Evaluación de la viabilidad técnica y optimización del algoritmo de Shor en procesadores NISQ

Un estudio comparativo de fidelidad y transpilación en IBM Quantum

Christopher Andrey Villota Freire

Departamento de Física, Facultad de Ciencias Exactas y Naturales
Universidad de Nariño
Director: Jorge Hernán López Melo

Sustentación de Trabajo de Grado – San Juan de Pasto, Colombia
Junio de 2026

Contenido

Introducción y objetivos

Introducción: del algoritmo ideal al hardware actual

Universidad de Nariño
FUNDADA EN 1904

En 1994, Peter Shor demostró que un computador cuántico ideal factoriza enteros en tiempo polinomial, abriendo una brecha exponencial frente a los métodos clásicos [Shor, 1994].

- Tres décadas después, la ventaja cuántica útil sigue condicionada por las restricciones físicas de los dispositivos disponibles [Preskill, 2018].
- Hoy es posible ejecutar circuitos en procesadores reales de IBM Quantum vía Qiskit, accesibles de forma remota desde cualquier lugar.
- Existe una escasez de trabajos experimentales en computación cuántica reportados desde la Universidad de Nariño y la región.

Caso de estudio

Factorización de $N = 15$, la instancia pedagógica canónica de Shor: pequeña, pero con circuitos de profundidad considerable que la convierten en una prueba exigente para el hardware NISQ.

Planteamiento del problema

Universidad de Nariño
FUNDADA EN 1904

El algoritmo de Shor exige circuitos profundos (exponenciación modular + QFT). Su ejecución directa tiende a exceder los tiempos de coherencia del hardware actual, produciendo salidas dominadas por ruido.

La brecha entre la formulación teórica y la ejecución en hardware NISQ no son dos problemas independientes, sino la manifestación de una misma causa física.

Pregunta de investigación

¿Cuál es el desempeño y la viabilidad técnica de implementar versiones optimizadas para NISQ del algoritmo de Shor (enteros pequeños) en procesadores de IBM, comparando su fidelidad, profundidad de circuito y probabilidad de éxito frente a simulaciones ideales y modelos de ruido?

Objetivo general

Universidad de Nariño
FUNDADA EN 1904

Evaluar la viabilidad técnica y el desempeño de implementaciones educativas y optimizadas del algoritmo de Shor para enteros pequeños ($N = 15$) en procesadores cuánticos NISQ de IBM, mediante un análisis comparativo de fidelidad, profundidad de circuito y probabilidad de éxito entre simuladores con modelos de ruido y la ejecución en hardware real.

Objetivos específicos

Universidad de Nariño
FUNDADA EN 1904

1. **Diseñar y caracterizar** en Qiskit los bloques fundamentales (QFT y exponenciación modular), analizando el impacto de la transpilación avanzada (SABRE) sobre la profundidad y el conteo de CNOT.
2. **Implementar optimización híbrida** con pre-procesamiento clásico para identificar los coeficientes a que minimicen la complejidad del circuito y hagan viable el caso $N = 15$.
3. **Ejecutar** las versiones optimizadas en *Fake Backends* (simuladores con ruido real) y en hardware de IBM, aplicando supresión de errores (*Dynamical Decoupling* y *Pauli Twirling*).
4. **Cuantificar la degradación** de la señal frente a la simulación ideal usando la *Fidelidad de Hellinger* y la *Probabilidad de Éxito Experimental*.

Marco referencial

La era NISQ: qubits físicos frente a qubits lógicos

Universidad de Nariño
FUNDADA EN 1904

El término *Noisy Intermediate-Scale Quantum* (NISQ) designa los procesadores actuales: número intermedio de qubits y ausencia de corrección de errores (QEC) a escala completa [Preskill, 2018]. Cada qubit del chip es un qubit físico expuesto directamente al ruido, donde 3 limitaciones operan en simultáneo:

Decoherencia
Tiempos T_1, T_2 finitos ($\sim$decenas–cientos de μs). Limitan la profundidad ejecutable.

Error de compuerta
2-qubit: $\epsilon_{2q} \sim 10^{-3}$ en el mejor de los casos.
$\mathcal{F} \sim (1 - \epsilon_{\mathrm{CZ}})^{N_g}$

Lectura
Errores asimétricos de medición (*readout*), típicamente 1–5% por qubit.

Idea clave

Unos pocos cientos de compuertas de dos qubits bastan para que la distribución medida sea indistinguible del ruido.

Arquitectura de IBM: qubits superconductores transmon

- Basados en la unión Josephson: un elemento no lineal que separa los dos niveles $|0\rangle$, $|1\rangle$ del resto del espectro.
- Operados con pulsos de microondas a la frecuencia de resonancia ($\omega_{01} \sim 4$–5 GHz).
- Familia **Heron r1** (este trabajo): compuerta nativa de dos qubits CZ; de un qubit, *RZ*, *SX*, *X*.

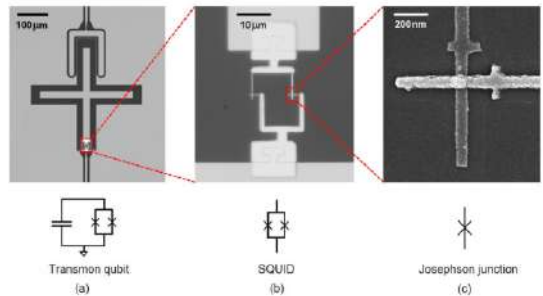


Fig. 1: Qubit transmon sintonizable: capacitancia en "+" en paralelo con un SQUID; detalle de la unión Josephson.

Topología Heavy-Hex y conectividad

Universidad de Nariño
FUNDADA EN 1904

En la red Heavy-Hex, cada qubit se acopla a lo sumo a 2–3 vecinos. No hay conectividad *all-to-all*.

Si el circuito lógico necesita una 2-qubit entre qubits no vecinos, el transpilador inserta compuertas SWAP.

El cuello de botella

En Heron r1, cada SWAP ≈ 3 compuertas CZ. Un circuito con N_{2Q} puertas se infla a $\sim 3N_{\text{SWAP}} + N_{2Q}$: la principal limitación física para algoritmos profundos como Shor.

Topología Heavy-Hex de IBM (Heron r1)

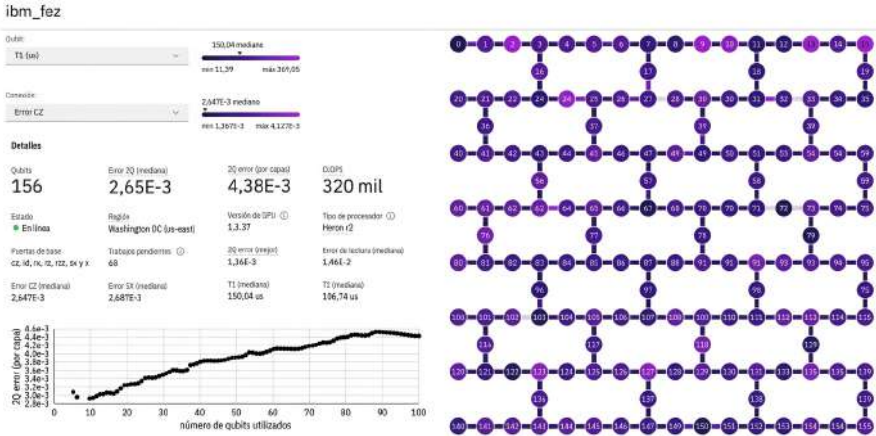


Fig. 2: Topología Heavy-Hex de un procesador IBM y sus datos de calibración.

Marco teórico

Estructura del algoritmo de Shor

La idea es indirecta: en lugar de factorizar directamente, Shor reduce el problema a la búsqueda de orden.

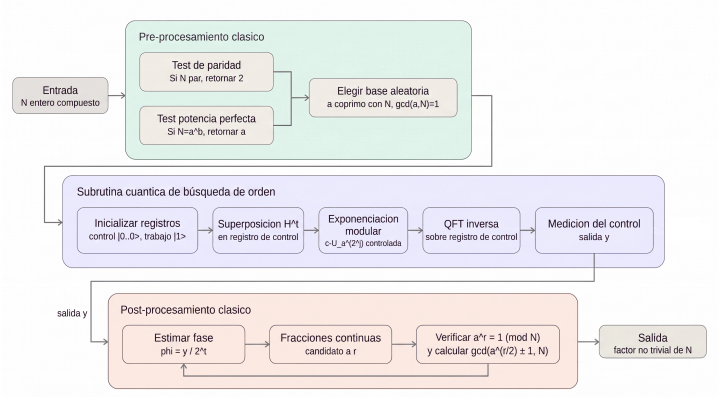


Fig. 3: Estructura general del algoritmo de Shor. Tres bloques: pre-procesamiento clásico, subrutina cuántica de búsqueda de orden y post-procesamiento clásico.

Transformada Cuántica de Fourier (QFT)

Operación lineal cuántica que mapea las amplitudes de un estado cuántico desde la base computacional hacia la base de frecuencias con $N = 2^n$:

$$\mathrm{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \mathrm{e}^{2\pi i jk/N}\ |k\rangle\,.$$

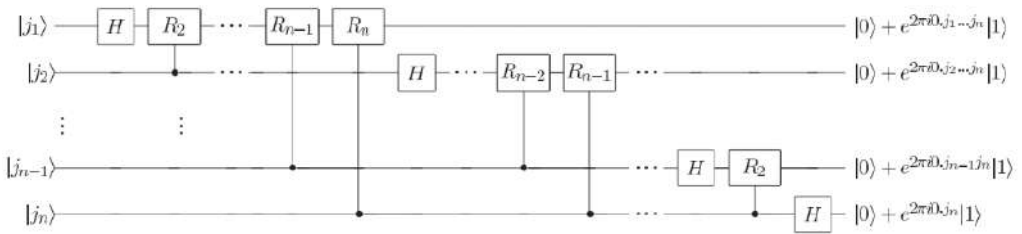


Fig. 4: Circuito eficiente de la QFT: Hadamard locales y rotaciones de fase controladas R_k (sin SWAP finales).

Estimación de Fase Cuántica (QPE)

Extrae el autovalor $e^{2\pi i\varphi}$ de U. Tras Hadamard y las controladas $c\text{-}U^{2^j}$:

$$|\psi_1\rangle = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i\varphi k} |k\rangle \otimes |u\rangle .$$

La fase queda codificada como una QFT $\Rightarrow$ aplicar $\mathrm{QFT}^\dagger$ lee directamente φ.

Cota de precisión (n bits, éxito $\geq 1-\epsilon$):

$$t = n + \left\lceil \log_2\left(2 + \tfrac{1}{2\epsilon}\right) \right\rceil .$$

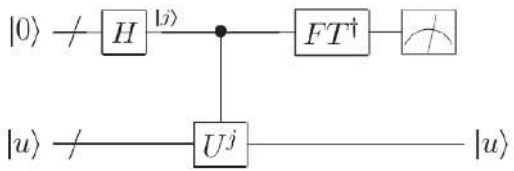


Fig. 5: Esquema general de la estimación de fase.

Estimación de Fase Cuántica (QPE)

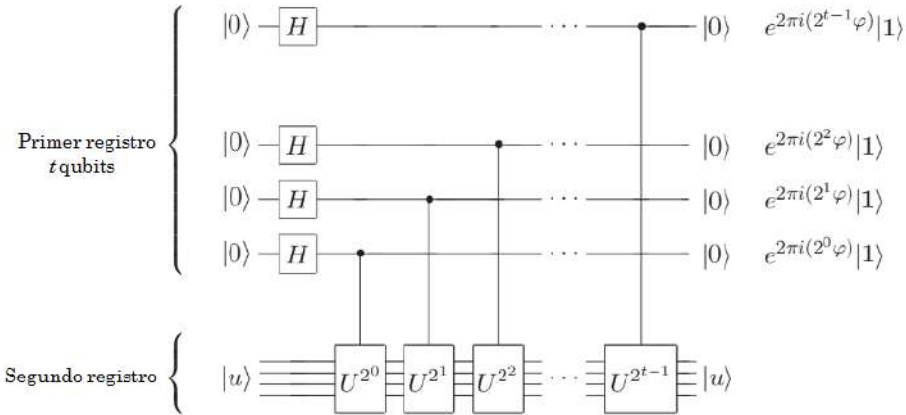


Fig. 6: Esquema de registros de la estimación de fase.

Búsqueda de orden y exponenciación modular

El algoritmo de búsqueda de orden es la QPE aplicada al operador de multiplicación modular:

$$U_{a,N}\,|y\rangle \equiv |\,ay \bmod N\,\rangle\,.$$

Sus autovectores $|u_s\rangle$ están definidos como:

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \left|a^k \bmod N\right\rangle$$

cuyos autovalores son $\lambda_s = e^{2\pi i s/r}$, de modo que la QPE estima las fases $\varphi_s = s/r$.

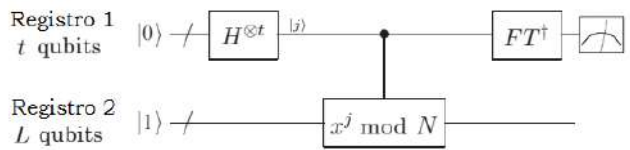


Fig. 7: Subrutina cuántica de búsqueda de orden.

Inicialización del estado $|1\rangle$

La preparación directa de los autovectores $|u_s\rangle$ de $U_{a,N}$ exigiría conocer r a priori (precisamente lo que se busca). Esta dificultad se elude mediante una propiedad estructural: el estado computacional $|1\rangle$ admite la descomposición

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle$$

Es decir, $|1\rangle$ es una superposición uniforme de todos los autovectores $|u_s\rangle$.

- Al preparar el registro inferior en $|1\rangle$, la QPE se ejecuta simultáneamente sobre todos los autovectores.
- Tras medir el registro de control, se obtiene una estimación de la fase $\varphi \approx s/r$ para un s aleatorio uniformemente distribuido.

La exponenciación modular controlada ($\mathcal{O}(L^3)$) domina la profundidad del circuito frente a la QFT ($\mathcal{O}(t^2)$).

Extracción del orden mediante fracciones continuas

La extracción de r a partir de la fase medida $\tilde{\varphi} = y/2^t$ se garantiza clásicamente:

Theorem (Convergencia de fracciones continuas)

Si s/r es un número racional tal que $\left|\frac{s}{r} - x\right| \leq \frac{1}{2r^2}$, entonces s/r es un convergente de la expansión en fracción continua de x.

Método de extracción

1. Expandir la fase medida en fracción continua: $y/2^t = [a_0; a_1, a_2, \dots]$.
2. Calcular los convergentes p_k/q_k mediante la recursión:

$$p_k = a_k p_{k-1} + p_{k-2}, \quad q_k = a_k q_{k-1} + q_{k-2}$$

3. Para cada denominador $q_k < N$, verificar si $a^{q_k} \equiv 1 \pmod{N}$. Si se cumple, el orden es $r = q_k$.

Reducción de la factorización a búsqueda de orden

Universidad de Nariño
FUNDADA EN 1904

Theorem (Reducción de factorización a búsqueda de orden)

Si N es compuesto y x es solución no trivial de $x^2 \equiv 1 \pmod{N}$, entonces al menos uno de

$$\gcd(x-1, N) \quad y \quad \gcd(x+1, N)$$

es un factor no trivial de N, computable en $\mathcal{O}(L^3)$.

Probabilidad de éxito

Para x aleatorio coprimo con N (con m factores primos):

$$p\left(r \text{ par y } x^{r/2} \not\equiv -1\right) \geq 1 - \frac{1}{2^m}.$$

Todo es clásico y eficiente, salvo la búsqueda de orden, que clásicamente es exponencial.

El proceso de transpilación

La transpilación transforma el circuito abstracto a instrucciones físicas del hardware NISQ en 6 etapas:

1. **Init:** Configura el *backend*.
2. **Layout:** Mapea qubits lógicos a físicos.
3. **Routing:** Inserta SWAPs para acatar topología.
4. **Translation:** Convierte a compuertas nativas.
5. **Optimization:** Cancela y reduce compuertas.
6. **Scheduling:** Asigna tiempos de ejecución.

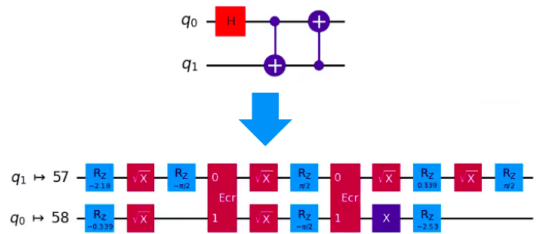


Fig. 8: Circuito lógico (arriba) adaptado a la topología y base nativa del hardware (abajo).

Niveles de optimización del transpilador

Table 1: Niveles de optimización en Qiskit.

Nivel	Transformaciones clave	Compuertas	Profundidad	T. Compilación
0	Solo traducción y enrutamiento mínimo. Sin reducciones.	Alta	Alta	Mínimo
1	Optimizaciones ligeras: cancelación adyacente y plegado de compuertas.	Moderada	Moderada	Bajo
2	Optimizaciones medias: consolidación de bloques unitarios, cancelación conmutativa y re-síntesis.	Significativa	Significativa	Moderado
3	Optimizaciones exhaustivas: nivel 2 + optimización adaptativa al ruido (como SABRE con parámetros agresivos).	Máxima	Máxima	Alto

El nivel de optimización ($\texttt{opt} \in \{0, 1, 2, 3\}$) controla el balance entre tiempo de compilación y reducción del circuito.

El algoritmo SABRE

SABRE (*SWAP-based BidiRectional heuristic search*) reduce la sobrecarga de SWAP con una heurística local:

- Precalcula la distancia topológica entre qubits físicos.
- Aísla un frente de interacciones inmediatas y genera SWAP candidatos solo alrededor de los qubits implicados.
- Es bidireccional: usa la reversibilidad cuántica para optimizar también el *layout* inicial.

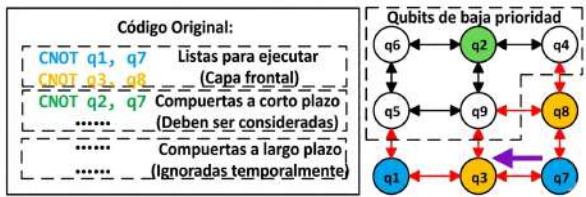
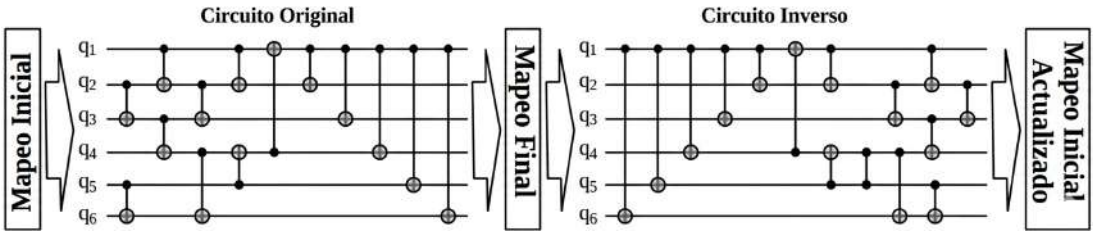


Fig. 9: Búsqueda heurística basada en SWAP.

El algoritmo SABRE



Menos intercambios ⇒ menos CZ ruidosas ⇒ menor decoherencia acumulada ⇒ mayor viabilidad experimental.

Estrategias de supresión: DD y Pauli Twirling

Universidad de Nariño
FUNDADA EN 1904

Desacoplamiento Dinámico (DD)

- Actúa en los intervalos de inactividad (*idle*).
- Pulsos rápidos que promedian a cero el acoplamiento con el entorno.
- Secuencia XY4: cancela a primer orden ruido de fase y de bit.

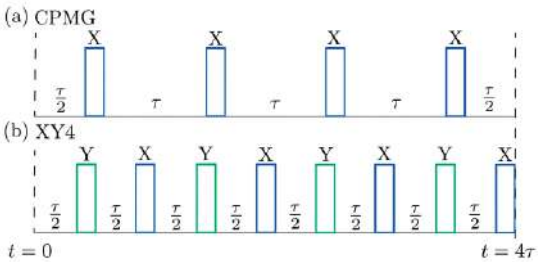
Pauli Twirling (PT)

- Actúa durante las compuertas de dos qubits.
- Envuelve cada CZ entre pares de Pauli aleatorios.
- Convierte error coherente (crece $\propto N_g^2$) en ruido estocástico (crece $\propto N_g$).

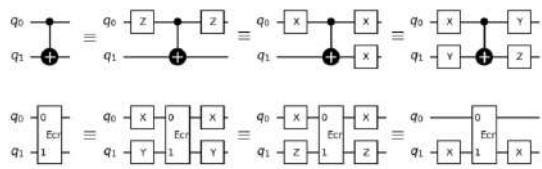
Supresión de errores: DD y Pauli Twirling

Universidad de Nariño
FUNDADA EN 1904

Desacoplamiento Dinámico (DD)



Pauli Twirling (PT)



DD protege los qubits ociosos; PT aleatoriza los errores de las compuertas activas. Atacan fuentes de error distintas.

Métricas de evaluación: PST y Fidelidad de Hellinger

Universidad de Nariño
FUNDADA EN 1904

Fidelidad de Hellinger

$$\mathcal{F}_H(P, Q) = \left(\sum_x \sqrt{p_x q_x} \right)^2 \in [0, 1]$$

Mide la similitud global de las distribuciones (cota superior de la fidelidad de estado).

PST (Probabilidad de Éxito de la Señal)

$$PST = \sum_{y \in \mathcal{Y}_{\text{ideal}}} q_y$$

Mide la masa de probabilidad medida sobre los picos teóricos del QPE (éxito de la señal).

Metodología

Selección de la base a y caracterización del grupo

Para $N = 15$, el grupo multiplicativo $\mathbb{Z}_{15}^*$ tiene 8 elementos coprimos. Descartando el caso trivial $a = 1$, se pre-calculan clásicamente el orden r y los factores para las 7 bases candidatas.

Table 2: Caracterización de bases candidatas para $N = 15$.

Base a	**Orden** r	r **par**	$a^{r/2} \bmod 15$	$\gcd(a^{r/2} - 1, 15)$	$\gcd(a^{r/2} + 1, 15)$
2	4	Sí	4	3	5
4	2	Sí	4	3	5
7	4	Sí	4	3	5
8	4	Sí	4	3	5
11	2	Sí	11	3	5
13	4	Sí	4	3	5
14	2	Sí	14	*trivial*	*trivial*

- Las bases con $r = 2$ ($a = 4, 11$) exigen diferenciar menos fases y generan circuitos cuánticos menos profundos.
- La base $a = 14$ es filtrada de antemano por el pre-procesamiento clásico al generar únicamente factores triviales.

Protocolo experimental en tres fases

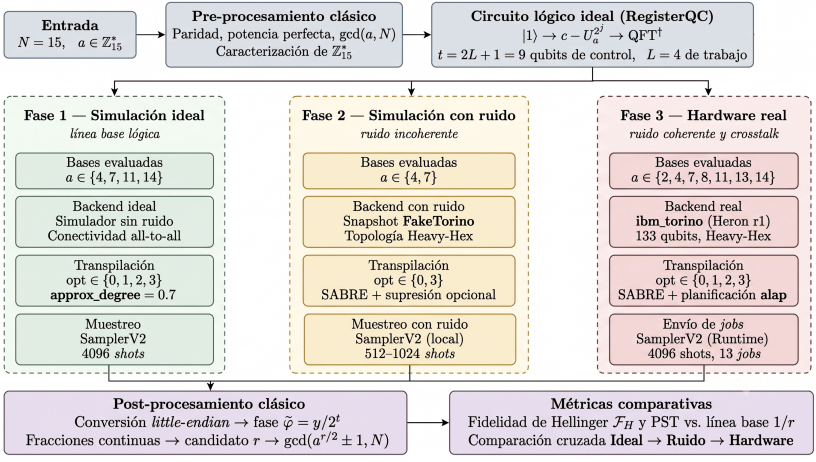


Fig. 10: Flujo del protocolo experimental por fases.

Resultados

Fase 1 — Caracterización de bloques

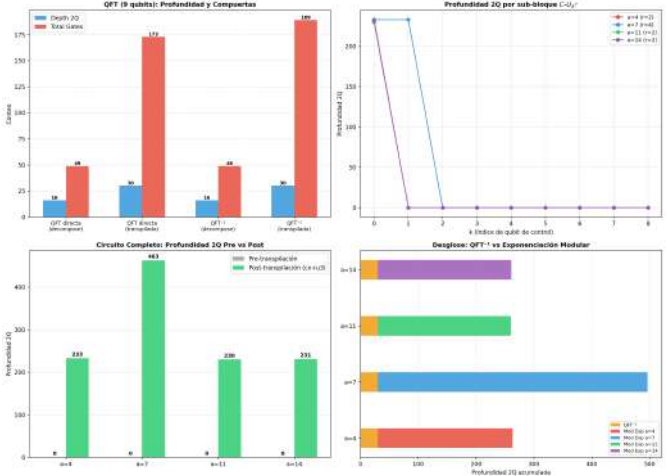


Fig. 11: Caracterizacion de los bloques fundamentales del algoritmo de Shor para $N = 15$.

Fase 1 — Caracterización de bloques

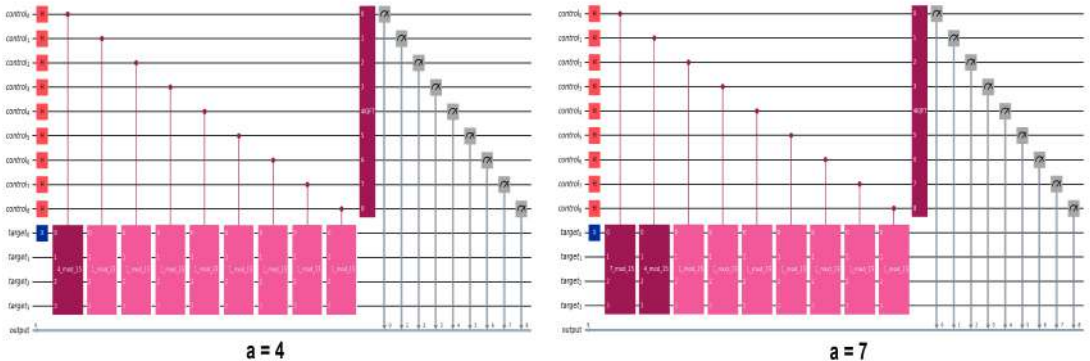


Fig. 12: Diagrama esquematico del circuito para las bases $a = 4$ (izquierda) y $a = 7$ (derecha) con $N = 15$

Fase 1 — Simulación ideal pura

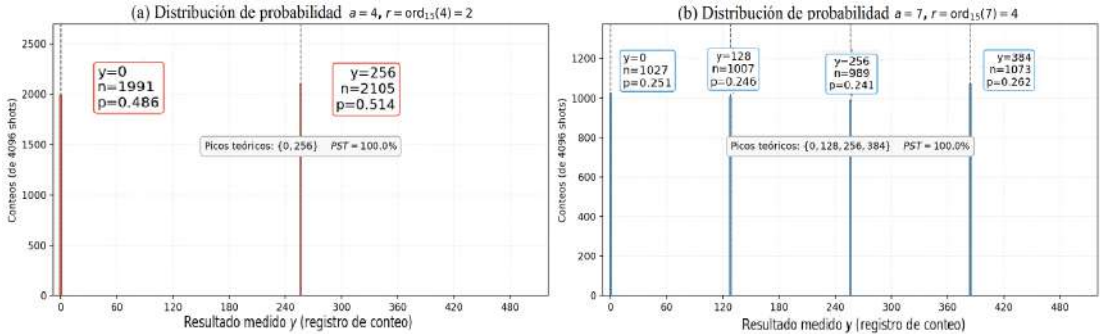


Fig. 13: Histogramas de medición del registro de conteo en simulación ideal del algoritmo de Shor ($N = 15$, 4096 shots)

Fase 1 — Simulación ideal pura

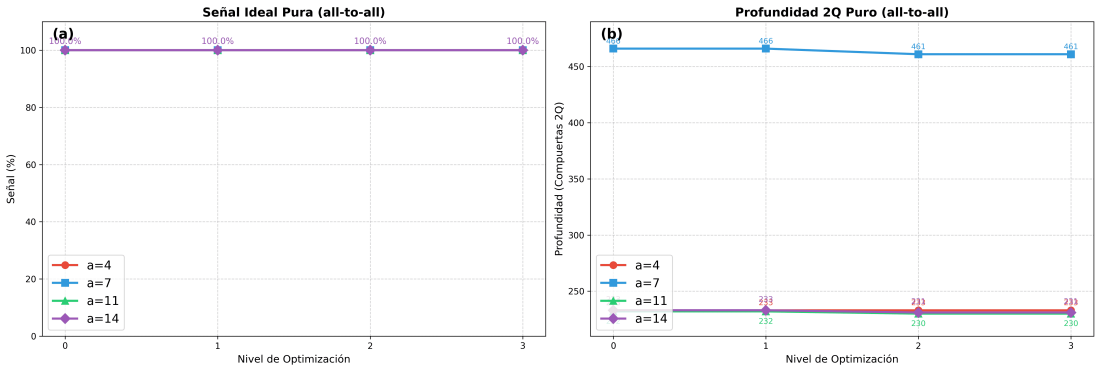


Fig. 14: Resultados de la simulación ideal pura (all-to-all). Izquierda: histogramas de medición del porcentaje de señal para cada base y nivel de Opt. Derecha: profundidad de dos qubits D_{2Q} por base y nivel de optimización

Fase 1 — Simulación ideal pura

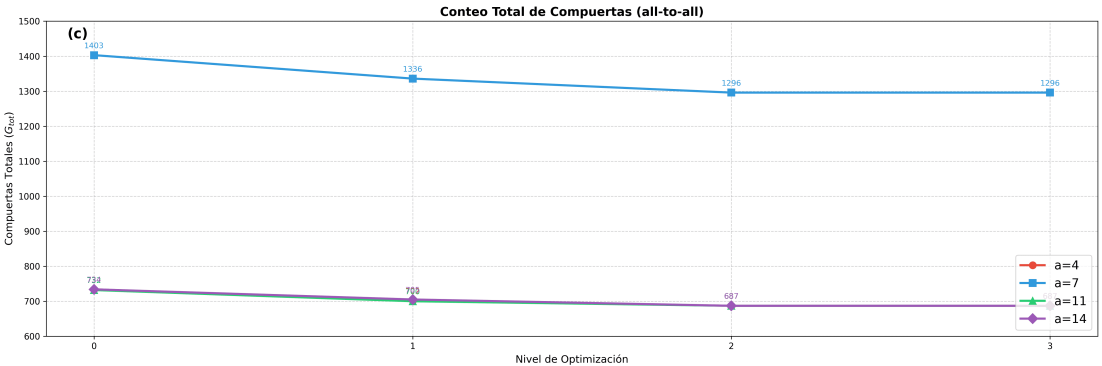


Fig. 15: Resultados de la simulación ideal pura (all-to-all). Conteo total de compuertas por base y nivel de optimización

Fase 1 — Simulación ideal con topología

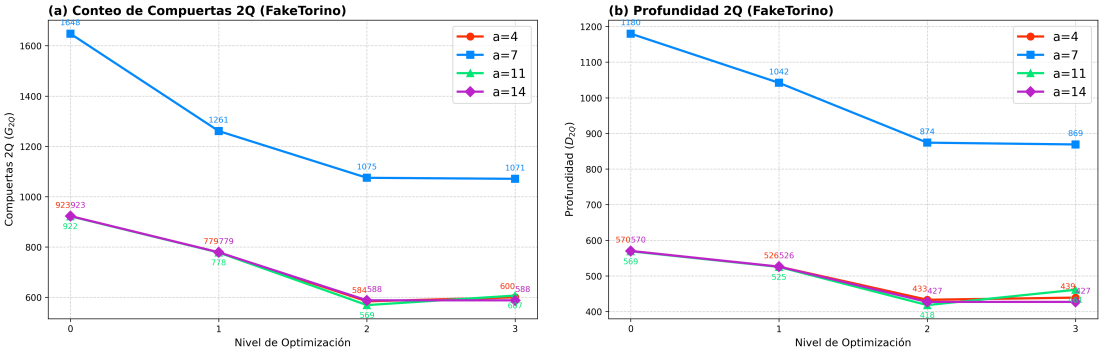


Fig. 16: Resultados de la simulación ideal del algoritmo de Shor para $N = 15$ transpilado a la topología Heavy-Hex de FakeTorino.

Fase 1 — Simulación ideal con topología

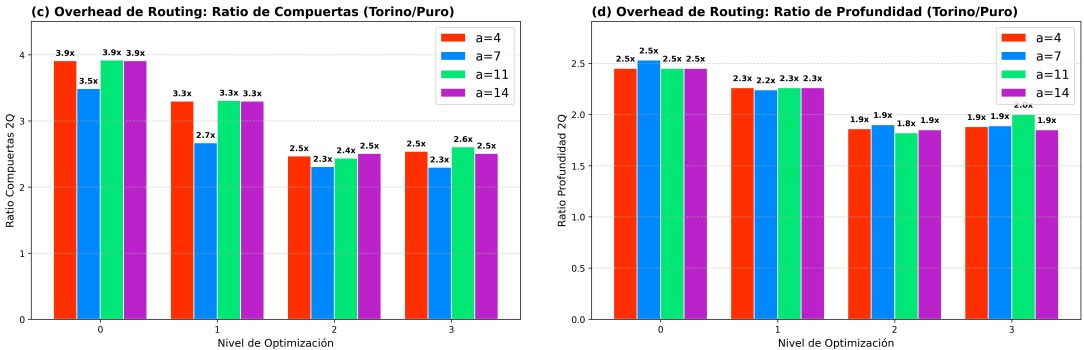


Fig. 17: Resultados de la simulación ideal del algoritmo de Shor para $N = 15$ transpilado a la topología Heavy-Hex de FakeTorino.

Fase 1 — Overhead topológico y ganancia de SABRE

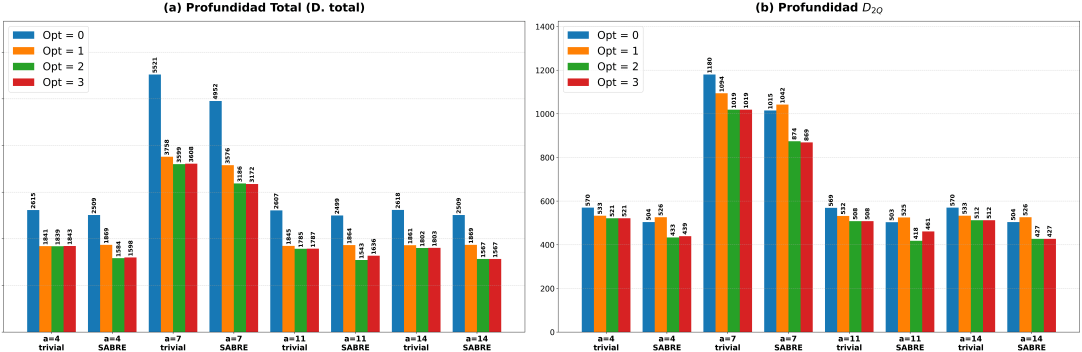


Fig. 18: Comparación de profundidad total y profundidad de compuertas de dos qubits entre el mapeo trivial y SABRE para el algoritmo de Shor ($N = 15$) sobre la topología Heavy-Hex de FakeTorino.

Fase 1 — Overhead topológico y ganancia de SABRE

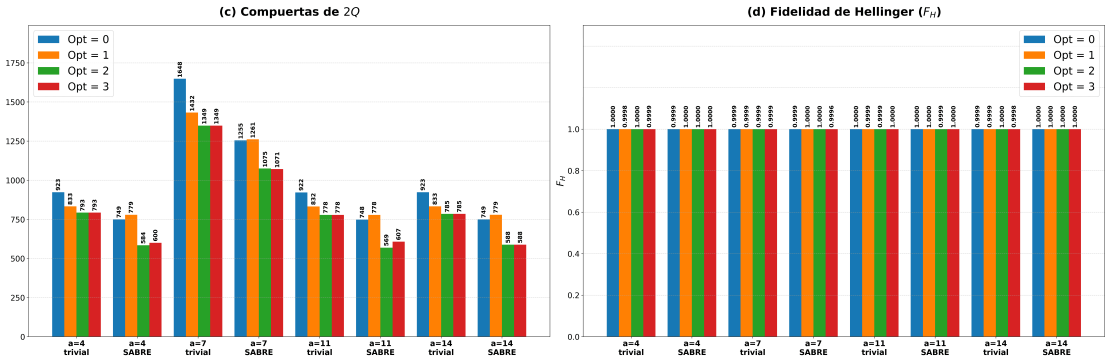


Fig. 19: Comparación del numero de compuertas de dos qubits y fidelidad de Hellinger entre el mapeo trivial y SABRE para el algoritmo de Shor ($N = 15$) sobre la topología Heavy-Hex de FakeTorino.

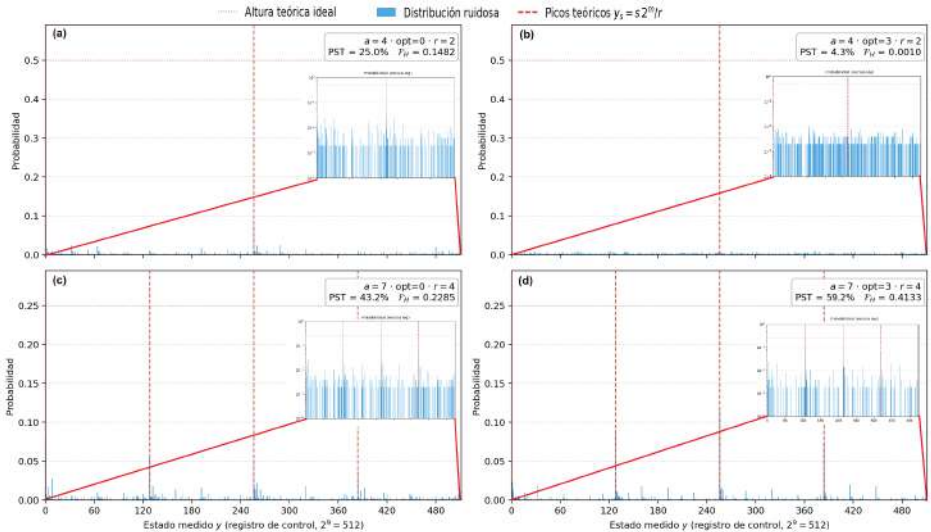
Fase 1 — Post-procesamiento clásico

Se debe verificar que la cadena clásica de post-procesamiento extrae efectivamente el período r y los factores no triviales de N para cada pico informativo ($y \neq 0$).

Table 3: Cadena de post-procesamiento clásico para los resultados con $y \neq 0$ obtenidos en simulación ideal pura ($N = 15, t = 9, M = 4096$ shots).

a	r	y	$\tilde{\varphi}$	FC	p_k/q_k	a^{q_k} mód N	Resultado clásico
4	2	256	0.500	$[0; 2]$	1/2	1	$\gcd = 3, 5$ ✓
7	4	128	0.250	$[0; 4]$	1/4	1	$\gcd = 3, 5$ ✓
7	4	384	0.750	$[0; 1, 3]$	3/4	1	$\gcd = 3, 5$ ✓
7	4	256	0.500	$[0; 2]$	1/2	$4\,(\neq 1)$	Convergente no válido †
11	2	256	0.500	$[0; 2]$	1/2	1	$\gcd = 5, 3$ ✓
14	2	256	0.500	$[0; 2]$	1/2	1	Triviales: $\gcd = 1, 15$ △

Fase 2 – Choque con el ruido (`FakeTorino`)



Fase 2 — Métricas de evaluación

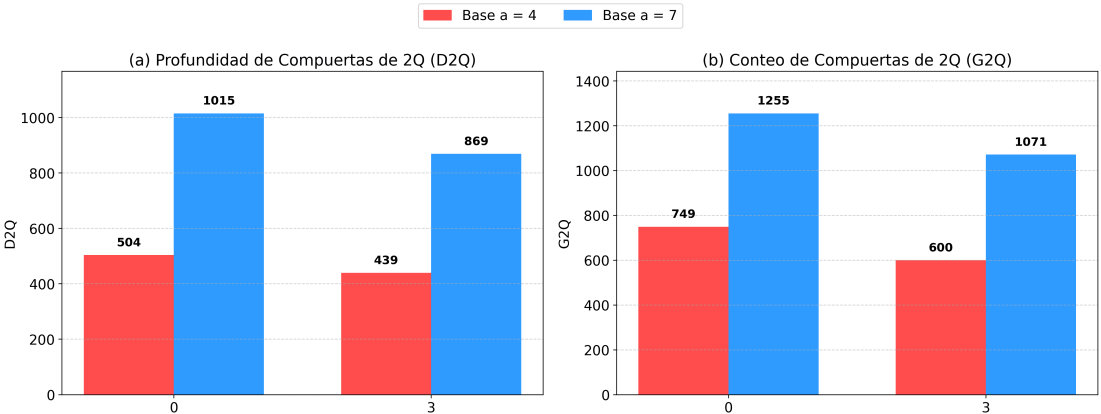


Fig. 21: Resultados para $a \in \{4, 7\}$ con `FakeTorino`, `opt=3`, $t = 9$ y 512 shots.

Fase 2 — Métricas de evaluación

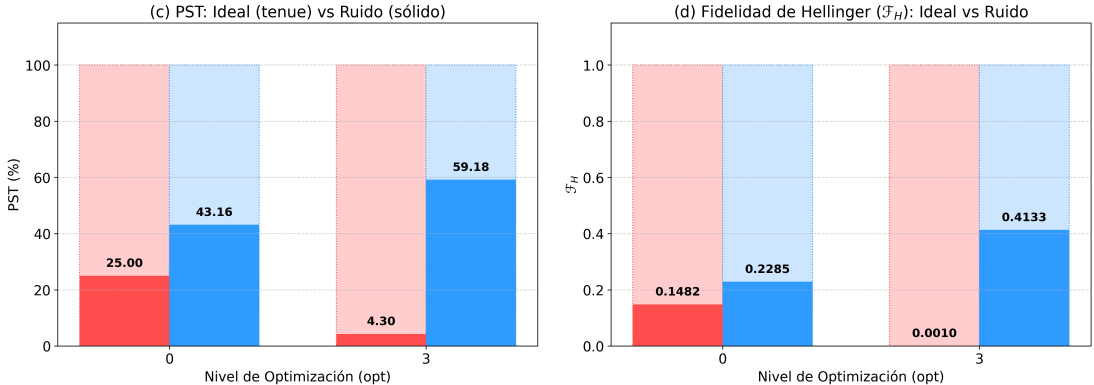


Fig. 22: Resultados para $a \in \{4, 7\}$ con `FakeTorino`, `opt`=3, $t = 9$ y 512 shots.

Fase 2 – Efecto de las técnicas de supresión

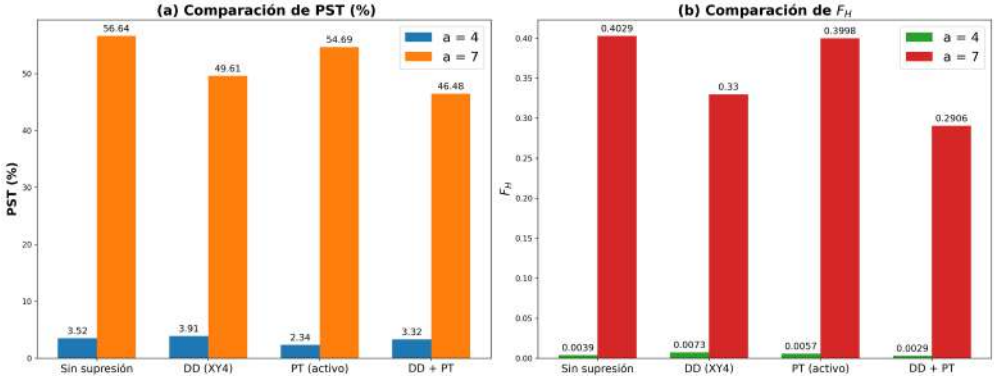


Fig. 23: Evaluación de técnicas de supresión en simulación ruidosa ($opt = 3$, shots=512, `FakeTorino`).

Fase 2 — La robustez del filtro clásico

Universidad de Nariño
FUNDADA EN 1904

Resultado central de la fase

El algoritmo de fracciones continuas extrae $\{3, 5\}$ en todas las configuraciones, incluso con $a = 4$, $PST \approx 4\%$ y $\mathcal{F}_H \approx 0.001$.

- Basta que los picos teóricos asomen sobre el piso de ruido para que el filtro clásico los seleccione.
- El ruido desordena la masa entre estados aislados que no pasan la verificación modular $a^{q_k} \equiv 1$.
- Asimetría cuántico (frágil) / clásico (robusto): lo cuántico aporta el indicio; lo clásico, el filtro.

Sobre las técnicas de supresión en el simulador

DD y PT no mejoran (incluso degradan ligeramente) las métricas: el ruido de `FakeTorino` es puramente estocástico, sin errores coherentes que estabilizar.

Fase 3 — Ejecución en hardware real: `ibm_torino`

Table 4: Configuración del backend.

Parámetro	Valor
QPU	`ibm_torino` (Heron r1)
Qubits	133
Topología	Heavy-Hex
Nativa 2Q	CZ
Nativas 1Q	RZ, SX, X
T_2	$\sim 132\ \mu$s
Fid. CZ	$\sim 99.5\%$
`approx`	0.7
Shots	4096

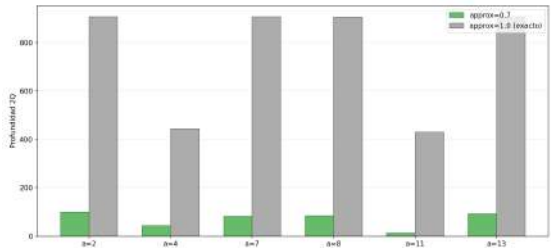


Fig. 24: `approximation_degree`=0.7 comprime D_{2Q} un orden de magnitud (de 430–910 a 12–100 capas CZ): clave para caber en el tiempo de coherencia.

Fase 3 — El nivel de optimización es *determinante*

Table 5: Estudio A2 — base $a = 4$, variando `opt`.

`opt`	D_{2Q}	CZ	*PST*	$\mathcal{F}_H$
0	770	1220	1.1%	0.011
1	526	779	3.1%	0.031
2	116	116	70.8%	0.529
3*	116	116	**82.8%**	**0.547**

* Referencia tomada de la línea base sin supresión (estudio A6/A7).

Hallazgo

Solo cambiando `opt`: *PST* salta de 1.1% a 70.8% (∼2 órdenes de magnitud).

- `opt`=0 → 113 SWAP estimadas → 770 capas CZ. $\mathcal{F}_{\text{est}} \sim (0.995)^{1220} \approx 0.002$.
- `opt`=2 con SABRE → 0 SWAP extra → 116 CZ. $\mathcal{F}_{\text{est}} \sim (0.995)^{116} \approx 0.56 \approx \mathcal{F}_H$ observada.

La compilación no es accesoria: es pieza estructural del algoritmo en NISQ.

Fase 3 — Resultados por base a (PT activado)

Table 6: `opt=3`, `approx=0.7`.

a	r	D_{2Q}	*PST*	Factores
8	4	212	**79.4%**	3×5
4	2	116	75.5%	3×5
7	4	244	67.5%	3×5
13	4	212	45.3%	3×5
2	4	225	42.4%	3×5
11	2	75	35.5%	3×5
14	2	117	75.1%	*triviales*

Éxito en 6 de 7 bases ($a = 14 \equiv -1$ da factores triviales por construcción). Profundidad y señal no son estrictamente monótonas: pesa la calibración local de los qubits asignados.

Fase 3 — Distribución de resultados por base *a*

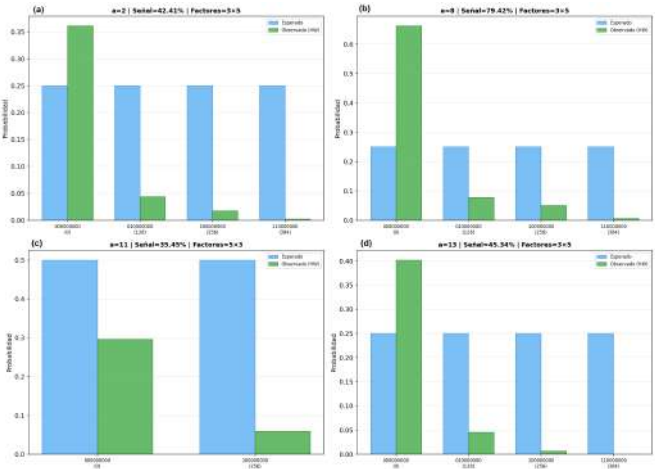


Fig. 25: HW (verde) vs. teórica (azul). La asimetría refleja el sesgo de relajación T_1 hacia $|0\rangle^{\otimes n}$.

Fase 3 — Hallazgo inesperado: DD *contraproducente*

Table 7: Supresión de errores en hardware ($a = 4$, `opt`=3, 116 CZ, 4096 *shots*).

Configuración	*PST*	$\mathcal{F}_H$	**Outcomes**	**Factores**
Sin supresión	**82.8%**	**0.547**	30	3×5
Solo DD (XY4)	20.3%	0.154	163	No hallados
PT + DD (XY4)	0.8%	0.007	469	$(3 \times 5)^{\dagger}$

† Extracción por coincidencia estadística; con *PST* $< 1\%$ la distribución es indistinguible del ruido (no es un éxito real).

¿Por qué el DD degrada aquí?

Para un circuito moderado (116 CZ) en hardware de alta fidelidad, los *idle times* son breves. Los pulsos de DD añaden error de calibración propio y crosstalk, que exceden el beneficio de la refocalización. PT+DD no es aditivo: puede romper cancelaciones coherentes favorables.

Fase 3 – Cascada: ideal → FakeTorino → HW

Table 8: *PST* (%) por entorno, `opt`=3.

a	*r*	Ideal	`FakeTorino`	**HW**
4	2	100	4.3	**75.5**
7	4	100	59.2	**67.5**
8	4	100	–	**79.4**
2	4	100	–	**42.4**
11	2	100	–	**35.5**
13	4	100	–	**45.3**

Hallazgo metodológico

En circuitos cortos estudiados, el *Fake Backend* es un predictor pesimista de su propio hardware: subestima la señal real.

Tres causas (no excluyentes): (1) el *snapshot* es un modelo estocástico aditivo (cotas pesimistas); (2) deriva de calibración entre ventanas (mejor fidelidad CZ el día de ejecución); (3) cancelaciones coherentes destructivas en la cadena de CZ que el modelo estocástico da por perdidas (se atenúa con la profundidad: fuerte en $a{=}4$, leve en $a{=}7$).

Señal promedio en HW: $\overline{PST} \approx 57.6\% \Rightarrow$ suficiente para extraer los factores en todos los casos válidos.

Conclusiones y perspectivas

Conclusiones principales

Universidad de Nariño
FUNDADA EN 1904

Demostración experimental

Se factorizó $N = 15 = \mathbf{3} \times \mathbf{5}$ en `ibm_torino` con señal máxima $PST = \mathbf{82.8}\%$ ($\mathcal{F}_H = 0.547$) y éxito $\approx 90\%$ sobre las bases coprimas válidas.

- **Bloques validados:** QFT y exponenciación modular con error de unitariedad $< 10^{-13}$.
- **Pre-procesamiento clásico:** las bases con $r = 2$ generan circuitos más cortos $\Rightarrow$ ventaja medible ($a = 4$: 75.5% vs. $a = 7$: 67.5%).
- La transpilación supera a la supresión de errores: elegir bien `opt` cambia el *PST* en $\sim$2 órdenes de magnitud; aplicar DD/PT indiscriminadamente degrada la señal.
- El modelo $\mathcal{F} \sim (1 - \epsilon_{CZ})^{N_g}$ predice bien la fidelidad en profundidad moderada.

Limitaciones, recomendaciones y perspectivas futuras

Universidad de Nariño
FUNDADA EN 1904

Limitaciones

- Restricción al caso $N = 15$ (extrapolación criptográfica inviable hoy).
- Cupo del *Open Plan* + retirada de `ibm_torino`: 13 *jobs*, sin réplicas.
- Parte de las diferencias pequeñas ($\sim$pp) son compatibles con deriva de calibración.

Recomendaciones / futuro

- **QPE iterativa:** reciclar 1 qubit de control ($t = 9 \to 1$) $\Rightarrow$ menos profundidad y conectividad.
- **DD adaptativo:** secuencias más cortas (X-X) o activadas solo si los *idle times* superan un umbral.
- **Hardware más reciente:** Heron r2 / Flamingo, mejor fidelidad y coherencia.
- **Metodología reproducible:** repositorio público + `job_id` para futuros trabajos en la Udenar.

Reflexión

En NISQ, el éxito no depende solo del hardware ni de la corrección formal, sino de la ingeniería de transpilación, la selección de parámetros clásicos y la evaluación empírica de cada técnica.

Referencias I

P. W. Shor (1994).
Algorithms for quantum computation: discrete logarithms and factoring.
Proc. 35th Annual Symp. on Foundations of Computer Science, 124–134.

M. A. Nielsen and I. L. Chuang.
Quantum Computation and Quantum Information.
Cambridge University Press.

J. Preskill (2018).
Quantum Computing in the NISQ era and beyond.
Quantum 2, 79.

Referencias II

G. Li, Y. Ding and Y. Xie (2019).
Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices (SABRE).
ASPLOS '19, 1001–1014.

L. M. K. Vandersypen *et al.* (2001).
Experimental realization of Shor's quantum factoring algorithm using NMR.
Nature 414, 883–887.

U. Skosana and M. Tame (2021).
Demonstration of Shor's factoring algorithm for $N = 21$ on IBM quantum processors.
Scientific Reports 11, 16599.

Referencias III

J. J. Wallman and J. Emerson (2016).
Noise tailoring for scalable quantum computation via randomized compiling.
Phys. Rev. A 94, 052325.

L. Viola, E. Knill and S. Lloyd (1999).
Dynamical decoupling of open quantum systems.
Phys. Rev. Lett. 82, 2417.

N. Ezzell *et al.* (2023).
Dynamical decoupling for superconducting qubits: a performance survey.
Phys. Rev. Applied 20, 064027.

Referencias IV

M. Abu-Ghanem (2025).
IBM Quantum Computers: Evolution, Performance, and Future Directions.
Preprint (verificar referencia en bibliografia.bib).

Referencia base de metodología NISQ (Shor en IBM, SABRE, Fake Backends).
Completar datos exactos desde bibliografia.bib.

C. A. Villota Freire (2026).
Repositorio público del proyecto: código, configuraciones y `job_id` de IBM Quantum.

¡Gracias por su atención!

Universidad de Nariño
FUNDADA EN 1904

Universidad de Nariño
ACREDITADA EN ALTA CALIDAD

121 AÑOS