

Proyecto Andrómeda - Software de astrología Armon2

Implementación de técnica astrológica de Progresiones Secundarias



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 para la técnica de Progresiones Secundarias usadas en astrología.

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 Progresiones Secundarias en Armon2.

Marco conceptual general de implementación

Las progresiones secundarias surgen de una intuición antigua: los primeros días tras el nacimiento “codifican” un patrón de desarrollo que se despliega a lo largo de la vida. El método se volvió canónico en la astrología moderna por su consistencia empírica interpretativa:

- el **Sol progresado** marca maduración,
- la **Luna progresada** describe ciclos emocionales,

- y los cambios de **ángulos/casas** describen reorientaciones vitales.

El principio 1 día = 1 año aparece sistematizado como técnica estándar en recursos modernos (calculadoras y manuales contemporáneos lo describen así). Las progresiones secundarias entonces son una técnica de “tiempo simbólico” donde se asume la equivalencia:

1 día después del nacimiento = 1 año de vida

La idea (en su formulación moderna) es que la carta natal se “desarrolla” internamente a lo largo del tiempo: la dinámica del desarrollo psicológico/biográfico se modela sustituyendo el tiempo cronológico por un **tiempo simbólico**. En términos astronómicos, esto equivale a evaluar las efemérides planetarias **pocos días** después del nacimiento para representar **décadas** de vida. En la implementación, esta equivalencia se concreta como:

- **Planetas progresados:** se toman desde una fecha simbólica (natal + edad_en_años días).
- **Casas progresadas (Placidus):** se calculan usando un enfoque “estilo Astro.com / Solar Fire” que es progresar el ARMC (Ascensión Recta del Medio Cielo) con clave Naibod, y desde ese ARMC obtener las cúspides Placidus.

El problema astronómico: cómo progresar los ángulos y las casas? Con planetas, el “día=año” es directo (efemérides). Pero para las casas hay un problema: las casas dependen de la rotación terrestre (tiempo sideral y lugar). Si se aplica “ingenuamente” la regla de 1 día=1 año a la rotación diaria, el MC avanzaría ~361° por año simbólico (porque un día solar no equivale a un día sideral). Por eso existen varios métodos para progresar ejes/casas, y servicios como Astro.com los distinguen explícitamente (p. ej. Naibod vs “ARMC 361°/prog day”).

En ese contexto, para esta implementación de Armon2, la decisión de progresar por ARMC (en RA) con una clave estándar es una solución profesional: separa el símbolo (planetas) de la “cinemática” terrestre (casas).

El ARMC, LST y oblicuidad son la geometría mínima para casas, las casas (especialmente Placidus) dependen de latitud ϕ , oblicuidad ε y el meridiano local expresado en el ecuador celeste (ARMC).

Swiss Ephemeris formaliza esto: si se tiene ARMC, latitud y oblicuidad, se pueden derivar casas (y posiciones en casa) sin necesidad de calcular otra vez todo desde fecha/hora. Esto explica por qué el enfoque tomando en este código es robusto: se calcula ARMC natal (vía sidtime + lon), éste se progresa, se obtiene así ε verdadera (ECL_NUT) y se hace uso de `houses_armc()`.

Por último, el uso de “aparente” en *Skyfield* tiene sentido en astrología moderna, *Skyfield* distingue entre posición astrométrica y aparente; la aparente aplica aberración y deflexión gravitacional, y por tanto corresponde a un “punto del cielo” físicamente distinto. En astrología práctica, la mayoría de efemérides modernas y programas comerciales trabajan con posiciones “aparentes geocéntricas”, de modo que la decisión tomada en esta implementación es coherente con el estándar contemporáneo y facilita la comparabilidad.

Flujo en la implementación

UI (armon2.py) + módulo (secondary_progressions.py)

Se creó un módulo **secondary_progressions.py** que hace de “pegamento” entre la interfaz gráfica y el motor de cálculo:

Desde la interfaz se toma

- **Fecha/hora natal** (QDate + QTime).
- **Fecha objetivo** de progresión (QDate), usando **la misma hora natal**.
- **Latitud/longitud** de los spinboxes.
- **Localidad** (texto).
- **TZ** (por helpers existentes en MainWindow: `_tz_field_value()` y `_tz_to_zoneinfo()`).

con lo que se construyen cadenas de caracteres en formato ISO para las fechas

- `birth_iso = "YYYY-MM-DDTHH:MM:SS" (naive local)`
- `target_iso = "YYYY-MM-DDTHH:MM:SS" (naive local, misma hora natal)`

y con eso se llama al motor `astro_engine.secondary progressed(birth_iso, lat, lon, tz_str, target_iso)`

Definición matemática de “día = año”

En `astro_engine.secondary progressed()` el código define la **conversión a UTC y cálculo de edad (en años)**. Se interpreta `birth` en zona horaria `tz` (si se define `naive`, se asume “hora local”), se hace lo mismo con `target_date` y ambas se convierten a **UTC**. La edad se calcula como fracción real:

$$\text{years} = \frac{\Delta t_{\text{UTC}}}{\text{SECONDARY_YEAR} \cdot 86400}$$

donde en el código

$$\text{SECONDARY_YEAR} = 365.2425$$

esto es cercano al año trópico/promedio usado en varios contextos (omo sugerencia se podría experimentar con 365.25 si se quieren comparar variantes).

Fecha simbólica (núcleo de la progresión secundaria)

Una vez obtenida la edad en años (`years`), se construye la **fecha simbólica**:

$$\text{symbolic_date} = \text{dob_utc} + \text{years días}$$

es decir: si la persona tiene 26.13 años, se calculan posiciones planetarias alrededor de ~26.13 días después del nacimiento.

Planetas progresados: efemérides y criterio astronómico usado

Fuente astronómica

El código usa la librería de *Skyfield* (con un conjunto de efemérides cargado en `PLANETS`) para posiciones geocéntricas. La longitud eclíptica se calcula en esas funciones con lo siguiente:

- Observador: `EARTH` (si `loc is None`, que es el caso por defecto para progresiones secundarias de planetas).

- Se usa posición **aparente** (`.apparent()`), lo que incorpora aberración.
- Se transforma al marco eclíptico (`frame_latlon(ecliptic_frame)`).
- Se toma `lon.degrees % 360`.

En resumen, la longitud eclíptica progresada es:

$$\lambda_p = \text{mod}(\lambda_{\text{aparente, geocéntrica}}(\text{symbolic_date}), 360^\circ)$$

Planetas incluidos en la carta progresada

La lista es: Sol, Luna, Mercurio, Venus, Mars Barycenter, Jupiter Barycenter, Saturn Barycenter, Uranus Barycenter, Neptune Barycenter, Pluto Barycenter. Esto es una decisión técnica común en efemérides modernas: usar barycenter para algunos cuerpos cuando la librería lo expone así.

Retrogradación (criterio computacional)

El código marca un planeta como retrógrado si estima su “velocidad” longitudinal alrededor del instante: Se calcula

$$\lambda(t - 6h) \text{ y } \lambda(t + 6h),$$

y se obtiene el delta mínimo en $[-180^\circ, 180^\circ]$, luego se divide por días del paso para aproximar grados/día. Se considera retrógrado si `speed < 0`. Formalmente (aprox.):

$$v \approx \frac{\Delta\lambda}{\Delta t}, \quad \text{retrógrado si } v < 0$$

Casas progresadas: “Naibod en ARMC” + Placidus desde ARMC

Esta parte es la más estilo “Astro.com/Solar Fire” de la implementación. Se construye el **Julian Day** natal en UT (UTC) con Swiss Ephemeris:

$$JD_{\text{birth,UT}} = \text{swe.julday}(Y, M, D, h_{\text{UT}})$$

donde $h_{\text{UT}} = \text{hour} + \text{minute}/60 + \text{second}/3600$. Luego, el Tiempo sideral natal y ARMC natal. Tiempo sideral en Greenwich (en horas):

$$ST_G = \text{swe.sidtime}(JD_{\text{birth,UT}})$$

El tiempo sideral local (LST) en horas (con `lon/15`; convención: `lon>0` Este, `lon<0` Oeste):

$$LST = (ST_G + \frac{\text{lon}}{15}) \text{mod} 24$$

Y el ARMC natal (en grados):

$$\boxed{ARMC_{\text{nat}} = (LST \cdot 15) \text{mod} 360}$$

Progresión del ARMC con clave Naibod

La constante que el código utiliza es:

$$k_{\text{Naibod}} = 0.98564733^\circ/\text{año}$$

por lo tanto

$$ARMC_{\text{prog}} = (ARMC_{\text{nat}} + \text{years} \cdot k_{\text{Naibod}}) \bmod 360$$

Esto separa el **tiempo simbólico** para planetas (día=año) del **avance sideral** de las casas mediante un arco anual estándar (Naibod).

Oblicuidad verdadera

Para coherencia con *Swiss Ephemeris* (y su modelado de nutación), se calcula la oblicuidad verdadera en la fecha simbólicam se calcula $JD_{\text{prog,UT}}$ de `symbolic_date`, `swe.calc_ut(jd_prog_ut, swe.ECL_NUT)` y se toma `eps = xx[0]`. Es decir:

$$\varepsilon = \varepsilon_{\text{true}}(JD_{\text{prog,UT}})$$

Casas Placidus desde ARMC

Finalmente:

$$(\text{cusps}, \text{ascmc}) = \text{swe.houses_armc}(ARMC_{\text{prog}}, \phi, \varepsilon, \text{hsys})$$

donde ϕ = latitud y `hsys` en viene de `normalize_house_system()`; por defecto termina siendo "P" (Placidus) si no se especifica otra cosa. El resultado que se regresa incluye: `houses_raw: {cusps[12], asc, mc}` y versión formateada de `houses` con `fmt()` para mostrarlo en gráfico

Aspectos y armónico predominante sobre la carta progresada

Aunque estrictamente no “define” la técnica, el resultado de progresiones secundarias incluye análisis posterior: `calcular_aspectos(planets_raw)` sobre posiciones progresadas y `calcular_armonico_predominante(planets_raw, soloPersonales=True)` como métrica de énfasis armónico. Esto convierte la carta progresada en un objeto listo para interpretación dentro del ecosistema Armon2 (armónicos/aspectos igual que en natal o tránsitos).

Ejemplo numérico

Suponemos que hay un nacimiento: 2000-01-01 12:00:00 hora local Monterrey (America/Monterrey) y que la fecha objetivo a calcular es: 2026-02-16 12:00:00 hora local Monterrey, el código (usando 365.2425) obtiene aproximadamente

$$\text{years} \approx 26.1278466$$

y la fecha simbólica:

$$\text{symbolic_date} \approx 2000-01-27 21:04:06 \text{ UTC}$$

Interpretación: las posiciones planetarias progresadas para 2026 se leen como “las posiciones del cielo” alrededor del **día 27–28 después del nacimiento**, pero proyectadas simbólicamente a los **26 años**. Y el arco Naibod aplicado al ARMC sería:

$$\Delta ARMC \approx 26.1278466 \times 0.98564733 \approx 25.752842^\circ$$

Con eso se calcula $ARMC_{prog}$, se obtiene ε verdadera en la fecha simbólica, y se generan las cúspides Placidus progresadas desde `swe.houses_armc()`.

Pseudocódigo

PROGRESIONES_SECUNDARIAS(main):

```
# --- UI / MÓDULO (secondary_progressions.py) ---

qdate_natal = main.de_fecha.date()

qtime_natal = main.te_hora.time()

qdate_target = main.de_ps_fecha.date()

lat = main.lat_spin.value()

lon = main.lon_spin.value()

loc_text = trim(main.ed_loc.text())

birth_iso = build_iso_datetime(qdate_natal, qtime_natal) # "YYYY-MM-DDTHH:MM:SS"

target_iso = build_iso_datetime(qdate_target, qtime_natal) # misma hora natal

tz_val = main._tz_field_value(main.ed_tz)

tz_str = main._tz_to_zoneinfo(tz_val, loc_text) or "Etc/GMT"

return astro_engine.secondary_progressed(birth_iso, lat, lon, tz_str, target_iso)
```

SECONDARY_PROGRESSED(birth_iso, lat, lon, tz_str, target_iso):

```
tzinfo = ZoneInfo(tz_str)

# 1) Interpretar birth y target como hora local tz_str y convertir a UTC

dob_local = parse_iso(birth_iso) as tzinfo

dob_utc = dob_local -> UTC

tgt_local = parse_iso(target_iso) as tzinfo (si None: now(tzinfo))

tgt_utc = tgt_local -> UTC

# 2) Edad en años (año medio = 365.2425 días)

SECONDARY_YEAR = 365.2425

years = seconds(tgt_utc - dob_utc) / (SECONDARY_YEAR * 86400)

# 3) Fecha simbólica: dob_utc + years días

symbolic_date_utc = dob_utc + timedelta(days=years)

# 4) Skyfield time para esa fecha simbólica

t_sym = TS.utc(symbolic_date_utc)

# 5) Planetas progresados (longitud eclíptica geocéntrica aparente)

bodies = [sun, moon, mercury, venus, Mars Barycenter, ... Pluto Barycenter]

planets_raw[body] = planet_long(body, t_sym, loc=None) % 360

planets_rx = compute_planets_retrograde(bodies, t_sym, loc=None)
```

```

planets_fmt          = fmt(planets_raw)

# 6) Casas progresadas (Naibod sobre ARMC natal + Placidus desde houses_armc)
jd_birth_ut = swe.julday(dob_utc Y,M,D, UT_hours)

sid_g_nat  = swe.sidtime(jd_birth_ut)          # horas

LST_nat_h  = (sid_g_nat + lon/15) % 24

ARMC_nat   = (LST_nat_h * 15) % 360            # grados

arco_naibod = years * 0.98564733              # grados

ARMC_prog  = (ARMC_nat + arco_naibod) % 360

jd_prog_ut = swe.julday(symbolic_date_utc Y,M,D, UT_hours)

eps_true   = swe.calc_ut(jd_prog_ut, ECL_NUT)[0] # oblicuidad verdadera

(cusps, ascmc) = swe.houses_armc(ARMC_prog, lat, eps_true, house_system_bytes)

houses_raw = {cusps[1..12], asc=ascmc[0], mc=ascmc[1]}

houses_fmt = fmt(houses_raw)

# 7) Post-proceso interpretativo

aspects = calcular_aspectos(planets_raw)

arm_pred = calcular_armonico_predominante(planets_raw, soloPersonales=True)

# 8) Salida unificada

return {

    utc: symbolic_date_utc,

    date_local: symbolic_date_utc in tzinfo,

    planets_raw, planets_rx, planets_fmt,

    houses_raw, houses_fmt,

    aspects, armonico,

    edad: years,

    target_date: tgt_utc

}

```

Glosario Técnico de Progresiones Secundarias

1.1 Progresión Secundaria

Técnica simbólica basada en la equivalencia:

1 día después del nacimiento $\equiv$ 1 año de vida

Matemáticamente, la fecha simbólica se define como:

$$\text{symbolic_date} = \text{fecha_natal} + \text{edad_en_años días}$$

En Armon2:

- La edad se calcula con un año medio de **365.2425 días**
- Las posiciones planetarias se calculan en la fecha simbólica
- Las casas se progresan con Naibod sobre el ARMC natal

1.2 Julian Day (JD)

Número continuo de días desde el 1 de enero del 4713 a.C. (escala astronómica). Se usa para:

- Calcular tiempo sideral
- Obtener oblicuidad verdadera
- Evaluar casas con Swiss Ephemeris

En el código se usa:

$$JD = \text{swe.julday}(Y, M, D, UT)$$

1.3 Tiempo Sideral (Sidereal Time)

El tiempo sideral mide la rotación de la Tierra respecto a las estrellas fijas, no respecto al Sol. El Tiempo Sideral de Greenwich (GST)

$$ST_G = \text{swe.sidtime}(JD)$$

el resultado es en horas.

1.4 LST — Local Sidereal Time

Tiempo sideral local.

$$LST = (ST_G + \frac{\lambda}{15}) \bmod 24$$

donde

λ = longitud geográfica en grados y $15^\circ = 1$ hora sideral

1.5 ARMC — Ascensión Recta del Medio Cielo

ARMC = Right Ascension of the Midheaven. Es el punto clave para calcular casas desde `houses_armc`. En `Armon2`:

$$ARMC = (LST \times 15) \bmod 360$$

esto representa el meridiano local proyectado en el ecuador celeste.

1.6 Oblicuidad de la Eclíptica (ϵ)

Ángulo entre el ecuador celeste y la eclíptica. Actualmente $\approx 23^\circ 26'$, pero varía lentamente en el tiempo. En `Armon2` se usa la **oblicuidad verdadera**:

$$\epsilon = \text{swe.calc_ut}(JD, ECL_NUT)$$

Es necesaria para transformar coordenadas ecuatoriales a eclípticas y para calcular casas Placidus correctamente.

1.7 Clave Naibod

Constante usada en direcciones simbólicas:

$$k_{Naibod} = 0.98564733^\circ/\text{año}$$

Equivale aproximadamente al movimiento medio diario del Sol. Para progresiones secundarias, el código la usa para progresar el ARMC:

$$ARMC_{prog} = ARMC_{nat} + (\text{edad} \times k_{Naibod})$$

1.8 Longitud Eclíptica Aparente

Coordenada angular del planeta sobre la eclíptica corregida por aberración. En Skyfield: `apparent().frame_latlon(ecliptic_frame)`. Se obtiene:

$$\lambda = \text{lon.degrees} \bmod 360$$

Notas de Precisión Astronómica

¿Por qué 365.2425 días?

El código usa:

$$SECONDARY_YEAR = 365.2425$$

Este valor corresponde al año medio gregoriano y se aproxima al año trópico real (~365.2422 días).

Las decisiones de usar este valor son varias: Coincide con la base del calendario civil moderno, minimiza acumulación de error en décadas, es estándar en otros programas de astrología, mantiene coherencia con Solar Fire / Astro.com, diferencia acumulada en 50 años respecto a 365.25:

$$(365.25 - 365.2425) \times 50 = 0.375 \text{ días}$$

Lo cual equivale a menos de medio día simbólico.

¿Por qué usar `.apparent()` en Skyfield?

Skyfield permite varias posiciones:

- `.at()` → geométrica
- `.observe()` → astrométrica
- `.apparent()` → corregida por aberración

Armon2 usa posición aparente geocéntrica. La “justificación” de esto es: es el estándar en astrología moderna, coincide con *Swiss Ephemeris* en modo aparente, integra aberración de la luz, produce coherencia con software profesional. No se usa posición topocéntrica en progresiones secundarias (decisión deliberada).

Uso de Oblicuidad Verdadera

Se obtiene mediante:

$$\varepsilon = \text{swe.calc_ut}(JD, ECL_NUT)$$

lo cual incluye: Nutación y correcciones modernas. Esto mejora la precisión del cálculo Placidus desde ARMC progresado.

Estimación de Retrogradación

Se calcula derivada aproximada:

$$v \approx \frac{\lambda(t + 6h) - \lambda(t - 6h)}{0.5 \text{ día}}$$

Si $v < 0 \Rightarrow$ retrógrado. Método numérico estable y suficiente para progresiones.

Comparación técnica contra Solar Fire

En esta implementación los cálculos de progresiones secundarias combina dos “relojes”:

- Planetas progresados: “día = año”, evaluando efemérides en una fecha simbólica (natal + edad_en_años días).
- Casas progresadas (Placidus): progresar el ARMC y derivar cúspides desde `houses_armc()` (Placidus) usando la **oblicuidad verdadera**.

Este enfoque coincide con el estándar más usado en programas profesionales: progresar el eje/ángulos por ARMC (RA) y mantener el criterio “día=año” para planetas.

Solar Fire ofrece varias opciones de progresión de ángulos/casas (p. ej. Naibod/RA, Solar Arc/RA, etc.). En la práctica, para obtener resultados equivalentes a lo que el código de Armon2 implementa, lo clave es que en Solar Fire se seleccione una opción equivalente a:

- Secundarias para planetas: 1 día = 1 año
- Ángulos/casas por RA/ARMC (Naibod en RA o equivalente)

Esto no es exclusivo de Solar Fire; Astro.com expone explícitamente opciones del tipo “ARMC 1 Naibod / prog. day” y otras variantes para progresiones de ejes/casas, lo cual refleja que el “problema” está precisamente en cómo progresar el ARMC y derivar casas.

Si al comparar esta implementación con Solar Fire y “no da igual”, casi siempre proviene de alguno de estos puntos:

1. Método de ángulos/casas

- Armon2: ARMC progresado con una clave (Naibod) y casas desde `houses_armc()`.
- En Solar Fire se podría estar usando otra variante (p. ej. “MC from solar arc”, o algún ajuste distinto). Astro.com muestra que existen varias.

2. Planetas: aparente vs astrométrico

- Armon2: usa `.apparent()` en Skyfield (incluye aberración y deflexión gravitacional), que es un “punto del cielo” físicamente distinto del astrométrico.

- Si Solar Fire estuviera configurado distinto (o la comparación usa datos exportados con otro modo), se ven pequeñas diferencias.

3. Tiempos y escalas

- Armon2: convierte a UTC y usa fracción de año con 365.2425.
- Si Solar Fire usa otra convención (p. ej. 365.25 o un redondeo de edad), puede mover la fecha simbólica minutos/horas y eso se nota especialmente en la Luna y en cúspides.

4. Oblicuidad/nutación

- Armon2 calcula oblicuidad verdadera con ECL_NUT para alimentar houses_armc().
- Diferencias en “true” vs “mean” obliquity también pueden mover cúspides (a veces poco, a veces suficiente para cambiar casa cerca de cúspide).

Análisis de sensibilidad: qué cambia si se usa 365.25 para cálculo de edad:

El código define la edad como

$$\text{years} = \frac{\Delta t}{365.2425 \cdot 86400}$$

y la fecha simbólica como:

$$\text{symbolic_date} = DOB_{UTC} + \text{years días}$$

Si se cambia el valor de **365.2425** a **365.25**, cambia ligeramente la fracción de años y por tanto la **fecha simbólica**, y eso impacta planetas (sobre todo Luna), oblicuidad (mínimo) y casas (vía ARMC progresado y oblicuidad, normalmente leve pero puede ser relevante cerca de cúspides). Cuánto se mueve la fecha simbólica? La discrepancia acumulada por N años:

$$\Delta D \approx N \cdot (365.25 - 365.2425) = 0.0075N \text{ días}$$

Ejemplos:

- 10 años $\rightarrow 0.075 \text{ días} \approx 1 \text{ h } 48 \text{ min}$
- 30 años $\rightarrow 0.225 \text{ días} \approx 5 \text{ h } 24 \text{ min}$
- 50 años $\rightarrow 0.375 \text{ días} \approx 9 \text{ h}$

Ese desplazamiento se traduce en movimiento planetario según la velocidad del cuerpo:

- Luna ($\sim 13^\circ/\text{día}$); a 30 años ($\approx 5.4\text{h}$): $13^\circ/\text{día} \cdot 0.225 \approx 2.9^\circ$
- Esto es enorme interpretativamente (aspectos, cambios de casa si cae cerca de cúspide).
- Sol ($\sim 1^\circ/\text{día}$): $\sim 0.225^\circ$ ($13'30''$ aprox.)
- Planetas lentos: casi despreciable.

Impacto en casas (vía ARMC + Naibod)

Como el ARMC se progresa con:

$$ARMC_{prog} = ARMC_{nat} + (years \cdot 0.98564733^\circ)$$

Si *years* cambia un poco, *ARMC_prog* cambia proporcionalmente. A 30 años la diferencia en *years* ≈ 0.0006 (orden de magnitud) y la diferencia en *ARMC_prog* $\approx 0.0006 \times 0.9856 \approx \mathbf{0.0006^\circ}$ (muy pequeña).

Como conclusión, podemos decir que el gran impacto del 365.25 vs 365.2425 suele verse en planetas, sobre todo la Luna (y puntos rápidos). El impacto en el ARMC por Naibod suele ser pequeño; lo que sí puede moverse más en casas es si se recalculan los ejes por otro método o si el programa de comparación usa una convención distinta (por ejemplo “ARMC 361°/prog day” en vez de Naibod).

Decisiones de diseño generales

Separación Planetas / Casas

Armon2 adopta un enfoque híbrido de Planetas → Día = Año y Casas → Naibod sobre ARMC, porque reproducen el comportamiento de otros programas como Astro.com, Solar Fire y Morinus (configuración estándar). Si se usara “día = año” también para casas (es decir, recalcular casas desde fecha simbólica completa), se produciría un desplazamiento de casas diferente al estándar “profesional”.

¿Por qué progresar el ARMC y no recalcular LST simbólico?

La progresión secundaria tradicional considera que el desarrollo interno (planetas) es simbólico, pero la rotación terrestre (casas) debe progresarse por arco equivalente anual. Naibod representa:

movimiento medio solar diario

al aplicarlo al ARMC produce continuidad angular, evolución suave y una consistencia técnica con direcciones simbólicas.

Uso de `houses_armc()`

En lugar de recalcular el JD simbólico, Sidereal time simbólico y LST simbólico, el código calcula ARMC natal, le suma arco Naibod, genera casas con `swe.houses_armc`. Esto trae algunas ventajas:

- Control directo del ARMC
- Mayor coherencia matemática
- Evita acumulación de error de redondeo
- Permite reproducibilidad exacta

Por qué no topocéntrico en progresiones

Se decidió mantener la posición geocéntrica aparente porque las progresiones secundarias son simbólicas, usar Topocéntrico introduce variación mínima en este contexto. Otros software de astrología tampoco usan topocéntrico en progresadas estándar.

Consistencia interna del sistema Armon2

Las progresiones devuelven: `planets_raw`, `houses_raw`, `aspects` y `armonico` predominante. Esto permite que el módulo progresado pueda ser compatible con armónicos, se integre con análisis natal y tránsitos, y mantenga la arquitectura unificada

Conclusión Técnica

El sistema de progresiones secundarias en la implementación está diseñado bajo tres principios:

1. Precisión astronómica moderna (Skyfield + Swiss Ephemeris)
2. Coherencia con software profesional
3. Separación conceptual entre evolución interna (planetas) y evolución angular (casas)

El resultado es un modelo matemáticamente consistente, astronómicamente preciso y técnicamente reproducible.