

Proyecto Andrómeda - Software de astrología Armon2

Implementación de Armogramas



Por Héctor Jiménez – Febrero 2026

Introducción

Este documento tiene como fin explicar las definiciones, decisiones y pasos de implementación que se llevaron a cabo en el desarrollo en código de programación de computadora, de la técnica llamada Armogramas basados en astrología armónica.

Historia

El propósito del proyecto Andrómeda es muy Neptuniano, me salen estas “visiones” de cómo integrar lo mejor de los mundos, conceptos y tecnología, en una sola “burbuja”. Si veo atrás en los caminos que he tomado en mi vida, en mi interior surge el deseo de unir mis dos conocimientos que en realidad me apasionan: la astrología y las ciencias computacionales.

Para el tema de los Armogramas, está inherente la innovación, desde John Addey hasta Tito Maciá, Miguel García Ferrández y Demetrio Santos.

El proyecto produce Armon2, inspirado en estas teorías y trabajos (materialización de la teoría) alrededor de la astrología armónica, misma que fue plasmada en el software o programa Armon (Explorador Armónico), cómo crear una nueva “burbuja” que contenga algo de dicho trabajo, a manera de preservar, continuarlo, que, aunque no tenga nada de innovador, si tiene mucho del deseo de unir mis 2 pasiones, de seguir aprendiendo y ofreciendo algo que pueda ser de utilidad y beneficio de otros.

Armon2 implementa Armogramas, este documento intenta explicar este proceso y detalles (técnicos) de su implementación. Lo Neptuniano fue confrontado por Saturno durante el proceso, tuve confusión, falta de información y falta de conocimiento, y por lo tanto, frustración y freno en el proceso.

Este documento nace de querer equilibrar esa confusión con un poco de orden, dejando estas notas con el fin de brindar un poco de luz en el tema de Armogramas, esperando que Neptuno no se asome entre líneas y confunda más.

Palabras clave en el documento

- *Código, implementación, programa, Armon2* = el trabajo realizado como implementación de un programa de cómputo en un lenguaje de programación, de la técnica de Armograma.
- *H2, H3, H4, etc.* = Se refiere al armónico 2, 3 y 4, etc. correspondientes

Qué es un armograma?

Lo primero que quiero expresar es el concepto de lo que es un Armograma, y aquí lo pongo en el contexto de la implementación realizada, defino lo siguiente como *armograma*:

“Un armograma es una representación temporal cuantitativa de la intensidad geométrica de una serie armónica, obtenida a partir de la densidad y coherencia de conjunciones en la carta armónica correspondiente entre un conjunto de factores astrológicos.”

Dicho esto, detallo algunos de los significados de la definición anterior (a que se refiere):

1) “representación temporal”

El armograma **no es una carta**, ni un gráfico estático. Es una **serie temporal**, con el eje X = tiempo y el eje Y = intensidad armónica. Sin tiempo, no hay armograma. El código recorre fechas (funciones en el programa como `compute_series_daily`, `dp`, `secondary`, etc.) y calcula valores por instante.

2) “cuantitativa”

No es binario (aspecto sí/no). No es simbólico. Produce valores numéricos continuos: agregados, normalizados, comparables. Esto está directamente en el código como: `intensity = 100.0 * (total / count)`. La intención del armograma es **medir**, no opinar.

3) “intensidad geométrica”

Esta es la clave conceptual más importante. No mide “*fuerza interpretativa*”. No mide “*energía psicológica*”. No mide “*importancia*”. Mide: **qué tan alineada está la geometría planetaria con una estructura armónica específica**. Esa geometría se expresa como: cercanía angular multiplicada por el armónico n y evaluada dentro de un orbe armónico (AOH).

4) “serie armónica”

Cada curva corresponde a un armónico n (1..12), definido formalmente por la carta armónica n, no por una lista arbitraria de aspectos. Esto conecta directamente con: John Addey (cartas armónicas), y con los conceptos de ARMON (conjunción en la armónica).

5) “densidad y coherencia de conjunciones”

Esto describe exactamente el código: **densidad**: cuántos pares planetarios están simultáneamente activados; **coherencia**: qué tan exactas son esas conjunciones (`taper / orbe`). No basta con un aspecto aislado, el armograma sube cuando **muchas relaciones** “resuenan” con el mismo armónico.

6) “en la carta armónica correspondiente”

Esto pretende explicar que no se están sumando aspectos de la carta radical, sino evaluando la **proyección armónica**:

$$\lambda_h = (\lambda \cdot n) \bmod 360$$

Todo pasa por esa transformación.

7) “entre un conjunto de factores astrológicos”

Porque el código permite filtrar planetas, cambiar `pairing` (same vs all-to-all), usar tránsito↔natal, tránsito↔tránsito, progresado↔natal, etc. La definición no queda atada a una sola técnica.

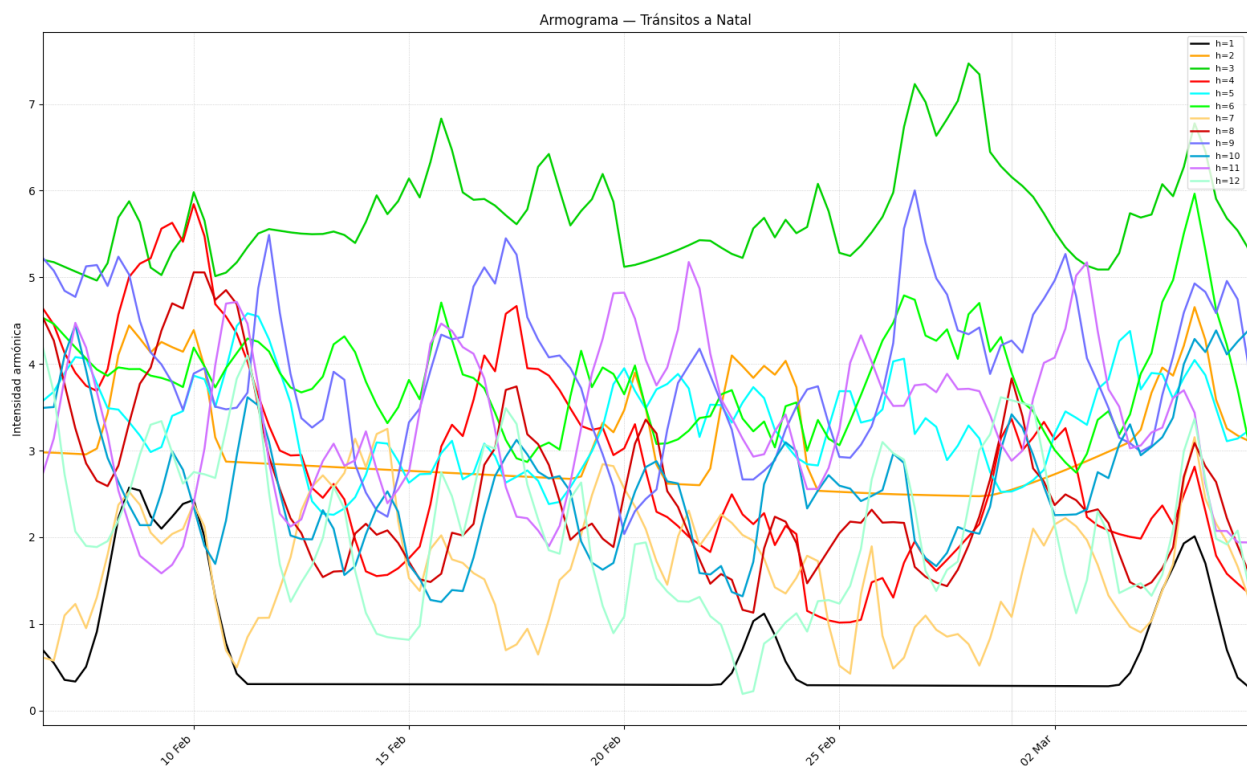
En resumen, el código implementado define al Armograma: gráfico temporal que cuantifica la activación de un armónico mediante la intensidad promedio de conjunciones en su carta armónica, calculada sobre un conjunto de relaciones planetarias a lo largo del tiempo.

Epistemológicamente: El armograma no es un instrumento interpretativo en sí mismo, sino un detector formal de patrones armónicos recurrentes. Su función es identificar momentos de máxima coherencia geométrica para una serie armónica determinada, dejando la interpretación simbólica como una capa posterior.

Intensidad Armónica

Los cálculos de armogramas que el código realiza no están diseñados para medir “**fuerza de aspectos**” en el sentido clásico astrológico, sino la **densidad y coherencia geométrica de una serie armónica en un instante dado**.

Lo que el código muestra en el eje Y (altura de las curvas) se llama **intensidad armónica**.



El código no resuelve la pregunta: “*Qué tan fuerte es este aspecto específico?*”, más bien contesta la pregunta: “*Cuántas relaciones planetarias están simultáneamente alineadas con esta estructura armónica?*” Eso es **estructura**, no valoración.

Menciono lo anterior porque pienso que vale la pena explicar el surgimiento de una posible confusión con el contexto histórico de los armogramas de John Addey y posterior trabajo de Miguel García Ferrández/Tito Maciá, en particular en relación con el concepto de Astrodinas.

En mi experiencia, he visto en común que en el ecosistema del programa Armon original, los conceptos de armogramas, astrodinas, ondas, curvas, picos se han usado muchas veces en paralelo o combinados en la práctica, pero no son el mismo concepto, ni nacen del mismo modelo. No se requiere este paralelismo para explicar armogramas.

Las astrodinas se usan como una medida cuantitativa del peso de los aspectos, útil para interpretación directa y práctica, pero, un astrólogo puede mirar un pico (armograma / onda) y luego leerlo con astrodinas, lo cual es una decisión interpretativa posterior, no la definición del armograma.

Miguel García plantea **el armograma como una representación temporal de resonancia armónica basada en cartas armónicas** (Addey), donde se observa cuándo un armónico “sube”, cuándo “baja”, cuándo domina sobre otros, dicho en términos muy simples. Los armogramas no son, en su origen, un desarrollo de las astrodinas, ni dependen conceptualmente de ellas.

El armograma tiene un enfoque estructural y fenomenológico, no es una interpretación en sí, no es un listado de aspectos, no es un juicio de fuerza psicológica. Es una imagen del comportamiento armónico del cielo en el tiempo. Esto encaja con lo que hace la implementación descrita en este documento. Por lo tanto lo único que quiero expresar es “*uso astrodinas para interpretar un pico*” no significa “*el armograma está calculado con astrodinas*”.

La implementación en código que explico se basa en un enfoque “Addey /Miguel García”

- usa transformación armónica ($\Delta \rightarrow dh$)
- mide cercanía a conjunción en carta armónica
- suma contribuciones geométricas
- produce curvas temporales por armónico
- no pondera aspectos clásicos,
- no asigna pesos interpretativos,
- no pretende decir qué es “más importante”.

Por tanto, conceptualmente, el armograma implementado se puede ver como **un detector de estructura armónica**, no como un índice interpretativo.

La intensidad armónica que el código usa pretende estar normalizada, esto es, **usar la misma función de ganancia para todos los armónicos**, lo cual permite comparar formas, ritmos, coactivaciones. No se introduce el concepto de pesos, por ejemplo, como las astrodinas, de lo contrario H2, H5 y H12 ya no serían comparables porque estarían “contaminados” por pesos subjetivos. Con esto, el código pretende ser más universal, quizá más cercano a la teoría de John Addey, pero intentando ser a la vez compatible con Armon.

El código hace algo mucho más básico para este propósito, usa una **función continua de intensidad**:

$$\text{intensidad} = f(\text{cercanía geométrica})$$

donde:

- exactitud $\Rightarrow$ más intensidad,
- lejanía $\Rightarrow$ menos intensidad,
- fuera del orbe $\Rightarrow$ cero.

Eso es un poco más matemático, neutral, reproducible, e intentar ser coherente con el concepto de armónico y de ahí que la implementación muestra **intensidad armónica** = “cuánto está presente esto en la geometría”

y de una manera más “académica”, podría decir:

“El presente modelo de armogramas no incorpora ponderaciones interpretativas del tipo astrodinas. En su lugar, la intensidad armónica se define como una magnitud geométrica, basada en la cercanía promedio a conjunción en la carta armónica correspondiente. Esta elección metodológica permite mantener la neutralidad formal del armograma, preservar la comparabilidad entre armónicos y evitar la introducción de interpretaciones dependientes de una línea conceptual específica.”

En un armograma **Tránsitos $\rightarrow$ Natal**, por ejemplo, calculado para un periodo de una semana, si observamos que el **armónico 6** muestra un “pico alto” un día concreto, eso significa que, en ese día, existe una alta coherencia geométrica de relaciones planetarias cuya separación angular está próxima a múltiplos de 60° ($360/6$), es decir, sextiles, trígonos y conjunciones, evaluadas en conjunto.

El armograma no indica “Este sextil X-Y es muy fuerte”, el armograma sí indica: “Ese día, el patrón armónico 6 está globalmente activado en muchas relaciones simultáneas”, de tal manera que es una **lectura de estructura**, no de fuerza puntual.

Dado lo anterior, un procedimiento posible para su uso podría ser:

1. El armograma se usa como detector temporal
 - Indica *cuándo* mirar con atención.
 - Reduce el problema de “revisar todo el cielo todos los días”.
2. Observar a la carta del día del pico, según la técnica usada
 - Tránsitos del día vs carta natal.
 - No para “confirmar el pico”, sino para identificar qué relaciones concretas lo generan.
3. Buscar en dicha carta relaciones compatibles con H6
 - Sextiles (60°)
 - Trígonos (120°)
 - Conjunciones (0°)
 - especialmente repetidas, encadenadas o involucrando planetas clave.

4. Realizar la interpretación astrológica con el “lenguaje habitual” del astrólogo que use (no dependiente de algún concepto particular): planetas, casas, significados, temas vitales, ciclos, etc.

Si en el paso 2 se encuentra, por ejemplo: Marte tránsito sextil Venus natal, Sol tránsito trígono Luna natal y Mercurio tránsito conjunción Júpiter natal, una posible interpretación puede ser algo como *“Ese día favorece acciones, intercambios, acuerdos y flujos armónicos, porque varias relaciones compatibles con el armónico 6 se activan simultáneamente.”*

No porque el armograma “diga” eso sino porque **esas relaciones** explican **por qué** H6 subió. El armograma **llevó al día correcto**, la carta **te dio el contenido**.

En este sentido el armograma **no reemplaza la carta**, la carta **da contenido** a lo que el armograma detecta como estructura activa. El armograma no interpreta; orienta. No jerarquiza; revela estructura. No sustituye la carta; la enfoca.

Qué es un Armónico?

La implementación usa “armónico” en el sentido de **Harmonic Astrology / Aspect Families**:

- Por un lado, existe la teoría **Addey/armónicos**: armónico n como **carta armónica n** , donde se multiplican longitudes por $n \pmod{360}$ y se observan conjunciones.
- Por otro lado, existe la teoría **Tito Maciá-Miguel García/armogramas**: armónico n como **familia de aspectos** asociada a la serie $k \cdot 360/n$, reducida a ángulos $0..180$.

El código soporta ambas, pero **por defecto** usa la de “conjunción en carta armónica n ”, pero al mismo tiempo alineada al programa Armon original de Miguel García. Esto se define en el código con:

```
USE_HARMONIC_CONJUNCTION: bool = True
AOH_DEGREES: float = 11.88
```

Estos parámetros dentro del programa establecen por defecto **el método de “conjunción en la carta armónica n ”**, que es estilo Addey en lo matemático (armónicos: multiplicar por n y ver conjunciones), pero también es estilo Armon (Miguel/Tito en práctica de harmogramas/armogramas), Armon suele leer armónicos como “conjunciones en armónicos”, la implementación pretende “no ser Addey vs Tito Maciá” como excluyentes. La implementación por defecto es el mismo núcleo matemático (cartas armónicas) que Addey formaliza y que ARMON usa en la práctica.

El código puede usar un modo alterno, cuando `USE_HARMONIC_CONJUNCTION=False` digamos que es un modo “legacy” de **familias por ángulos explícitos** (más cercano a la lectura “serie de aspectos” estilo Maciá/harmogramas tradicionales), lo cual es comparar Δ con los ángulos de `family_angles(n)` dentro del código.

Implementación técnica

Pseudo-algoritmo de la implementación

A fines de dar un entendimiento de la implementación en código, aquí describo, en forma de algoritmo simplificado, basada en el **código** `armogramas.py + armon2.py`. La idea es que, aunque el documento tenga secciones con temas por separado, aquí tengas la **secuencia real** y puedas ubicar conceptos

como `USE_HARMONIC_CONJUNCTION=True`, `AOH`, `taper(...)`, `_weight_for_angle()` y otras referencias al código y sus funciones.

A. Entradas, configuración y constantes relevantes

Entradas principales (desde UI / `armon2.py` → `armogramas.py`):

- `start_dt`, `end_dt` (fechas del rango, en hora local)
- `include_asc (bool)`: incluir Ascendente “ASC” como factor adicional
- `pairing_mode`: “ALL_TO_ALL” o “SAME_PLANET”
- `normalize_pairs (bool)`: normalizar por número de pares

Configuración global / contexto (definida en `armogramas.py`):

- `TIMEZONE` (zona horaria, p.ej. America/Mexico_City)
- `NATAL_DT_LOCAL` (fecha/hora natal local)
- `GEOLAT`, `GEOLON` (lat/lon natal)
- `SE_PATH` (ruta de efemérides Swiss Ephemeris)
- filtros opcionales:
 - `ARMOGRAM_T_PLANETS` (conjunto permitido lado tránsito)
 - `ARMOGRAM_N_PLANETS` (conjunto permitido lado natal)

Constantes del modelo armónico (núcleo):

- `USE_HARMONIC_CONJUNCTION = True` (modo por defecto)
- `AOH_DEGREES = 11.88` - orbe de conjunción en **carta armónica por defecto**
 - El método por defecto sigue siendo el de “conjunción en la carta armónica n”, pero el parámetro `AOH` (Armon Orbe Harmónico) ya no debe entenderse como una constante rígida del sistema. En la implementación actual de Armon2, el usuario puede modificar el valor de `AOH` desde Configuración → Ajustes Generales. El programa conserva 11.88 como valor inicial por defecto, pero permite cambiarlo, incluyendo como opción rápida el valor 27.69231 usado en el programa Armon. Por tanto, el cálculo del armograma depende del `AOH` activo en tiempo de ejecución, y el título del gráfico muestra explícitamente qué valor de `AOH` se utilizó en cada cálculo. Esto implica que `AOH` forma parte de la configuración operativa del armograma y no de una constante fija e inmutable del modelo.
- Función de atenuación:
 - `taper(distance_deg, orb_deg, power)`
 - en modo armónico usa `power = 1.2`
- `PLANET_POWER` (peso por planeta): en el original es esencialmente 1.0 para todos (neutral)

B. Flujo de alto nivel: qué hace armon2.py

1. La interfaz gráfica captura opciones (rango, modo, filtros, etc.)
2. Elige qué “familia de cálculo” usar según el tipo de armograma:
 - **Tránsito ↔ Natal:** `compute_series_daily(...)`
 - **Tránsito ↔ Tránsito:** `compute_series_daily_transit_to_transit(...)`
 - **Progresiones secundarias (día=año):** `compute_series_daily_secondary(...)`
 - **Direcciones simbólicas Naibod:** `compute_series_daily_dp(...)`
3. Recibe:
 - `times`: lista de datetimes muestreados dentro del rango
 - `series`: diccionario {n: [valores...]} para n=1..12
4. Dibuja:
 - modo “classic” (curvas sobre un eje Y común), o
 - modo “armon” (bandas visuales; *solo afecta visualización*).

C. Generación de tiempos: `generate_times_auto(start_dt, end_dt)`

El armograma no necesariamente se calcula “un valor por día”, sino con densidad variable:

1. Calcula `span_days = (end_dt - start_dt).days + 1`
2. Decide un step:
 - si `span_days <= 14`: cada 3 horas (8 muestras/día)
 - si `span_days <= 62`: cada 6 horas (4 muestras/día)
 - si `span_days <= 365`: 1 muestra/día
 - si 1–5 años: cada 3 días
 - si >5 años: semanal

Salida:

- `times = [t0, t1, ..., tk]` (en hora local)

D. Cálculo de longitudes planetarias: `get_planet_longitudes(dt_local, include_asc)`

Este es el “motor astronómico” de `armogramas.py` (*Swiss Ephemeris*).

D.1 Inicialización *Swiss Ephemeris* (una sola vez):

- `swe.set_ephe_path(SE_PATH)`
- `tabla name → swe.planet_id (SUN, MOON, ... PLUTO)`

- **flags:** `swe.FLG_SWIEPH` (sin velocidad, solo longitudes)

D.2 Convertir tiempo local a Julian Day UT:

1. `dt_utc = to_utc(dt_local, TIMEZONE)`
2. `jd_ut = swe.julday(y, m, d, hour_float, swe.GREG_CAL)`

D.3 Calcular longitudes (λ) para cada planeta:

Para cada planeta `p` en `PLANETS`:

- `(xx, retflag) = swe.calc_ut(jd_ut, planet_id, flag)`
- `lon = normalize_deg(xx[0])` (asegura 0..360)
- guardar en dict: `longs[p] = lon`

D.4 Ascendente (opcional):

Si `include_asc=True`:

- calcula casas (Placidus) con fallback:
 - intenta `swe.houses_ex(...)` con flags,
 - si no funciona por firma $\rightarrow$ `swe.houses_ex` sin flags,
 - fallback final $\rightarrow$ `swe.houses(...)`
- toma `asc = normalize_deg(ascmc[0])`
- `longs["ASC"] = asc`

Salida:

- `longs = {"SUN":  $\lambda$ , "MOON":  $\lambda$ , ..., "PLUTO":  $\lambda$ , ("ASC":  $\lambda$ )}`

E. Preparación de natal y filtros: `compute_series_daily(...)` (Tránsito $\leftrightarrow$ Natal)

E.1 Natal una sola vez

1. `natal_raw = get_natal_longitudes(include_asc)`
2. `natal = _apply_filter(natal_raw, ARMOGRAM_N_PLANETS)`

E.2 Para cada tiempo `t`

Para cada `dt_local` in times:

1. `trans_raw = get_planet_longitudes(dt_local, include_asc)`
2. `trans = _apply_filter(trans_raw, ARMOGRAM_T_PLANETS)`
3. Para cada armónico `n = 1..12`:
4. `val = _intensity_prefiltered(trans, natal, n, pairing_mode, normalize_pairs)`
 - agregar a `series[n].append(val)`

Salida:

- `times`
- `series[n]` (lista de floats) para $n=1..12$

F. Núcleo matemático: `_intensity_prefiltered(trans_lons, natal_lons, n, pairing, normalize_pairs, ...)`

Esta función es la **definición computacional** de “intensidad armónica”.

F.1 Diferencia angular por par

Para cada par de longitudes:

- $dt = (\lambda_T - \lambda_N) \bmod 360$

F.2 Elección del método (default vs legacy)

- `use_harmonic_conjunction = USE_HARMONIC_CONJUNCTION`
- `aoh = AOH_DEGREES (11.88)`

Modo por defecto (`USE_HARMONIC_CONJUNCTION=True`): conjunción en carta armónica n

1. Proyección armónica:

$$dh = (dt \cdot n) \bmod 360$$

2. Reducir a distancia mínima a 0°:

$$dh = \min(dh, 360 - dh) \Rightarrow dh \in [0, 180]$$

3. Atenuación por orbe AOH (en la armónica):

$$g = \text{taper}(dh, AOH, power = 1.2)$$

con:

- si $dh > AOH \rightarrow g = 0$
- si $dh \leq AOH \rightarrow$

$$g = \left(1 - \frac{dh}{AOH}\right)^{1.2}$$

Nota conceptual: esto implementa exactamente la idea “armónico n = conjunción en carta armónica n ”. Es el modelo “ARMON/Addey” aplicado a series temporales.

Modo legacy (`USE_HARMONIC_CONJUNCTION=False`): familia de ángulos explícitos

1. Para cada ángulo `ang` en `family_angles(n)`:

- distancia mínima:

$$d = |((dt - ang + 180) \bmod 360) - 180| \in [0, 180]$$

- ganancia candidata:

$$g_{ang} = \text{taper}(d, orb_override, power = 1.5)$$

2. Se toma el máximo de la familia:

$$g = \max_{ang \in family} g_{ang}$$

Nota: en la implementación, el “peso por ángulo” no se multiplica aquí (ver sección H).

F.3 Agregación por pares (pairing)

- Si pairing = "SAME_PLANET":
 - compara solo mismo planeta T con mismo planeta N
- Si pairing = "ALL_TO_ALL" (default conceptual del armograma):
 - compara cada planeta T con cada planeta N

Acumulación:

- total += (peso_planeta_T * peso_planeta_N) * g
- y count += 1 por cada par evaluado

Normalización (si normalize_pairs=True):

$$total = \frac{total}{count}$$

Escala final:

$$I_n(t) = 100 \cdot total$$

Este $I_n(t)$ es el valor que se grafica como “altura” de H_n .

G. Variantes del armograma (mismo núcleo, distintas longitudes)

G.1 Tránsito ↔ Tránsito: `compute_series_daily_transit_to_transit(...)`

- Para cada t: calcula longitudes del mismo instante
- Aplica filtro lado T y filtro lado N (pueden ser distintos)
- Llama al mismo `_intensity_prefiltered(...)`

G.2 Progresiones secundarias día=año: `compute_series_daily_secondary(...)`

Para cada t:

1. calcula `years = (t - NATAL_DT_LOCAL).days / 365.2422`

2. fecha simbólica:

$$t_{sym} = NATAL_DT_LOCAL + \text{timedelta}(\text{days} = \text{years})$$

3. longitudes progresadas = `get_planet_longitudes(t_sym, include_asc=False)`
(omite ASC por rendimiento)

4. compara progresado ↔ natal con `_intensity_prefiltered(...)`

G.3 Direcciones simbólicas Naibod: `compute_series_daily_dp(...)`

Aquí no se usa efemérides en el bucle.

Para cada t:

1. `years = (t - NATAL_DT_LOCAL).days / 365.2422`

2. arco Naibod:

$$arco = years \cdot NAIBOD_KEY$$

donde NAIBOD_KEY se toma de `astro_engine` si existe, si no usa 0.98564733

3) longitudes dirigidas:

$$\lambda_{dir} = (\lambda_{natal} + arco) \bmod 360$$

4. compara dirigido ↔ natal con `_intensity_prefiltered(...)`

H. Cómo se usan `family_angles`, `_weight_for_angle()` y `FAMILY_SPECS`?

`family_angles(n)`

Define, para cada n, la lista de ángulos “de familia” (0..180) y explicita subestructuras (p.ej. H6 incluye 60/120/180).

En el cálculo actual:

- Solo se usa si `USE_HARMONIC_CONJUNCTION=False` (modo legacy) o en utilidades de distancia.

`FAMILY_ANGLE_WEIGHTS` y `_weight_for_angle(n, ang)`

Definen pesos por ángulo dentro de familias compuestas (para evitar que un armónico compuesto sea “eco” de mayores).

`_build_family_specs()` y `FAMILY_SPECS`

Construye:

5. `FAMILY_SPECS[n] = [(ang, orb, weight_ang), ...]`

En la **rama por defecto** (conjunción armónica), no se usa `FAMILY_SPECS` ni `_weight_for_angle()`.

En la **rama legacy** (familias explícitas), el algoritmo actual usa un `w_override global` (por armónico), pero **no multiplica el peso por ángulo** (aunque está definido y listo para usarse). Esto permite estructura definida en el diseño, no aplicada en el flujo principal actual, disponible para extensión futura/experimental.

Mini “mapa” de funciones clave

Orquestación (armograma por día):

- `compute_series_daily(...)` → genera times → para cada t calcula longitudes → para cada n llama `_intensity_prefiltered(...)`

Astronomía (longitudes):

- `get_planet_longitudes(dt_local, include_asc)`
 - `swe_jd_from_local(dt_local)`
 - `swe.calc_ut(...)`
 - `_houses_with_fallback(...)` si ASC

Matemática armónica (intensidad):

- `_intensity_prefiltered(trans_lons, natal_lons, n, ...)`
 - `gain_for_pair(dt)`
 - **default:** `dh=(dt*n)%360, taper(dh, AOH, power=1.2)`
 - **legacy:** min distancia a `family_angles(n), taper(d, orb, power=1.5)`
 - agregación por pares + normalización + escala ×100

Modelo matemático

El modelo matemático se implementa en el módulo de código llamado *armogramas.py*, básicamente la obtención de longitudes eclípticas, definición de “familias armónicas”, orbes, pesos, y el cálculo de una **intensidad por armónico** para cada fecha del rango dado para los cálculos.

El programa principal *armon2.py*, recoge parámetros (rango, armónicos, filtros de planetas, modo) que un usuario proporciona en la interface gráfica y manda a calcular `times, series` llamando a una de estas funciones:

- `compute_series_daily()` para Tránsitos ↔ Natal
- `compute_series_daily_dp()` para Direcciones primarias simbólicas por arco Naibod, “natal dirigido” ↔ natal
- `compute_series_daily_transit_to_transit()` para Tránsito ↔ Tránsito)
- `compute_series_daily_secondary()` Para progresiones secundarias día=año)

Después de realizar el cálculo, el código crea un gráfico en 2 estilos posibles: **classic** (curvas superpuestas) o **armon** (bandas con “baseline” fijo por armónico).

El “motor” de cálculos contenido en `astro_engine.py` se usa **opcionalmente** (según el valor de `PS_ENGINE`) para calcular longitudes (por ejemplo, con la librería astronómica de *Skyfield*) y proveer constantes (p.ej. `NAIBOD_KEY`), además de utilidades astronómicas extra, no esenciales, al núcleo del armograma.

Datos astronómicos: qué se calcula y con qué efemérides

Longitudes planetarias (λ)

El armograma trabaja con **longitudes eclípticas geocéntricas** en grados 0..360 para: *Sun, Moon, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto* y opcionalmente **ASC** (ascendente).

Para armogramas de tipo **Tránsitos a Natal** (modo principal “T”), las longitudes se calculan usando *Swiss Ephemeris*:

- Se convierte la fecha local a UTC con `to_utc(dt_local, TIMEZONE)`
- Se obtiene el **Julian Day UT** con `swe.julday(...)` (`swe_jd_from_local`)
- Se calcula cada planeta con `swe.calc_ut(jd_ut, planet_id, flags)` y se normaliza a 0..360.

Para el **ASC**, el código usa casas Placidus (`hsys=b'P'`) con validación de retroceso por error:

- intenta `swe.houses_ex(...)` con flags de Swiss Ephemeris
- si falla por firma, intenta `houses_ex` sin flags,
- y si falla, usa `swe.houses(...)`
- De ahí toma `ascmc[0]` como ASC.

Nota metodológica: esto reproduce la práctica típica de programas astrológicos en cuanto a efemérides y longitudes.

Definición de “familias armónicas” (ángulos objetivo)

La función `family_angles(n)` define, para cada armónico, un conjunto de **ángulos objetivo** en 0..180:

- $n=1 \rightarrow [0]$ (conjunción)
- $n=2 \rightarrow [180]$ (oposición)
- $n=3 \rightarrow [120]$ (trígono)
- $n=4 \rightarrow [90, 180]$
- $n=5 \rightarrow [72, 144]$ (quintil/biquintil)
- $n=6 \rightarrow [60, 120, 180]$
- $n=7 \rightarrow [360/7, 2 \cdot 360/7, 3 \cdot 360/7]$ (septiles)
- $n=8 \rightarrow [45, 90, 135, 180]$

- $n=9 \rightarrow [40, 80, 120, 160]$
- $n=10 \rightarrow [36, 72, 108, 144, 180]$
- $n=11 \rightarrow [(1..5) \cdot 360/11]$
- $n=12 \rightarrow [30, 60, 90, 120, 150, 180]$

Esto refleja la idea “serie armónica = particiones del círculo” (Addey) y, a la vez, el uso práctico de familias con submúltiplos que suele emplearse en armogramas (Maciá).

Manejo de Subarmónicos: cómo se calculan para cada armónico y cómo se grafica

El armograma calcula la activación del armónico n midiendo la cercanía a conjunción en la carta armónica n , equivalente a que la separación angular entre dos factores esté próxima a múltiplos de $360/n$ en la rueda radical.

Esta propiedad introduce de forma natural la inclusión de ángulos compartidos por armónicos compuestos (p.ej., 180° es múltiplo de 90° y 30°), lo que explica la coactivación simultánea de series como H2, H4, H6 y H12 ante oposiciones fuertes.

Esto es exactamente el concepto de “*subarmónicos*” en la implementación.

En la implementación *por defecto* (`USE_HARMONIC_CONJUNCTION=True`), no se “eligen subarmónicos” explícitamente, más bien el cálculo del armónico n se hace así:

1. Para cada par planetario (Tránsito vs Natal, o el modo que se elija):
 - Δ = diferencia angular entre longitudes (0..360)
2. Se “proyecta” al armónico n :
 - $dh = (\Delta \cdot n) \bmod 360$, y luego se reduce a distancia mínima a 0 (0..180)
3. Se mide que tan cerca está dh de 0° usando el orbe armónico AOH (11.88°):
 - $gain = taper(dh, AOH, power=1.2)$

En este enfoque, los subarmónicos no se listan o especifican como “H4 usa H2 y H1”, etc. Lo que ocurre es más dentro de un enfoque matemático.

Un armónico n “dispara” cuando Δ está cerca de **cualquier múltiplo** de $360/n$ (en la rueda radical), porque eso hace que $(\Delta \cdot n)$ caiga cerca de 0° en la armónica.

Ejemplos (radical):

- **H4:** se activa cuando Δ está cerca de $0^\circ, 90^\circ, 180^\circ, 270^\circ$ (porque $360/4 = 90$)
- **H12:** se activa cerca de $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 180^\circ$... (porque $360/12 = 30$)

Entonces, sí hay “subestructuras” o subarmónicos, pero aparecen **automáticamente** como consecuencia de la aritmética de los múltiplos:

- H12 incluye 60/90/120/180 porque son múltiplos de 30.

- H6 incluye 120/180 porque son múltiplos de 60.
- H4 incluye 180 porque es múltiplo de 90.

A eso le llamamos “*relación con subarmónicos*”: no porque el código llame a H2 o H3, sino porque **un armónico compuesto contiene ángulos que coinciden con armónicos menores**.

La implementación, por lo tanto, contiene 2 maneras de calcular subarmónicos, y entonces “sabe” cuales ángulos pertenecen a un armónico dependiendo del modo:

Modo default (conjunción armónica) – Si el código tiene (por defecto) `USE_HARMONIC_CONJUNCTION=True`

No usa una lista de ángulos. El criterio es: Pertenecer a H_n cualquier Δ que esté cerca de $k \cdot (360/n)$. Esto está implícito en $dh = (\Delta \cdot n) \bmod 360$ y medir distancia a 0.

Modo alterno (familias explícitas) - Si el código tiene `USE_HARMONIC_CONJUNCTION=False`

El código usa una lista explícita: `family_angles(n)` devuelve ángulos tipo:

- H5 → 72, 144
- H8 → 45, 90, 135, 180
- H12 → 30, 60, 90, 120, 150, 180

Esto es la manera “aspectual” (familias) de expresar subarmónicos: un compuesto trae varios ángulos. Aunque la lista existe en el código, no se usa porque está en el modo default.

Cómo se grafica en relación con subarmónicos?

Es importante separar **cálculo** vs **visualización**. Con respecto al *cálculo*, el código calcula 12 series independientes: `series[n][t]` para $n=1..12$. Cada una es “cuánta conjunción armónica” hay para ese n en esa fecha.

No hay existe una curva “H12 descompuesta en H6+H4+H3...”, la relación con subarmónicos se manifiesta porque, si hay un aspecto mayor fuerte (p.ej. 180), **puede levantar varias series** (H2, H4, H6, H12) a la vez. Como se explicó anteriormente.

Para la *Visualización*, se implementan dos modos en `armon2.py`, no se dibujan subarmónicos como “componentes”, el código lo puede hacer de dos maneras posibles:

1. Modo classic (por defecto - visual)

- Dibuja las 12 curvas (H1..H12) sobre el mismo eje Y (“intensidad”).
- Aquí se aprecia la relación con subarmónicos porque se ve, por ejemplo, picos simultáneos en H2/H4/H6/H12 cuando hay una oposición fuerte.

2. Modo armon (estilo bandas)

- Cada armónico se dibuja como una “banda” con un *baseline* fijo (separación vertical por armónico).

- Además, se hace un re-escalado robusto por porcentajes (p5–p95) solo para legibilidad (para que no quede una banda plana si ese armónico tiene rango pequeño).
- Esto **no cambia el cálculo**, solo “acomoda” la lectura.

En ambos modos, la relación con subarmónicos está no porque el programa dibuje subarmónicos, sino porque un mismo evento angular (p.ej. 90 u 180) pertenece a varios armónicos compuestos, y por eso se ven correlaciones/picos cruzados.

El código usa una **función continua** $\text{taper}(d, \text{orb})^{\text{power}}$ que asigna un peso entre 0 y 1 según la cercanía dentro del orbe. Eso da una “energía” continua por par planetario, que luego se agrega y se normaliza. Dentro de la implementación:

- No hay una tabla de **ponderación por tipo de aspecto** (conjunción vs oposición vs trígono, etc.) en el modo por defecto.
- No hay pesos planetarios diferenciales (todos los planetas pesan 1.0).
- En el modo por defecto ni siquiera se distingue “qué aspecto fue”: todo es “conjunción armónica”.

La implementación incluye código con estructuras como `FAMILY_ANGLE_WEIGHTS` (pesos por ángulo dentro de un armónico compuesto), que es lo que podría aproximarse como la definición de “astrodinias por aspecto dentro del armónico” pero, actualmente, el modo defecto `USE_HARMONIC_CONJUNCTION=True` **no usa ángulos explícitos**, así que no puede aplicar esos pesos. Aún en el modo legacy por ángulos, la función de ganancia no está aplicando `_weight_for_angle()`.

Este código existe sólo por cuestiones de uso futuro, pero no es utilizado activamente en los cálculos.

La implementación es entonces un modelo de **intensidad continua por cercanía a conjunción armónica** (más un promedio/normalización por cantidad de pares), intensidad continua basada en la distancia dentro de un orbe armónico (AOH) mediante una función de decaimiento (`taper`).

En el gráfico generado de los armogramas, la “intensidad armónica” que aparece en el eje Y es una magnitud cuantitativa que expresa **el grado promedio de conjunción armónica (para un armónico n) entre los pares planetarios evaluados en un instante dado**.

Matemáticamente, para cada fecha t y cada armónico n :

1. Se calcula, para cada par planetario (p, q) , una **ganancia** $g_{pq,n}(t) \in [0,1]$ basada en la cercanía a conjunción en la carta armónica n (usando AOH y la función `taper`).
2. Se agregan todas esas ganancias:

$$G_n(t) = \sum_{p,q} g_{pq,n}(t)$$

3. Se normaliza por el número de pares evaluados (si está activo):

$$I_n(t) = 100 \cdot \frac{G_n(t)}{N_{\text{pares}}}$$

$I_n(t)$ es exactamente el valor que se grafica en el eje Y.

La altura de una curva, cuando es mayor en el gráfico significa que más pares planetarios simultáneamente cercanos a conjunción en la armónica n, y/o conjunciones más exactas (menor distancia angular).

Una **altura menor** significa menos pares activados, o activaciones más débiles (cercanas al borde del orbe). Es una medida **continua**, no binaria.

En el modo “armon” de visualización, `armogram_view_mode = "armon"`, el valor interno $I_n(t)$ no cambia, lo que cambia es cómo se presenta gráficamente, cada armónico se dibuja con un *baseline* vertical propio, y se hace un re-escalado visual (porcentaje 5–95) solo para lectura. El eje Y se vuelve una proyección por banda. La señal es intensidad armónica; la coordenada vertical está re-proyectada para legibilidad por armónico.

Entonces, el modo de visualización estilo armon `armogram_view_mode = "armon"` en el código, introduce una separación vertical y un re-escalado gráfico con fines de legibilidad, sin alterar los valores de intensidad armónica calculados.

Estos modos son comparables entre armónicos, en términos relativos, porque todos están normalizados (0–100), usan el mismo tipo de función de ganancia, y el mismo criterio de agregación. Pero no implica que $H_{12} = 50$ sea “equivalente psicológicamente” a $H_4 = 50$. La comparación es estructural y dinámica, no interpretativa directa.

En resumen, el eje Y del armograma representa la intensidad armónica, definida como el promedio normalizado de la cercanía a conjunción en la carta armónica n entre los pares planetarios evaluados. La altura de cada curva o banda corresponde directamente al valor numérico de dicha intensidad y constituye el resultado cuantitativo del cálculo armónico, no una escala simbólica o arbitraria.

Modos de visualización implementados

En la implementación, no hay que confundir `USE_HARMONIC_CONJUNCTION = False` con el “modo armon de visualización” que corresponde a `armogram_view_mode = "armon"`,

El modo de visualización es un parámetro de visualización leído por `armon2.py` para dibujar en 'classic' o 'armon'. No afecta los valores numéricos de `series[n][t]`.

Concepto	Codigo	Afecta	Uso
<code>USE_HARMONIC_CONJUNCTION</code>	<code>armogramas.py</code>	Cálculos	Define cómo se calcula la intensidad armónica
<code>armogram_view_mode = "armon"</code>	<code>armon2.py</code>	Visualización	Decide cómo se dibujan las curvas

`USE_HARMONIC_CONJUNCTION` cuando es `True`, el valor por defecto, el armónico se calcula como **conjunción en la carta armónica n**, matemáticamente:

- se multiplica Δ por n
- se mide cercanía a 0°

Es la opción defecto, es el núcleo tipo Addey / pero considerando Armon.

Cuando `USE_HARMONIC_CONJUNCTION` es `False` el armónico se calcula como: cercanía a los **ángulos explícitos** de la familia (30, 60, 90, etc.), usa `family_angles(n)`. Es un enfoque **aspectual por familias**, no es el default, no cambia cómo se dibuja nada.

`armogram_view_mode` cuando es "classic", todas las curvas comparten el mismo eje Y, es la gráfica "tradicional estilo armon" donde cada armónico se dibuja en su **propia banda vertical**, se usa un *baseline* fijo por armónico y se re-escala solo para legibilidad (porcentajes), no cambia los valores calculados, es puramente **gráfico**, no matemático.

Orbes y pesos: cómo se controla qué cuenta y cuánto cuenta

AOH (Armon Orbe Armónico)

El AOH (Armon Orbe Harmónico) es el **orbe permitido para la conjunción dentro del cálculo de la carta armónica** y tiene una consecuencia importante: dado que se multiplica por n , un pequeño Δ en la rueda radical se vuelve grande en la armónica. Por eso, mucha gente explica que "el orbe efectivo en la radical" se comporta aproximadamente como AOH / n (intuición útil, aunque el código mide directamente en la armónica).

En la implementación **AOH_DEGREES = 11.88°** está puesto como un valor práctico/caligrafiado, es un parámetro empírico/configurable (no derivado por una fórmula dentro del código) para aproximar 12° y así poder calibrar el comportamiento del armograma para que "se parezca" a la forma en que lo manejan/explican los armogramas tipo Armon, donde es configurable.

En el código, el modo por defecto (`USE_HARMONIC_CONJUNCTION=True`), para un par planetario se calcula:

- $dh = (\Delta \cdot n) \bmod 360 \rightarrow$ distancia en la carta armónica n

Δ = diferencia angular real entre dos factores astrológicos en la carta original (radical), esto es, cuántos grados separan a dos puntos sobre el círculo zodiacal. En el código se maneja formalmente como

$$\Delta = (\lambda_T - \lambda_N) \bmod 360$$

donde

λ_T = longitud eclíptica del planeta en tránsito (o progresado, dirigido, etc.)

λ_N = longitud eclíptica del planeta natal

$$\Delta \in [0^\circ, 360^\circ)$$

dh = es la distancia angular en la carta armónica n , derivada de Δ . Mide qué tan cerca está una relación planetaria de una conjunción en la carta armónica n . Determina qué tan cerca está dh de 0° usando: `taper(dh, AOH_DEGREES, power=1.2)`. Formalmente:

$$dh = \min((\Delta \cdot n) \bmod 360, 360 - ((\Delta \cdot n) \bmod 360))$$

con

$$dh \in [0^\circ, 180^\circ]$$

Δ vive en la **carta radical**

dh vive en la **carta armónica**

En el modo por defecto (`USE_HARMONIC_CONJUNCTION=True`) el armónico n se activa entonces cuando:

$$dh = (\Delta \cdot n) \bmod 360$$

quedando cerca de 0°, y eso se mide con:

$$g = \text{taper}(dh, AOH, \text{power} = 1.2)$$

AOH por lo tanto controla qué tan “ancho” es el margen para considerar “conjunción” en la carta armónica. **11.88° = 11°52.8'**, aproximadamente 12 grados, pero con una cifra un poco más fina. En el código no hay una fórmula que derive 11.88; se usa como **parámetro empírico/de configuración** para que el sistema “calce” con el comportamiento deseado.

Como el orbe se aplica en la armónica, el “orbe efectivo” aproximado en la rueda radical se percibe como:

$$\text{orbe_radical} \approx \frac{AOH}{n}$$

Por ejemplo, para **n=6**, sería $\sim 11.88/6 \approx 1.98^\circ$. Para familias con subarmónicos (6 incluye 60/120/180; 12 incluye 30/150 y mayores).

Si en la carta armónica se tiene

$$dh = \Delta \cdot n$$

Solo cuando la distancia armónica dh cae dentro del orbe armónico AOH , la relación aporta intensidad armónica distinta de cero. Es una **condición computacional**, al “despejar”:

$$dh = (\Delta \cdot n) \bmod 360$$

y observar que para que haya contribución debe cumplirse

$$dh \leq AOH$$

se obtiene, como consecuencia matemática, que:

$$\Delta \lesssim \frac{AOH}{n}$$

La función de atenuación asigna contribución armónica únicamente a valores de dh comprendidos dentro del orbe armónico AOH .

Si queremos entender cuándo hay contribución distinta de cero, observamos:

$$dh = \Delta \cdot n$$

y como la contribución ocurre solo dentro del intervalo $dh \leq AOH$, se deduce:

$$\Delta \cdot n \lesssim AOH \Rightarrow \Delta \lesssim \frac{AOH}{n}$$

Todo este razonamiento se hace en el entorno donde

$$dh \approx 0$$

es decir, cerca de una conjunción en la carta armónica, fuera de eso el valor de dh “salta” por el módulo 360, y ya no se puede linealizar $dh \approx \Delta \cdot n$, la intuición “ $\Delta \approx AOH/n$ ” deja de ser útil.

De lo anterior, podemos determinar que cuanto mayor es el armónico n , más pequeño debe ser el Δ original para que “caiga” dentro del mismo AOH armónico. Por eso $AOH = 11.88^\circ$ en H6 $\rightarrow \Delta \lesssim 2^\circ$, en H12 $\rightarrow \Delta \lesssim 1^\circ$, esto es válido solo cerca de conjunción armónica.

En resumen, el parámetro Δ_{OH} (Armon Orbe Harmónico) define el margen angular permitido para considerar una conjunción en la carta armónica n . Este orbe se aplica en el espacio armónico transformado y es independiente del armónico considerado. Sin embargo, dado que la diferencia angular radical se multiplica por n antes de evaluar la cercanía a la conjunción, el efecto práctico es que, a mayor valor de n , se requiere una mayor exactitud geométrica en la carta radical para que se produzca una contribución significativa al armograma. Esta relación no implica la definición de orbes radicales decrecientes, sino que emerge directamente de la transformación armónica.

Relación conceptual entre Δ y dh

La transformación clave del modelo armónico es

$$\lambda_n = (\lambda \cdot n) \bmod 360$$

Aplicada a una diferencia angular

$$dh = (\Delta \cdot n) \bmod 360$$

Esto tiene una consecuencia esencial: Una separación Δ en la carta radical corresponde a una conjunción ($dh \approx 0^\circ$) en la carta armónica n **si y solo si** Δ está cerca de un múltiplo de $\frac{360^\circ}{n}$.

Ejemplo con Armónico 4

- $\frac{360}{4} = 90^\circ$
- $\Delta \approx 0^\circ, 90^\circ, 180^\circ, 270^\circ$
→ $dh \approx 0^\circ$ → alta activación de H4

Ejemplo con Armónico 6

- $\frac{360}{6} = 60^\circ$
- $\Delta \approx 0^\circ, 60^\circ, 120^\circ, 180^\circ$
→ $dh \approx 0^\circ$ → alta activación de H6

Ejemplo con Armónico 12

- $\frac{360}{12} = 30^\circ$
- $\Delta \approx 0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 180^\circ$
→ $dh \approx 0^\circ$ → alta activación de H12

Por eso Δ identifica la **geometría radical** y dh evalúa la **resonancia armónica**. El código trabaja con dh y no directamente con Δ porque el objetivo del armograma no es detectar aspectos individuales, sino responder: *“Esta relación pertenece a la estructura armónica n ? lo cual se responde así:*

- Si $dh \approx 0^\circ$ → sí pertenece (con cierta intensidad)
- Si $dh \gg 0^\circ$ → no pertenece

y esa cercanía se mide con el orbe armónico **AOH** y la función `taper`. dh entra en el cálculo de la intensidad, una vez calculado dh con la función `taper` del código se tiene

$$g = \text{taper}(dh, AOH, power)$$

- $dh = 0^\circ \rightarrow g = 1$ (máxima contribución)
- $dh = AOH \rightarrow g = 0$
- $dh > AOH \rightarrow g = 0$

Por lo tanto, dh es la variable geométrica central que gobierna la intensidad armónica.

Orbe por armónico (FAMILY_ORB)

La tabla `FAMILY_ORB` define un orbe base en grados para cada armónico 1..12 (más estrecho conforme sube n) de acuerdo con el criterio usado en Armon (Tito Maciá / Miguel F.) de la pauta: $360^\circ/(h+1) * \text{orbe_base}$, donde en Armon aparece 'Armónico Orbe = 27.69231'. Esto encaja con el criterio práctico de que armónicos altos requieren orbes más cerrados-

El código lo incluye, pero no es usado a menos que `USE_HARMONIC_CONJUNCTION = False`, lo cual no es el funcionamiento por defecto del código. Existe ya que fue utilizada durante el desarrollo del código y se decidió dejarlo, pero la implementación no la usa. En vez, ver la explicación de AOH anterior.

```
FAMILY_ORB: Dict[int, float] = {
    # Esta tabla ya está convertida a grados de orbe angular para cada armónico.
    1: 6.50,    # conjunción
    2: 4.33,    # oposición
    3: 3.24,    # trigono
    4: 2.59,    # cuadratura
    5: 2.16,    # quintil / biquintil (familia 5)
    6: 1.85,    # sextil (familia 6; incluye 120/180 como sub-armónicos)
    7: 1.62,    # septil (51°26', 102°51', 154°17')
    8: 1.44,    # semicadratura / sesquicadratura (familia 8)
    9: 1.29,    # novil (40°) y derivados (80°, 120°, 160°)
    10: 1.18,   # decil (36°) y derivados
    11: 1.08,   # oncil (360/11) y derivados
    12: 0.99,   # semisextil (30°) + quincuncio (150°) + tradicionales (60/90/120/180) en armónico
}
12
}
```

Pesos por planeta (PLANET_POWER)

Dentro del código, no se consideran pesos para cada planeta para los cálculos, los planetas tienen peso neutral: todos 1.0, es decir: la forma del armograma la dominan las **relaciones angulares**, no la jerarquía planetaria.

Pesos por ángulo dentro de una familia (FAMILY_ANGLE_WEIGHTS)

El modo de familias explícitas utiliza `FAMILY_ORB` cuando el código se fija `USE_HARMONIC_CONJUNCTION = False` para definir el orbe por armónico, pero los pesos por ángulo definidos en la tabla `FAMILY_ANGLE_WEIGHTS` no se aplican en la implementación actual, aunque la estructura de datos para ello (`FAMILY_SPECS`) esté presente como soporte para extensiones futuras.

En la implementación actual no se aplica al cálculo de la ganancia por par. En *armogramas.py* se construye `FAMILY_SPECS` con `_weight_for_angle()`, pero `gain_for_pair()` no multiplica por ese peso en ninguna rama. Por lo tanto, es 'estructura disponible' más que 'mecánica activa'.

Aunque no se usa, este código es parte del estilo Armon/Maciá-García. En armónicos compuestos (6,8,9,10,12), el código **reduce el peso** de ángulos que también son “mayores” (p.ej. 120 o 180) para que el armónico no se dispare solo por heredar aspectos fuertes de series inferiores. Ejemplos:

- En H6: 60° peso 1.0, 120° peso 0.70, 180° peso 0.40
- En H12: 30° y 150° peso 1.0, pero 60/90/120 bajan a 0.35 y 180 a 0.20

Esto es una **decisión de ingeniería** para estabilizar la señal por armónico y mantener interpretabilidad, si no, muchos armónicos altos se volverían “proxies” de oposición/trígono.

“Proxy” aquí significa: una señal que parece del armónico alto, pero en realidad está reflejando un aspecto mayor de armónicos bajos: En familias compuestas, como el 12 por ejemplo, que incluye ángulos como **180°** (oposición) dentro de la serie del 12 (porque 180 es múltiplo de 30), si hay una oposición fuerte en el cielo, podría activar H12 aunque no haya nada “realmente” del armónico 12 (30°/150°) dominante. En ese caso, la curva de H12 se vuelve un **proxy** (un sustituto/indicador indirecto) de la oposición (H2), en lugar de representar “lo propio” del 12.

Cómo se calcula la “intensidad” diaria de un armónico

La función clave es `_intensity_prefiltered(trans_lons, natal_lons, n, ...)`.

La intensidad que se calcula viene de `armogramas.py` y es el resultado de dicha función que calcula una ganancia por par planetario dentro de un orbe, agrega, normaliza (si `NORMALIZE_BY_NUM_PAIRS=True`) y entrega algo tipo **0..100**.

Eso es la “intensidad matemática” (la serie numérica real). Ésta se dibuja en `armon2.py`, el `plot` tiene dos “vistas”: `view_mode = getattr(armo, "armogram_view_mode", "classic")` mencionadas anteriormente. Si no se indica nada, por defecto es `classic` donde todas las curvas comparten el mismo eje (y=“intensidad armónica”). Si se cambia (en `armogramas.py`) a:

```
armogram_view_mode = "armon"
```

entonces `armon2.py` dibuja en “bandas” (baselines por armónico) y hace un **re-escalado robusto por percentiles (p5–p95)** solo para que “se lea bonito”, pero **no cambia la serie calculada**, solo cambia la proyección visual. `armon2.py` siempre usa la intensidad matemática calculada, el modo “armon” es solo el estilo de dibujo.

Pares planetarios (pairing)

Hay 2 modos:

- **ALL_TO_ALL** (por defecto): compara cada planeta de Tránsitos contra cada planeta de Natal. Si hay 10 planetas, son 100 pares (o 121 si se incluye ASC en ambos lados).
- **SAME_PLANET**: compara solo el mismo planeta con su natal (Sol con Sol, Luna con Luna, etc.).

6.2. Medida angular básica

Para cada par, se calcula:

- $dt = (\lambda_T - \lambda_N) \bmod 360$

A partir de dt , el algoritmo calcula “cercanía a aspecto” con una de dos estrategias:

Estrategia A (por defecto): Conjunción en la carta armónica n (estilo ARMON)

Controlada por `USE_HARMONIC_CONJUNCTION = True`.

La idea (clásica de Addey y típica en software armónico) es:

1. Convertir el diferencial a coordenada armónica:
 - $dh = (dt * n) \bmod 360$
 - se reduce a $0..180$ (distancia mínima al $0/360$)
2. Evaluar cercanía a 0° (conjunción) con un orbe **AOH** (“Armón Orbe Harmónico”):
 - `AOH_DEGREES = 11.88`
3. Convertir distancia a “ganancia” suave con una campana:
 - `g = taper(dh, AOH, power=1.2)`

Donde `taper(d, orb, power)` es:

 - si $d > orb \rightarrow 0$
 - si $d \leq orb \rightarrow (1 - d/orb)^{power}$

La interpretación técnica de lo anterior es que cada par planetario aporta una “probabilidad/intensidad” continua (no binaria) de estar en conjunción en la carta armónica n , esto se aproxima a Armon / Addey por:

- Addey populariza el uso de **cartas armónicas** (multiplicar longitudes por n) como la forma “limpia” de representar familias de aspectos.
- Armon (García/Maciá) usa armónicos como lectura de **conjunciones en la armónica**, que equivalen a aspectos de la radical en esa serie.

El código lo implementa literalmente (multiplica diferencia por n y mide cercanía a 0°), con un orbe AOH configurable.

Estrategia B (legacy): distancia al conjunto explícito de ángulos

Si `USE_HARMONIC_CONJUNCTION = False`, entonces:

1. Para cada ángulo ang en `family_angles(n)`:
 - d = distancia angular mínima entre dt y ang
 - `g = taper(d, orb_override, power=1.5)`
2. se toma el **máximo** g de la familia.

Este modo se parece más a un “detector de aspectos” clásico por familia.

Agregación, normalización y escala final

- Se acumula total += (peso_planeta_T * peso_planeta_N) * g (en ALL_TO_ALL)
- count se incrementa por cada par evaluado
- si NORMALIZE_BY_NUM_PAIRS=True, entonces:
 - total = total / count
- y finalmente:
 - intensidad = 100 * total

Así, cada armónico n produce una serie 0..100 aproximadamente (dependiendo de orbes y taper), comparable día a día.

Construcción del armograma: series temporales y muestreo

Tránsitos ↔ Natal (`compute_series_daily`)

Para cada día/hora de la rejilla temporal:

1. calcula longitudes del tránsito con *Swiss Ephemeris*
2. calcula longitudes natales (una vez)
3. por n=1..12, calcula `_intensity_prefiltered(...)`, devuelve:
 - times: [datetime...]
 - series: {n: [float...]}

Direcciones en armogramas

Conviene distinguir dos niveles dentro del programa:

1. Direcciones Simbólicas (DS) como técnica general de Armon2

La técnica general de Direcciones Simbólicas usa actualmente un arco fijo:

- 1° por año por defecto
- opción Naibod mediante checkbox

2. Direcciones Primarias (DP) dentro del módulo de Armogramas

En el armograma, la variante de DP se presenta explícitamente como:

“Direcciones primarias zodiacales Naibod a Natal”

Esto significa que la curva DP del armograma no representa cualquier modalidad de direcciones disponible en la aplicación, sino una variante concreta, etiquetada de forma explícita, para mantener claridad metodológica y comparabilidad de resultados.

Direcciones primarias simbólicas (`compute_series_daily_dp`) - NOTA: Esta función está deshabilitada temporalmente

Implementa un modelo “rápido”:

- Desplaza todas las longitudes natales por un arco:

- `arco = edad_en_años * NAIBOD_KEY`

- `NAIBOD_KEY ~ 0.98564733°/año` (tomado de *astro_engine.py* si está definido en el código, si no usa ese por defecto)

- Luego compara “natal dirigido” ↔ natal con el mismo motor armónico.

Importante: actualmente este cálculo es **simbólico por arco** (no calcula primarias astronómicas completas), pero sirve como “armograma de direcciones” coherente con un enfoque tipo Naibod.

Tránsito ↔ Tránsito (`compute_series_daily_transit_to_transit`)

En esta variante se comparan posiciones de tránsito del mismo instante entre sí, pero la implementación actual evita dos sesgos matemáticos importantes:

1. No se cuentan auto-pares del tipo P→P.
2. Cada par no ordenado se cuenta una sola vez, evitando duplicar simultáneamente P→Q y Q→P.

Esto permite que la curva de Tránsito ↔ Tránsito represente mejor la coherencia armónica global del conjunto de tránsitos, sin introducir una base artificial constante ni inflar la intensidad por duplicación de pares simétricos. Mismo día, mismas longitudes, pero:

- lado T se filtra con `ARMOGRAM_T_PLANETS`
- lado N se filtra con `ARMOGRAM_N_PLANETS` y se comparan entre sí.

Progresiones secundarias (`compute_series_daily_secondary`)

Implementa el principio “día simbólico = 1 año real”, comparando posiciones progresadas ↔ natal con el mismo núcleo armónico.

En la implementación actual, el armograma de Progresiones Secundarias puede incluir opcionalmente el Ascendente progresado. Cuando se incluye ASC, el cálculo respeta el mismo modo de casas activo en la técnica principal de Progresiones Secundarias:

- por defecto: `ARMC natal + años * clave Naibod`
- opcional: Casas/MC reales de la fecha simbólica

De esta forma, el armograma mantiene coherencia con la técnica mostrada en la interfaz general del programa.

Por razones de rendimiento, si el rango temporal del armograma supera 20 años, el sistema desactiva automáticamente la inclusión del ASC progresado. Esta desactivación no altera el cálculo planetario del armograma, sino únicamente evita el costo adicional del Ascendente progresado en rangos muy extensos.

- calcula `years` desde natal

- **construye** `symbolic_local = natal_local + timedelta(days=years)`
- **calcula longitudes** en esa fecha simbólica (según valor de `PS_ENGINE`):
 - `swe`: usa *Swiss Ephemeris* (coherencia con tránsitos)
 - `ae`: usa funciones dentro de *astro_engine.py* (*Skyfield/TS/planet_long*) y mapea nombres
- **compara progresado** ↔ natal.

Además, adapta el muestreo según el periodo a calcular dado:

- hasta 1 año: diario
- 1–5 años: cada 3 días
- 5 años: semanal

Filtros de planetas (relevante para “estilo armon”)

La interfaz gráfica permite elegir qué planetas entran “del lado T” y “del lado N” con un **código de letras** (`ARMOGRAM_DEFAULT_CODES = "elhvmsunpa"`), y `set_planet_filters_for_armogram()` convierte eso a conjuntos internos.

Esto reproduce una práctica común en armogramas: crear armogramas “por subconjuntos” (p.ej. solo luminarias, o solo sociales/transpersonales).

Suavizado y presentación (armon2.py)

La interfaz actual muestra en el título del gráfico información operativa relevante para interpretar correctamente el cálculo realizado. En particular, el título del armograma incluye:

- el tipo de armograma calculado
- la referencia técnica correspondiente
- el valor de AOH activo en ese cálculo

Por ejemplo, el gráfico puede indicar explícitamente:

“Tránsitos a Natal — AOH 11.88”

o

“Direcciones primarias zodiacales Naibod a Natal — AOH 27.69231”

Esto permite que el gráfico conserve trazabilidad metodológica visual inmediata, especialmente al comparar capturas, pruebas o variantes del mismo rango temporal.

- `SMOOTH_WINDOW_DAYS` existe, pero el `postprocess_series` está neutral (compatibilidad: no aplica `display gain` ni re-escala).
- La interfaz gráfica tiene dos modos de vista:
 - `classic`: todas las curvas en el mismo eje.

- **armon:** “bandas” por armónico con *baseline* fijo y re-escalado robusto (percentiles 5–95) solo para visualización, sin alterar el cálculo base.

La intensidad armónica que el programa maneja es “física/matemática” (0..100 promedio de pares) y el modo ARMON es una proyección gráfica para una lectura cómoda.

Cómo se aproxima el armograma implementado con las teorías de Addey, Hamblin, García Fernández y Tito Maciá

John Addey (Harmonic Astrology)

Coincidencia con la implementación: La idea de que un armónico n se evalúa mirando **conjunciones en la carta armónica n** se implementa directamente con $dh = (dt * n) \bmod 360$ y distancia a 0° . Eso es prácticamente la traducción algorítmica de la operación armónica.

David Hamblin (enfoque práctico/interpretativo sobre armónicos)

La implementación es congruente con el uso de armónicos como “capas” de lectura, series temporales para ver “activación”, y el uso de orbes más finos conforme sube n .

Miguel García Ferrández (Armon) y Tito Maciá (armogramas)

La aproximación de la implementación a esa técnica es por estos rasgos:

- **Conjunción armónica** como criterio principal (`USE_HARMONIC_CONJUNCTION=True`)
- **AOH** como orbe en la armónica (`AOH_DEGREES`)
- **Corrección de pesos en compuestos** (con `FAMILY_ANGLE_WEIGHTS`) para que armónicos altos no sean “eco” de oposición/trígono/cuadratura.

La idea es exactamente la de los “proxies” mencionada anteriormente, en armónicos compuestos (4,6,8,10,12), aparecen ángulos que también pertenecen a armónicos “más básicos”.

- H4 incluye 180° (que es H2).
- H6 incluye 120° (H3) y 180° (H2).
- H12 incluye $60/90/120/180$ (H6/H4/H3/H2).

La corrección de pesos significa darles menos peso a esos ángulos heredados (como 180, 120, 90) dentro del armónico compuesto, para que el armónico compuesto no sea “eco” de los mayores. En la tabla que se incluye en la implementación, por ejemplo:

- H4: 90 pesa 1.0, pero 180 pesa 0.60
- H6: 60 pesa 1.0, 120 pesa 0.70, 180 pesa 0.40
- H12: 30 y 150 pesan 1.0 (lo “propio” del 12), mientras 60/90/120 bajan a 0.35 y 180 a 0.20

En la implementación actual, hay que recordar que el código lo tiene definido, pero no se está aplicando al cálculo por defecto, el cálculo **por defecto** usa `USE_HARMONIC_CONJUNCTION=True`

donde `_intensity_prefiltered()` solo aplica: `taper(dh, AOH, power=1.2)` y un `w_override` global si se pasa por parámetro, pero no aplica pesos por ángulo.

Y aun en el modo `legacy` (`USE_HARMONIC_CONJUNCTION=False`), ahora mismo la función `gain_for_pair()` toma el mejor ángulo por proximidad, pero tampoco consulta `_weight_for_angle()`.

`FAMILY_ANGLE_WEIGHTS` está definido (y se usa para construir `FAMILY_SPECS`), pero el cálculo actual no lo usa para afectar el valor final, entonces H4 usa o no “subarmónicos” dependiendo del modo:

- En modo **conjunción armónica** (default): sí, porque la condición, por ejemplo, “conjunción en armónico 4” equivale a que Δ esté cerca de múltiplos de 90° (incluye 90/180/270/0).
- En modo **familias explícitas**: el armónico (ejemplo) H4 usa lo que se define en `family_angles(4)` (donde en la tabla de pesos incluye 90 y 180).

La corrección de pesos sería *un paso extra* para que dentro de, por ejemplo, el H4, el 180 no domine (eso es lo que intenta representar la tabla), y, recordar que este modo está disponible en la implementación, pero actualmente no lo está usando. Existe en el código porque a través del proceso de desarrollo se realizaron muchos intentos de cálculos y se preservaron esas funciones, aunque no se usan.

Otras decisiones menores realizadas en la implementación

Vale la pena declarar explícitamente algunas decisiones que se realizaron durante la implementación:

- La **intensidad** no es “conteo binario” de aspectos, sino un promedio continuo usando `taper()` (campana dentro de orbe).
- La normalización por número de pares (`NORMALIZE_BY_NUM_PAIRS`) hace comparables series con distintos filtros planetarios.
- En progresiones secundarias, el ASC progresado puede incluirse opcionalmente y respeta el modo de casas activo de la técnica principal. Sin embargo, si el rango temporal del armograma supera 20 años, el programa desactiva automáticamente el ASC progresado por razones de rendimiento.
- `FAMILY_ORB` está “ya tabulado” en grados (en vez de recalcularlo dinámicamente por una fórmula única).
- Nota de versión metodológica: La implementación actual de Armon2 mantiene como núcleo del cálculo la conjunción en la carta armónica n , pero permite ajustar el parámetro AOH desde la configuración general, corrige el conteo matemático de la variante Tránsito $\leftrightarrow$ Tránsito para evitar auto-pares y duplicaciones simétricas, y sincroniza el comportamiento del armograma de Progresiones Secundarias con el modo real de casas usado por la técnica principal.

Pseudocódigo formal del cálculo

Para cada fecha t del rango:

1. Obtener longitudes:
 - $\lambda_T(p, t)$ para cada planeta p en el set T
 - $\lambda_N(q)$ para cada planeta q en el set N

2. Para cada armónico $n \in \{1..12\}$:
 - Para cada par (según `pairing`):
 - $\Delta = (\lambda_T - \lambda_N) \bmod 360$
 - Modo armon (por defecto):
 - $d_h = \min((\Delta \bmod 360), 360 - (\Delta \bmod 360))$
 - $g = \max(0, (1 - d_h/AOH)^\alpha)$ con $\alpha \approx 1.2$
 - acumular g (ponderado por pesos planetarios si aplica)
 - normalizar: $I_n(t) = 100 \cdot \frac{1}{K} \sum g$

Resultado: $I_n(t)$ como serie temporal = **armograma** del armónico n .

Apéndice A - Armograma (intensidad armónica)

Discusión final

El armograma dentro de la tradición de la astrología armónica

La astrología armónica, tal como fue formulada por John Addey, introdujo un cambio conceptual decisivo en la comprensión de los aspectos astrológicos, al proponer que las relaciones angulares podían entenderse como expresiones de series armónicas subyacentes. En este marco, la carta armónica **n** no constituye una técnica auxiliar, sino una transformación formal que revela estructuras geométricas específicas, invisibles en la carta radical, pero coherentes desde el punto de vista matemático y simbólico.

El desarrollo posterior de los armogramas y harmogramas, particularmente en el trabajo de Miguel García Ferrández y Tito Maciá, trasladó este enfoque estructural al dominio temporal. Mediante la representación gráfica de la activación de distintas series armónicas a lo largo del tiempo, estos autores propusieron herramientas capaces de identificar períodos de especial resonancia armónica, incorporando en muchos casos sistemas de ponderación interpretativa como las astrodinas, con el fin de jerarquizar la relevancia de los aspectos y planetas implicados.

El sistema presentado en este trabajo se sitúa conscientemente dentro de esta tradición, pero introduce una delimitación metodológica entre detección estructural e interpretación astrológica. Al igual que en la formulación original de Addey, el núcleo del modelo se basa en la evaluación de conjunciones en la carta armónica correspondiente, entendidas como el criterio formal que define la pertenencia de una relación planetaria a una serie armónica determinada. Esta elección preserva la coherencia matemática del enfoque armónico y evita la fragmentación del análisis en listas discretas de aspectos.

A diferencia de otros desarrollos posteriores, la implementación aquí descrita define la intensidad armónica como una magnitud puramente geométrica y cuantitativa, calculada a partir de la densidad y coherencia de conjunciones armónicas entre un conjunto de factores astrológicos. Esta intensidad no pretende medir fuerza interpretativa, significado psicológico ni relevancia simbólica, sino únicamente el grado de alineación de la geometría planetaria con una estructura armónica específica en un momento dado.

En este sentido, el armograma funciona como un detector temporal de patrones armónicos, señalando los instantes en los que una determinada serie se encuentra especialmente activa. La interpretación astrológica

—incluyendo la valoración de aspectos concretos, planetas implicados, casas y significados— queda deliberadamente fuera del cálculo, y se realiza como una etapa posterior, mediante el análisis directo de la carta correspondiente al momento señalado por el armograma.

Esta decisión metodológica implica una diferencia clara respecto a los modelos basados en astrodinas. Mientras que las astrodinas integran ponderaciones interpretativas dentro del propio cálculo, asignando pesos diferenciados a tipos de aspecto y factores planetarios, el presente sistema opta por mantener la neutralidad formal del armograma. Esta elección permite preservar la comparabilidad entre armónicos, facilita el análisis estructural y reduce la dependencia de supuestos interpretativos específicos de una escuela o autor.

No obstante, esta misma neutralidad implica que el armograma, por sí solo, no jerarquiza la importancia relativa de los contactos ni ofrece una interpretación inmediata del significado astrológico de un pico de intensidad. Lejos de constituir una limitación, esta característica define el rol epistemológico del armograma como instrumento de exploración y enfoque, más que como herramienta interpretativa cerrada. En este sentido, el armograma no sustituye la lectura de la carta, sino que la orienta, indicando cuándo y bajo qué estructura armónica resulta más pertinente realizarla.

Desde esta perspectiva, el sistema aquí presentado no compite con otros modelos, sino que se sitúa en un nivel distinto del proceso astrológico. El armograma identifica estructura y ritmo; la interpretación —sea mediante ciclos planetarios, tránsitos, significados puros de influencias planetarias, astrodinas u otros criterios— aporta significado. Todos estos enfoques pueden considerarse complementarios, siempre que se mantenga clara la distinción entre detección geométrica e interpretación simbólica.

En conclusión, la implementación desarrollada en este trabajo representa una continuidad coherente de la tradición armónica inaugurada por Addey y desarrollada en aplicaciones como ARMON, incorporando herramientas computacionales modernas y una definición explícita de la intensidad armónica como magnitud estructural. Su principal aportación reside en la formalización clara del armograma como detector temporal de patrones armónicos, preservando la neutralidad geométrica del cálculo y dejando abierta, de manera consciente, la capa interpretativa como un proceso posterior y contextual.

Armograma implementado

- *¿Cuándo* una estructura armónica está globalmente activa?
- *¿Qué serie armónica* domina un período?
- *¿Cómo evoluciona* esa activación en el tiempo?

A. Ventajas del modelo de Armograma implementado

a. Neutralidad geométrica

Se basa únicamente en relaciones angulares, orbes, agregación matemática. No introduce jerarquías interpretativas previas.

Ventaja: Permite detectar patrones sin sesgo interpretativo.

b. 2. Alta reproducibilidad

Misma entrada, mismo resultado. No depende de: tablas subjetivas, escuelas, criterios variables entre autores.

Ventaja: Apto para análisis comparativos y estudios longitudinales.

c. 3. Lectura estructural del tiempo

Visualiza picos, ciclos, resonancias simultáneas entre armónicos. Ideal para ver ritmos, no eventos aislados.

Ventaja: Excelente para planificación, investigación y detección de ventanas temporales.

d. 4. Separación clara entre detección e interpretación

El cálculo detecta. El astrólogo interpreta después.

Ventaja: Evita confundir estructura con significado.

e. 5. Comparabilidad entre armónicos

Intensidades normalizadas. permite observar coactivaciones (ej. H2 + H6), cambios de dominancia armónica.

Ventaja: Da una visión sistémica del cielo.

B. Limitaciones del modelo de Armograma implementado

a. No jerarquiza automáticamente los factores

No dice qué planeta “manda”, qué aspecto es más importante.

Limitación: Requiere volver a la carta para interpretación detallada.

b. No distingue cualitativamente tipos de aspecto

Un sextil y un trígono contribuyen al mismo armónico.

Limitación: La semántica específica queda fuera del cálculo.

c. Menor inmediatez interpretativa

El número no “significa” por sí solo.

Limitación: Exige mayor formación del intérprete.