

Proyecto Andrómeda - Software de astrología Armon2

Implementación de técnica astrológica de Carta Natal



Por Héctor Jiménez – Mayo 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 el cálculo de una carta natal astrológica.

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.

Trasfondo

La intención de este documento es que sirva de entendimiento sobre el proceso que se lleva a cabo para implementar en código de programación y que los usuarios de Armon2 sepan con mayor exactitud cómo y porqué ha sido implementada.

Este documento describe, con enfoque técnico, la implementación de Cartas Natales en Armon2.

Resumen Del Motor de Cálculo

Armon2 calcula la carta natal como una representación zodiacal tropical del cielo para una fecha, hora y lugar determinados. El procedimiento general es:

- Convierte la fecha/hora local de nacimiento a UTC usando la zona horaria IANA, por ejemplo America/Mexico_City.
- Convierte ese instante UTC a Día Juliano UT.
- Calcula longitudes eclípticas de planetas y puntos.

- Calcula casas, Ascendente, Medio Cielo y cúspides.
- Añade nodos, partes arábicas, estrellas fijas, retrogradación, aspectos y armónico predominante.
- Devuelve dos versiones de los datos: valores crudos en grados y valores formateados por signo zodiacal.

La carta natal se genera con `natal_chart()`, que delega en `chart_for_date()`.

Librerías Usadas

La librería de cálculos central es **Swiss Ephemeris**, vía `pyswisseph`. Se usa para hacer cálculos de:

- posiciones planetarias tropicales aparentes;
- velocidad eclíptica y retrogradación;
- casas astrológicas;
- nodos lunares;
- estrellas fijas;
- eclipses;
- retornos solares;
- Luna fuera de curso;
- cálculos de tiempo sideral, oblicuidad y Día Juliano.

En el código:

```
import swisseph as swe

FLAGS = swe.FLG_SWIEPH

swe.set_ephe_path(str(base_path() / "sweph"))
```

El programa incluye archivos locales de efemérides `sepl_18.se1`, `semo_18.se1` y `sefstars.txt`.

También se usa la librería **Skyfield** con el archivo JPL: `de440s.bsp`

Skyfield aparece como motor auxiliar para algunas técnicas, progresiones, lunaciones y cálculos con efemérides JPL. Pero en la carta natal principal, el código actual prioriza Swiss Ephemeris para las posiciones planetarias porque busca mayor compatibilidad con programas como Astro.com o Solar Fire.

Conversión De Fecha

El usuario introduce una fecha local. El programa resuelve la zona horaria y la convierte a UTC:

```
dt_local -> dt_utc
```

Luego calcula el Día Juliano:

```
h = hora + minuto / 60 + segundo / 3600

jd_ut = swe.julday(año, mes, día, h)
```

Ese `jd_ut` es la base de casi todos los cálculos astronómicos.

Posiciones Planetarias

Para cada planeta se llama a Swiss Ephemeris:

```
res = swe.calc_ut(jd_ut, planeta, FLAGS | swe.FLG_SPEED)
```

La salida incluye, entre otros valores:

- λ = longitud eclíptica
- β = latitud eclíptica
- r = distancia
- $v\lambda$ = velocidad en longitud

El programa toma principalmente:

```
lon = xx[0] % 360
```

```
speed = xx[3]
```

La retrogradación se determina con:

```
retrogrado = speed < 0
```

Es decir, si la velocidad longitudinal aparente es negativa, el planeta se marca como retrógrado.

Los cuerpos principales calculados son: Sol, Luna, Mercurio, Venus, Marte, Júpiter, Saturno, Urano, Neptuno, Plutón. Además añade:

```
Nodo Norte = swe.MEAN_NODE
```

Nodo Sur = Nodo Norte + 180°

Zodiaco

El programa trabaja en longitud eclíptica tropical de 0° a 360°. Luego traduce esa longitud a signo:

```
signo = int(longitud // 30)
```

```
grado_en_signo = longitud % 30
```

Por ejemplo:

218° = Escorpio 8°

porque Escorpio empieza en 210°.

Casas

El sistema de casas es configurable. Aparecen estos sistemas:

- P = Placidus
- T = Topocéntrico
- R = Regiomontanus
- E = Casas iguales

La interfaz permite elegirlos desde el menú de configuración. El valor se guarda y se aplica al motor con:

```
ae.set_house_system(code)
```

El cálculo base usa Swiss Ephemeris:

```
cusps, ascmc = swe.houses(jd_ut, lat, lon, hsys)
```

De ahí salen:

cusps = cúspides de casas

ascmc[0] = Ascendente

ascmc[1] = Medio Cielo

ascmc[2] = ARMC, cuando está disponible

La fórmula conceptual detrás del ARMC es:

$$ARMC = Tiempo Sideral Local \times 15^\circ$$

donde:

$$Tiempo Sideral Local = Tiempo Sideral Greenwich + longitud\ geográfica / 15$$

En grados:

$$ARMC = (sidtime(JD) \times 15 + longitud_geográfica) \bmod 360$$

Para el sistema topocéntrico, el programa incluye un ajuste propio estilo Marr/Polich/Page. Usa polos derivados de la latitud:

$$p_1 = atan(tan \varphi / 3)$$

$$p_2 = atan(2 \tan \varphi / 3)$$

y calcula cúspides intermedias con una fórmula del tipo:

$$x = (\sin \varepsilon \cdot \tan p - \cos \varepsilon \cdot \cos OA) / \sin OA$$

$$\lambda = atan(x) + cuadrante$$

donde:

- φ = latitud geográfica
- ε = oblicuidad de la eclíptica
- OA = ascensión oblicua usada para la cúspide
- λ = longitud eclíptica resultante

Ascendente Y Medio Cielo

Aunque la mayoría de casas se delega a Swiss Ephemeris, el código contiene fórmulas auxiliares. Para el Medio Cielo se usa la relación entre tiempo sideral local y oblicuidad:

$$\tan(MC) = \tan(LST) / \cos(\varepsilon)$$

con corrección de cuadrante.

Para el Ascendente aparece una forma equivalente:

$$Asc = atan2(-\cos \theta, \sin \theta * \cos \varepsilon - \tan \varphi * \sin \varepsilon)$$

donde:

- θ = tiempo sideral local en grados
- ϕ = latitud geográfica
- ε = oblicuidad

Partes Arábigas

El programa calcula **Parte de la Fortuna y Parte del Infortunio**. Primero determina si la carta es diurna o nocturna usando la posición del Sol respecto al eje Ascendente/Descendente.

Parte de la Fortuna:

Carta diurna:

$$Fortuna = Asc + Luna - Sol$$

Carta nocturna:

$$Fortuna = Asc + Sol - Luna$$

Parte del Infortunio:

Carta diurna:

$$Infortunio = Asc + Marte - Saturno$$

Carta nocturna:

$$Infortunio = Asc + Saturno - Marte$$

Todo se normaliza a 0°-360°:

`valor = valor mod 360`

Aspectos

Los aspectos se calculan por distancia angular mínima entre dos longitudes zodiacales. Primero:

$$\delta = |\lambda_1 - \lambda_2| \bmod 360$$

Luego:

$$distancia = \delta \text{ si } \delta \leq 180$$

$$distancia = 360 - \delta \text{ si } \delta > 180$$

Después se compara con una tabla de aspectos y orbes. El programa incluye:

- conjunción 0° orbe 5°
- semicuadratura 45° orbe 3°

- sesquicuadratura 135° orbe 3°
- novil 40° orbe 2°
- septil 360/7° orbe 2°
- quincuncio 150° orbe 2°
- sextil 60° orbe 5°
- cuadratura 90° orbe 8°
- trígono 120° orbe 8°
- oposición 180° orbe 8°

La condición es:

```
|distancia - ángulo_del_aspecto| <= orbe
```

El orbe que se guarda es:

```
orbe = distancia - ángulo_exacto
```

Armónico Predominante

El programa calcula armónicos del 1 al 12 con un método vectorial. Para cada armónico n, multiplica las longitudes por n y suma componentes seno/coseno:

$$X_n = \sum \cos(n \cdot \lambda_i)$$

$$Y_n = \sum \sin(n \cdot \lambda_i)$$

$$R_n = \sqrt{X_n^2 + Y_n^2}$$

El armónico predominante es el n con mayor R_n .

Esto mide concentración angular: si varios planetas se agrupan según una misma familia armónica, el vector resultante crece.

Estrellas Fijas

Las estrellas fijas se calculan con Swiss Ephemeris:

```
swe.fixstar2_ut(nombre, jd_ut, FLAGS)
```

El programa trae un catálogo mínimo de estrellas behenianas, como Algol, Aldebaran, Sirius, Regulus, Spica, Antares, Vega, etc. Se guarda su longitud eclíptica zodiacal, no su constelación visual.

En Síntesis

El motor de Armon2 no “dibuja” la carta a partir de fórmulas simplificadas de órbitas, sino que consulta efemérides astronómicas de alta precisión, principalmente **Swiss Ephemeris**. A partir de esas posiciones astronómicas calcula estructuras astrológicas: zodiaco tropical, casas, ángulos, aspectos, partes arábicas, estrellas fijas y armónicos.

La parte matemática propia del programa aparece sobre todo en:

- normalización angular;
- distancia mínima entre puntos;
- detección de aspectos por orbe;

- cálculo de partes;
- armónicos vectoriales;
- ajustes topocéntricos;
- búsquedas por bisección en retornos, ciclos y fases.

Así, el núcleo combina astronomía computacional delegada a efemérides profesionales con reglas astrológicas tradicionales implementadas explícitamente en lenguaje de programación Python en su versión 3.13.