

Quantum Analisys Frameworks

Estas explicaciones la realizamos mediante la implementación de un framework de análisis QAF

Qué es el Quantum Analysis Framework (QAF)

Un marco de análisis y decisión para sistemas complejos donde las variables (tangibles e intangibles) se tratan como estados que coexisten e interactúan. QAF combina:

- Observación sistemática (sistemas complejos);
- Índices compuestos y optimización variacional (mínimo de “energía” del sistema);
- Dinámica cuántica (superposición, entrelazamiento, colapso) para modelar cómo evoluciona el estado del sistema con el tiempo.

Cómo se implementa el método (pasos)

1. Definir el campo: contexto, variables claves y restricciones.
2. Medir estado inicial: normalizar variables e índices (0-1).
3. Plantear ecuaciones:
 - Índice compuesto QAF (ej. mezcla de coherencia, tracción y eficiencia).
 - Principio variacional: minimizar una funcional F (entropía/desorden/“energía”).
 - Dinámica de Schrödinger: evolución temporal del vector de estado.
4. Simular/optimizar: resolver ODEs, explorar escenarios, sensibilidad.
5. Colapsar en decisiones: elegir acciones con mayor impacto esperado.
6. Retroalimentar: medir, recalibrar parámetros, iterar.

Cómo usamos las ecuaciones (incluida Schrödinger)

- QAF (índice compuesto): escalar del “estado de impacto” del sistema.
- Variacional: $\delta F=0 \Rightarrow \delta F=0$ para hallar configuraciones estables/óptimas.
- Schrödinger (genérica):

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = \hat{H} |\psi(t)\rangle$$

donde $|\psi\rangle$ es el vector de estado de las “modalidades” del sistema y \hat{H} es la matriz de acoplamientos/influencias.

Resultado de la simulación (genérica)

Usé un sistema de 3 estados con Hamiltoniano hermítico y un estado inicial puro.

Evolucioné $|\psi(t)\rangle$ y calculé:

- Probabilidades de cada estado en el tiempo (oscilaciones → intercambio de dominancia).
- Coherencia (suma de |off-diagonal| de la matriz densidad): mide superposición efectiva.
- Impacto (proxy): favorece balance entre estados y coherencia moderada.

Hallazgos claves (del panel “headline” que ejecuté):

- Autovalores de \hat{H} : [0.598, 0.8637, 1.5383] → tres frecuencias naturales del sistema.
- Máximo impacto y máxima coherencia coinciden en $t \approx 16.14$.
- Punto de estabilidad (cambio mínimo del impacto) en $t \approx 30.476$.

Sabiduría (qué aprender de esto)

- Acoplamientos moderados entre “modos” producen pulsos de coherencia: momentos óptimos donde el sistema está listo para colapsar en acción.
- El timing importa: hay ventanas donde la coherencia e impacto alcanzan picos simultáneos (nuestro $t \approx 16t \setminus \text{approx} 16t \approx 16$).
- La estabilidad no coincide necesariamente con el máximo impacto; sirve para consolidar más que para expandir.
- Medir dominancia por estado en el tiempo revela ciclos naturales (evita forzar decisiones fuera de fase).

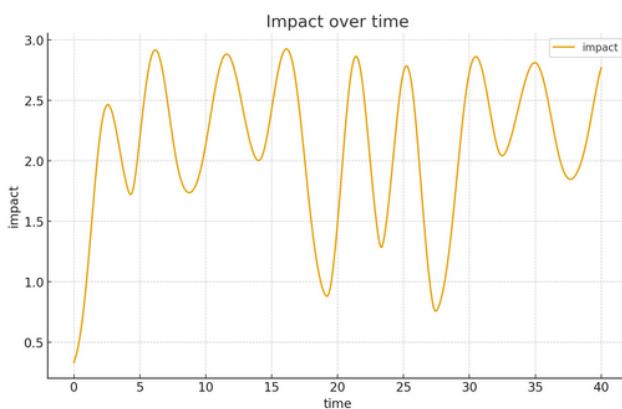
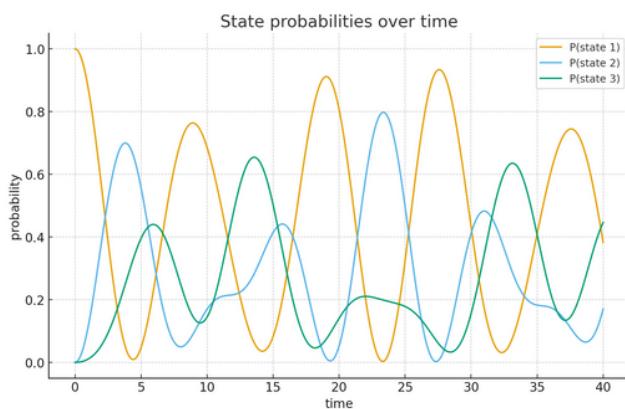
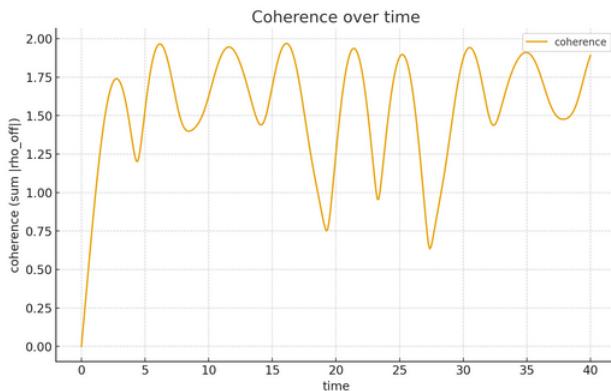
Conclusiones

1. QAF ofrece una lupa dinámica: no solo “qué hacer”, sino cuándo hacerlo.
2. La ecuación de Schrödinger es útil como motor temporal para visualizar la transferencia de “energía de estado” (probabilidad) entre opciones.
3. Un índice compuesto simple + coherencia ya permite priorizar ventanas de acción sin perder el carácter genérico del marco.

Accionables

- Estandariza el pipeline: (definir variables → normalizar → H hermítico → simular ttt → leer picos de coherencia/impacto).
- Documenta autovalores y fases: son tu “huella digital” de dinámica interna.
- Decide por ventanas: actúa en picos de impacto; consolida en tramos de estabilidad.

QAF—Schrödinger Mini Simulation (Sample)						
t	P_state_1	P_state_2	P_state_3	coherence	impact	dominant_state
1 0.0	0.9999999999999999	3.08148791101957	74e-33	3.774828909989	4.9990038108132	0
2 41	0.0268713384865	0.6951515867258	0.19	0.2779770747876	1.32537374321341	1
3 83	0.0205005125132	0.7043505730638	0.83	0.0495965801195	0.24605284468165	0
4 924	0.2987283874496	0.2163848985988	5474	0.4848867130515	1.9175047780485	2
5 568	0.2793004203676	0.4340841992266	6864	0.2866053804056	1.9877036972186	1
6 207	0.7959272706604	0.0597561105306	458	0.1443166188088	1.2997372588564	0
7 85	0.0743969226709	0.7343820182044	9395	0.19122106112455	1.45551161478331	1
8 49	0.9065708394820	0.0555033507536	829	0.0378258097542	0.9106778855016	0
9 13	0.0360208476331	0.4125943766252	16415	0.5513847757416	1.47961573001806	2
10 77	0.61568533763909	0.373687592622	271	0.2106270700994	1.75678162745032	0



```

# ----- QAF – Schrödinger Mini Simulation (Generic) -----
# Este script demuestra un pipeline mínimo inspirado en QAF:
# 1) Define un Hamiltoniano hermítico 3x3 (interacciones del sistema).
# 2) Evoluciona un estado inicial con la ecuación de Schrödinger.
# 3) Calcula probabilidades, una métrica simple de coherencia y un índice de "impacto".
# 4) Exporta CSV y figuras (una por gráfico, sin estilos personalizados).

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import json

h = 1.0 # escala unitaria para la demo

def evolve_state(H, psi0, ts, h=1.0):
    """
    Evolución unitaria  $|\psi(t)\rangle = \exp(-i H t / \hbar) |\psi(0)\rangle$  via descomposición espectral.
     $H = V \text{diag}(w) V^\dagger$ , con  $V$  unitaria. Luego  $\exp(-i H t / \hbar) = V \exp(-i w t / \hbar) V^\dagger$ 
    """
    w, V = np.linalg.eigh(H) # autovalores y autovectores (H hermítico)
    Vinv = V.conj().T
    states = np.zeros((len(ts), len(psi0)), dtype=np.complex128)
    for k, t in enumerate(ts):
        phase = np.exp(-1j * w * t / h)
        states[k] = V @ (phase * (Vinv @ psi0))
    return states, w, V

def density_matrix(psi):
    """
     $\rho = |\psi\rangle\langle\psi|$ 
    """
    return np.outer(psi, psi.conj())

def main():
    # Carpeta de salida
    outdir = "qaf_demo_assets"
    os.makedirs(outdir, exist_ok=True)

    # 1) Hamiltoniano 3x3 hermítico (interacciones genéricas entre 3 modos)
    H = np.array([
        [1.20, 0.35, 0.10],
        [0.35, 1.00, 0.25],
        [0.10, 0.25, 0.80]
    ], dtype=np.complex128)
    assert np.allclose(H, H.conj().T), "H debe ser hermítico"

    # Estado inicial puro (normalizado): todo el peso en el modo 1 (genérico)
    psi0 = np.array([1.0, 0.0, 0.0], dtype=np.complex128)
    psi0 = psi0 / np.linalg.norm(psi0)

    # 2) Malla temporal
    T = 40.0
    N = 400
    ts = np.linspace(0, T, N)

    # 3) Evolución de Schrödinger
    states, evals, evecs = evolve_state(H, psi0, ts, h)
    probs = np.abs(states)**2 # probabilidades en la base canónica

    # 4) Métrica de coherencia: suma de  $|\text{off-diagonales}|$  de  $\rho(t)$ 
    coherence = np.zeros(N)
    for k in range(N):
        rho = density_matrix(states[k])
        mask = ~np.eye(3, dtype=bool)
        coherence[k] = np.abs(rho[mask]).sum()

    # 5) Índice "impact": favorece equilibrio probabilístico y coherencia moderada
    balance = 1 - sum((p_i - 1/3)**2 for p_i in evals) # (máximo cuando p_i = 1/3)
    impact = balance * (1.0 + coherence)

    # Dominancia (modo con mayor probabilidad en cada t)
    dominant = np.argmax(probs, axis=1)

    # 6) Empaquetar resultados
    df = pd.DataFrame({
        "t": ts,
        "P_state_1": probs[:, 0],
        "P_state_2": probs[:, 1],
        "P_state_3": probs[:, 2],
        "coherence": coherence,
        "impact": impact,
        "dominant_state": dominant
    })

    headline = {
        "eigenvalues(H)": np.round(evals, 4).tolist(),
        "t_max_impact": float(np.round(ts[np.argmax(impact)], 3)),
        "t_max_coherence": float(np.round(ts[np.argmax(coherence)], 3)),
        "t_stability": float(np.round(ts[np.argmin(np.gradient(impact)**2)], 3)),
        "max_impact": float(np.round(impact.max(), 4)),
        "max_coherence": float(np.round(coherence.max(), 4))
    }

    # 7) Exportar CSV y JSON
    df.to_csv(os.path.join(outdir, "qaf_schrodinger_timeseries.csv"), index=False)
    with open(os.path.join(outdir, "qaf_schrodinger_headline.json"), "w") as f:
        json.dump(headline, f, indent=2)

    # 8) Figuras (una por gráfico, sin colores personalizados)
    plt.figure()
    plt.plot(df["t"], df["P_state_1"], label="P(state 1)")
    plt.plot(df["t"], df["P_state_2"], label="P(state 2)")
    plt.plot(df["t"], df["P_state_3"], label="P(state 3)")
    plt.xlabel("time"); plt.ylabel("probability"); plt.title("State probabilities over time"); plt.legend()
    plt.savefig(os.path.join(outdir, "plot_probabilities.png"), dpi=180, bbox_inches="tight")

    plt.figure()
    plt.plot(df["t"], df["coherence"], label="coherence")
    plt.xlabel("time"); plt.ylabel("coherence (sum |rho_off|)"); plt.title("Coherence over time"); plt.legend()
    plt.savefig(os.path.join(outdir, "plot_coherence.png"), dpi=180, bbox_inches="tight")

    plt.figure()
    plt.plot(df["t"], df["impact"], label="impact")
    plt.xlabel("time"); plt.ylabel("Impact"); plt.title("Impact over time"); plt.legend()
    plt.savefig(os.path.join(outdir, "plot_impact.png"), dpi=180, bbox_inches="tight")

    # 9) Salidas de consola
    print("==== QAF – Schrödinger Mini Simulation (Generic) ===")
    print("Eigenvalues(H):", np.round(evals, 4))
    print("Headlines:", headline)
    print(f"Saved files in ./[{outdir}]")

if __name__ == "__main__":
    main()
•

```

Quantum Analysis Framework (QAF)

El método que convierte la intuición en ciencia.

Todo sistema vibra entre orden y caos.

QAF observa la energía invisible que conecta datos, decisiones y emociones.

Su propósito: revelar cuándo y dónde actuar con precisión

LA DANZA DE SCHRÖDINGER



LA ECUACIÓN SCHRIÖDINGER
DESCRIBE LA EVOLUCIÓN ARMÓNICA DE LOS ESTADOS DE PENSAMIENTO

$$i\hbar \frac{\partial \psi}{\partial t} = \hat{H}$$



Método QAF

Siete pasos para alinear energía, información y acción.



- 1 Observa el campo
- 2 Identifica variables
- 3 Mide el estado
- 4 Calcula la ecuación
- 5 Colapsa en decisión
- 6 Mide la resonancia
- 7 Expande el campo

Coherencia = el arte de sincronizar realidades.

Sabiduría Cuántica

"No controlamos la realidad; la sincronizamos."

La coherencia es más valiosa que la fuerza.

El impacto óptimo ocurre cuando el sistema y el observador vibran al mismo ritmo.

El QAF revela el instante en que la acción se vuelve inevitable y natural.

El colapso no destruye: revela propósito.