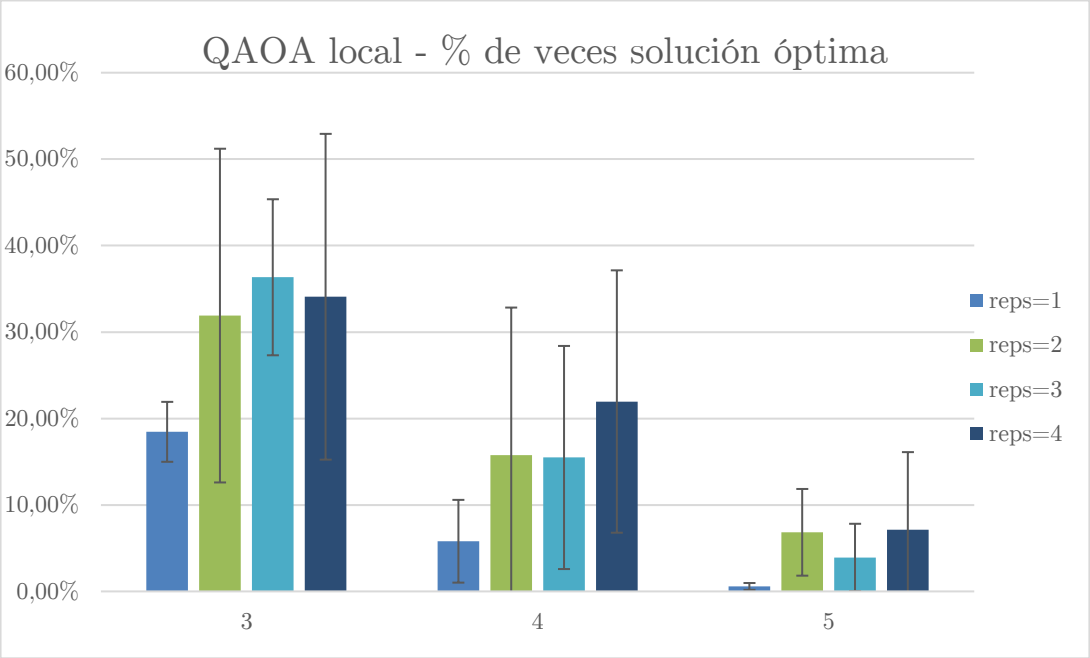


Figura 7.31. QAOA local: porcentaje de soluciones óptimas (Graph C.)

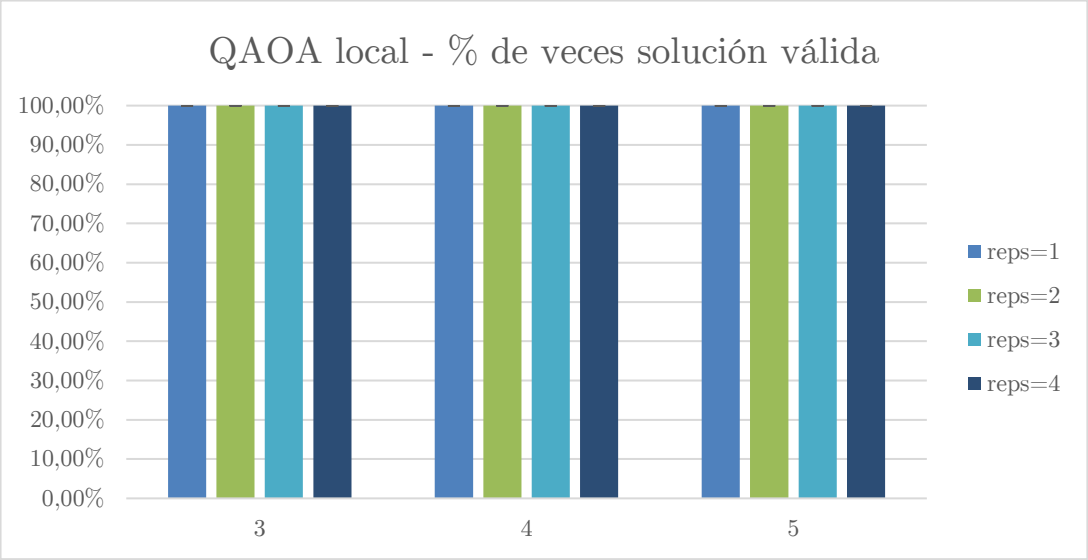


Figura 7.32. QAOA local: porcentaje de soluciones válidas (Graph C.)

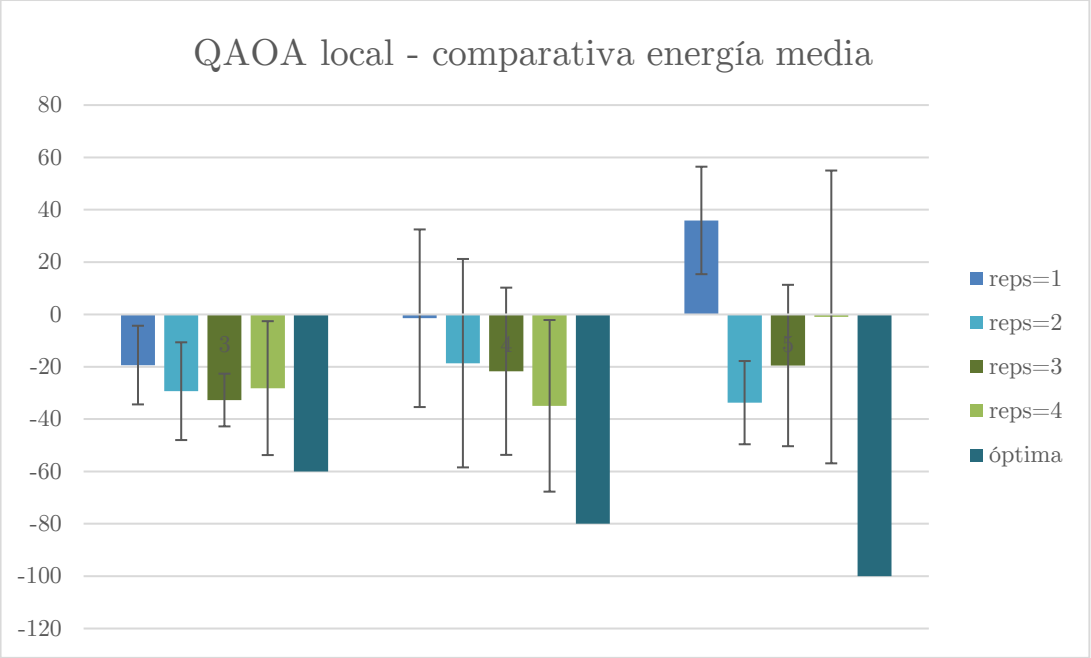


Figura 7.33. QAOA local: comparativa entre energía media y óptima (Graph C.)

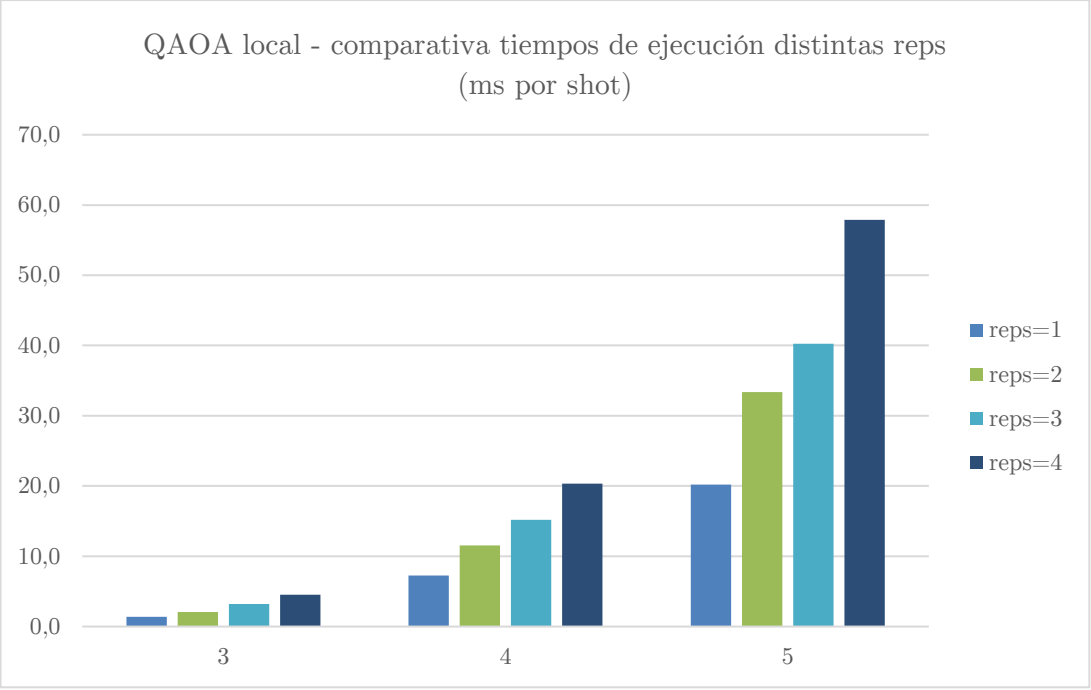


Figura 7.34. QAOA local: comparativa del tiempo de ejecución (Graph C.)

7.1.4.2. QAOA – Simulador remoto

A continuación se muestran los gráficos, que se han hecho en base a las 122880 muestras obtenidas.

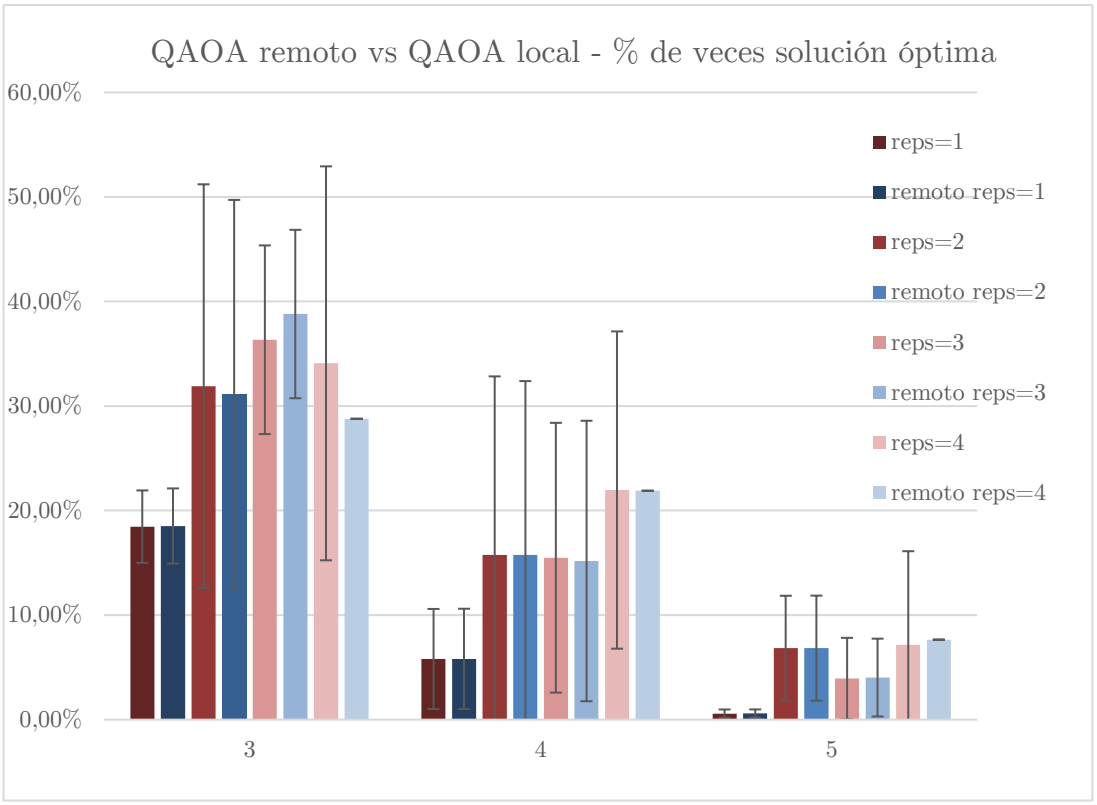


Figura 7.35. QAOA remoto: comparativa soluciones óptimas (Graph C.)

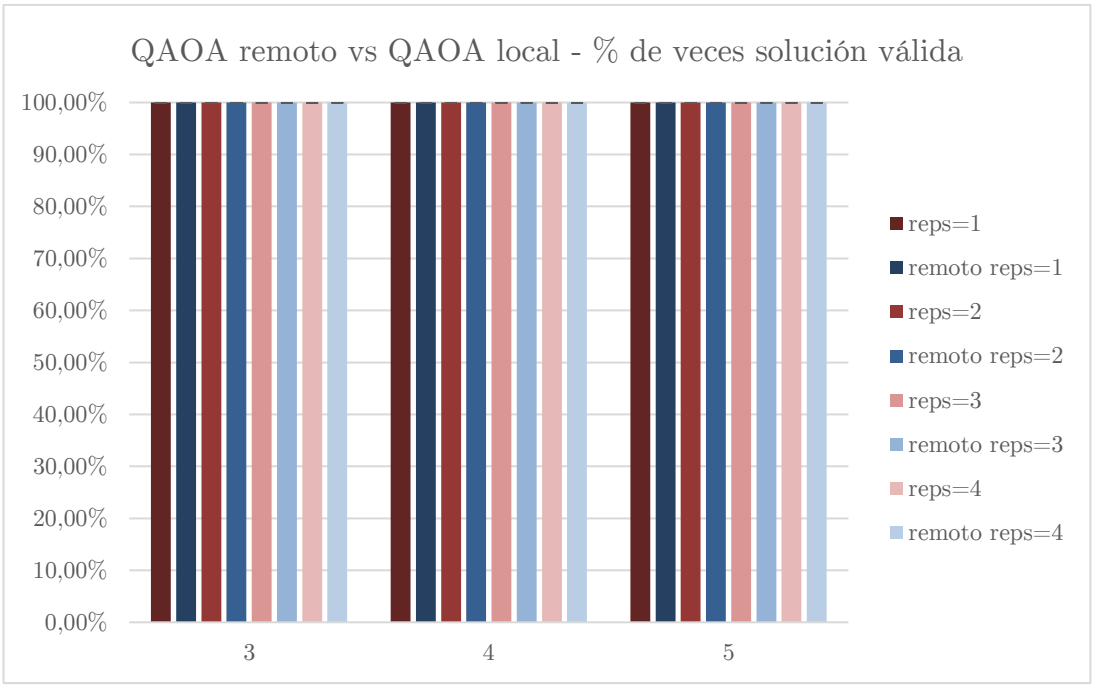


Figura 7.36. QAOA remoto: comparativa soluciones válidas (Graph C.)

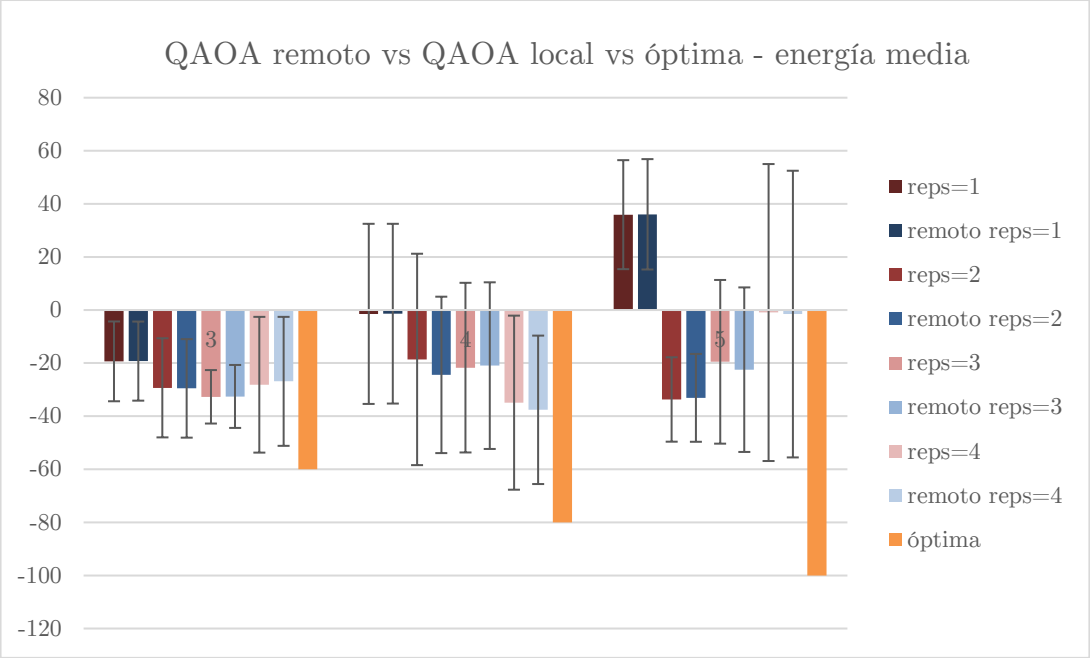


Figura 7.37. QAOA remoto: comparativa energía media (Graph C.)

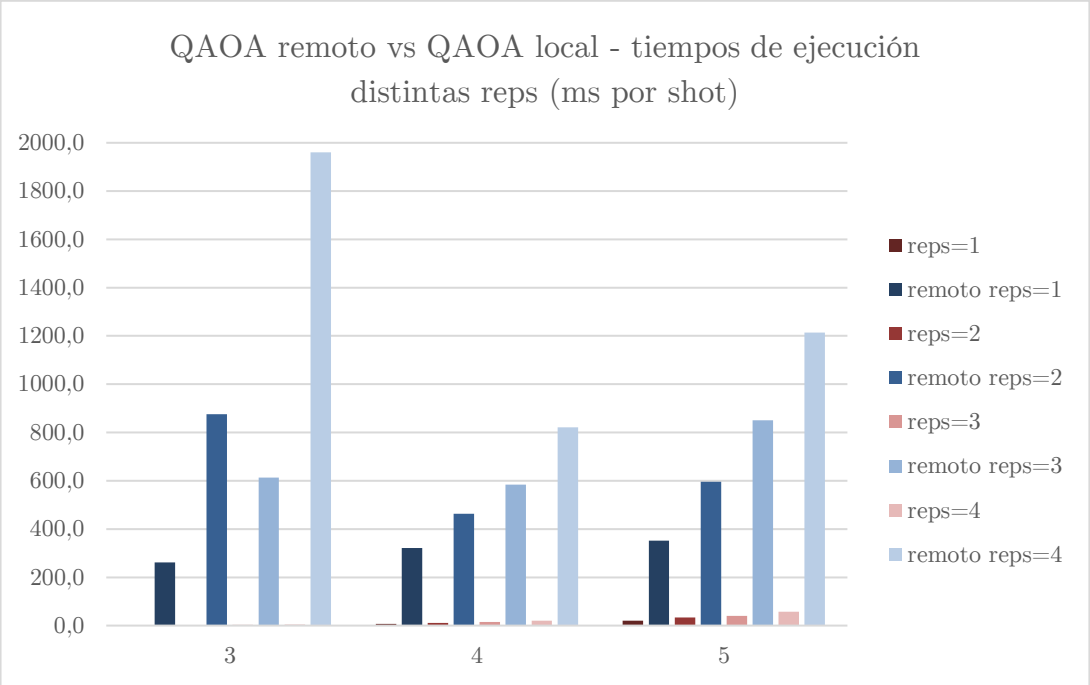


Figura 7.38. QAOA remoto: comparativa del tiempo de ejecución (Graph C.)

7.1.4.3. Annealer

En total se ha trabajado con 3000 muestras del annealer.

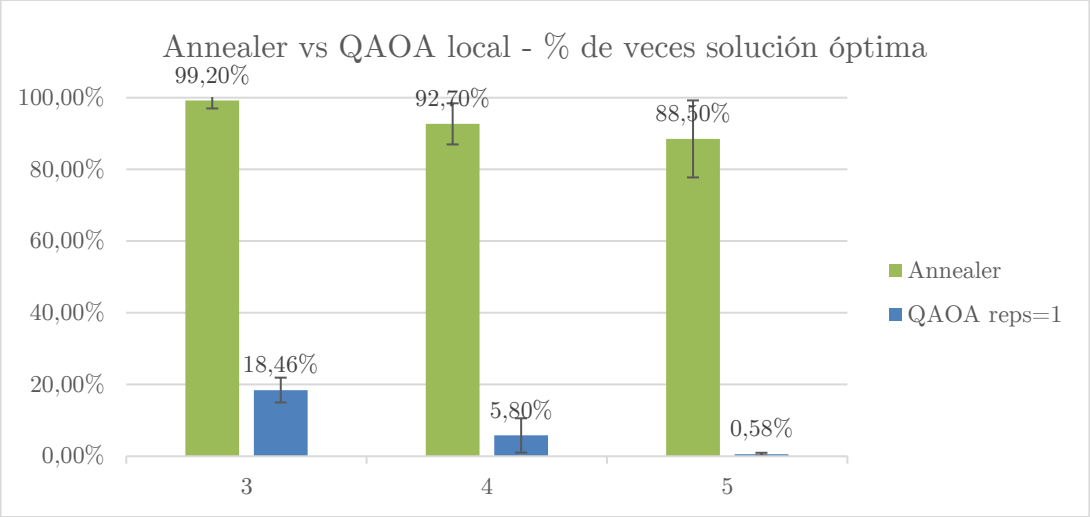


Figura 7.39. Annealer: comparativa de soluciones óptimas (Graph C.)

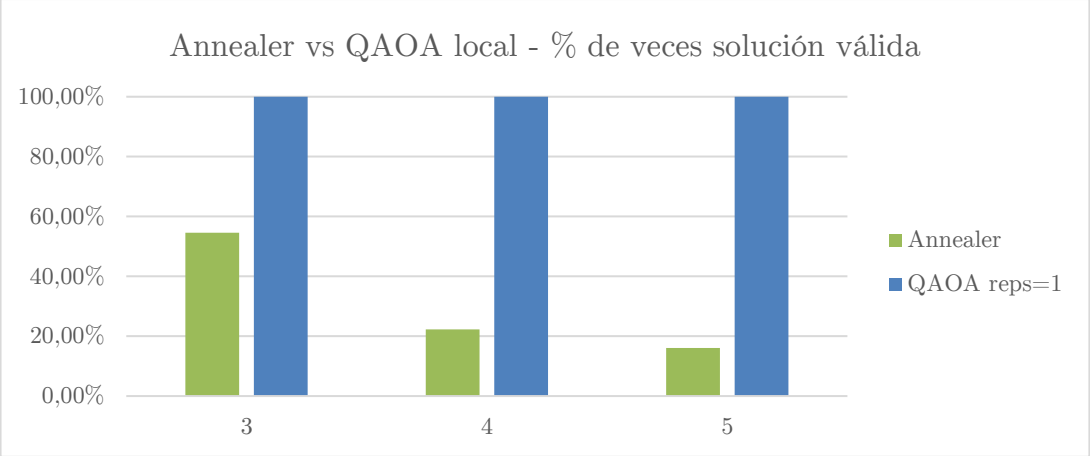


Figura 7.40. Annealer: comparativa de soluciones válidas (Graph C.)

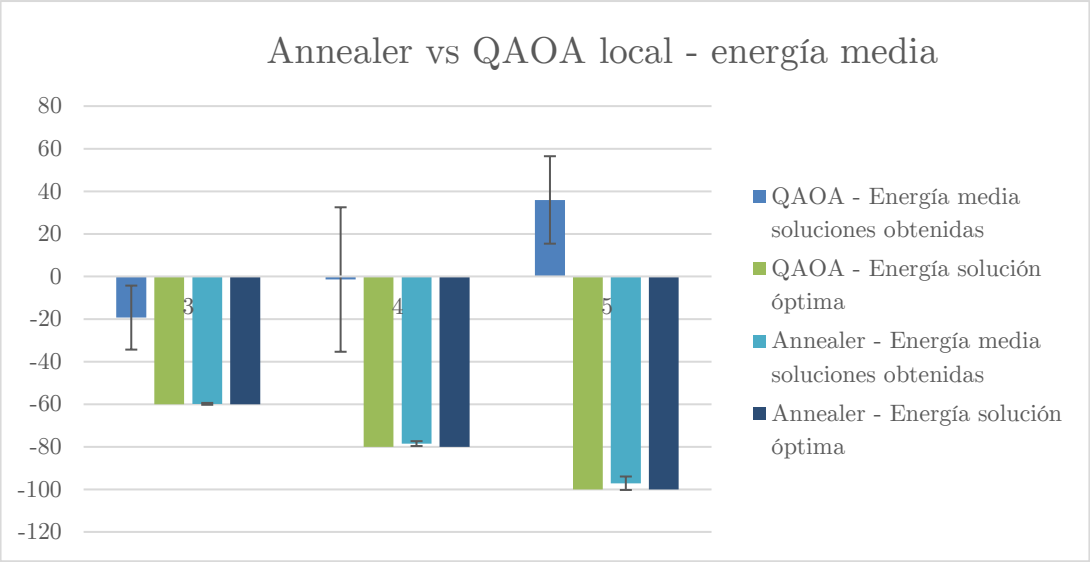


Figura 7.41. Annealer: comparativa de energía media (Graph C.)

7.2. Discusión de los resultados

Tras haber expuesto los resultados obtenidos, hay varias conclusiones a las que podemos llegar.

La mejor opción

El annealer es el método más efectivo y fiable en la gran mayoría de los casos. Supera ampliamente a las otras dos alternativas tanto en el porcentaje de veces que encuentra la solución óptima como en el porcentaje de veces que encuentra una solución válida que cumpla las restricciones (con algunas excepciones, como por ejemplo en el problema del coloreado de grafos, en el que el QAOA en ambos simuladores encuentra un mayor porcentaje de soluciones válidas).

Dependiendo del problema, la diferencia entre el rendimiento del annealer y los simuladores del QAOA puede ser más reducida (por ejemplo el porcentaje de soluciones óptimas en el problema de la mochila es bastante parejo), o más pronunciada (en el Max-Cut y el problema del coloreado de grafos el annealer es mucho más efectivo encontrando soluciones óptimas).

Esta diferencia de rendimiento entre el annealer y los simuladores no es de extrañar, puesto que el annealer cuenta con unas especificaciones mucho mejores (∼5000 qubits frente a ∼30). Como se expuso en 5.3.1.2. Annealing cuántico, la naturaleza del annealer permite construir dispositivos con un mayor número de qubits, a cambio de estar más restringido en el tipo de problemas que puede resolver.

Una constante que se puede apreciar en todos los métodos y problemas es que, al incrementar el número de nodos de la instancia, y por tanto también subir la dificultad del problema, las métricas tienden a empeorar. El número de soluciones óptimas y válidas decrece, y la energía media empieza a distanciarse más de la óptima, indicando que los resultados tienen una menor calidad.

Esto también tiene sus excepciones, como por ejemplo en el problema de la mochila, en el que el número de soluciones que cumplen las restricciones no se ve afectado de forma negativa por el crecimiento en el tamaño del problema, y la energía media tampoco se distancia considerablemente de la óptima.

El efecto del parámetro p sobre el algoritmo QAOA

Centrándonos en el QAOA, podemos observar que incrementar el valor de p generalmente se traduce en una mayor efectividad, aunque este no siempre es el caso. Dependiendo del problema y del número de nodos este efecto puede variar.

Por ejemplo, en el problema de la mochila, el mayor porcentaje de soluciones óptimas con 3 y 4 nodos en el simulador local se consigue con $p = 1$, pero con 5 nodos pasa a

ser el segundo peor (y en el simulador remoto pasa de ser el que mejores resultados produce a ser el peor).

En el Max-Cut, al disponer de más datos, podemos observar de manera clara las tendencias que se producen. Por ejemplo, en el simulador local, a partir de 6 nodos, el porcentaje de solución óptima sube hasta $p = 3$, y para $p = 4$ empeora un poco. En el simulador remoto la tendencia es distinta. Los mejores resultados se obtienen con $p = 1$ y con $p = 3$, mientras que los otros valores, en general, son peores.

Esto nos demuestra que incrementar el valor de p no implica necesariamente una mejoría en la efectividad del QAOA a la hora de encontrar más soluciones óptimas, pero lo que sí sufre un incremento directo a raíz de aumentar este valor es el tiempo de ejecución, que se puede ver en todos los problemas y ambos simuladores.

Con esto en mente, podemos concluir que no existe un valor de p que garantice conseguir los mejores resultados posibles para cualquier tipo de problema. Para descubrir qué valor funciona mejor con un problema en concreto es necesario realizar las pruebas pertinentes para evaluar el comportamiento del algoritmo al incrementar o reducir la profundidad del mismo.

Además de la efectividad, en ocasiones puede ser importante también sopesar la duración del tiempo de ejecución ya que, dependiendo de la tarea a realizar, utilizar valores mayores de p puede ser inviable debido al incremento en el tiempo de ejecución que conlleva.

Simulador local vs simulador remoto

El simulador local y el remoto del QAOA comparten características muy similares (ambos tienen en torno a 30 qubits y aplican el mismo algoritmo para encontrar la solución a problemas), y esto se traduce en resultados parecidos.

En algunos casos, el simulador local ofrece mejores resultados (véase, en el problema del coloreado de grafos, con 3 nodos la diferencia en el porcentaje de soluciones óptimas entre los dos métodos con $p = 4$), y en otros, el simulador remoto es una opción más indicada (véase, en el Max-Cut, con 5 nodos la diferencia en el porcentaje de soluciones óptimas con $p = 1$), pero la tendencia general es que sus resultados son prácticamente iguales.

El aspecto en el que más difieren estos métodos es en el tiempo de ejecución, donde el simulador remoto demuestra ser una opción mucho más costosa en cuanto a tiempo se refiere. Esto tiene sentido, puesto que para utilizar el simulador remoto es necesario conectarse a él, y dicha conexión es un tiempo que se suma al de ejecución.

Cabe mencionar la existencia del módulo Runtime, que no se ha probado en este estudio, ya que puede ayudar a reducir en gran manera este tiempo de conexión haciendo que el programa entre en cola menos veces. Además del aumento en el

tiempo de ejecución, al utilizar el simulador remoto estamos generando una dependencia con los servicios proporcionados por IBM Quantum, los cuales son externos a nuestro control, y pueden sufrir caídas o retrasos.

Esto se puede apreciar, por ejemplo, en el gráfico del problema del coloreado de grafos, donde, para grafos de 3 nodos, con $p = 2$ y $p = 4$ hay una anomalía en los tiempos obtenidos, que son mucho mayores de lo que deberían. La razón de esto es que los servicios de IBM Quantum tuvieron un periodo de mantenimiento, durante el cual las tareas se mantuvieron en la cola de espera del simulador, y no pudieron completarse. Estos mantenimientos son un ejemplo de las incidencias que este tipo de servicios pueden tener, que escapan al control de los usuarios.

Con el simulador local, por otro lado, este riesgo se evita completamente, ya que no es necesario realizar esta conexión. Un punto a tener en cuenta sobre este simulador es que su eficiencia depende de los recursos de la máquina en la que se esté ejecutando, por lo que en caso de no contar con especificaciones de cierto calibre es posible que las capacidades del algoritmo se vean comprometidas.

Teniendo en cuenta estos puntos, la elección de un simulador u otro dependerá de la situación particular y las restricciones con las que se esté trabajando. Cada opción tiene sus ventajas y desventajas y deberán tenerse en cuenta a la hora de tomar una decisión.

Capítulo 8

Conclusiones y trabajo futuro

Durante este estudio, se han realizado pruebas con distintos algoritmos cuánticos sobre varios problemas diferentes, y se han analizado los resultados para poder comparar el rendimiento de cada uno.

Este experimento nos ha dado información sobre cómo se comportan estos algoritmos frente a los problemas variando distintos parámetros como pueden ser el tamaño del problema o parámetros específicos del algoritmo (por ejemplo, el valor de p en el QAOA)

8.1. Trabajo futuro

Para investigaciones próximas sobre cómo se desenvuelven estos algoritmos en problemas de optimización combinatoria hay muchas vías que se podrían explorar para aumentar el conocimiento que tenemos sobre ellos.

En primer lugar, existe una gran variedad de métodos que se pueden probar además de los que se han visto en este trabajo (existen artículos que recopilan este tipo de algoritmos [17]), por lo que una opción sería resolver problemas con ellos y analizar los resultados que proporcionen.

Otro factor importante que puede afectar a la calidad o la velocidad de las soluciones obtenidas es la formulación de los problemas. Utilizando el mismo método y trabajando sobre el mismo problema, pueden obtenerse unos resultados muy diferentes cambiando la manera en la que se formula el problema. La diferencia que puede venir en los recursos necesarios a causa de la formulación puede ser suficiente como para decidir si el ordenador puede resolverlo o no, por lo que puede llegar a ser un factor muy decisivo.

Dentro de los algoritmos hay también muchos parámetros que se pueden modificar o afinar para mejorar el rendimiento, como se vio por ejemplo en el capítulo anterior con el valor de p en el QAOA.

El annealer, por ejemplo, también es sensible a cambios en los parámetros como el tiempo de annealing, o la fuerza de la cadena, de la cual ya hay algunos estudios que buscan encontrar el valor óptimo [18].

Un último punto a resaltar sería recordar que el campo de la computación cuántica todavía se encuentra en una fase muy preliminar, en la que se están produciendo cambios a una velocidad notable, y en la que todavía se pueden producir diversas

innovaciones. Muchas partes de la disciplina evolucionan continuamente, por lo que las posibilidades para la investigación son abundantes, y están en constante alteración.

Como ya se explicó previamente, uno de los factores limitantes en la actualidad en este ámbito es el hardware disponible. A medida que este vaya progresando, las posibilidades en investigación también se irán expandiendo.

En este estudio, por ejemplo, los simuladores de QAOA utilizados cuentan con en torno a 30 qubits con los que trabajar, que está muy lejos de ser suficiente para que el algoritmo muestre todo su potencial.

El annealer, en cambio, dispone de más de 5000 qubits, y esto deriva en una mejoría apreciable en los resultados, pero sigue sin ser suficiente. Para que los algoritmos cuánticos sean capaces de lograr mejores resultados que las alternativas clásicas, estos necesitarían tener cerca del orden de 10^5 qubits [20].

Hasta llegar a ese punto, los investigadores tienen que hacer todo lo que puedan con las herramientas que haya disponibles, buscando nuevas formas de utilizarlas que saquen el mejor rendimiento posible.

Este mismo estudio, si se repitiera en unos años, con los mismos métodos y problemas, pero con un hardware mejor, posiblemente tendría resultados muy diferentes a los que se han obtenido, por lo que, una de las posibilidades de trabajo futuro, es continuar investigando y trabajando con los métodos y dispositivos más novedosos con los que se cuenta, analizarlos, y seguir buscando sus límites.

8.2. Difusión de resultados

Los resultados de este trabajo no han sido publicados en ninguna revista científica ni se han presentado en ninguna conferencia. El código utilizado para llevar a cabo los experimentos se puede encontrar en GitHub.

Capítulo 9

Bibliografía

[1] E. F. Combarro y S. G. Castillo, *Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on approach to modern quantum algorithms.* Packt Publishing Limited, 2023.

[2] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, y A. Zeilinger, "Experimental quantum teleportation," *Nature*, vol. 390, no. 6660, pp. 575–579, Jan. 2019, doi: 10.1038/37539.

[3] A. Aspect, J. Dalibard, y G. Roger, "Experimental test of Bell's inequalities using time- varying analyzers," *Phys Rev Lett*, vol. 49, no. 25, pp. 1804–1807, Dec. 1982, doi: 10.1103/PhysRevLett.49.1804.

[4] J. F. Clauser, M. A. Horne, A. Shimony, y R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Phys Rev Lett*, vol. 23, no. 15, pp. 880–884, Oct. 1969, doi: 10.1103/PhysRevLett.23.880.

[5] S. J. Freedman y J. F. Clauser, "Experimental test of local hidden-variable theories," *Phys Rev Lett*, vol. 28, no. 14, pp. 938–941, Apr. 1972, doi: 10.1103/PhysRevLett.28.938.

[6] E. Farhi, J. Goldstone, y S. Gutmann, "A Quantum Approximate Optimization Algorithm," *Proceedings of the 3rd International Workshop on Post-Moore's Era Supercomputing*, p. 3, Nov. 2014, Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/1411.4028

[7] G. E. Crooks, "Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem," Nov. 2018, Accessed: Sep. 13, 2023. [Online]. Available: http://arxiv.org/abs/1811.08419

[8] M. X. Goemans y D. P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995, doi: 10.1145/227683.227684.

[9] M. Streif y M. Leib, "Comparison of QAOA with Quantum and Simulated Annealing," Jan. 2019, Accessed: Sep. 14, 2023. [Online]. Available: http://arxiv.org/abs/1901.01903

[10] B. Ghimire, A. Mahmood, y K. Elleithy, "Hybrid Quantum Approximate Optimization Using Enhanced Ant Colony Optimization to Solve Large-Scale Combinatorial Optimization Problems," in *2021 8th International Conference on Soft Computing and Machine Intelligence, ISCMI 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 108–113. doi: 10.1109/ISCMI53840.2021.9654824.

[11] E. Villar-Rodriguez, E. Osaba, y I. Oregi, "Analyzing the behaviour of D-Wave quantum annealer: Fine-tuning parameterization and tests with restrictive Hamiltonian formulations," in *Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence, SSCI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 938–946. doi: 10.1109/SSCI51031.2022.10022300.

[12] O. Titiloye y A. Crispin, "Quantum annealing of the graph coloring problem," *Discrete Optimization*, vol. 8, no. 2, pp. 376–384, May 2011, doi: 10.1016/j.disopt.2010.12.001.

[13] E. Farhi, J. Goldstone, S. Gutmann, y M. Sipser, "Quantum Computation by Adiabatic Evolution," Jan. 2000, Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/quant-ph/0001106

[14] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, y O. Regev, "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation," *SIAM Review*, vol. 50, no. 4, pp. 755–787, May 2004, Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/quant-ph/0405098

[15] B. Barak *et al.*, "Beating the random assignment on constraint satisfaction problems of bounded degree," *Leibniz International Proceedings in Informatics, LIPIcs*, vol. 40, pp. 110–123, May 2015, Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/1505.03424

[16] "What Is the Pegasus Topology? – D-Wave Systems." Accessed: Oct. 29, 2023. [Online]. Available: https://support.dwavesys.com/hc/en-us/articles/360054564874-What-Is-the-Pegasus-Topology-

[17] K. Bharti *et al.*, "Noisy intermediate-scale quantum (NISQ) algorithms," *Rev Mod Phys*, vol. 94, no. 1, Jan. 2021, doi: 10.1103/RevModPhys.94.015004.

[18] D. Willsch *et al.*, "Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems," *Quantum Inf Process*, vol. 21, no. 4, May 2021, doi: 10.1007/s11128-022-03476-y

[19] Project Management Institute., *A guide to the project management body of knowledge : (PMBOK guide)*. PMI, 2013.

[20] N. C. Jones et al., "Layered architecture for quantum computing," Phys Rev X, vol. 2, no. 3, Oct. 2010, doi: 10.1103/PhysRevX.2.031007.

[21] Redondo, José. (2007). Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo.

[22] Redondo, José. (2020). Creación y evaluación de plantillas para trabajos fin de grado como buena práctica docente. Revista de Innovación y Buenas Prácticas Docentes. 9. 17-35. 10.21071/ripadoc.v9i2.12988.

Capítulo 10

Anexos

Anexo I. Plan de gestión de riesgos

El objetivo de este plan es el de establecer un protocolo de respuesta frente a los riesgos que se podrían producir en el proyecto. Esto permite reducir el impacto que puedan ocasionar, o en caso de que se traten de riesgos positivos, podría ayudarnos a sacar más provecho de ellos.

El plan, tal y como se define en el PMBOK [19], se divide en varios pasos: planificación, identificación, análisis y monitorización de riesgos.

Planificación de la gestión de riesgos

Durante esta fase se presentan los modelos, las bases y las políticas a seguir en un proyecto para llevar a cabo las actividades de gestión de riesgos. Se definen los conceptos que se van a necesitar para explicar la manera en la que se va a lidiar con los riesgos.

La **metodología** que se va a utilizar para gestionar los riesgos es la propuesta por el PMBOK, y se compone de los siguientes pasos:

1. Planificar la gestión de riesgos
2. Identificar los riesgos
3. Analizar los riesgos
    a. Analizar los riesgos cualitativamente
    b. Analizar los riesgos cuantitativamente
4. Planificar la respuesta a los riesgos
5. Monitorizar y controlar los riesgos

Las **categorías de riesgo** con las que se van a trabajar y agrupar los riesgos se pueden ver en la figura siguiente:

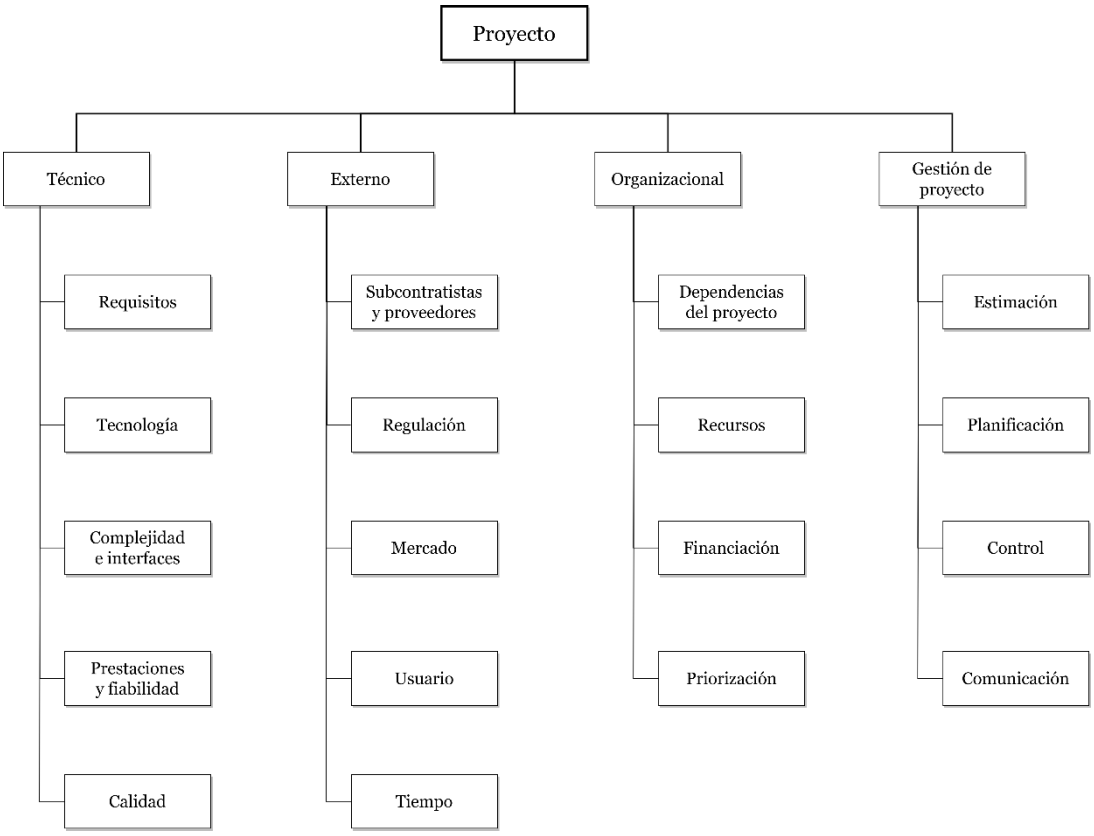


Figura 10.1. Categorías de riesgos

Otro concepto importante sobre el que hablar son las **definiciones de probabilidad e impacto**. Hay que asignarle un valor a los efectos que puede tener un riesgo sobre el proyecto, para poder medir su impacto. A continuación se muestra la tabla de definición de probabilidades que se va a tener en cuenta, la cual muestra cinco etiquetas y los rangos de probabilidad asociados a ella

Etiqueta	Rango	Valor usado para los cálculos (y en la matriz de probabilidad / impacto)
Muy bajo	[0% ... 20%]	10%
Bajo	[20% ... 40%]	30%
Medio	[40% ... 60%]	50%
Alto	[60% ... 80%]	70%
Muy alto	[80% ... 100%]	90%

Tabla 11. Definición de probabilidades.

Para el impacto se utiliza una tabla parecida, pero que, en lugar de asignar una etiqueta para un rango de probabilidades, lo haga en función de la repercusión que tenga el riesgo en una métrica u objetivo en concreto del proyecto (como puede ser el coste o el tiempo). Para este proyecto, la tabla de impacto que se considera es la siguiente

Condiciones definidas para las escalas de impacto de un riesgo (negativo) sobre los objetivos principales del proyecto					
Objetivos de proyecto	Escalas relativas o numéricas				
	Inapreciable / 5%	Bajo / 10%	Medio / 20%	Alto / 40%	Crítico / 80%
Presupuesto	Incremento del coste < 1%	Incremento del coste < 5%	Incremento del coste entre 5% y 10%	Incremento del coste entre 10%-20%	Incremento del coste > 20%
Planificación	Incremento del tiempo < 1%	Incremento del tiempo entre 1% y 3%	Incremento del tiempo entre 3% y 5%	Incremento del tiempo entre 5% y 10%	Incremento del tiempo > 10%
Alcance	Reducción del alcance sin importancia	Reducción del alcance poco notable	Reducción del alcance notable pero no importante	Reducción del alcance notable pero no crucial	Reducción de una parte importante del alcance
Calidad	No se aprecia la bajada de calidad	Bajada de calidad sin demasiada relevancia	Bajada de calidad pero se mantiene aceptable	Bajada de calidad notable	Bajada de calidad que compromete de forma grave el producto final

Tabla 12. Escalas de impacto para los principales factores del proyecto

Para priorizar los riesgos utilizaremos la **matriz de probabilidad e impacto**, cuyos valores se obtienen teniendo en cuenta la probabilidad de que ocurra el riesgo y su impacto. En este caso, la matriz que se utiliza es la siguiente

MATRIZ DE PROBABILIDAD E IMPACTO										
Probabilidad	*Amenazas*					*Oportunidades*				
0,90	0,05	0,09	0,18	0,36	0,72	0,72	0,36	0,18	0,09	0,05
0,70	0,04	0,07	0,14	0,28	0,56	0,56	0,28	0,14	0,07	0,04
0,50	0,03	0,05	0,10	0,20	0,40	0,40	0,20	0,10	0,05	0,03
0,30	0,02	0,03	0,06	0,12	0,24	0,24	0,12	0,06	0,03	0,02
0,10	0,01	0,01	0,02	0,04	0,08	0,08	0,04	0,02	0,01	0,01
	0,05 Muy bajo	0,10 Bajo	0,20 Moderado	0,40 Alto	0,80 Muy alto	0,80 Muy alto	0,40 Alto	0,20 Moderado	0,10 Bajo	0,05 Muy bajo
	Impacto negativo					**Impacto positivo**				

Tabla 13. Matriz de probabilidad e impacto

También hay que identificar las personas que van a estar involucradas en la gestión de riesgos del proyecto. En este caso, el alumno tomará el rol de **responsable de riesgos**, y se encargará de las responsabilidades asociadas en este aspecto.

Identificación de riesgos

En este proceso se utilizan una serie de técnicas para encontrar riesgos potenciales del proyecto. Se realiza de manera iterativa porque los riesgos no son estáticos. A medida que el proyecto avanza, pueden aparecer riesgos nuevos o desaparecer.

Se pueden identificar riesgos nuevos de distintas maneras. Para este trabajo se han utilizado las siguientes técnicas:

1. **Tormenta de ideas (brainstorming):** al inicio del proyecto se consideraron riesgos potenciales teniendo en cuenta las actividades que se iban a realizar, resultando en una lista.

2. **Juicio de expertos:** debido a la experiencia del desarrollador en algunos proyectos anteriores, este pudo deducir y tener en cuenta algunos riesgos que podrían darse también en este proyecto.

Análisis de riesgos

El objetivo de esta fase es asignar un valor numérico a la probabilidad y el impacto que podría tener un riesgo en el proyecto.

El PMBOK habla de dos fases de análisis, el análisis cualitativo y el cuantitativo.

Análisis cualitativo

El análisis cualitativo permite calcular la probabilidad, impacto e importancia de un riesgo. Esto sirve para tener una comprensión inicial sobre cuáles son los riesgos que más podrían comprometer el desarrollo del proyecto, y priorizarlos en base a ello.

Análisis cuantitativo

Este análisis complementa al cualitativo, ya que proporciona una evaluación más precisa y cuantificada, asignando valores numéricos a los riesgos.

Una técnica para realizar esto, y que se ha utilizado en este proyecto, es la matriz de probabilidad e impacto. Se tienen en cuenta valores, que en este caso están comprendidos en 5 rangos, y se obtiene el valor a partir de la tabla de definición de probabilidades (véase Tabla 11. Definición de probabilidades.), y dicho valor posteriormente se utiliza en la matriz de probabilidad e impacto (véase Tabla 13. Matriz de probabilidad e impacto).

Planificación de la respuesta a los riesgos

Habiendo identificado y priorizado los riesgos, el siguiente paso es elaborar un plan en concreto para su gestión, preferiblemente de los de mayor importancia.

En este plan se especifica la respuesta frente a este riesgo, y los recursos (temporales, económicos...) que se van a dedicar en esta labor.

Las posibles estrategias que se van a considerar en este proyecto frente a los riesgos son 4:

1. **Eliminar el riesgo:** se suprimen las posibles causas de manifestación del riesgo, evitando por completo que pueda aparecer.

2. **Transferir el riesgo:** hay casos en los que hay un tercero que se puede encargar de responder frente al riesgo, liberándonos a nosotros de esa responsabilidad.

3. **Mitigación del riesgo:** reducimos el impacto del riesgo en la medida de lo posible, conociendo de antemano los aspectos en los que puede afectar y estando preparados para minimizar el impacto en caso de que el riesgo suceda.

4. **Asumir el riesgo:** si el riesgo no se puede evitar de ningún modo, una opción es aceptarlo y no elaborar una respuesta específica.

Monitorización y control de los riesgos

El plan de gestión de riesgos no se limita solamente al principio del proyecto. Es importante mantener un control de la evolución de los riesgos y de su tratamiento a lo largo de todo su desarrollo, y ajustar el plan acorde a los cambios que vayan surgiendo.

Las decisiones y respuestas que se tomarán durante la monitorización del proyecto variarán en función de los cambios que se produzcan.

Anexo II. Presupuesto

El presupuesto de costes y el de cliente no son los únicos componentes del presupuesto. Hay otros aspectos que intervienen en el resultado final que no se han mencionado en el capítulo 3.

El propósito de este anexo es el de exponer estos datos restantes, tanto para la versión inicial como para la final.

Presupuesto inicial

En primer lugar tenemos la definición de la empresa encargada de llevar a cabo este trabajo. El objetivo de crear esta empresa es el de simular un presupuesto realista si este estudio fuese encargado a una entidad.

A continuación se listan todos los elementos que se han tenido descrito para la definición de la empresa:

Personal	Núm.	Sueldo bruto año	Coste salarial año	TOTAL
Jefe de proyecto	1	43.500 €	57.855 €	57.855 €
Arquitecto	1	39.600 €	52.668 €	52.668 €
Analista	1	39.600 €	52.668 €	52.668 €
Investigador	1	33.000 €	43.890 €	43.890 €
Ingeniero de software	1	33.000 €	43.890 €	43.890 €
TOTAL	5			250.971 €

Tabla 14. Definición de empresa: personal

Personal	TOTAL	Prod (%)	Coste Directo	CI (%)	Coste Indirecto
Jefe de proyecto	57.855 €	80,00%	46.284,00 €	20,00%	11.571 €
Arquitecto	52.668 €	75,00%	39.501,00 €	25,00%	13.167 €
Analista	52.668 €	75,00%	39.501,00 €	25,00%	13.167 €
Investigador	43.890 €	80,00%	35.112,00 €	20,00%	8.778 €
Ingeniero de software	43.890 €	80,00%	35.112,00 €	20,00%	8.778 €
TOTAL	250.971 €		195.510 €		55.461 €

Tabla 15. Definición de empresa: productividad

Servicio	Coste mes	Coste año
Alquiler oficina	300 €	3.600 €
Consumo de electricidad	50 €	600 €

Mantenimiento, reparación, conservación	15 €	180 €
Primas de seguros	100 €	1.200 €
Gastos de comunicaciones	10 €	120 €
Gastos en material de oficina	20 €	240 €
Gastos en conexión a Internet	30 €	360 €
TOTAL	525 €	6.300 €

Tabla 16. Definición de empresa: costes indirectos

Equipo / Licencia	Unid.	Precio	Coste total	Coste año	Tipo	Plazo
Licencias de software de ofimática	1	7 €	7 €	60 €	Alquiler	1
Portátil	1	1.000 €	1.000 €	1.000 €	Amortización	3
Periféricos	1	50 €	50 €	50 €	Amortización	-
Licencia de IDE	1	60 €	60 €	600 €	Alquiler	-
TOTAL				1.710 €		

Tabla 17. Definición de empresa: medios de producción

Personal	Product (%)	Horas / año	Horas productivas / año (por persona)	Horas productivas (total empresa)
Jefe de proyecto	70,00%	1776	1243,2	1243,2
Arquitecto	75,00%	1776	1332	1332
Analista	75,00%	1776	1332	1332
Investigador	80,00%	1776	1420,8	1420,8
Ingeniero de software	80,00%	1776	1420,8	1420,8
TOTAL				5505,6

Tabla 18. Definición de empresa: horas productivas

Personal	Precio / hora	Horas productivas (total empresa)	Facturación
Jefe de proyecto	53,75 €	1243,2	66.822,00 €
Arquitecto	50,00 €	1332	66.600,00 €
Analista	50,00 €	1332	66.600,00 €
Investigador	43,75 €	1420,8	62.160,00 €
Ingeniero de software	43,75 €	1420,8	62.160,00 €
TOTAL		6748,8	324.342,00 €

Tabla 19. Definición de empresa: precio por hora (1)

Precio / hora (sin beneficios)
43,00 €
40,00 €
40,00 €
35,00 €
35,00 €

Tabla 20. Definición de empresa: precio por hora (2)

Nº	CONCEPTO	IMPORTE
1	Total costes directos	195.510 €
2	Total costes indirectos	63.471 €
3	Suma costes directos + indirectos	258.981 €
4	Beneficio deseado (25%)	64.745 €
5	Coste total (costes + beneficios)	323.726 €
6	Facturación posible en función de horas de producción y de los precios por hora calculados	324.342,00 €
7	Margen entre el coste total y la facturación	0,19%

Tabla 21. Definición de empresa: resumen

Tras haber descrito la empresa, otro de los componentes que no se ha detallado previamente y que aportan información importante sobre la elaboración del presupuesto es el detalle de las partidas que se han utilizado.

Hasta ahora sólo se había visto el nombre de las mismas, y un desglose simplificado en el presupuesto del cliente, pero esto deja fuera mucho detalle acerca de cómo se ha calculado el precio de estas partidas, por lo que se incluyen también en este anexo.

En total se han creado cinco partidas:

1. Partida para las reuniones de inicio y cierre, y para la formación.
2. Partida para las tareas referidas al estudio.
3. Partida para las tareas del proceso de desarrollo.
4. Partida para la documentación
5. Partida para otros costes del proyecto.

Debido al extenso tamaño de las partidas, estas se presentan en páginas horizontales en lugar de verticales.

Partida 1: Reuniones y formación										
l1	**l2**	**l3**	**l4**	**l5**	**Descripción**	**Cantidad**	**Unidades**	**Precio**	**Subtotal**	**Total**
01					Reunión de arranque					43,00 €
	001				Jefe de proyecto	1	hora	43,00 €	43,00 €	
02					**Formación en computación cuántica**					**3.885,00 €**
	001				Formación en las bases de la computación cuántica					
		0001			Investigador	40	horas	35,00 €	1.400,00 €	
	002				**Formación en optimización cuántica**					
		0001			**Formación en formulaciones y problemas**					
			00001		Formación en el framework QUBO					
				000001	Investigador	6	horas	35,00 €	210,00 €	
			00002		Formación en el modelo de Ising					
				000001	Investigador	5	horas	35,00 €	175,00 €	
			00003		Formación en paso de Ising a QUBO					
				000001	Investigador	3	horas	35,00 €	105,00 €	
			00004		Formación en representación de problemas de optimización combinatoria					
				000001	Investigador	20	horas	35,00 €	700,00 €	
		0002			**Formación en annealing cuántico**					
			00001		Formación en computación adiabática					
				000001	Investigador	7	horas	35,00 €	245,00 €	
			00002		Formación en annealing cuántico					
				000001	Investigador	5	horas	35,00 €	175,00 €	
			00003		Formación en uso de annealers de D-Wave					
				000001	Investigador	5	horas	35,00 €	175,00 €	
		0003			**Formación en QAOA**					
			00001		Formación en paso de annealing cuántico a QAOA					
				000001	Investigador	5	horas	35,00 €	175,00 €	
			00002		Formación en resolución de problemas utilizando el QAOA					
				000001	Investigador	15	horas	35,00 €	525,00 €	
03					Reunión de cierre					43,00 €
	001				Jefe de proyecto	1	horas	43,00 €	43,00 €	
									TOTAL	3.971,00 €

Tabla 22. Presupuesto de costes: partida de reuniones y de formación

Partida 2: Estudio								
I1	**I2**	**I3**	**Descripción**	**Cantidad**	**Unidades**	**Precio**	**Subtotal**	**Total**
01			**Estudio del Max-Cut**					**105,00 €**
	001		Generación de grafos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Resolución de los grafos					
		0001	Equipo	30	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
02			**Estudio del TSP**					**105,00 €**
	001		Generación de grafos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Resolución de los grafos					
		0001	Equipo	15	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
03			**Estudio del problema de la mochila**					**70,00 €**
	001		Generación de grafos					
		0001	Investigador	1	horas	35,00 €	35,00 €	
	002		Resolución de los grafos					
		0001	Equipo	12	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
04			**Estudio del problema del coloreado de grafos**					**105,00 €**
	001		Generación de grafos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Resolución de los grafos					
		0002	Equipo	15	horas			
	003		Procesamiento de los resultados					
		0003	Investigador	1	horas	35,00 €	35,00 €	
							TOTAL	385,00 €

Tabla 23. Presupuesto de costes: partida del estudio

Partida 3: Desarrollo								
I1	**I2**	**I3**	**Descripción**	**Cantidad**	**Unidades**	**Precio**	**Subtotal**	**Total**
01			Formación en Flask					175,00 €
	001		Ingeniero de software	5	horas	35,00 €	175,00 €	
02			Implementación página principal					105,00 €
	001		Ingeniero de software	3	horas	35,00 €	105,00 €	
03			**Página Max-Cut**					**630,00 €**
	001		Implementación de las distintas formas de resolver los problemas					
		0001	Ingeniero de software	10	horas	35,00 €	350,00 €	
	002		Parametrización de los problemas					
		0001	Ingeniero de software	5	horas	35,00 €	175,00 €	
	003		Carga de problema a través de fichero					
		0001	Ingeniero de software	3	horas	35,00 €	105,00 €	
04			**Página TSP**					**315,00 €**
	001		Implementación de las distintas formas de resolver los problemas					
		0001	Ingeniero de software	1	horas	35,00 €	35,00 €	
	002		Parametrización de los problemas					
		0001	Ingeniero de software	5	horas	35,00 €	175,00 €	
	003		Carga de problema a través de fichero					
		0001	Ingeniero de software	3	horas	35,00 €	105,00 €	
05			**Página del problema de la mochila**					**280,00 €**
	001		Implementación de las distintas formas de resolver los problemas					
		0001	Ingeniero de software	1	horas	35,00 €	35,00 €	
	002		Parametrización de los problemas					
		0001	Ingeniero de software	3	horas	35,00 €	105,00 €	
	003		Carga de problema a través de fichero					
		0001	Ingeniero de software	4	horas	35,00 €	140,00 €	
06			**Página del problema de coloreado de grafos**					**350,00 €**

001		Implementación de las distintas formas de resolver los problemas						
	0001	Ingeniero de software	1	horas	35,00 €	35,00 €		
002		Parametrización de los problemas						
	0001	Ingeniero de software	5	horas	35,00 €	175,00 €		
003		Carga de problema a través de fichero						
	0001	Ingeniero de software	4	horas	35,00 €	140,00 €		
							TOTAL	1.855,00 €

Tabla 24. Presupuesto de costes: partida de desarrollo

Partida 4: Documentación										
l1	**l2**	**l3**	**l4**	**l5**	**Descripción**	**Cantidad**	**Unidades**	**Precio**	**Subtotal**	**Total**
01					**Abstract**					**35,00 €**
	001				Investigador	1	horas	35,00 €	35,00 €	
02					**Introducción**					**1.225,00 €**
	001				Motivación					
		0001			Investigador	4	horas	35,00 €	140,00 €	
	002				Finalidad del proyecto					
		0001			Investigador	1	horas	35,00 €	35,00 €	
	003				Marco teórico					
		0001			Investigador	30	horas	35,00 €	1.050,00 €	
03					**Fijación de objetivos**					**175,00 €**
	001				Investigador	5	horas	35,00 €	175,00 €	
04					**Planificación y gestión del TFG**					**4.996,50 €**
	001				Planificación del proyecto					
		0001			Identificación de interesados					
			00001		Analista	1	horas	40,00 €	40,00 €	
			00002		Jefe de proyecto	1	horas	43,00 €	43,00 €	
		0002			OBS					

		00001		Arquitecto	1	horas	40,00 €	40,00 €
		00002		Jefe de proyecto	1	horas	43,00 €	43,00 €
	0003			PBS				
		00001		Arquitecto	8	horas	40,00 €	320,00 €
		00002		Jefe de proyecto	8	horas	43,00 €	344,00 €
	0004			WBS				
		00001		Arquitecto	20	horas	40,00 €	800,00 €
		00002		Jefe de proyecto	20	horas	43,00 €	860,00 €
	0005			Riesgos				
		00001		Plan de gestión de riesgos				
			000001	Arquitecto	2	horas	40,00 €	80,00 €
			000002	Jefe de proyecto	2	horas	43,00 €	86,00 €
		00002		Identificación de riesgos				
			000001	Arquitecto	8	horas	40,00 €	320,00 €
			000001	Jefe de proyecto	8	horas	43,00 €	344,00 €
	0006			Presupuesto inicial del TFG				
		00001		Presupuesto de costes				
			000001	Jefe de proyecto	6	horas	43,00 €	258,00 €
		00002		Presupuesto de cliente				
			000001	Jefe de proyecto	2	horas	43,00 €	86,00 €
002				Ejecución del proyecto				
	0001			Plan de seguimiento de planificación				
		00001		Arquitecto	5	horas	40,00 €	200,00 €
		00002		Jefe de proyecto	5	horas	43,00 €	215,00 €
	0002			Bitácora de incidencias del proyecto				
		00001		Arquitecto	1	horas	40,00 €	40,00 €
		00002		Jefe de proyecto	1	horas	43,00 €	43,00 €
	0003			Riesgos				
		00001		Arquitecto	3,5	horas	40,00 €	140,00 €

		00002	Jefe de proyecto	3,5	horas	43,00 €	150,50 €	
003			Cierre del proyecto					
	0001		Planificación final					
		00001	Arquitecto	1	horas	40,00 €	40,00 €	
		00002	Jefe de proyecto	1	horas	43,00 €	43,00 €	
	0002		Informe final de riesgos					
		00001	Arquitecto	1	horas	40,00 €	40,00 €	
		00002	Jefe de proyecto	1	horas	43,00 €	43,00 €	
	0003		Presupuesto final de costes					
		00001	Jefe de proyecto	3	horas	43,00 €	129,00 €	
	0004		Informe de lecciones aprendidas					
		00001	Arquitecto	3	horas	40,00 €	120,00 €	
		00002	Jefe de proyecto	3	horas	43,00 €	129,00 €	
05			**Estado del arte**					**700,00 €**
001			Investigador	20	horas	35,00 €	700,00 €	
06			**Descripción del sistema**					**1.120,00 €**
001			Formulaciones					
	0001		Framework QUBO					
		00001	Investigador	1	horas	35,00 €	35,00 €	
	0001		Modelo de Ising					
		00001	Investigador	2	horas	35,00 €	70,00 €	
002			Problemas					
	0001		Max-Cut					
		00001	Investigador	2	horas	35,00 €	70,00 €	
	0002		Programación lineal binaria					
		00001	Investigador	3	horas	35,00 €	105,00 €	
	0003		TSP					
		00001	Investigador	3	horas	35,00 €	105,00 €	
	0004		Problema de la mochila					

		00001		Investigador	2	horas	35,00 €	70,00 €	
	0005			Problema del coloreado de grafos					
		00001		Investigador	3	horas	35,00 €	105,00 €	
003				Métodos					
	0001			Annealing cuántico					
		00001		Computación adiabática cuántica					
			000001	Investigador	3	horas	35,00 €	105,00 €	
		00002		Annealing cuántico					
			000001	Investigador	2	horas	35,00 €	70,00 €	
	0002			QAOA					
		00001		Definición					
			000001	Investigador	2	horas	35,00 €	70,00 €	
		00002		Algoritmo					
			000001	Investigador	3	horas	35,00 €	105,00 €	
		00003		Paso a circuitos					
			000001	Investigador	6	horas	35,00 €	210,00 €	
07				**Metodología de trabajo**					**352,92 €**
001				Explicación de métodos a utilizar					
	0001			Investigador	2	horas	35,00 €	70,00 €	
002				Max-Cut					
	0001			Datos utilizados					
		00001		Investigador	0,25	horas	35,00 €	8,75 €	
	0002			Formulación del problema					
		00001		Ingeniero de software	1	horas	35,00 €	35,00 €	
	0003			Ejecución del problema					
		00001		Ingeniero de software	2	horas	35,00 €	70,00 €	
003				TSP					
	0001			Datos utilizados					
		00001		Investigador	0,25	horas	35,00 €	8,75 €	

	0002		Formulación del problema					
		00001	Ingeniero de software	2	horas	35,00 €	70,00 €	
	0003		Ejecución del problema					
		00001	Ingeniero de software	1	horas	35,00 €	35,00 €	
004			Problema de la mochila					
	0001		Datos utilizados					
		00001	Investigador	0,25	horas	35,00 €	8,75 €	
	0002		Formulación del problema					
		00001	Ingeniero de software	0,5	horas	35,00 €	17,50 €	
	0003		Ejecución del problema					
		00001	Ingeniero de software	0,25	horas	35,00 €	8,75 €	
005			Problema del coloreado de grafos					
	0001		Datos utilizados					
		00001	Investigador	0,25	horas	35,00 €	8,75 €	
	0002		Formulación del problema					
		00001	Ingeniero de software	0,17	horas	35,00 €	5,83 €	
	0003		Ejecución del problema					
		00001	Ingeniero de software	0,17	horas	35,00 €	5,83 €	
08			**Resultados obtenidos**					**245,00 €**
001			Presentación de los resultados					
	0001		Max-Cut					
		00001	Investigador	1	horas	35,00 €	35,00 €	
	0002		TSP					
		00001	Investigador	1	horas	35,00 €	35,00 €	
	0003		Problema de la mochila					
		00001	Investigador	1	horas	35,00 €	35,00 €	
	0004		Problema del coloreado de grafos					
		00001	Investigador	1	horas	35,00 €	35,00 €	
002			Discusión de los resultados					

		0001	Investigador	3	horas	35,00 €	105,00 €	
09			**Conclusiones y trabajo futuro**					**78,75 €**
	001		Trabajo futuro					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Difusión de los resultados					
		0001	Investigador	0,25	horas	35,00 €	8,75 €	
10			**Bibliografía**					**8,75 €**
	001		Investigador	0,25	horas	35,00 €	8,75 €	
11			**Anexos**					**635,00 €**
	001		Plan de gestión de riesgos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Presupuesto					
		0001	Jefe de proyecto	5	horas	43,00 €	215,00 €	
	003		Aplicación web					
		0001	Investigador	10	horas	35,00 €	350,00 €	
							TOTAL	9.571,92 €

Tabla 25. Presupuesto de costes: partida de documentación

Partida 5: Otros gastos							
I1	**I2**	**Descripción**	**Cantidad**	**Unidades**	**Precio**	**Subtotal**	**Total**
01		**Reuniones**					**60,00 €**
	001	Arranque	1	hora	30,00 €	30,00 €	
	002	Cierre	1	hora	30,00 €	30,00 €	
						TOTAL	60,00 €

Tabla 26. Presupuesto de costes: partida de otros gastos

Presupuesto final

El presupuesto final comparte muchas similitudes con el inicial. Los cambios que se han hecho son los que provienen también de cambios en la planificación, que han afectado a algunas partidas.

Las revisiones que se hicieron a las partidas en el presupuesto nuevo son las siguientes:

Partida 2: Estudio

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal	Total
01			**Estudio del Max-Cut**					**105,00 €**
	001		Generación de grafos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Resolución de los grafos					
		0001	Equipo	45	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
02			**Estudio del TSP**					**385,00 €**
	001		Generación de grafos					
		0001	Investigador	10	horas	35,00 €	350,00 €	
	002		Resolución de los grafos					
		0001	Equipo	25	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
03			**Estudio del problema de la mochila**					**70,00 €**
	001		Generación de grafos					
		0001	Investigador	1	horas	35,00 €	35,00 €	
	002		Resolución de los grafos					
		0001	Equipo	30	horas			
	003		Procesamiento de los resultados					
		0001	Investigador	1	horas	35,00 €	35,00 €	
04			**Estudio del problema del coloreado de grafos**					**105,00 €**
	001		Generación de grafos					
		0001	Investigador	2	horas	35,00 €	70,00 €	
	002		Resolución de los grafos					
		0002	Equipo	35	horas			
	003		Procesamiento de los resultados					
		0003	Investigador	1	horas	35,00 €	35,00 €	
							TOTAL	665,00 €

Tabla 27. Presupuesto de costes final: correcciones de la partida del estudio

Y en la partida de documentación, se hicieron estas modificaciones

04			**Planificación y gestión del TFG**					**4.955,00 €**
001			Planificación del proyecto					
	0001		Identificación de interesados					
		00001	Analista	1	horas	40,00 €	40,00 €	
		00002	Jefe de proyecto	1	horas	43,00 €	43,00 €	
	0002		OBS					
		00001	Arquitecto	2	horas	40,00 €	80,00 €	
		00002	Jefe de proyecto	2	horas	43,00 €	86,00 €	
	0003		PBS					
		00001	Arquitecto	6	horas	40,00 €	240,00 €	
		00002	Jefe de proyecto	6	horas	43,00 €	258,00 €	
	0004		WBS					
		00001	Arquitecto	20	horas	40,00 €	800,00 €	
		00002	Jefe de proyecto	20	horas	43,00 €	860,00 €	
	0005		Riesgos					
		00001	Plan de gestión de riesgos					
			000001 Arquitecto	2	horas	40,00 €	80,00 €	
			000002 Jefe de proyecto	2	horas	43,00 €	86,00 €	
		00002	Identificación de riesgos					
			000001 Arquitecto	8	horas	40,00 €	320,00 €	
			000001 Jefe de proyecto	8	horas	43,00 €	344,00 €	
	0006		Presupuesto inicial del TFG					
		00001	Presupuesto de costes					
			000001 Jefe de proyecto	6	horas	43,00 €	258,00 €	
		00002	Presupuesto de cliente					
			000001 Jefe de proyecto	2	horas	43,00 €	86,00 €	
002			Ejecución del proyecto					
	0001		Plan de seguimiento de planificación					
		00001	Arquitecto	4	horas	40,00 €	160,00 €	
		00002	Jefe de proyecto	4	horas	43,00 €	172,00 €	
	0002		Bitácora de incidencias del proyecto					
		00001	Arquitecto	2	horas	40,00 €	80,00 €	
		00002	Jefe de proyecto	2	horas	43,00 €	86,00 €	
	0003		Riesgos					
		00001	Arquitecto	2	horas	40,00 €	80,00 €	
		00002	Jefe de proyecto	2	horas	43,00 €	86,00 €	
003			Cierre del proyecto					
	0001		Planificación final					

	00001	Arquitecto	2	horas	40,00 €	80,00 €
	00002	Jefe de proyecto	2	horas	43,00 €	86,00 €
0002		Informe final de riesgos				
	00001	Arquitecto	2	horas	40,00 €	80,00 €
	00002	Jefe de proyecto	2	horas	43,00 €	86,00 €
0003		Presupuesto final de costes				
	00001	Jefe de proyecto	3	horas	43,00 €	129,00 €
0004		Informe de lecciones aprendidas				
	00001	Arquitecto	3	horas	40,00 €	120,00 €
	00002	Jefe de proyecto	3	horas	43,00 €	129,00 €
		...				
					TOTAL	9.530,42 €

Tabla 28. Presupuesto de costes final: correcciones partida documentación

Las demás partidas no sufrieron cambios, aunque hay un ligero incremento de algunos céntimos en su precio debido a un cambio en el factor de ponderación a causa de haber modificado el coste de la partida del estudio y la de la documentación.

Anexo III. Aplicación web.

Descripción de la aplicación web

A modo de complemento del estudio, se ha desarrollado una aplicación web en la que se pueden introducir instancias de los distintos problemas que se han analizado en el trabajo, bien por medio de un archivo o a través de la propia página web de forma manual, y se pueden resolver a través de numerosos métodos.

De esta forma el usuario puede observar el rendimiento y las capacidades de estos métodos de primera mano.

Alcance de la aplicación web

Para conseguir que la aplicación cumpla el propósito descrito previamente, ha sido necesario implementar la siguiente funcionalidad:

1. El usuario puede registrarse en su cuenta de IBM Quantum Experience y de D-Wave a través de los tokens para conectarse a las API proporcionados por las empresas.

2. El usuario puede cargar en la aplicación sus propias instancias de los problemas estudiados.

    a. De forma manual, especificando los parámetros a través de la propia aplicación.

    b. A través de un archivo, siguiendo un formato concreto.

3. El usuario puede elegir el método a utilizar (tanto clásico como cuántico), y utilizarlo para encontrar la solución al problema.

Requisitos de la aplicación web

A continuación se presenta de manera breve los requisitos que debe cumplir la aplicación web, así como los actores implicados en su uso, y una serie de casos de uso importantes que aclaren el comportamiento del sistema ante ciertas situaciones.

Obtención de requisitos

En este apartado se listarán los requisitos del sistema, los cuales se agrupan en funcionales y no funcionales, y también se agrupan por funcionalidades concretas.

Requisitos funcionales

Requisitos de la carga en sesión de plataformas:

R-Ses-1: El sistema permite al usuario cargar en sesión las credenciales de Qiskit utilizando el token para conectarse a la API que proporciona la empresa.

> *R-Ses-1.1:* El sistema notificará al usuario en caso de que los credenciales se hayan cargado correctamente.
>
> *R-Ses-1.2:* El sistema notificará al usuario en caso de que no sea posible cargar los credenciales.

R-Ses-2: El sistema permite al usuario cargar en sesión las credenciales de D-Wave por medio del token que pone a disposición la empresa para conectarse a la API.

> *R-Ses-2.1:* El sistema notificará al usuario en caso de que los credenciales se hayan cargado correctamente.
>
> *R-Ses-2.2:* El sistema notificará al usuario en caso de que no sea posible cargar los credenciales.

Requisitos de la carga de problemas:

R-Carga-1: El sistema permite al usuario cargar instancias de los problemas a través de archivos.

> *R-Carga-1.1:* El sistema permite al usuario cargar instancias del problema del Max-Cut a través de archivos.
>
> *R-Carga-1.2:* El sistema permite al usuario cargar instancias del problema del TSP a través de archivos.
>
> *R-Carga-1.3:* El sistema permite al usuario cargar instancias del problema de la mochila a través de archivos.
>
> *R-Carga-1.4:* El sistema permite al usuario cargar instancias del problema del coloreado de grafos a través de archivos.

R-Carga-2: El sistema permite al usuario cargar instancias del problema del Max-Cut de forma manual, solicitando los siguientes datos obligatorios:

> *R-Carga-2.1:* El número de nodos del problema.
>
> > *R-Carga-2.1.1:* Tiene que ser un número positivo.
> >
> > *R-Carga-2.1.2:* Tiene que ser un número mayor que 0.
>
> *R-Carga-2.2:* El número de repeticiones.
>
> > *R-Carga-2.2.1:* Tiene que ser un número positivo.

R-Carga-2.2.2: Tiene que ser un número mayor que 0.

R-Carga-2.3: Las conexiones entre los nodos, que constan de las siguientes partes obligatorias:

R-Carga-2.3.1: El nodo desde el que sale la conexión.

R-Carga-2.3.2: El nodo al que entra la conexión.

R-Carga-2.3.3: El valor de esta conexión.

R-Carga-2.3.3.1: Tiene que ser un número positivo.

R-Carga-2.3.3.2: Tiene que ser un número mayor que 0.

R-Carga-2.3.4: En caso de que alguno de estos datos no se proporcione, el sistema:

R-Carga-2.3.4.1: Impedirá la generación de la conexión.

R-Carga-2.3.4.2: Notificará al usuario de que debe introducir todos los datos de la conexión.

R-Carga-2.4: En caso de que alguno de estos datos no se proporcione o tenga un valor incorrecto y se intente resolver el problema, el sistema:

R-Carga-2.4.1: Impedirá la resolución de la instancia.

R-Carga-2.4.2: Notificará al usuario de que debe introducir todos los parámetros del problema de manera correcta.

R-Carga-3: El sistema permite al usuario cargar instancias del problema del TSP de forma manual, solicitando los siguientes datos obligatorios:

R-Carga-3.1: El número de nodos del problema.

R-Carga-3.1.1: Tiene que ser un número positivo.

R-Carga-3.1.2: Tiene que ser un número mayor que 0.

R-Carga-3.2: El número de repeticiones.

R-Carga-3.2.1: Tiene que ser un número positivo.

R-Carga-3.2.2: Tiene que ser un número mayor que 0.

R-Carga-3.3: Las conexiones entre los nodos, que constan de las siguientes partes obligatorias:

R-Carga-3.3.1: El nodo desde el que sale la conexión.

R-Carga-3.3.2: El nodo al que entra la conexión.

R-Carga-3.3.3: El valor de esta conexión.

R-Carga-3.3.3.1: Tiene que ser un número positivo.

R-Carga-3.3.3.2: Tiene que ser un número mayor que 0.

R-Carga-3.3.4: En caso de que alguno de estos datos no se proporcione, el sistema:

R-Carga-3.3.4.1: Impedirá la generación de la conexión.

R-Carga-3.3.4.2: Notificará al usuario de que debe introducir todos los datos de la conexión.

R-Carga-3.4: En caso de que alguno de estos datos no se proporcione o tenga un valor incorrecto y se intente resolver el problema, el sistema:

R-Carga-3.4.1: Impedirá la resolución de la instancia

R-Carga-3.4.2: Notificará al usuario de que debe introducir todos los parámetros del problema de manera correcta.

R-Carga-4: El sistema permite al usuario cargar instancias del problema del problema de la mochila de forma manual, solicitando los siguientes datos obligatorios:

R-Carga-4.1: El número de repeticiones.

R-Carga-4.1.1: Tiene que ser un número positivo.

R-Carga-4.1.2: Tiene que ser un número mayor que 0.

R-Carga-4.2: El peso máximo.

R-Carga-4.2.1: Tiene que ser un número positivo.

R-Carga-4.2.2: Tiene que ser un número mayor que 0.

R-Carga-4.3: Los objetos, que tienen los siguientes componentes obligatorios.

R-Carga-4.3.1: El valor del objeto.

R-Carga-4.3.1.1: Tiene que ser un número positivo.

R-Carga-4.3.1.2: Tiene que ser un número mayor que 0.

R-Carga-4.3.2: El peso del objeto.

R-Carga-4.3.2.1: Tiene que ser un número positivo.

R-Carga-4.3.2.2: Tiene que ser un número mayor que 0.

R-Carga-4.3.3: En caso de que alguno de estos datos no se proporcione, el sistema:

R-Carga-4.3.3.1: Impedirá la creación del objeto.

R-Carga-4.3.3.2: Notificará al usuario de que debe introducir todos los datos del objeto.

R-Carga-4.4: En caso de que alguno de estos datos no se proporcione o tenga un valor incorrecto y se intente resolver el problema, el sistema:

R-Carga-4.4.1: Impedirá la resolución de la instancia

R-Carga-4.4.2: Notificará al usuario de que debe introducir todos los parámetros del problema de manera correcta.

R-Carga-5: El sistema permite al usuario cargar instancias del problema del coloreado de grafos de forma manual, solicitando los siguientes datos obligatorios:

R-Carga-5.1: El número de vértices del problema.

R-Carga-5.1.1: Tiene que ser un número positivo.

R-Carga-5.1.2: Tiene que ser un número mayor que 0.

R-Carga-5.2: El número de colores del problema.

R-Carga-5.2.1: Tiene que ser un número positivo.

R-Carga-5.2.2: Tiene que ser un número mayor que 0.

R-Carga-5.3: El número de repeticiones.

R-Carga-5.3.1: Tiene que ser un número positivo.

R-Carga-5.3.2: Tiene que ser un número mayor que 0.

R-Carga-5.4: Las conexiones entre los nodos, que constan de las siguientes partes obligatorias:

R-Carga-5.4.1: El nodo desde el que sale la conexión.

R-Carga-5.4.2: El nodo al que entra la conexión.

R-Carga-5.4.3: En caso de que alguno de estos datos no se proporcione, el sistema:

R-Carga-5.4.3.1: Impedirá la generación de la conexión.

R-Carga-5.4.3.2: Notificará al usuario de que debe introducir todos los datos de la conexión.

R-Carga-5.5: En caso de que alguno de estos datos no se proporcione o tenga un valor incorrecto y se intente resolver el problema, el sistema:

R-Carga-5.5.1: Impedirá la resolución de la instancia

R-Carga-5.5.2: Notificará al usuario de que debe introducir todos los parámetros del problema de manera correcta.

Requisitos de la resolución de problemas:

R-Sol-1: El sistema permite al usuario utilizar algoritmos cuánticos para resolver los problemas

R-Sol-1.1: El sistema permite al usuario utilizar el algoritmo del QAOA en un simulador local para resolver los problemas

R-Sol-1.2: El sistema permite al usuario utilizar el algoritmo del QAOA en un simulador remoto para resolver los problemas

R-Sol-1.3: El sistema permite al usuario utilizar el algoritmo del QAOA en un ordenador cuántico real para resolver los problemas.

R-Sol-1.4: El sistema permite al usuario utilizar el annealer para resolver los problemas.

R-Sol-2: El sistema permite al usuario utilizar algoritmos clásicos para resolver los problemas

R-Sol-2.1: El sistema permite al usuario utilizar el algoritmo del annealing simulado para resolver los problemas.

R-Sol-2.2: El sistema permite al usuario utilizar el algoritmo de búsqueda local para resolver los problemas.

R-Sol-2.3: El sistema permite al usuario utilizar el algoritmo de descenso de gradiente para resolver los problemas.

R-Salida-1: El sistema proporcionará la siguiente información dependiendo de la posibilidad de resolver el problema utilizando el método seleccionado:

R-Salida-1.1: En caso de que la resolución se ejecute correctamente, el sistema proporcionará la siguiente información sobre la resolución de la instancia del problema:

R-Salida-1.1.1: El sistema proporcionará los valores asignados a las variables en la solución obtenida.

R-Salida-1.1.2: El sistema proporcionará el valor de la energía de la solución obtenida.

R-Salida-1.1.3: El sistema proporcionará el estado de resolución de la instancia del problema con el método utilizado.

R-Salida-1.1.3.1: En el caso de que se haya utilizado un método proporcionado por IBMQ, el sistema elegirá un estado dentro de una serie de valores:

R-Salida-1.1.3.1.1: En caso de que la ejecución sea correcta, el estado será "SUCCESS".

R-Salida-1.1.3.1.2: En caso de que la ejecución no se pueda realizar, el estado será "INFEASIBLE".

R-Salida-1.1.3.1.3: En caso de que la ejecución sea errónea, el estado será "FAILURE".

R-Salida-1.1.3.2: En el caso de que se haya utilizado un método ofrecido por D-Wave y la ejecución sea correcta, el sistema proporcionará el estado "SUCCESS".

R-Salida-1.2: En caso de que la resolución no haya podido realizarse debido a un error, el sistema notificará al usuario de que se ha producido un problema.

Requisitos no funcionales

RNF-Nav-1: El sistema será compatible con los principales navegadores web.

RNF-Nav-1.1: El sistema será compatible con Google Chrome.

RNF-Nav-1.2: El sistema será compatible con Mozilla Firefox.

RNF-Nav-1.3: El sistema será compatible con Microsoft Edge.

RNF-Nav-1.4: El sistema será compatible con Opera.

Identificación de actores del sistema

Se han identificado los siguientes actores:

- IBM Quantum Experience (IBMQ)

  Este actor interviene en numerosos casos de uso, como en la carga en sesión de la cuenta de IBMQ, o a la hora de utilizar los métodos que nos proporciona IBM, como por ejemplo los simuladores de QAOA, entre otros.

- D-Wave

  Aparece en los casos de uso en los que utilizamos servicios de D-Wave. Es necesario para la carga en sesión de las credenciales, y para el uso del annealer, entre otros métodos.

- Usuarios de la aplicación

  Representa el usuario que utiliza la aplicación web. Este hará uso de todas las funcionalidades que ofrece el sistema.

Especificación de casos de uso

Caso de uso 1 – Usuario carga sesión en plataformas

Nombre del caso de uso	Usuario carga la sesión en las plataformas
Descripción	El usuario quiere utilizar el token que le ha proporcionado la plataforma en la aplicación. Una vez introducido y cargado el token a través de la interfaz de la aplicación, se habrá registrado correctamente y podrá hacer uso de los métodos asociados a la plataforma.

Tabla 29. Caso de uso 1: cargar sesión

Caso de uso 2 – Usuario carga problema a través de archivo

Nombre del caso de uso	Usuario carga problema a través de archivo

Descripción	Un usuario busca introducir datos sobre una instancia de un problema a través de un archivo, el cual está formateado correctamente, de acuerdo a las indicaciones. Tras especificar el archivo que contiene los datos y cargarlo, la aplicación actualizará los parámetros de la instancia con la que se está trabajando.

Tabla 30. Caso de uso 2: cargar problema a través de archivo

Caso de uso 3 – Usuario carga problema de forma manual

Nombre del caso de uso	Usuario carga problema de forma manual
Descripción	Un usuario busca introducir datos sobre una instancia de un problema de manera manual. Para ello, hará uso de los componentes que ofrece la página del problema para ir construyendo la instancia, y esta irá actualizándolos a medida que se cambian.

Tabla 31. Caso de uso 3: cargar problema manualmente

Caso de uso 4 – Usuario elige método para resolver el problema

Nombre del caso de uso	Usuario elige método para resolver el problema
Descripción	Tras haber cargado el problema, ya sea de forma manual o a través de un archivo, el usuario puede elegir el método que quiere utilizar a través de la interfaz ofrecida por la aplicación, que da a elegir de entre varias opciones. Una vez se ha seleccionado el método, el usuario puede, por medio de la interfaz, iniciar la resolución de la instancia utilizando ese método.

Tabla 32. Caso de uso 4: elegir método para resolver el problema

A continuación se muestran los diagramas de casos de uso:

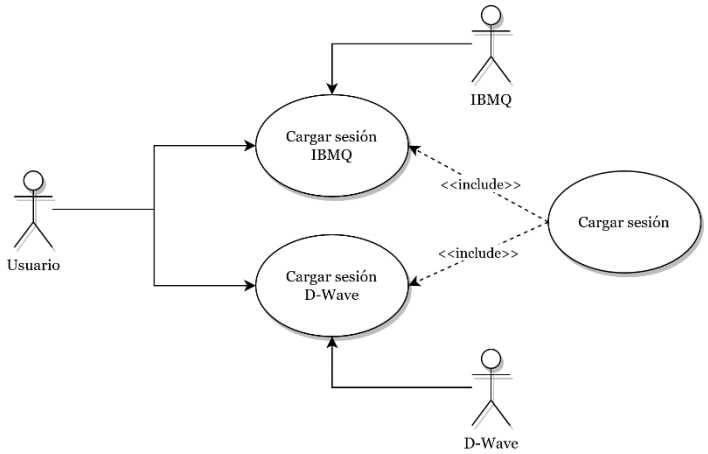


Figura 10.2. Casos de uso: carga de sesión

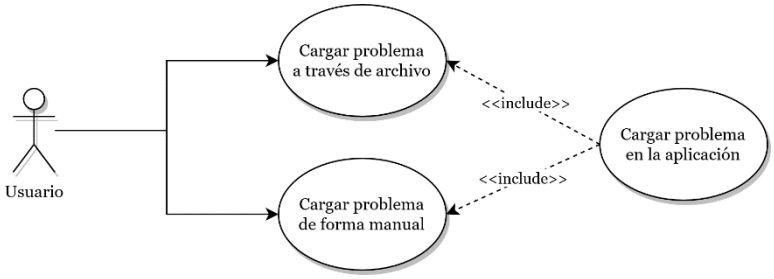


Figura 10.3. Casos de uso: cargar problema

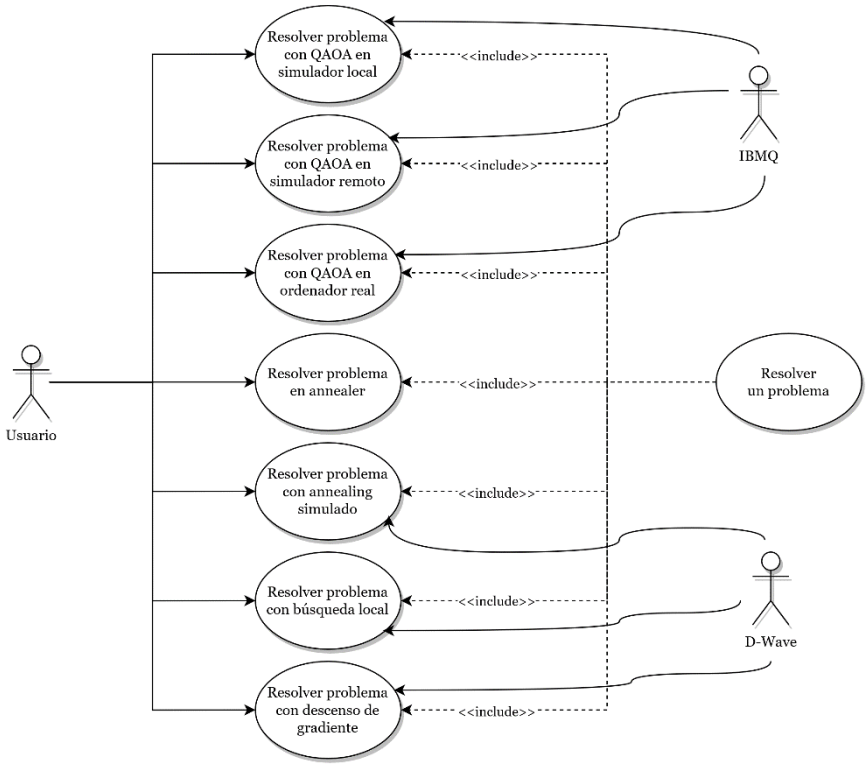


Figura 10.4. Casos de uso: resolver problema

Tecnologías utilizadas

En esta sección se detallan las tecnologías que se han escogido para la realización de la aplicación web. Esto incluye lenguajes de programación, herramientas y programas.

Lenguajes de programación, librerías y frameworks

Los lenguajes con los que se ha construido la aplicación son

- Python (versión 3.9.13)
- AJAX
- CSS
- JavaScript
- HTML5

Es importante tener la versión exacta de Python ya que no usar la misma puede resultar en errores, por ejemplo al no tener compatibilidad con las librerías.

Las librerías y frameworks a destacar son:

- *Flask:* framework para la creación de la aplicación web. Permite gestionar solicitudes HTTP y gestionar rutas. Es una opción muy buena para proyectos de esta índole, ya que es muy sencillo y rápido de utilizar.

- *Qiskit:* framework desarrollado por IBM para el trabajo con ordenadores cuánticos. Ofrece funcionalidad para la programación, simulación y ejecución de algoritmos cuánticos, ya sea en dispositivos reales o simulados. También permite trabajar con circuitos cuánticos. Es una de las herramientas más potentes que existen para la experimentación con computación cuántica ahora mismo.

- *dimod:* librería que permite la resolución de problemas de optimización cuadrática (QUBO) y modelos de Ising. A través de las herramientas que proporciona se pueden formular y resolver problemas que pueden mapearse a estructuras cuadráticas binarias. También proporciona acceso a diversos solucionadores, tanto los basados en computación cuántica que ofrece la empresa D-Wave como los solucionadores clásicos. Es una opción muy completa para la resolución de problemas de optimización.

- *DOcplex:* librería desarrollada por IBM que contiene muchas herramientas para resolver problemas de optimización. En concreto el uso que se le ha dado es el de modelar las formulaciones de los problemas usando las restricciones y términos de penalización propios de la definición de los problemas que se ha expuesto en este trabajo.

Programas

Otras herramientas que fueron muy útiles en la realización de la aplicación y la documentación incluyen:

- *Draw.io:* para la creación de los numerosos diagramas y mockups.
- *GitHub:* para el control de versiones.
- *Microsoft Word:* para la elaboración de la documentación.

Cabe mencionar que muchas de las elecciones sobre las tecnologías que se han tomado han sido debido a que el alumno ya tenía experiencia utilizándolas. Al tratarse de una aplicación simple bastó en gran parte utilizar estas herramientas, sin necesidad de adentrarse en nuevas.

Definición de interfaces de usuario

La interfaz del usuario es una de las partes fundamentales de una aplicación, ya que es la que conecta a la persona que la esté utilizando con el resto de la funcionalidad, por lo que su diseño es un paso importante.

Descripción de la interfaz

Una técnica que se puede utilizar para describir inicialmente el diseño de las pantallas es el uso de bocetos, o mockups. Estos representan, de forma algo general y simplificada, la estructuración que va a tener la pantalla, y dónde se colocaría cada elemento.

Esta técnica es útil ya que permite al cliente hacerse una idea del aspecto que va a tener la aplicación sin necesidad de construirla, o de hacer un prototipo, abaratando costes y reduciendo el tiempo. A continuación se presentan los mockups que se han construido para las pantallas de la interfaz.

Página home (principal)

La primera página que se ha diseñado es la que primero visita el usuario una vez ejecuta la aplicación.

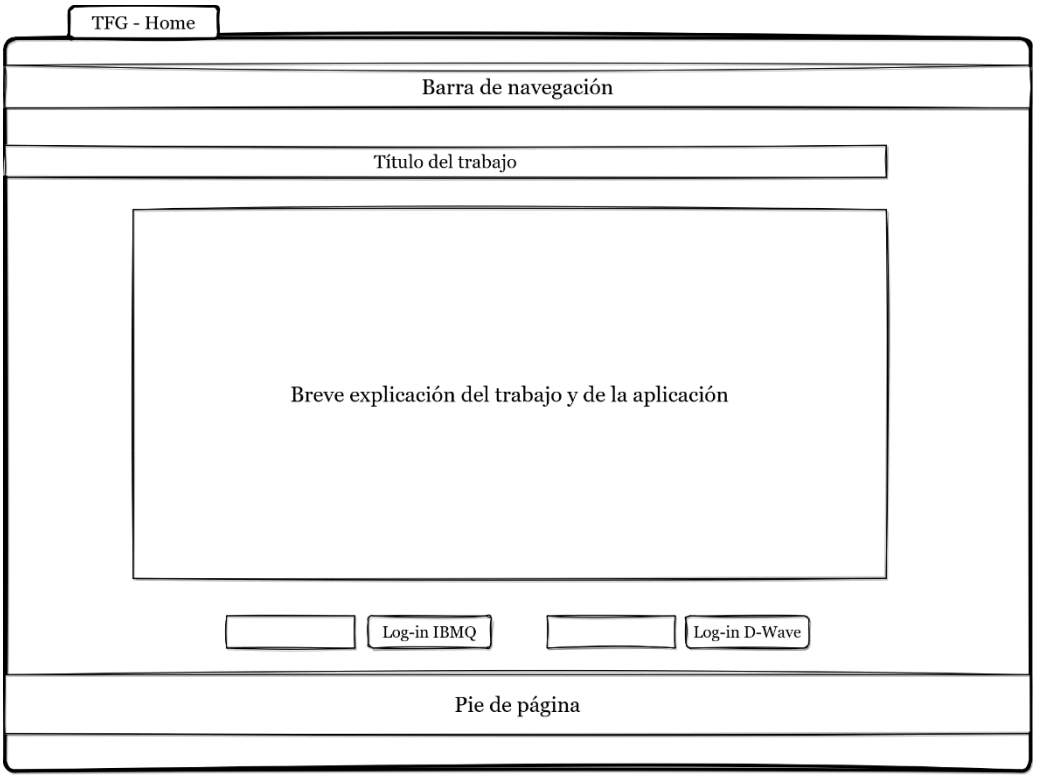


Figura 10.5. Mockup de la página home

Esta pantalla cuenta, mayoritariamente, con información sobre este TFG y la funcionalidad de la aplicación, para explicar el contexto en el que se ha realizado.

También cuenta con dos inputs y dos botones debajo de dicha explicación, cuya funcionalidad es la de cargar en sesión las cuentas de IBMQ y D-Wave, necesarias para la posterior ejecución de los métodos que hacen uso de las máquinas que proporcionan estas empresas.

Por último, la página tiene en la parte superior una barra de navegación, y en la inferior un pie de página.

Página de un problema (general)

El otro tipo de página con el que cuenta la aplicación web es la de cada problema. Cada una de ellas sigue una estructura general, muy parecida, variando solamente en la parte de la carga manual de las instancias y en los datos de la misma, que cambian dependiendo de los parámetros que se necesiten para el problema.

En esta página (véase Figura 10.6) tenemos algunos componentes más que en la página principal. En la mitad superior tenemos de nuevo una barra de navegación, y por debajo de ella un recuadro que contendrá información sobre el problema en específico, así como un ejemplo de resultado, para explicar cómo interpretarlo, y la explicación del formato que debe tener el archivo, en caso de que queramos cargar un problema a través de él.

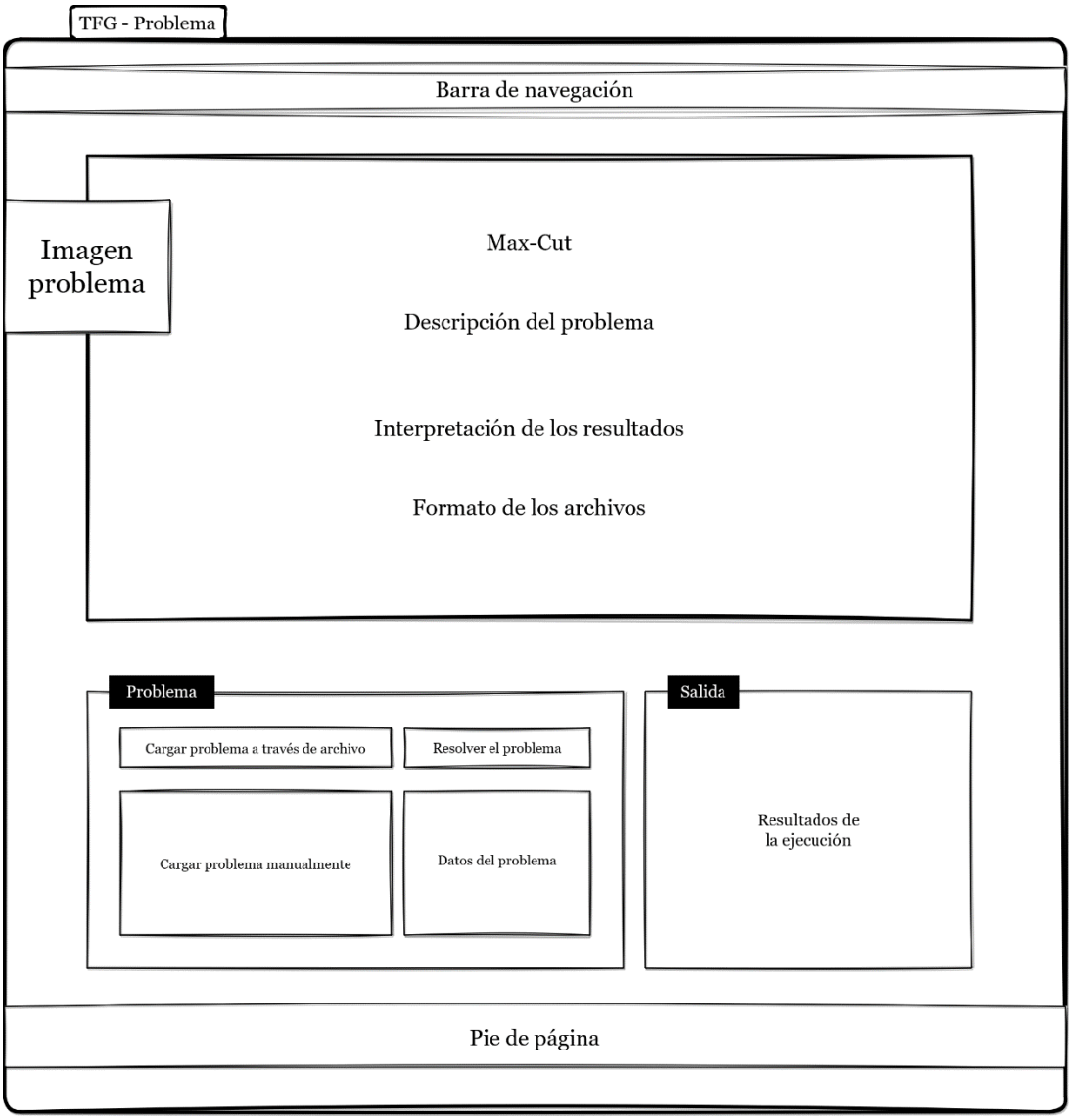


Figura 10.6. Mockup de la página de los problemas

En la mitad inferior, contamos con dos recuadros principales y un pie de página. En el recuadro etiquetado como "Problema" tendremos todo lo necesario para cargar la instancia que queremos resolver, y para elegir el método que queremos utilizar.

Dentro de ese recuadro, el usuario tendrá los componentes necesarios para cargar el problema a través de un archivo, para ir construyendo el problema manualmente, para ver los datos del problema cargado, y para elegir un método y resolver el problema. En el recuadro siguiente, denominado como "Salida", aparecerá la solución generada por el método, o información de los errores que se produzcan.

A continuación se va a hablar de la página específica de cada problema, en la que se va a hacer hincapié en las partes que cambian respecto a la página del diseño general.

Página del Max-Cut

En esta página contamos con un botón de input para elegir el archivo que contiene la instancia del problema, y otro para cargarlo.

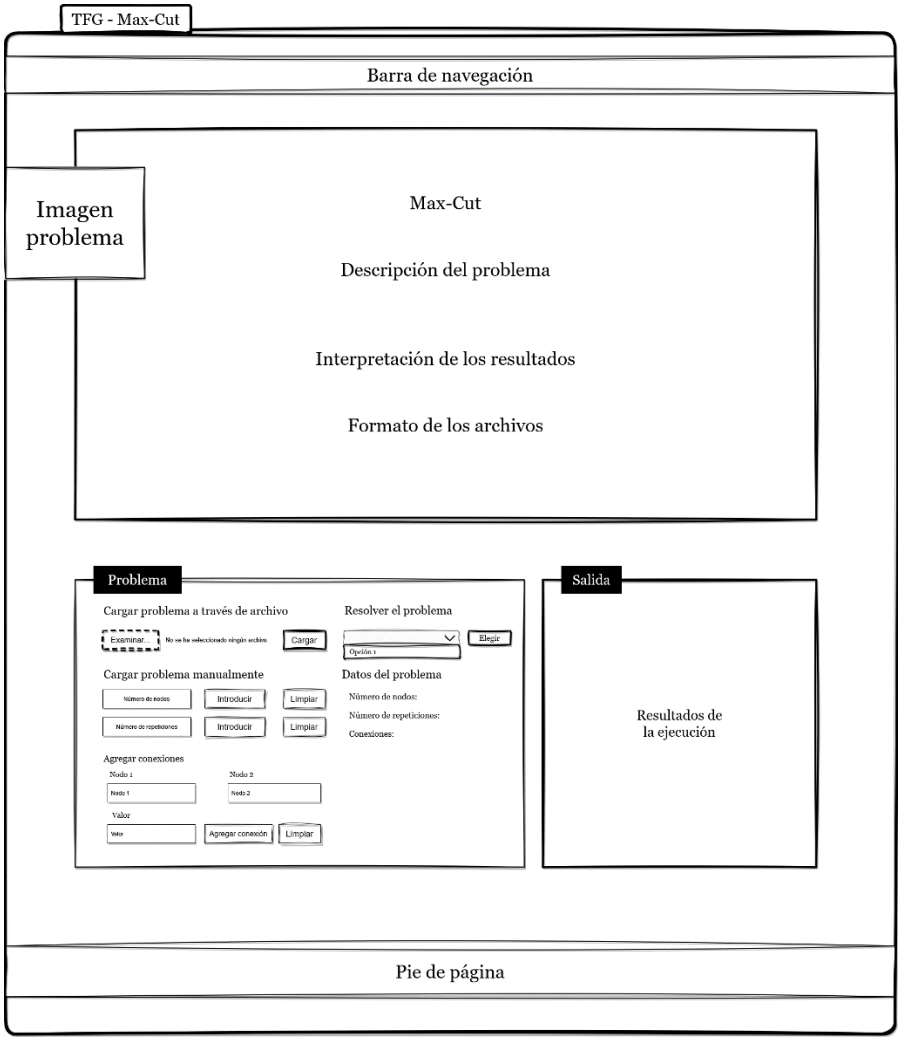


Figura 10.7. Mockup de la página del Max-Cut completa

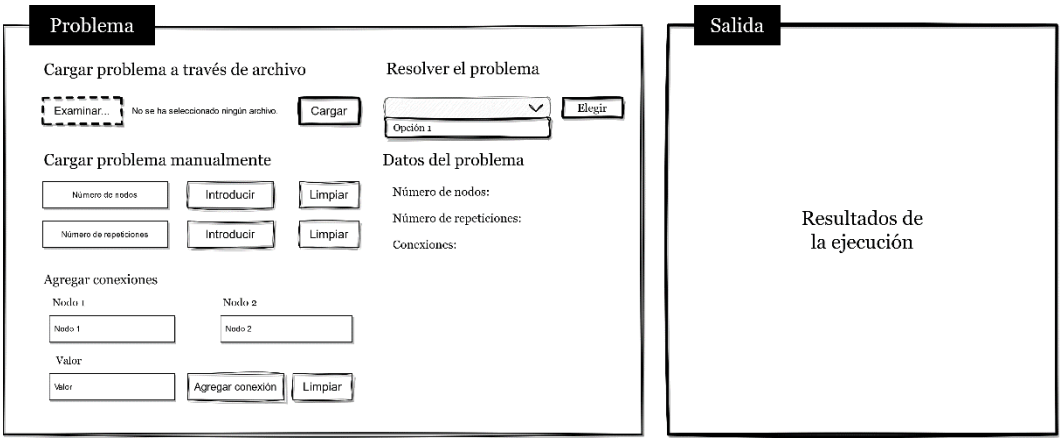


Figura 10.8. Mockup de la página del Max-Cut, parte inferior

Debajo tenemos la opción de cargar la página de manera manual, con botones para introducir o limpiar el número de nodos, el número de repeticiones, y para añadir conexiones, especificando los nodos de la conexión y el valor de la misma.

A la derecha tenemos un desplegable desde el que podemos elegir la opción que queremos utilizar para resolver el problema, y el botón de "Elegir", que iniciará el proceso de resolución con el método elegido.

Debajo tenemos información sobre el problema que hay cargado, la cual se irá actualizando a medida que cambiemos los parámetros.

Página del TSP

Esta página tiene muchas similitudes con la del Max-Cut, ya que utilizan los mismos parámetros.

Figura 10.9. Mockup de la página del TSP completa

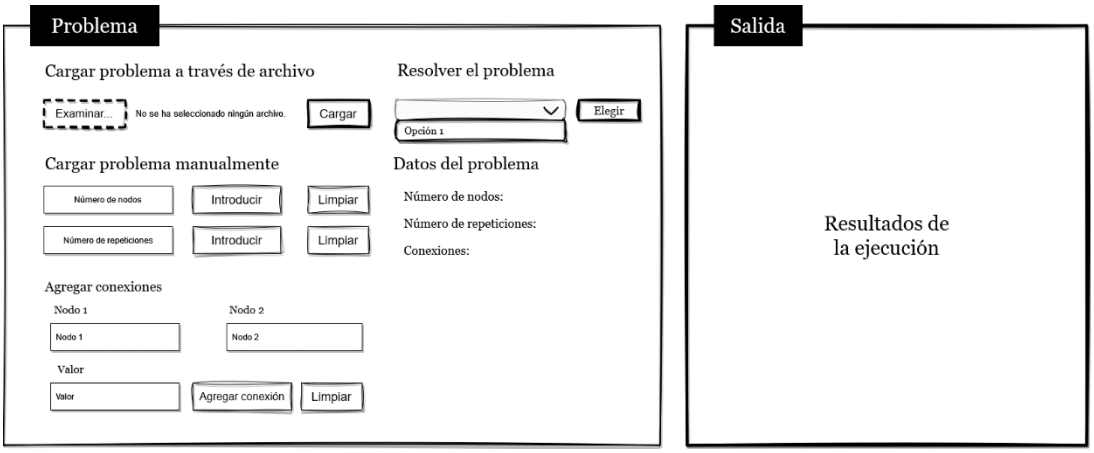


Figura 10.10. Mockup de la página del TSP, parte inferior

Ambas páginas cuentan con las mismas funcionalidades y botones en el mockup.

Página del problema de la mochila

En este caso tenemos algo de variación, ya que se usan otros parámetros.

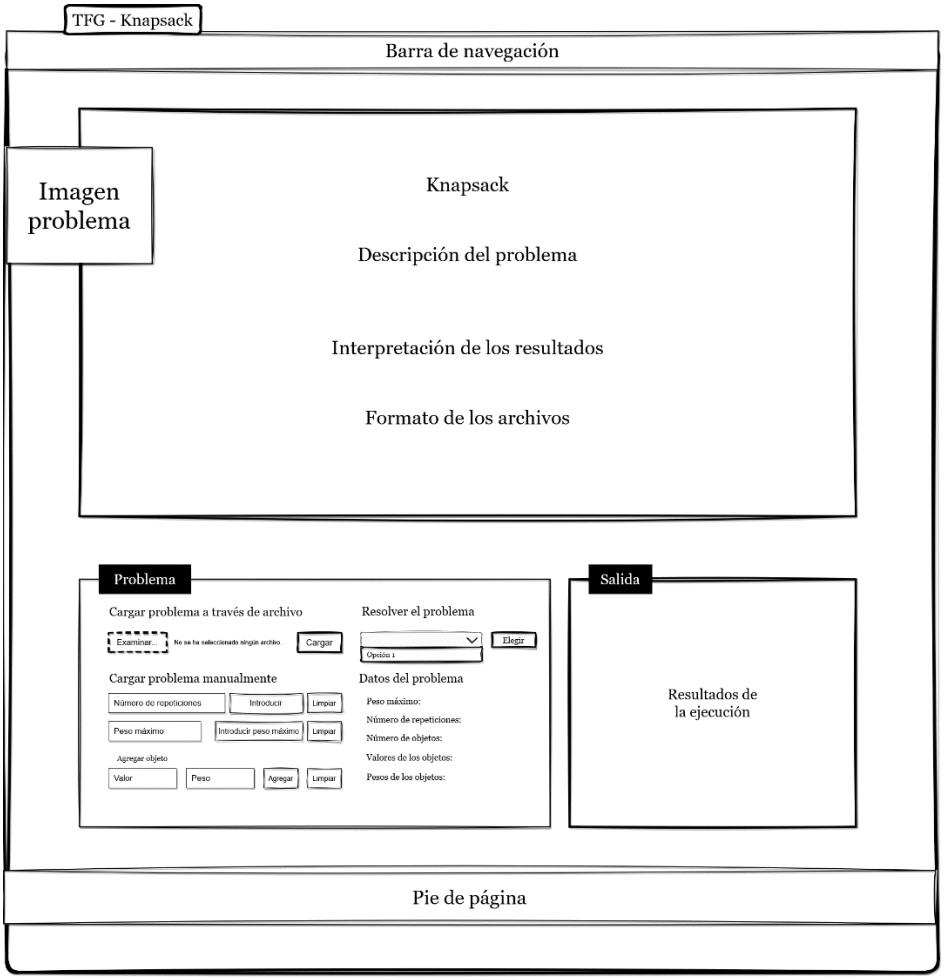


Figura 10.11. Mockup de la página del problema de la mochila completa

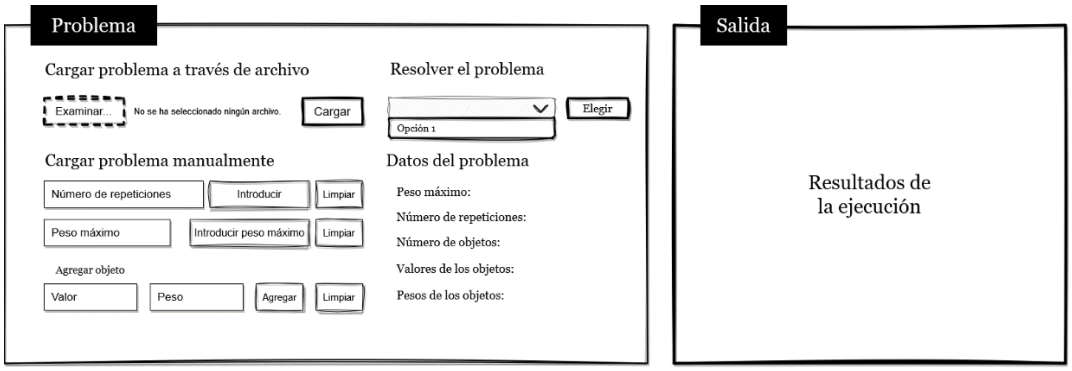


Figura 10.12. Mockup de la página del problema de la mochila, parte inferior

Tenemos botones y cuadros de texto para especificar y limpiar el número de repeticiones y el peso máximo del problema, y para los objetos, en los que se introduce su valor y peso asociados. También cambia la información que se muestra del problema.

Página del problema del coloreado de grafos

En este caso también cambian los parámetros necesarios.

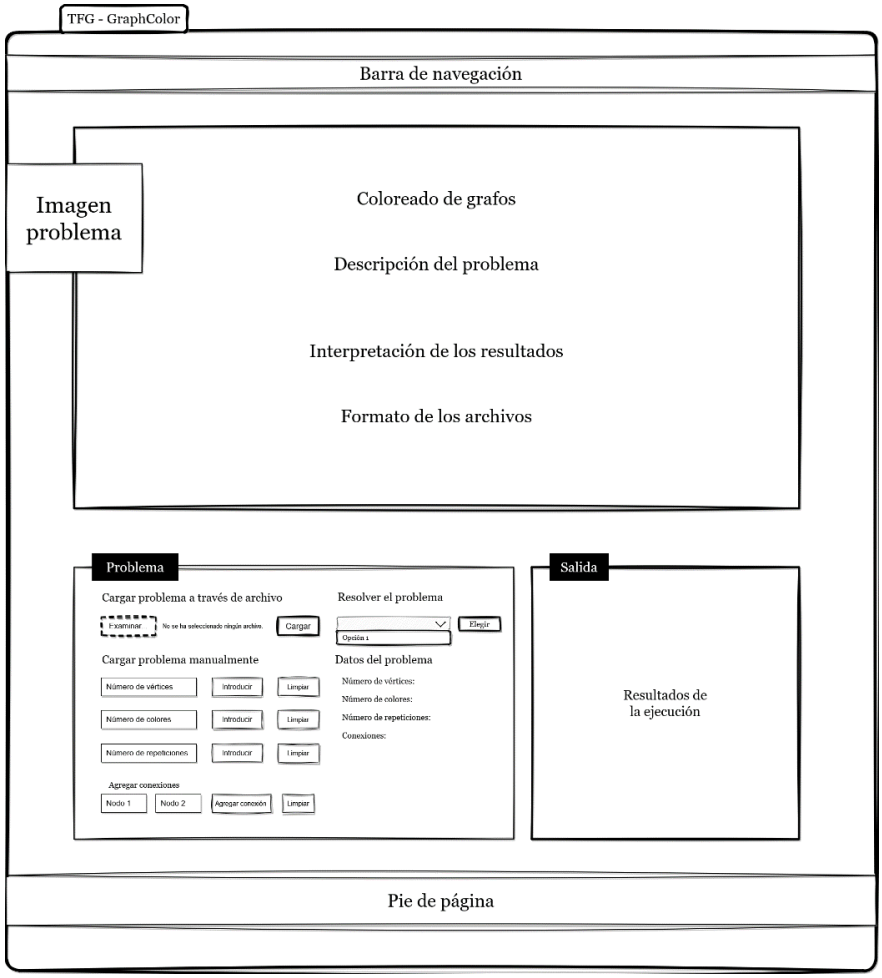


Figura 10.13. Mockup de la página del problema del coloreado completa

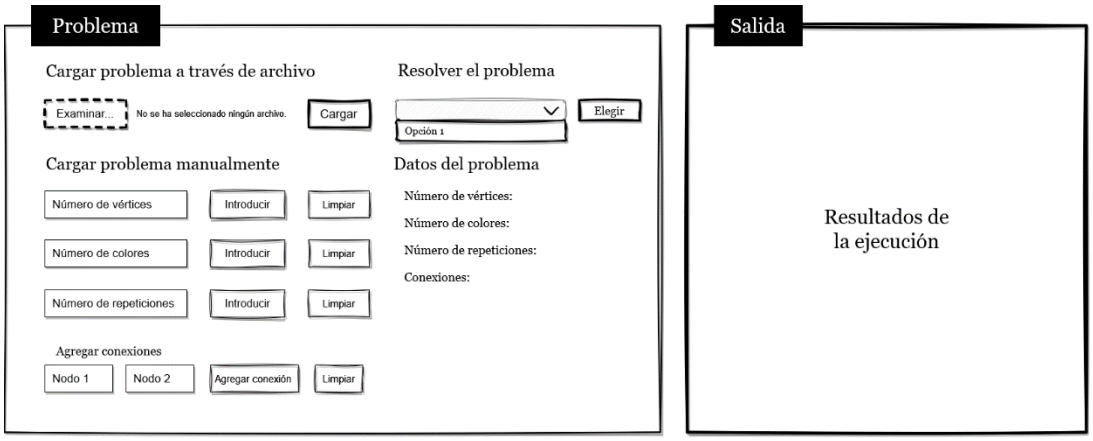


Figura 10.14. Mockup de la página del problema del coloreado, parte inferior

Al haber parámetros distintos, también cambian los botones y campos de texto que se necesitan. En este caso hay botones para introducir y limpiar el número de vértices, el de colores, el de repeticiones, y las conexiones entre nodos.

Todos estos datos se muestran a la derecha, debajo de "Datos del problema".

Descripción del comportamiento de la interfaz

En esta sección se detalla la manera en la que la aplicación procesa y informa al usuario de los errores.

En la página principal, para que el usuario pueda saber el estado de la sesión en los servicios de IBM y D-Wave, debajo de los inputs se especifica si está logeado o no en ellos. En caso de que un usuario no logeado introduzca los token erróneos, este seguirá en el estado "no logeado".

En las páginas de los problemas pueden ocurrir una variedad de errores diferentes. En primer lugar, los inputs que requieran de valores numéricos mayores de 0 están validados de manera que no se puedan introducir valores distintos a esos, ni escribiéndolos ni copiando y pegándolos, para prevenir errores.

También pueden ocurrir errores por introducir parámetros sin especificar todos los componentes que forman parte de él. Por ejemplo, para las conexiones entre nodos, es necesario introducir ambos nodos, y en algunos casos el valor de la conexión. En caso de que uno de estos valores falte, el sistema no añadirá dicha conexión e informará al usuario a través de un mensaje de que los datos del parámetros están incompletos.

De manera parecida, para la resolución de la instancia de un problema es necesario que hayan especificado todos los parámetros, y que estos tengan valores correctos. En el caso de que uno de estos parámetros no se haya especificado o no tenga un valor

esperado el problema no se resolverá, y el sistema informará al usuario de que tiene que completar todos los parámetros y los valores que se esperan en cada parámetro.

Diagrama de navegabilidad

A continuación se muestra el diagrama de navegabilidad, en el que se pueden ver las distintas formas de pasar de una pantalla a otra. Desde cualquier pantalla se puede acceder a las otras a través de la barra de navegación (navbar).

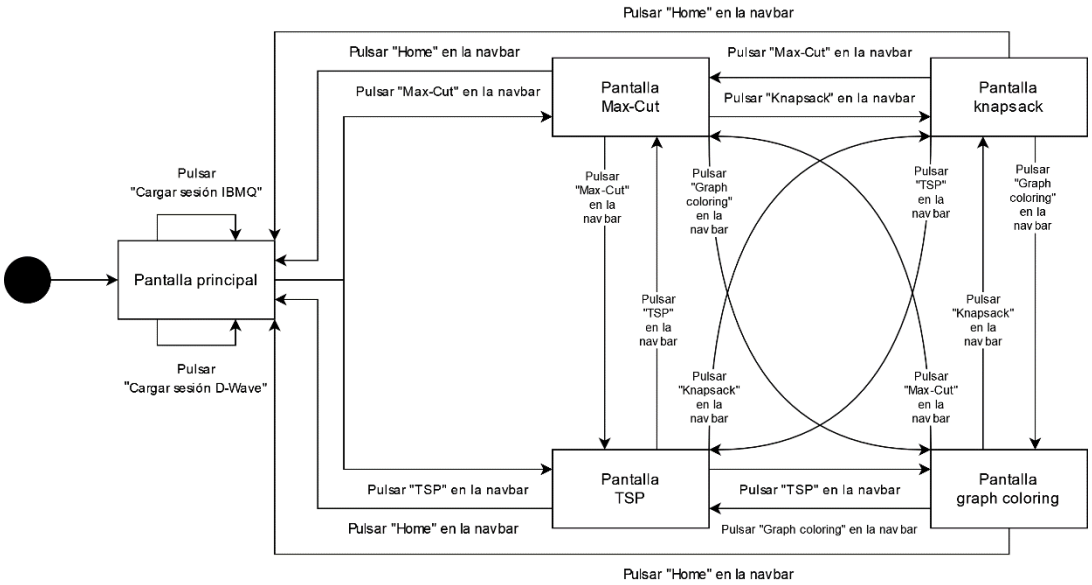


Figura 10.15. Diagrama de navegabilidad

Definición de manuales de usuario

A continuación se muestran numerosos manuales que explican varios aspectos del funcionamiento de la aplicación, desde cómo instalarla hasta cómo entender los resultados que se obtienen.

Manual de instalación y ejecución

En primer lugar hay que hablar de cómo se puede instalar la aplicación. Para poder hacer esto, primero tendremos que acceder al repositorio de GitHub (véase 8.2. Difusión de resultados).

Desde ahí podemos copiar la URL que nos permitirá clonar el repositorio en nuestro equipo (la cual tiene el aspecto de https://github.com/UO277653/TFG.git). También podemos descargar directamente los contenidos del repositorio en un ZIP desde GitHub, a través de la misma pestaña que nos muestra el enlace del repositorio, "Code", que se muestra en la siguiente ilustración:

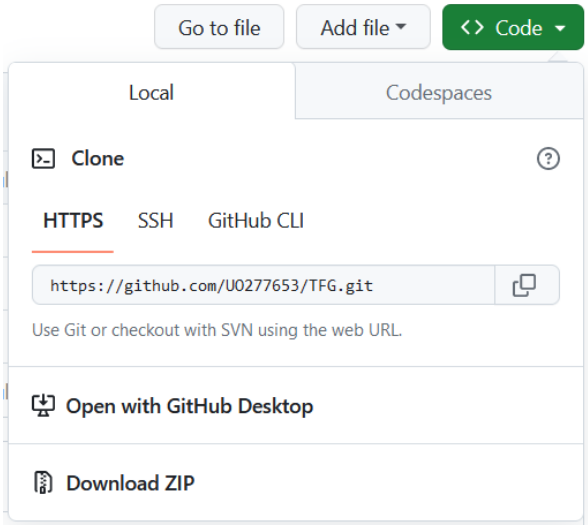


Figura 10.16. Pestaña "Code" de GitHub

Una vez tengamos el contenido en nuestro equipo, procederemos a instalar las librerías necesarias en el entorno para el correcto funcionamiento del código.

Para evitar tener que instalarlas manualmente de manera individual, se pone a disposición un archivo de texto "requirements.txt", el cual se puede utilizar en conjunto con pip para instalar automáticamente todas las librerías y versiones incluidas en él.

Esto se puede hacer a través del comando "pip install -r requirements.txt" ó "py -m pip install -r requirements.txt".

Cabe mencionar que también es recomendable tener instalada la versión de Python que se especifica en el apartado Metodología de trabajo, debido a que muchas de las librerías utilizadas están en constante desarrollo, y el no tener la misma versión puede dar lugar a errores.

Ya tenemos el entorno preparado con todo lo necesario para ejecutar la aplicación. Esto se puede hacer navegando en el terminal hasta la carpeta "webapp", por ejemplo, a través del comando "cd .\webapp\", y una vez ahí ejecutaremos la aplicación con "py .\app.py".

En caso de que no suceda ningún error, podremos acceder a la página a través de la URL que se especificará en la terminal (por ejemplo, http://127.0.0.1:5000/).

```
(base) PS C:\Users\adria\OneDrive\Escritorio\UNIVERSIDAD\TFG\App> cd .\webapp\
(base) PS C:\Users\adria\OneDrive\Escritorio\UNIVERSIDAD\TFG\App\webapp> py .\app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Figura 10.17. Ejemplo de aplicación ejecutándose correctamente

Manual de usuario

Obtener los tokens y cargar la sesión en la aplicación

En la página principal, hay unos botones para cargar en sesión las cuentas de IBM Quantum y D-Wave, necesarias para poder ejecutar los dispositivos que estas empresas nos ponen a nuestra disposición.

El usuario debe introducir los token de acceso a las API que se proporcionan en las páginas web de estos servicios, y posteriormente pulsar el botón respectivo para cargar la sesión. Para obtener estos token, es necesario registrarse en la página de IBM Quantum y en la de D-Wave.

Una vez tengamos la cuenta, el token se puede encontrar en el lugar señalado en las siguientes figuras:

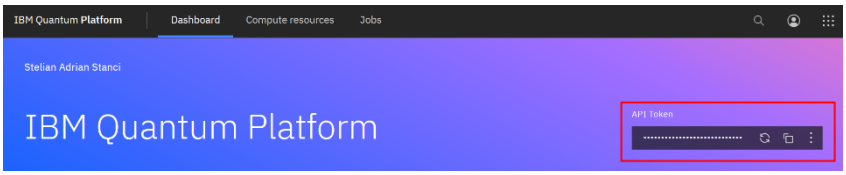


Figura 10.18. Localización del token de IBMQ

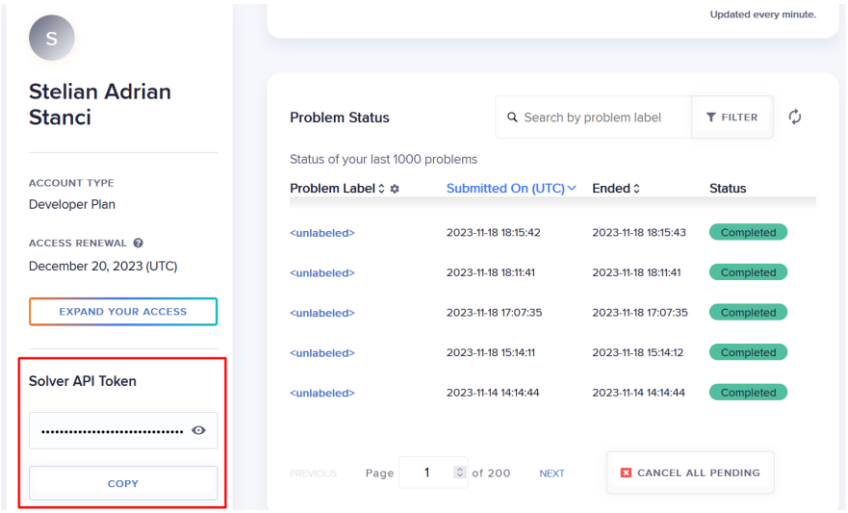


Figura 10.19. Localización del token de D-Wave

Al entrar a la página se verificará el estado de la sesión y se informará al usuario de si está logeado correctamente en ambos servicios.

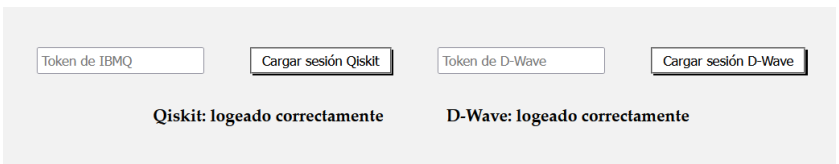


Figura 10.20. Ejemplo de carga en sesión correcta

Uso de las páginas de los problemas y respuesta a errores

En estas páginas hay mas posibilidades de interacción para el usuario. En la mitad inferior tiene la funcionalidad respectiva a la carga de problemas y la elección de un método.

Para cargar el problema a través de un archivo, el usuario debe pulsar el botón de "Examinar", el cual le abrirá la ventana de selección de archivo. El formato que tiene que seguir este archivo se especifica en la propia página del problema, y puede cambiar dependiendo de los parámetros que necesite (se ponen a disposición archivos de ejemplo para el formato de cada problema en el repositorio). Una vez seleccionado el archivo, hay que pulsar el botón "Cargar", y ya estaría listo para resolverse.

Para cargar el problema manualmente, los inputs del usuario variarán en función de los parámetros que se tengan que introducir. Estarán compuestos de diversos campos de textos y botones, cuyo contenido está explicado bien por títulos encima del campo o por *placeholders*, que desaparecerán una vez el usuario comience a rellenar el campo.

Respecto a la elección de un método, esta se hará a través de un combobox, que contiene todas las opciones posibles, y al lado un botón, "Elegir", que resuelve el problema cargado con el método seleccionado.

En el caso de que se produzca un error en la aplicación, el sistema notificará al usuario a través del recuadro de la salida.

Hay una variedad de errores que se pueden producir, como por ejemplo que el usuario intente añadir un parámetro que tenga varios componentes sin especificarlos todos (por ejemplo una conexión), que se intente resolver un problema sin haber introducido todos los parámetros, o que estos no tengan valores válidos (en cuyo caso el sistema indicará al usuario cuáles son los valores esperados), o que haya un error durante la ejecución del método.

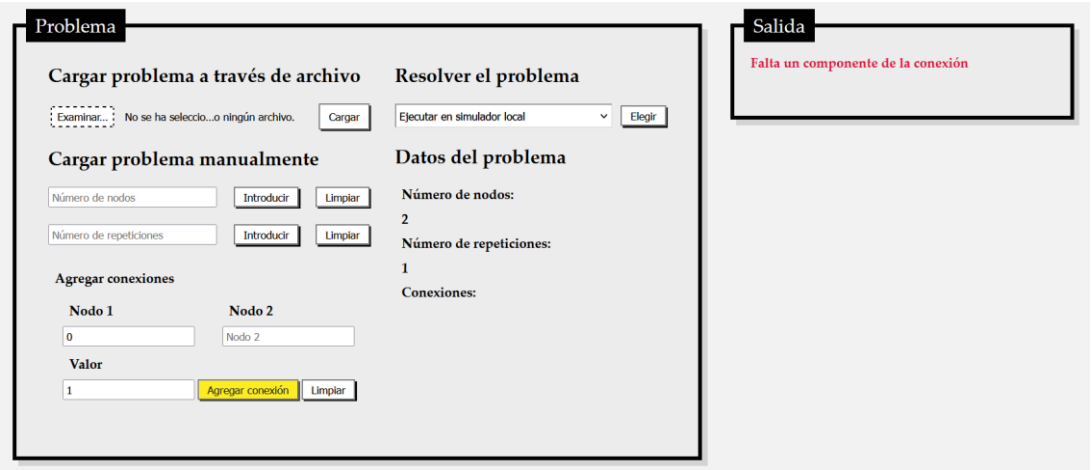


Figura 10.21. Ejemplo de error por parámetro incompleto

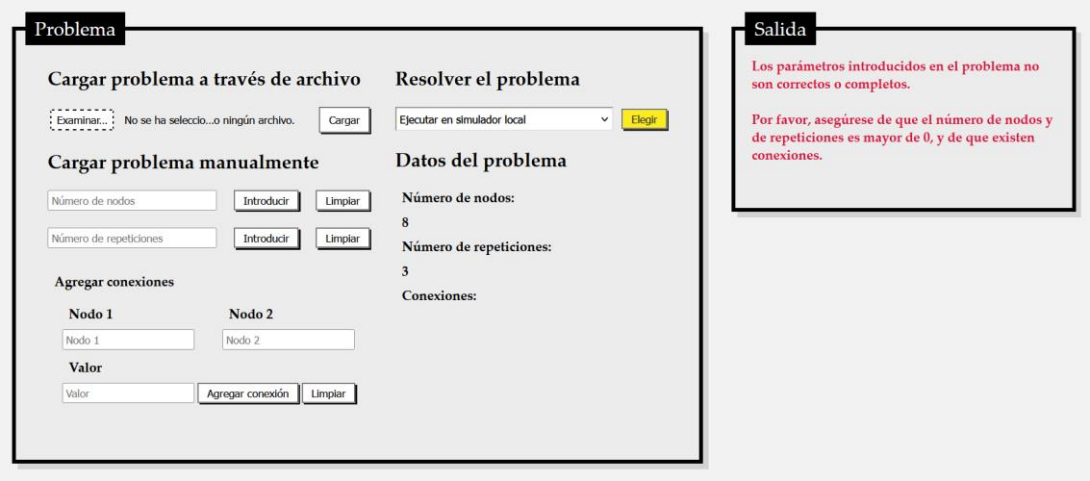


Figura 10.22. Ejemplo de error por parámetros incorrectos

Interpretación de los resultados

Los resultados que se obtienen tras resolver un problema pueden dar lugar a confusiones, especialmente las primeras veces. En ocasiones, también puede suceder que un método devuelva los resultados de una manera distinta a otro.

Para mitigar esta posible falta de claridad, en la aplicación se muestran una serie de ejemplos de salidas para cada problema que se pueden obtener utilizando los métodos en los que se explica su significado.

Ampliaciones

Las funcionalidades actuales de la aplicación cumplen con los objetivos que se tenían en mente en un principio. El usuario puede logearse en los servicios necesarios, cargar las instancias que quiera de los problemas que se han tratado en el estudio, y también puede resolverlos utilizando métodos que, entre otros, incluyen los mismos que se han analizado en este trabajo.

A pesar de que cumple con estos objetivos, la aplicación podría mejorarse de manera considerable, por ejemplo añadiendo explicaciones más detalladas del funcionamiento de los métodos, para que el usuario pueda comprender mejor cómo se obtienen los resultados.

En cuanto a los resultados, podría perfeccionarse la forma en la que se presentan, por ejemplo, generando un gráfico más visual en el que se pueda ver de manera más intuitiva la solución propuesta.

Un caso de esto podría ser, en el Max-Cut, teniendo en cuenta las asignaciones de los vértices a cada grupo, podría usarse ese output y procesarlo para generar a través de código un gráfico en el que se muestren los vértices y sus conexiones, y cada vértice reciba un color distinto dependiendo del grupo al que pertenece. Esto ayudaría a

entender mucho mejor las soluciones, comparado con la forma en la que se expresan ahora, que es más propensa a dar lugar a confusiones.

También se podría expandir la aplicación para abarcar más problemas y algoritmos, dando más opciones al usuario para que pueda experimentar. Esto enriquecería el valor de la aplicación en general.

Al no ser la aplicación el objetivo central del TFG, hay algunos aspectos que no están tan optimizados como debería, o que se podrían mejorar. En futuras versiones estos se refinarían, con el fin de dar lugar a un producto final de mayor calidad.