

Proyecto Andrómeda - Software de astrología Armon2

Implementación de técnica astrológica de Direcciones Primarias



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 Direcciones Primarias 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 actual de Direcciones Primarias en Armon2. Se documentan las dos variantes expuestas en la interfaz: Zodiacales y AR (Carta Ascensional), las dos claves disponibles para el arco (Ptolomeo y Naibod), y la manera en que el motor usa el sistema de casas actualmente seleccionado en la carta base. También se aclaran las decisiones prácticas observables en el código y algunas etiquetas históricas que permanecen por compatibilidad interna.

Fuentes de implementación:

- Función de entrada desde interfaz: `tecnica_direcciones_primarias(window)` en `armon2.py`; motor astronómico principal: `astro_engine.py`, función `primary_directions_placidus_zodiacal`.

- Funciones de apoyo y dependencias astronómicas: Swiss Ephemeris (swisseph), principalmente `houses()`, `houses_armc()`, `house_pos()` y `calc_ut()` para oblicuidad verdadera.

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 las técnicas de Direcciones Primarias.
- *ARMC/RAMC* = Ascensión Recta del Medio Cielo

Resumen conceptual

En la implementación actual se distinguen dos variantes explícitas de Direcciones Primarias:

- **Direcciones Primarias Zodiacales** - Se trabaja sobre la eclíptica y se conserva la posición de casa natal del significador dentro del sistema de casas activo en la carta base. El punto dirigido final se resuelve nuevamente sobre la eclíptica mediante inversión numérica, de modo que reproduzca bajo el ARMC dirigido la misma house position natal.
- **Direcciones Primarias Ascensionales en Ascensión Recta (AR)** - Se trabaja en coordenadas ecuatoriales. Las casas se expresan como sectores uniformes de 30° de Ascensión Recta contados desde el ARMC dirigido, y cada planeta se dirige en AR real a partir de su posición natal, incluyendo la latitud eclíptica propia del cuerpo.

Interfaz Gráfica

La interfaz gráfica construye las entradas en `tecnica_direcciones_primarias(window)`:

- `birth_iso`: fecha/hora natal (QDate + QTime)
- `target_iso`: fecha objetivo con la hora natal "casteada" (imitando la práctica común como en Solar Fire)
- `converse`: checkbox "converso"
- `use_naibod`: checkbox "Clave Naibod" (si no, Ptolomeo)
- `mode`: radio-button que selecciona una de las dos variantes expuestas en la interfaz:
 - Zodiacales
 - AR (Carta Ascensional)

La interfaz construye la llamada al motor de Direcciones Primarias con los parámetros natales, fecha objetivo, localidad, clave direccional, sistema de casas activo y sentido directo/converso. En la variante zodiacal, además, existe un checkbox específico para decidir si el promisor conserva su latitud natal en el speculum. La implementación interna devuelve una carta compatible con el formato general de Armon2, acompañada de metadatos específicos sobre arco dirigido, clave empleada, edad simbólica y sistema de coordenadas usado.

Definiciones astronómicas y notación

Notación usada en este documento (coincide con la usada en el código):

- λ (λ): longitud eclíptica (0–360°)
- β (β): latitud eclíptica
- ϵ (ϵ): oblicuidad verdadera de la eclíptica.
- RA (α): Ascensión Recta (0–360°)
- Dec (δ): declinación.
- ARMC/RAMC: Ascensión Recta del Medio Cielo; en *Swiss Ephemeris* suele aparecer como `ascmc[2]`.
- `house_pos()`: función de *Swiss Ephemeris* que devuelve la 'posición de casa' (`house position`) de un punto para un sistema de casas dado.

La conversión eclíptica a ecuatorial se implementada manualmente en `astro_engine.py` para evitar dependencias adicionales:

La fórmula para conversión de Eclípticas (λ, β) a ecuatoriales (RA, Dec) se realiza rotando el vector por ϵ alrededor del eje X (de eclíptico a ecuatorial):

```
x_e = cosβ cosλ
y_e = cosβ sinλ
z_e = sinβ
x_q = x_e
y_q = y_e cosε - z_e sinε
z_q = y_e sinε + z_e cosε
```

Cálculo del arco (llave) y claves disponibles

El arco dirigido se calcula a partir de la edad en años tropicales promedio. El código usa:

```
age_years = (dt_target_utc - dt_birth_utc).total_seconds() / (365.2422 * 86400.0)
```

Claves implementadas:

- Clave de Ptolomeo ("`year-for-degree`") que es 1° por año.
- Clave de Naibod: $360^\circ / 365.2422 \approx 0.98564733^\circ$ por año ($\approx 59'08.33''/\text{año}$).

En el código, `arc_per_year` se selecciona así:

```
arc_per_year = (360.0 / 365.2422) if use_naibod else 1.0
arc_unsigned = age_years * arc_per_year
arc_signed   = -arc_unsigned if converse else arc_unsigned
arc = arc_signed
arc_mod = arc_signed % 360.0 # solo para presentar 0..360
```

Observación práctica: esta forma de arco no usa el Sol del día objetivo para determinar el arco (a diferencia del Arco Solar clásico), sino una equivalencia tiempo→grados basada en la clave elegida. Por ello el valor depende únicamente de la diferencia temporal (edad) y de la constante 365.2422.

Esto quiere decir que, en la técnica implementada, el arco direccional no se calcula comparando la posición real del Sol en dos fechas (natal vs. fecha dirigida), se calcula aplicando una regla proporcional fija que traduce tiempo vivido a grados de arco, usando una clave constante. Es decir: *el arco depende solo de cuántos años han pasado*, no de dónde esté realmente el Sol en el cielo en la fecha objetivo.

Como contraste conceptual, existen dos formas distintas de obtener el arco:

Arco Solar clásico (no es lo que usa el código)

En el Arco Solar clásico tenemos

$$\text{Arco} = \lambda_{\odot}(\text{fecha objetivo}) - \lambda_{\odot}(\text{natal})$$

Donde $\lambda_{\odot}$ = longitud eclíptica real del Sol. se calcula con efemérides astronómicas. El arco varía ligeramente según irregularidades del año trópico, ecuación del tiempo o pequeñas variaciones orbitales.

Dos personas con la misma edad pueden tener arcos solares distintos.

Direcciones primarias por clave temporal (lo que usa el código)

En la implementación tenemos

$$\text{Arco} = \text{Edad} \times \text{Clave}$$

La clave puede ser una **constante astronómica**:

- **Ptolomeo:**

$$1^{\circ} = 1 \text{ año}$$

- **Naibod:**

$$\frac{360^{\circ}}{365.2422} \approx 0.98564733^{\circ} \text{ por año}$$

Por lo tanto, en los cálculos de la implementación no se consulta la efeméride solar, no importa la fecha calendárica exacta, el arco depende únicamente del tiempo transcurrido. Es decir, el arco direccional no se obtiene a partir de la posición astronómica real del Sol en la fecha objetivo —como ocurre en la técnica de Arco Solar—, sino mediante una conversión directa del tiempo transcurrido en grados de arco, utilizando una clave fija (Ptolomeo o Naibod). En consecuencia, el valor del arco depende exclusivamente de la edad del nativo y de la constante astronómica adoptada (año trópico medio = 365.2422 días), independientemente de la posición real del Sol en la fecha dirigida.

El código implementa **direcciones primarias clásicas**, no Arco Solar moderno:

- **Tiempo → arco**, no **Sol → arco**
- Arco matemático, no efemérico

Esto es coherente con la tradición clásica de direcciones primarias; en la variante zodiacal la conservación de *house position* se evalúa sobre el sistema de casas activo en Armon2.

Como ejemplo numérico supongamos:

- Fecha natal: 31-oct-1970
- Fecha dirigida: 31-oct-2020
- Edad: **50 años**

Con Arco Solar clásico se calculan posiciones reales:

- Sol natal $\approx 7^\circ$ Escorpio
- Sol 2020 $\approx 8^\circ$ Escorpio

$$\text{Arco} \approx 31^\circ$$

(No necesariamente igual a la edad)

Mientras que con la técnica del código (usando Naibod)

$$\text{Arco} = 50 \times 0.98564733 = 49.2823665^\circ$$

Ese arco se suma al ARMC natal o se traduce a eclíptica según el método (zodiacal o ascensional), pero no cambia aunque la fecha exacta sea 2020-10-31 o 2020-11-02.

Depende únicamente de la edad porque la constante 365.2422 es el año trópico medio, usado como:

$$\text{factor de conversión} = \frac{360^\circ}{365.2422}$$

Por lo tanto 1 día simbólico = 1 año de vida y 1 rotación completa del cielo = vida completa simbolizada. Esto es coherente con el uso de claves temporales fijas en direcciones primarias clásicas.

Variante A: Primary Zodiacal — house position invariante

- Sistema de referencia: **Eclíptica**
- Casas: sistema de casas activo en Armon2 (Placidus, Topocéntrico, Regiomontanus o Iguales)
- Principio: **house position invariante**
- Requiere: cálculo de `house_pos` y de inversión numérica (muestreo + bisección)
- Subtemas: sistema de casas activo, tratamiento interno de latitud, proceso de inversión y claves (Naibod / Ptolomeo)

Esta variante corresponde a lo que muchos programas llaman "Primary Zodiacal". En Armon2, aunque la función conserva un nombre histórico asociado a Placidus, la geometría de casas usada en el cálculo no está fija a Placidus: se toma siempre el sistema de casas global actualmente seleccionado por el usuario.

El ARMC natal y el ARMC dirigido se obtienen a partir de la oblicuidad verdadera ε y de las casas natales calculadas por Swiss Ephemeris con el sistema de casas activo. En el código:

```
ecl_nut = swe.calc_ut(jd_birth, swe.ECL_NUT)
eps = float(ecl_nut[0][0])
cusps_nat, ascmc_nat = swe.houses(jd_birth, resolved_lat, resolved_lon, _house_system_bytes())
armc_nat = float(ascmc_nat[2]) % 360.0
```

Luego se aplica el arco (directo o converso) para obtener el ARMC dirigido:

```
armc_dir = (armc_nat + arc_unsigned) % 360.0    # directo
armc_dir = (armc_nat - arc_unsigned) % 360.0    # converso
```

Con el ARMC dirigido, las casas dirigidas se recalculan con `houses_armc()` usando el mismo sistema de casas activo. Esto fija el marco geométrico para resolver el resto de los puntos.

```
cusps_dir, ascmc_dir = swe.houses_armc(armc_dir, resolved_lat, eps, _house_system_bytes())
houses_raw_dir = {
    'cusps': list(cusps_dir),
    'asc': float(ascmc_dir[0]),
    'mc': float(ascmc_dir[1]),
    'armc': float(ascmc_dir[2]),
}
```

Tratamiento actual de latitud en la variante zodiacal.

Astronómicamente, cada planeta tiene:

- Longitud eclíptica λ
- Latitud eclíptica β

La latitud indica cuánto está el planeta por encima o debajo del plano de la eclíptica. Durante el desarrollo se evaluaron distintas posibilidades teóricas, pero la implementación actual ya no expone variantes de latitud al usuario.

En la implementación actual de la variante zodiacal, el cálculo se resuelve de manera estrictamente eclíptica para el speculum utilizado en la conservación de house position. Esto significa que la house position natal del significador se evalúa sobre la eclíptica, tomando latitud 0, y que la longitud dirigida final también se obtiene sobre la eclíptica mediante inversión numérica.

Esta decisión elimina ambigüedades metodológicas internas y mantiene la variante zodiacal como una formulación uniforme: house position conservada sobre la eclíptica, pero dentro del sistema de casas activo.

En consecuencia, la técnica zodiacal no está limitada a una sola formulación fija. La interfaz actual ofrece una opción explícita para decidir si el **promisor** se evalúa como punto eclíptico puro o si conserva su latitud natal en el speculum zodiacal.

Opción visible en la interfaz

La opción *"Usar latitud natal para promisor"* aparece dentro de la sección de Direcciones Primarias, solo tiene efecto en el modo Zodiacales y actualmente queda activado por defecto. Si el usuario lo desactiva, Armon2 vuelve a la variante zodiacal pura con $\beta = 0$.

Funcionamiento interno de ambas variantes

Cuando el checkbox está activado, hp_nat se calcula con la latitud natal real del promisor y la semilla lon_guess también se construye desplazando por RA con esa latitud. Cuando está desactivado, tanto hp_nat como la inversión dirigida trabajan con latitud 0 para el speculum zodiacal.

Consecuencia práctica

La inversión final sigue haciéndose en longitud eclíptica pura. Lo que cambia entre ambas variantes es el speculum usado para conservar house position y la semilla inicial por RA.

Esto permite dos comportamientos legítimos dentro de la misma técnica: una variante zodiacal pura, útil como referencia comparativa, y una variante con latitud del promisor, más cercana a software que declara explícitamente "promisor with latitude".

Modo	Resultado en la implementación actual
Zodiacales	Conservan house position bajo el sistema de casas activo (Placidus, Topocéntrico, Regiomontanus o Iguales) y resuelven la longitud dirigida sobre la eclíptica mediante inversión numérica.
AR (Carta Ascensional)	Trabajan en Ascensión Recta real, con 12 sectores de 30° desde el ARMC dirigido; no usan house_pos() ni inversión numérica.

Por ello, las diferencias entre cartas dirigidas pueden provenir no solo del sistema de casas activo, la clave direccional, el sentido directo/converso y el modo elegido (zodiacal o AR), sino también de este checkbox específico en la variante zodiacal.

Para cada planeta/punto con longitud natal λ_{nat} , el algoritmo procede así para el cálculo de dirección (y conservar house position):

- 1) Calcular house position natal hp_{nat} con `swe.house_pos()` sobre el sistema de casas activo, usando β natal real si el checkbox está activado o $\beta = 0$ si está desactivado.
- 2) Generar una semilla `lon_guess` desplazando por RA: se convierte $(\lambda_{\text{nat}}, \beta_{\text{speculum}})$ a (RA, Dec) , se suma el arco en RA y se regresa a longitud eclíptica. Esta semilla acelera la convergencia.
- 3) Resolver λ_{dir} tal que `house_pos(armc_dir,  $\lambda_{\text{dir}}$ , 0.0)  $\approx$   $hp_{\text{nat}}$` . La longitud final dirigida sigue quedando sobre la eclíptica.

Esquema en el código:

```
hp_nat = swe.house_pos(armc_nat, geolat, eps, (lon_nat, beta_speculum), _house_system_bytes())
lon_guess = dir_by_ra(lon_nat, beta_speculum, arc)
lon_dir = solve_lon_for_housepos(hp_nat, lon_guess)
```

Debe subrayarse que, en la implementación actual, tanto el cálculo de la house position natal como la inversión de la longitud dirigida se realizan sobre la eclíptica (latitud 0), pero siempre bajo el sistema de casas activo seleccionado por el usuario.

Proceso de inversión numérica (`solve_lon_for_housepos`) - Planteamiento matemático del problema

En la Variante A (Primary Zodiacal), cada punto dirigido debe conservar su house position natal bajo el ARMC dirigido y dentro del sistema de casas activo.

Dado que la función que relaciona longitud eclíptica y posición en casas (`swe.house_pos`) no admite una inversión analítica directa y presenta comportamiento no lineal y periódico, el código implementa un método numérico robusto para recuperar la longitud eclíptica correspondiente a una *house position* objetivo.

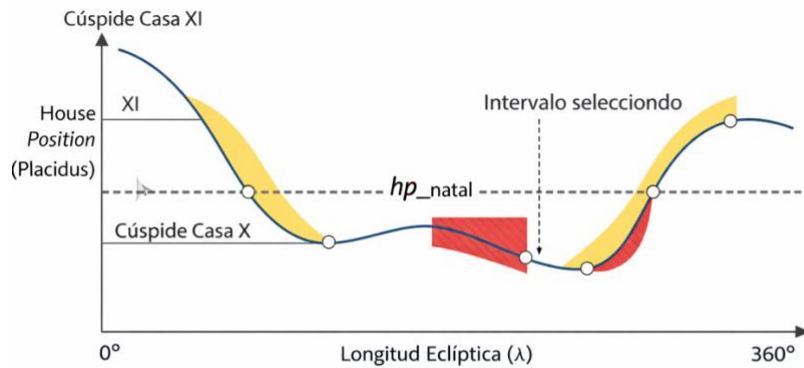


Gráfico que ilustra el proceso de inversión numérica para encontrar la longitud eclíptica que conserve la misma posición en casas del ponto natal. Se muestra la función de house position (hp) en casas Placidus y la línea horizontal del hp_{natal} . Los intervalos con cambio de signo se indican en amarillo, y el intervalo seleccionado para la bisección se resalta en rojo.

Paso 1: definir una función de diferencia

Se define implícitamente la función:

$$d(\lambda) = hp(\lambda) - hp_{objetivo}$$

Donde $hp(\lambda)$ se calcula con `swe.house_pos()`. El objetivo es encontrar un cero de esa función:

$$d(\lambda) = 0$$

Paso 2: muestreo global de 0° a 360°

El código recorre $\lambda = 0^\circ$ a 360° con paso fijo de **0.5°**, evalúa $d(\lambda)$ en cada punto. Esto genera una tabla aproximada del comportamiento de la función. Se elige 0.5° porque hace un compromiso entre precisión inicial y rendimiento suficiente para detectar “cruces” sin saltarse soluciones.

Paso 3: detección de cambios de signo

Se buscan intervalos $[\lambda_i, \lambda_{i+1}]$ tales que

$$d(\lambda_i) \cdot d(\lambda_{i+1}) < 0$$

Eso implica que la función cruzó por cero y hay al menos una solución dentro del intervalo. Este paso es crucial porque la función puede cruzar varias veces y cada cruce corresponde a una solución geométrica distinta.

Paso 4: selección del intervalo correcto

Si hay varios intervalos válidos, el código elige el intervalo más cercano a `lon_guess`. `lon_guess` proviene de la longitud eclíptica aproximada del punto antes de la inversión o de la posición previa dirigida. Esto introduce **continuidad geométrica** y evita saltos a la solución “equivocada” por periodicidad e inversiones a la casa opuesta o al hemisferio incorrecto.

Paso 5: refinamiento por bisección

Una vez elegido el intervalo se aplica **método de bisección** hasta **40 iteraciones** donde cada iteración reduce el intervalo a la mitad. El resultado es precisión sub-arcosegundo y convergencia garantizada (si hay cambio de signo). Matemáticamente, el error final es:

$$\frac{360^\circ}{2^{40}} \ll 10^{-6} \text{ grados}$$

Este enfoque se considera robusto y correcto porque no depende de derivadas (a diferencia de Newton-Raphson), funciona con funciones no lineales, no suaves, periódica, maneja múltiples soluciones correctamente, es compatible con latitudes extremas, casas interceptadas, zonas de curvatura alta cerca de cúspides.

Esto no es un hack, es un método numérico clásico de inversión funcional.

Variante B: AR (Carta Ascensional) — direcciones en Ascensión Recta

- Sistema de referencia: **Ecuador celeste**
- Casas: **Ascensionales (30° de AR)**
- Principio: **rotación uniforme del cielo**
- No requiere: `house_pos` ni inversión numérica

Casas ascensionales (*Direcciones en Ascensión Recta*)

En casas **ascensionales**, el sistema de referencia **ya no es la eclíptica**, sino el **ecuador celeste**. Esto implica un cambio radical:

Zodiacales	Ascensionales
Longitud eclíptica (λ)	Ascensión Recta (α)
Sistema de casas activo (P/T/R/E)	12 sectores iguales de 30° desde el ARMC
Dependen de latitud geográfica y del sistema de casas	No dependen del sistema global de casas
Requiere inversión de <code>house_pos()</code>	No requiere inversión

En casas ascensionales no existe inversión funcional de posiciones en casas, ya que la asignación se realiza directamente en Ascensión Recta. Esta capa es independiente del sistema global de casas usado por la carta natal, porque la carta ascensional define sus propias 12 divisiones uniformes de 30° de AR desde el ARMC dirigido.

$$(\lambda, \beta) \rightarrow \alpha$$

En la implementación actual no hay dos submodos visibles para calcular la AR del planeta. La variante ascensional usa la Ascensión Recta real del cuerpo natal, obtenida a partir de su longitud y latitud eclípticas natales.

$$\alpha = \text{RA}(\lambda, \beta)$$

No existe ya una variante operativa zodiacal_nolat en la interfaz para esta técnica.

$$\alpha = RA(\lambda, 0)$$

Por lo tanto, las capas reales de la implementación actual son simplemente:

1. Modo geométrico
 - Zodiacales
 - Ascensionales
2. Clave temporal
 - Ptolomeo
 - Naibod
3. Sentido
 - Directo
 - Converso

Y, para la variante zodiacal, el sistema de casas activo en la carta base.

Esto hace que la variante AR sea conceptualmente más cerrada: no depende del selector global de casas, no usa house_pos() y no realiza inversión numérica.

Esta variante aproxima una carta ascensional donde el sistema de casas es un círculo de 360° dividido en 12 sectores iguales de 30°, contados desde el ARMC. En consecuencia, casas y puntos quedan en coordenadas de Ascensión Recta (AR).

Las casas ascensionales dividen el ecuador en 12 sectores iguales de 30° de Ascensión Recta, comenzando en el ARMC. Sea **ARMC**: Ascensión Recta del Medio Cielo y **α** : Ascensión Recta del planeta (directa o dirigida) tenemos:

$$\text{Casa ascensional} = \left\lfloor \frac{\alpha - \text{ARMC}}{30^\circ} \right\rfloor + 10$$

(con ajuste módulo 12). No hay cúspides curvadas ni ecuaciones implícitas. Es geometría lineal sobre el ecuador. En la implementación:

1. Se dirige el **ARMC** usando la clave (Ptolomeo o Naibod).
2. Se calcula la **Ascensión Recta del planeta**:
 - a partir de la posición eclíptica natal completa del cuerpo (longitud y latitud)
 - y se transforma a Ascensión Recta real antes de aplicar el arco
3. Se calcula directamente la casa ascensional:
 - sin llamar a swe.house_pos

- sin inversión numérica

Se define MC = ARMC y ASC = ARMC+90°. Luego cada cúspide avanza 30° en AR:

```
for i in range(12):
    cusp[i] = (armc_dir + 90.0 + 30.0*i) % 360.0
asc = (armc_dir + 90.0) % 360.0
mc = armc_dir % 360.0
```

Puntos ascensionales

¿Qué puntos se dirigen y cómo se expresan en Ascensión Recta dentro de la carta ascensional?

En la técnica de direcciones primarias en Ascensión Recta, es esta Variante B, todos los significadores se expresan como valores de Ascensión Recta sobre el ecuador celeste. Los puntos ascensionales incluyen el MC y el ASC ascensionales, las cúspides de casas definidas como sectores iguales de 30° de AR, así como los planetas y demás puntos dirigibles, convertidos previamente desde coordenadas eclípticas a ecuatoriales.

A diferencia de las direcciones zodiacales, no se conserva la posición relativa en casas ni se requiere la inversión de funciones implícitas. La dirección se basa exclusivamente en la comparación directa entre la Ascensión Recta del punto y el ARMC dirigido, reflejando la rotación uniforme del cielo.

Todos los puntos se trabajan en **Ascensión Recta (AR)** sobre el **ecuador celeste**, por lo tanto, un *punto ascensional* es cualquier significador (planeta, cúspide, punto sensible) expresado como un valor de Ascensión Recta y comparado directamente con el ARMC dirigido.

No hay proyección eclíptica a casas, conservación de `house position` ó inversión numérica. Los tipos de puntos ascensionales en la implementación son:

1. MC y ASC ascensionales

Definición directa en el código para MC ascensional

$$MC = ARMC$$

Para el ASC ascensional

$$ASC = ARMC + 90^\circ$$

El ASC y MC ascensionales no corresponden a las intersecciones reales con la eclíptica, sino a puntos definidos puramente en el ecuador celeste.

2. Cúspides ascensionales

Las cúspides se definen como

$$C_i = ARMC + 90^\circ + 30^\circ \cdot i (i = 0 \dots 11)$$

esto genera 12 sectores iguales de 30° exactos de AR sin dependencia de latitud geográfica

```
for i in range(12):
    cusp[i] = (armc_dir + 90.0 + 30.0*i) % 360.0
```

3. Planetas como puntos ascensionales

Cada planeta se convierte en un punto ascensional a partir de su posición eclíptica natal completa, transformada a Ascensión Recta real.

AR real del cuerpo natal

sin submodos visibles de latitud en la interfaz

$$\alpha = RA(\lambda, 0)$$

Ese valor de AR es el que se usa directamente para aspectos por AR, cruces de cúspides y contactos con MC / ASC ascensionales.

4. Otros puntos dirigibles

Pueden ser Nodo lunar, Partes arábicas, Puntos medios, Puntos ficticios, siempre que puedan expresarse como valores dirigibles. En la implementación actual de esta técnica la carta dirigida trabaja principalmente con los puntos que ya forman parte del formato estándar de Armon2.

Los puntos ascensionales no son puntos eclípticos, no conservan house position, no usan `house_pos`, no participan en inversión numérica, por lo que en la Variante B no se realiza inversión de posiciones en casas, ya que todos los puntos se comparan directamente en Ascensión Recta.

Para cada punto, se toma su posición eclíptica natal completa, es decir, longitud y latitud eclípticas, y se transforma a Ascensión Recta real. Sobre esa Ascensión Recta natal se aplica directamente el arco dirigido. El valor resultante se entrega en `planets_raw` como una coordenada de AR.

```
if mode == "AR":
    ra_nat, dec_nat = ecl_to_equ(lon_nat, lat_nat, eps)
    point_dir = (ra_nat + arc) % 360.0
```

Representación y semántica del modo AR

La variante AR (Carta Ascensional) no debe interpretarse como una carta zodiacal convencional. Aunque conserva la estructura general de salida de una carta dentro de Armon2, sus valores representan coordenadas en Ascensión Recta y no longitudes zodiacales.

Por ello, en la implementación actual la carta dirigida AR se marca explícitamente como sistema de coordenadas ecuatoriales, distinguiéndose de la carta zodiacal tanto en el formateo de posiciones como en su representación gráfica. Esta decisión evita mezclar semánticamente coordenadas eclípticas y ecuatoriales bajo una misma apariencia visual.

Cálculo Converso y normalización angular

El cálculo converso se implementa invirtiendo el signo del arco. En todos los casos, las longitudes/AR finales se normalizan módulo 360°. El código conserva además `arc_mod` para reportar el arco en 0..360°.

Metadatos de salida y compatibilidad con el formato de carta

La función `primary_directions_placidus_zodiacal` devuelve un dict compatible con `chart_for_date()`, incluyendo `planets`, `planets_raw`, `houses`, `houses_raw`, `aspectos` y armónico predominante. Incluye metadatos específicos:

- `edad_actual` (años tropicales promedio)
- `directed_arc` (arco con signo) y `directed_arc_mod` (0..360)
- `key_name`: 'Ptolomeo' o 'Naibod'
- `mode`: etiqueta humana ('Primary Zodiacal (Placidus)' o 'AR (Carta Ascensional)'). En el caso zodiacal, la etiqueta es histórica y no implica que el sistema de casas esté forzado a Placidus.
- `coordinate_system`: 'zodiacal' o 'equatorial_ra'
- `chart_type`: identificador técnico de la carta dirigida dentro del sistema

Pseudocódigo (alto nivel)

```
if mode == "AR":
    # Variante B: Carta ascensional en Ascensión Recta
    ra_nat = RA(lon_nat, lat_nat, eps)
    point_dir = (ra_nat + arc) % 360.0
else:
    # Variante A: Primary Zodiacal sobre el sistema de casas activo
    hp_nat = house_pos(armc_nat, lon_nat, beta_speculum)
    lon_guess = dir_by_ra(lon_nat, beta_speculum, arc)
    lon_dir = solve_lon_for_housepos(armc_dir, hp_nat, lon_guess)
    point_dir = lon_dir % 360.0
```

11. Consideraciones de validación

El motor incluye un bloque de depuración activable con la variable de entorno `ARMON_PD_DEBUG` que imprime: fechas UTC, lat/lon, clave, edad, arco, RAMC natal y RAMC dirigido (directo/converso). Esta salida está pensada para contrastar con bibliografía o software de referencia.

Para calibración cruzada con software (p.ej. Solar Fire, Astro.com), los puntos más sensibles son:

- Convención de año (365.2422 vs 365.25) en la edad
- Uso de oblicuidad verdadera (`ECL_NUT`) vs oblicuidad media
- Definición del speculum zodiacal: Armon2 permite comparar la variante con $\beta = 0$ y la variante con latitud natal del promisor; actualmente esta última queda activada por defecto en la interfaz de Direcciones Primarias.
- Selección de sistema de casas en la carta base (Placidus, Topocéntrico, Regiomontanus o Iguales), ya que la variante zodiacal usa ese sistema activo.

Conclusión Técnica

La implementación actual de Direcciones Primarias en Armon2 distingue con claridad dos marcos geométricos de trabajo: una variante zodiacal resuelta por conservación de house position e inversión numérica dentro del sistema de casas activo, con dos subvariantes controladas por interfaz (promisor con latitud natal o zodiacal pura con $\beta = 0$), y una variante ascensional expresada en Ascensión Recta real, donde las casas son 12 sectores de 30° contados desde el ARMC dirigido.

Esta separación metodológica fortalece la coherencia interna del sistema: la variante zodiacal permanece íntegramente definida sobre la eclíptica, pero usa la geometría del sistema de casas elegido por el usuario y ahora puede trabajar tanto como punto eclíptico puro como con latitud natal del promisor; la variante AR, en cambio, opera en coordenadas ecuatoriales propiamente dichas y no depende del sistema global de casas. Además, la técnica de Rectificación reutiliza este mismo motor de Direcciones Primarias; por ello, cuando se desea trabajar al estilo Marr/Polaris, conviene seleccionar Topocéntrico + Naibod + modo Zodiacal en la carta base. Si además se busca acercarse a software que declara "promissor with latitude", debe mantenerse activado el checkbox "Usar latitud natal para promisor".