

# Plan de Validación Empírica

Del Modelo Teórico a los Datos Observacionales  
Cosmología Estocástica con Memoria Finita

Ernesto Cisneros Cino  
info@impulses.online

Noviembre 2025

*Documento Operacional para Investigadores Independientes*

## Abstract

Este documento presenta un protocolo operacional completo para validar empíricamente el modelo de cosmología estocástica con memoria finita usando datos públicos de supernovas tipo Ia (Pantheon+). Incluye código Python reproducible, criterios de falsabilidad explícitos, y estrategia de comunicación científica. Diseñado para investigadores independientes con recursos computacionales estándar.

**Duración estimada:** 4-6 semanas

**Recursos requeridos:** Python 3.8+, 8GB RAM, acceso a internet

**Costo:** \$0 (todo software y datos son públicos)

## Contents

<b>1</b>	<b>Introducción: Del Modelo a la Realidad</b>	<b>2</b>
1.1	Contexto del Proyecto . . . . .	2
1.2	Objetivo de este Documento . . . . .	2
1.3	Filosofía Metodológica . . . . .	2
<b>2</b>	<b>Fase 1: Análisis Mínimo Viable (2-4 semanas)</b>	<b>3</b>
2.1	Dataset: Pantheon+ (Supernovas Tipo Ia) . . . . .	3
2.2	Modelo Cosmológico a Probar . . . . .	3
2.2.1	Parametrización . . . . .	3
2.2.2	Parámetros Libres . . . . .	3
2.3	Implementación: Código Python Completo . . . . .	4
2.3.1	Paso 1: Descarga y Preparación de Datos . . . . .	4
2.3.2	Paso 2: Definición del Modelo Cosmológico . . . . .	6
2.3.3	Paso 3: Inferencia Bayesiana con EMCEE . . . . .	8
2.4	Instrucciones de Ejecución . . . . .	13
2.4.1	Instalación de Dependencias . . . . .	13
2.4.2	Ejecución Secuencial . . . . .	13
2.4.3	Monitoreo de Progreso . . . . .	13
<b>3</b>	<b>Fase 2: Interpretación de Resultados (1 semana)</b>	<b>14</b>
3.1	Criterios de Éxito Pre-Registrados . . . . .	14
3.2	Checklist de Análisis . . . . .	14
3.3	Tabla de Resultados Esperados (Template) . . . . .	16
3.4	Interpretación de $\Delta BIC$ . . . . .	16

<b>4 Fase 3: Pre-Print y Llamado a la Comunidad</b>	<b>17</b>
4.1 Condición para Publicación . . . . .	17
4.2 Estructura del Pre-Print (arXiv) . . . . .	17
4.2.1 Template de Abstract . . . . .	17
4.2.2 Secciones Requeridas . . . . .	17
4.3 Estrategia de Comunicación . . . . .	19
4.3.1 Paso 1: Pre-Registro en OSF (Open Science Framework) . . . . .	19
4.3.2 Paso 2: Difusión Científica Calibrada . . . . .	19
4.4 Respuestas Esperables y Cómo Manejarlas . . . . .	21
4.4.1 Escenario A: Silencio (más probable) . . . . .	21
4.4.2 Escenario B: Crítica técnica constructiva . . . . .	21
4.4.3 Escenario C: Interés genuino / oferta de colaboración . . . . .	21
<b>5 Fase 4: Validación Extendida (Opcional, 3-6 meses)</b>	<b>22</b>
5.1 Si los Resultados de Pantheon+ son Prometedores . . . . .	22
5.1.1 A. Oscilaciones Acústicas Bariónicas (BAO) . . . . .	22
5.1.2 B. Efecto Sachs-Wolfe Integrado Tardío (ISW) . . . . .	22
5.1.3 C. Tasa de Crecimiento de Estructura ( $f\sigma_8$ ) . . . . .	23
5.2 Protocolo de Validación Multi-Dataset . . . . .	24
5.3 Recursos Computacionales Necesarios . . . . .	24
<b>6 Timeline y Milestones</b>	<b>25</b>
6.1 Cronograma Realista . . . . .	25
6.2 Milestones de Decisión . . . . .	25
<b>7 Recursos y Soporte</b>	<b>26</b>
7.1 Recursos Educativos . . . . .	26
7.1.1 Python Científico (si necesitas nivelación) . . . . .	26
7.1.2 Inferencia Bayesiana . . . . .	26
7.1.3 Cosmología Observacional . . . . .	26
7.2 Comunidades de Soporte . . . . .	26
7.3 Herramientas Adicionales . . . . .	26
<b>8 Conclusión y Siguiente Paso</b>	<b>27</b>
8.1 Resumen Ejecutivo . . . . .	27
8.2 El Próximo Paso es Tuyo . . . . .	27
8.3 Compromiso del Autor de este Plan . . . . .	27
8.4 Nota Final . . . . .	28
<b>A Apéndice A: Troubleshooting Común</b>	<b>29</b>
A.1 Error: "No module named 'emcee'" . . . . .	29
A.2 Error: "HTTPError 404: Not Found" al descargar datos . . . . .	29
A.3 MCMC muy lento (¿2 horas) . . . . .	29
A.4 Posteriors "raros" (bimodales, degenerados) . . . . .	29
<b>B Apéndice B: Glosario de Términos</b>	<b>30</b>
<b>C Apéndice C: Checklist Pre-Publicación</b>	<b>30</b>

# 1 Introducción: Del Modelo a la Realidad

## 1.1 Contexto del Proyecto

El modelo de cosmología estocástica propone que la ecuación de estado de la energía oscura  $w(z)$  exhibe oscilaciones amortiguadas controladas por un parámetro de memoria  $\tau$ :

$$w(z) = -1 + Ae^{-z/z_\tau} \cos(\omega \ln(1+z) + \delta) \quad (1)$$

donde:

- $A$ : amplitud de oscilación
- $\omega$ : frecuencia en escala logarítmica de redshift
- $\delta$ : fase inicial
- $z_\tau = cH_0\tau$ : profundidad de memoria efectiva

**Predicción central:** El sistema alcanza máxima estabilidad cuando  $\tau H_0 \in [0.5, 3.5]$ .

## 1.2 Objetivo de este Documento

### Objetivo Principal

Determinar si existe señal estadística de oscilaciones en  $w(z)$  consistente con la predicción teórica, usando datos públicos y métodos reproducibles.

## 1.3 Filosofía Metodológica

1. **Transparencia radical:** Todo el código es público y reproducible
2. **Falsabilidad pre-registrada:** Criterios de éxito/fracaso definidos *antes* del análisis
3. **Honestidad epistémica:** Un resultado negativo es tan valioso como uno positivo
4. **Apertura colaborativa:** Invitación explícita a crítica y validación independiente

## 2 Fase 1: Análisis Mínimo Viable (2-4 semanas)

### 2.1 Dataset: Pantheon+ (Supernovas Tipo Ia)

¿Por qué Pantheon+?

- Es el dataset de SNe Ia más grande y preciso disponible (2022)
- Acceso público inmediato, sin solicitudes burocráticas
- Metodología bien documentada (Brout et al. 2022, ApJ)
- Cubre redshift  $z \in [0.01, 2.3]$  donde esperamos ver oscilaciones

**Especificaciones técnicas:**

Propiedad	Valor
Número de SNe Ia	1701
Rango de redshift	$0.01 < z < 2.3$
Precisión promedio	$\Delta m_B \sim 0.15$ mag
Formato	ASCII (.dat)
Tamaño	$\sim 2$ MB
URL	<a href="https://github.com/PantheonPlusSH0ES/DataRelease">https://github.com/PantheonPlusSH0ES/DataRelease</a>

Table 1: Características del dataset Pantheon+

### 2.2 Modelo Cosmológico a Probar

#### 2.2.1 Parametrización

La ecuación de estado oscilatoria se integra en la ecuación de Friedmann:

$$H^2(z) = H_0^2 \left[ \Omega_m (1+z)^3 + \Omega_\Lambda \exp \left( 3 \int_0^z \frac{1+w(z')}{1+z'} dz' \right) \right] \quad (2)$$

El módulo de distancia observable es:

$$\mu(z) = 5 \log_{10} [d_L(z)] + 25 \quad (3)$$

donde la distancia luminosidad:

$$d_L(z) = c(1+z) \int_0^z \frac{dz'}{H(z')} \quad (4)$$

#### 2.2.2 Parámetros Libres

Vector de parámetros:  $\theta = \{A, \omega, \delta, \tau H_0, \Omega_m, H_0\}$

Parámetro	Prior Min	Prior Max	Prior Type
$A$	0	0.2	Uniforme
$\omega$	1	5	Uniforme
$\delta$	0	$2\pi$	Uniforme
$\tau H_0$	0.5	5	Uniforme
$\Omega_m$	0.25	0.35	Gaussiano ( $\mu = 0.315, \sigma = 0.02$ )
$H_0$	65	75	Gaussiano ( $\mu = 70, \sigma = 3$ )

Table 2: Distribuciones a priori para los parámetros del modelo

## 2.3 Implementación: Código Python Completo

### 2.3.1 Paso 1: Descarga y Preparación de Datos

```

1 """
2 Descarga y prepara datos de Pantheon+ para an lisis
3 Ejecutar: python 1_download_data.py
4 Output: pantheon_clean.npz
5 """
6
7 import pandas as pd
8 import numpy as np
9 import urllib.request
10 import os
11
12 def download_pantheon():
13     """Descarga datos de Pantheon+ desde GitHub"""
14
15     url = "https://github.com/PantheonPlusSHOES/DataRelease/raw/main/Pantheon%2
16         B_Data/2_DISTANCES_AND_COVAR/Pantheon%2BSHOES.dat"
17
18     filename = 'Pantheon+SHOES.dat'
19
20     if not os.path.exists(filename):
21         print(f"Descargando {filename}...")
22         urllib.request.urlretrieve(url, filename)
23         print("Descarga completa.")
24     else:
25         print(f"{filename} ya existe. Omitiendo descarga.")
26
27     return filename
28
29 def prepare_data(filename):
30     """Limpia y prepara datos para an lisis"""
31
32     # Leer datos (formato específico de Pantheon+)
33     data = pd.read_csv(filename, delim_whitespace=True, comment='#')
34
35     # Extraer columnas relevantes
36     z = data['zHD'].values # Redshift heliocéntrico
37     mu_obs = data['m_b_corr'].values # Módulo de distancia observado
38     mu_err = data['m_b_corr_err_DIAG'].values # Error diagonal
39
40     # Filtrar datos válidos
41     mask = (z > 0.01) & (z < 2.5) & np.isfinite(mu_obs) & np.isfinite(mu_err)
42     z_clean = z[mask]
43     mu_clean = mu_obs[mask]
44     err_clean = mu_err[mask]
45
46     # Ordenar por redshift

```

```

46     sort_idx = np.argsort(z_clean)
47     z_clean = z_clean[sort_idx]
48     mu_clean = mu_clean[sort_idx]
49     err_clean = err_clean[sort_idx]
50
51     # Guardar en formato numpy
52     np.savez('pantheon_clean.npz',
53             z=z_clean,
54             mu=mu_clean,
55             mu_err=err_clean)
56
57     print(f"\nDatos preparados:")
58     print(f" - N supernovas: {len(z_clean)}")
59     print(f" - Rango z: [{z_clean.min():.3f}, {z_clean.max():.3f}]")
60     print(f" - Error promedio: {err_clean.mean():.3f} mag")
61
62     return z_clean, mu_clean, err_clean
63
64 if __name__ == "__main__":
65     filename = download_pantheon()
66     z, mu, err = prepare_data(filename)
67
68     # Gráfico de verificaci n
69     import matplotlib.pyplot as plt
70     plt.figure(figsize=(10, 6))
71     plt.errorbar(z, mu, yerr=err, fmt='o', markersize=2,
72                 alpha=0.3, elinewidth=0.5, label='Pantheon+')
73     plt.xlabel('Redshift $z$')
74     plt.ylabel(r'M dulo de distancia $\mu$ (mag)')
75     plt.title('Pantheon+: Diagrama de Hubble')
76     plt.legend()
77     plt.grid(alpha=0.3)
78     plt.savefig('hubble_diagram_raw.png', dpi=150)
79     print("\nGráfico guardado: hubble_diagram_raw.png")

```

Listing 1: 1\_download\_data.py: Descarga automática de Pantheon+

### 2.3.2 Paso 2: Definición del Modelo Cosmológico

```

1  """
2  Modelo cosmológico con ecuación de estado oscilatoria
3  """
4
5  import numpy as np
6  from scipy.integrate import cumulative_trapezoid, quad
7  from scipy.interpolate import interp1d
8
9  def w_oscillatory(z, A, omega, delta, tau_H0):
10     """
11     Ecuación de estado con memoria finita
12
13     Parameters:
14     -----
15     z : float or array
16         Redshift
17     A : float
18         Amplitud de oscilación [0, 0.2]
19     omega : float
20         Frecuencia angular [1, 5]
21     delta : float
22         Fase inicial [0, 2 ]
23     tau_H0 : float
24         Profundidad de memoria [0.5, 5]
25
26     Returns:
27     -----
28     w : float or array
29         Ecuación de estado w(z)
30     """
31     z_tau = tau_H0 # En unidades donde c*H0 = 1
32     return -1.0 + A * np.exp(-z/z_tau) * np.cos(omega * np.log1p(z) + delta)
33
34 def H_squared(z, H0, Omega_m, A, omega, delta, tau_H0):
35     """
36     Parámetro de Hubble al cuadrado: H^2(z)
37
38     Integra la ecuación de Friedmann con w(z) dinámico
39     """
40     Omega_Lambda = 1.0 - Omega_m # Universo plano
41
42     if np.isscalar(z):
43         z_array = np.linspace(0, z, 150)
44     else:
45         z_max = np.max(z)
46         z_array = np.linspace(0, z_max * 1.1, 150)
47
48     # Calcular w(z) en la grilla
49     w_array = w_oscillatory(z_array, A, omega, delta, tau_H0)
50
51     # Integrar: int_0^z [1+w(z')]/(1+z') dz'
52     integrand = (1.0 + w_array) / (1.0 + z_array)
53     integral = cumulative_trapezoid(integrand, z_array, initial=0)
54
55     # Interpolador para obtener valores en z arbitrarios
56     integral_func = interp1d(z_array, integral, kind='cubic',
57                             bounds_error=False, fill_value='extrapolate')
58
59     if np.isscalar(z):
60         integral_z = integral_func(z)
61     else:

```

```

62     integral_z = integral_func(z)
63
64     # Factor de energía oscura
65     factor_DE = np.exp(3.0 * integral_z)
66
67     # H(z) / H0
68     return Omega_m * (1.0 + z)**3 + Omega_Lambda * factor_DE
69
70 def luminosity_distance(z, H0, Omega_m, A, omega, delta, tau_H0):
71     """
72     Distancia luminosidad en Mpc
73
74     d_L(z) = c(1+z) * int_0^z dz'/H(z')
75     """
76     c_km_s = 299792.458 # Velocidad de la luz en km/s
77
78     def integrand(zp):
79         H2 = H_squared(zp, H0, Omega_m, A, omega, delta, tau_H0)
80         return 1.0 / (H0 * np.sqrt(H2))
81
82     if np.isscalar(z):
83         comoving_dist, _ = quad(integrand, 0, z, limit=100)
84     else:
85         comoving_dist = np.zeros_like(z)
86         for i, zi in enumerate(z):
87             comoving_dist[i], _ = quad(integrand, 0, zi, limit=100)
88
89     d_L = c_km_s * (1.0 + z) * comoving_dist
90
91     return d_L
92
93 def distance_modulus(z, H0, Omega_m, A, omega, delta, tau_H0):
94     """
95     Módulo de distancia observable: = 5 log (d_L/Mpc) + 25
96     """
97     d_L = luminosity_distance(z, H0, Omega_m, A, omega, delta, tau_H0)
98     mu = 5.0 * np.log10(d_L) + 25.0
99     return mu
100
101 # Caso especial: CDM (para comparación)
102 def distance_modulus_LCDM(z, H0, Omega_m):
103     """Módulo de distancia para CDM (w = -1 constante)"""
104     return distance_modulus(z, H0, Omega_m, A=0.0, omega=0.0,
105                             delta=0.0, tau_H0=1.0)

```

Listing 2: 2\_model.py: Funciones de integración cosmológica



### 2.3.3 Paso 3: Inferencia Bayesiana con EMCEE

```

1  """
2  Inferencia bayesiana usando emcee (MCMC)
3  Estima par metros del modelo y posteriors
4  """
5
6  import numpy as np
7  import emcee
8  import corner
9  import matplotlib.pyplot as plt
10 from model import distance_modulus, distance_modulus_LCDM
11 from multiprocessing import Pool
12 import time
13
14 # Cargar datos
15 data = np.load('pantheon_clean.npz')
16 z_obs = data['z']
17 mu_obs = data['mu']
18 mu_err = data['mu_err']
19
20 print(f"Datos cargados: {len(z_obs)} supernovas")
21
22 # ===== FUNCIONES DE PROBABILIDAD =====
23
24 def log_prior(theta):
25     """
26     Distribución a priori
27     theta = [A, omega, delta, tau_H0, Omega_m, H0]
28     """
29     A, omega, delta, tau_H0, Omega_m, H0 = theta
30
31     # Limites hard
32     if not (0.0 < A < 0.2): return -np.inf
33     if not (1.0 < omega < 5.0): return -np.inf
34     if not (0.0 < delta < 2*np.pi): return -np.inf
35     if not (0.5 < tau_H0 < 5.0): return -np.inf
36     if not (0.25 < Omega_m < 0.35): return -np.inf
37     if not (65.0 < H0 < 75.0): return -np.inf
38
39     # Priors gaussianos d biles (de Planck/CMB)
40     lp_Om = -0.5 * ((Omega_m - 0.315) / 0.02)**2
41     lp_H0 = -0.5 * ((H0 - 70.0) / 3.0)**2
42
43     return lp_Om + lp_H0
44
45 def log_likelihood(theta, z_obs, mu_obs, mu_err):
46     """
47     Verosimilitud: -0.5 * chi
48     """
49     A, omega, delta, tau_H0, Omega_m, H0 = theta
50
51     try:
52         # Calcular modulo de distancia te rico
53         mu_model = distance_modulus(z_obs, H0, Omega_m, A, omega, delta, tau_H0)
54
55         # Chi-cuadrado
56         chi2 = np.sum(((mu_obs - mu_model) / mu_err)**2)
57
58         # Protección contra valores infinitos
59         if not np.isfinite(chi2):
60             return -np.inf
61

```

```

62         return -0.5 * chi2
63
64     except Exception as e:
65         # Si la integraci n falla, retornar probabilidad nula
66         return -np.inf
67
68 def log_posterior(theta, z_obs, mu_obs, mu_err):
69     """Posterior = prior * likelihood"""
70     lp = log_prior(theta)
71     if not np.isfinite(lp):
72         return -np.inf
73     return lp + log_likelihood(theta, z_obs, mu_obs, mu_err)
74
75 # ===== CONFIGURACI N MCMC =====
76
77 ndim = 6 # N mero de par metros
78 nwalkers = 32 # N mero de caminantes
79 nsteps = 5000 # Pasos de MCMC
80
81 # Inicializaci n (cerca de valores esperados)
82 initial = np.array([
83     0.05, # A
84     2.5, # omega
85     np.pi, # delta
86     1.5, # tau_H0 (dentro de la banda predicha)
87     0.315, # Omega_m
88     70.0 # H0
89 ])
90
91 # Dispersi n peque a para inicializar walkers
92 p0 = [initial + 1e-3 * np.random.randn(ndim) for _ in range(nwalkers)]
93
94 # Verificar que todos los walkers tienen prior finito
95 for i, p in enumerate(p0):
96     if not np.isfinite(log_prior(p)):
97         print(f"Warning: Walker {i} tiene prior infinito. Reajustando...")
98         p0[i] = initial + 1e-4 * np.random.randn(ndim)
99
100 print("\nIniciando MCMC...")
101 print(f" - Par metros: {ndim}")
102 print(f" - Walkers: {nwalkers}")
103 print(f" - Pasos: {nsteps}")
104 print(f" - Tiempo estimado: 30-60 minutos\n")
105
106 # ===== EJECUTAR MCMC =====
107
108 # Opci n 1: Sin paralelizaci n (m s simple, m s lento)
109 sampler = emcee.EnsembleSampler(
110     nwalkers, ndim, log_posterior,
111     args=(z_obs, mu_obs, mu_err)
112 )
113
114 # Opci n 2: Con paralelizaci n (comentar l nea anterior, descomentar estas)
115 # with Pool() as pool:
116 #     sampler = emcee.EnsembleSampler(
117 #         nwalkers, ndim, log_posterior,
118 #         args=(z_obs, mu_obs, mu_err),
119 #         pool=pool
120 #     )
121 #     sampler.run_mcmc(p0, nsteps, progress=True)
122
123 start_time = time.time()
124 sampler.run_mcmc(p0, nsteps, progress=True)

```

```

125 elapsed = time.time() - start_time
126
127 print(f"\nMCMC completado en {elapsed/60:.1f} minutos")
128
129 # ===== AN LISIS DE CONVERGENCIA =====
130
131 # Descartar burn-in
132 discard = 2000
133 thin = 10
134 samples = sampler.get_chain(discard=discard, thin=thin, flat=True)
135
136 # Aceptaci n promedio (objetivo: 0.2-0.5)
137 acc_frac = np.mean(sampler.acceptance_fraction)
138 print(f"\nFracci n de aceptaci n promedio: {acc_frac:.3f}")
139
140 if acc_frac < 0.1:
141     print("      Baja aceptaci n. Considere ajustar propuesta.")
142 elif acc_frac > 0.6:
143     print("      Alta aceptaci n. Considere aumentar dispersi n.")
144 else:
145     print("      Aceptaci n en rango   ptime ")
146
147 # Autocorrelaci n
148 try:
149     tau = sampler.get_autocorr_time(discard=discard, quiet=True)
150     print(f"\nTiempo de autocorrelaci n: {np.mean(tau):.1f} pasos")
151 except:
152     print("\n      No se pudo calcular autocorrelaci n (cadenas cortas)")
153
154 # Guardar samples
155 np.save('mcmc_samples.npy', samples)
156 print("\nSamples guardados: mcmc_samples.npy")
157
158 # ===== VISUALIZACI N =====
159
160 labels = [r'$A$', r'$\omega$', r'$\delta$', r'$\tau_{H_0}$',
161           r'$\Omega_m$', r'$H_0$']
162 truths = None # No tenemos valores "verdaderos"
163
164 # Corner plot
165 fig = corner.corner(
166     samples, labels=labels,
167     quantiles=[0.16, 0.5, 0.84],
168     show_titles=True,
169     title_kwargs={"fontsize": 12},
170     truths=truths
171 )
172 plt.savefig('posteriors_corner.png', dpi=300, bbox_inches='tight')
173 print("Corner plot guardado: posteriors_corner.png")
174
175 # Imprimir resultados
176 print("\n" + "="*60)
177 print("RESULTADOS: Percentiles [16, 50, 84]")
178 print("="*60)
179
180 for i, label in enumerate(labels):
181     mcmc = np.percentile(samples[:, i], [16, 50, 84])
182     q = np.diff(mcmc)
183     print(f"{label:12s} = {mcmc[1]:.4f} +{q[1]:.4f} -{q[0]:.4f}")
184
185 print("="*60)
186
187 # Extraer valores ptimos (mediana del posterior)

```

```

188 A_best, omega_best, delta_best, tau_H0_best, Omega_m_best, H0_best = \
189     np.median(samples, axis=0)
190
191 # ===== COMPARACION CON CDM =====
192
193 print("\n" + "="*60)
194 print("COMPARACION CON CDM ")
195 print("="*60)
196
197 # Calcular chi del modelo oscilatorio
198 mu_model = distance_modulus(z_obs, H0_best, Omega_m_best,
199                             A_best, omega_best, delta_best, tau_H0_best)
200 chi2_osc = np.sum(((mu_obs - mu_model) / mu_err)**2)
201
202 # Calcular chi de CDM (A=0)
203 mu_LCDM = distance_modulus_LCDM(z_obs, H0_best, Omega_m_best)
204 chi2_LCDM = np.sum(((mu_obs - mu_LCDM) / mu_err)**2)
205
206 # Grados de libertad
207 dof_osc = len(z_obs) - 6 # 6 parametros
208 dof_LCDM = len(z_obs) - 2 # Solo Omega_m, H0
209
210 # Chi reducido
211 chi2red_osc = chi2_osc / dof_osc
212 chi2red_LCDM = chi2_LCDM / dof_LCDM
213
214 print(f"\nModelo Oscilatorio:")
215 print(f"      = {chi2_osc:.1f}")
216 print(f"      /dof = {chi2red_osc:.3f}")
217
218 print(f"\n CDM (w = -1):")
219 print(f"      = {chi2_LCDM:.1f}")
220 print(f"      /dof = {chi2red_LCDM:.3f}")
221
222 #
223 delta_chi2 = chi2_LCDM - chi2_osc
224 print(f"\n      = {delta_chi2:.1f}")
225
226 if delta_chi2 > 6:
227     print("      Mejora significativa respecto a CDM (>3 )")
228 elif delta_chi2 > 2:
229     print("      ~ Mejora marginal (~2 )")
230 else:
231     print("      Sin mejora significativa")
232
233 # BIC (Bayesian Information Criterion)
234 n = len(z_obs)
235 BIC_osc = chi2_osc + 6 * np.log(n)
236 BIC_LCDM = chi2_LCDM + 2 * np.log(n)
237 delta_BIC = BIC_LCDM - BIC_osc
238
239 print(f"\n BIC = {delta_BIC:.1f}")
240
241 if delta_BIC < -10:
242     print("      Evidencia muy fuerte a favor del modelo oscilatorio")
243 elif delta_BIC < -6:
244     print("      Evidencia fuerte")
245 elif delta_BIC < -2:
246     print("      ~ Evidencia debil")
247 else:
248     print("      Sin evidencia ( CDM preferido)")
249
250 print("="*60)

```

```

251
252 # ===== GR FICO DE RESIDUOS =====
253
254 fig, axes = plt.subplots(2, 1, figsize=(10, 8), sharex=True)
255
256 # Panel superior: Diagrama de Hubble
257 axes[0].errorbar(z_obs, mu_obs, yerr=mu_err, fmt='o',
258                 markersize=2, alpha=0.3, label='Pantheon+')
259 axes[0].plot(z_obs, mu_model, 'r--', linewidth=2,
260             label='Modelo Oscilatorio')
261 axes[0].plot(z_obs, mu_LCDM, 'b--', linewidth=2, alpha=0.7,
262             label=r'$\Lambda$CDM')
263 axes[0].set_ylabel(r'$\mu$ (mag)', fontsize=12)
264 axes[0].legend(fontsize=10)
265 axes[0].grid(alpha=0.3)
266
267 # Panel inferior: Residuos
268 residuals_osc = mu_obs - mu_model
269 residuals_LCDM = mu_obs - mu_LCDM
270
271 axes[1].errorbar(z_obs, residuals_osc, yerr=mu_err, fmt='o',
272                 markersize=2, alpha=0.5, color='red',
273                 label='Residuos (Oscilatorio)')
274 axes[1].errorbar(z_obs, residuals_LCDM, yerr=mu_err, fmt='o',
275                 markersize=2, alpha=0.3, color='blue',
276                 label=r'Residuos ($\Lambda$CDM)')
277 axes[1].axhline(0, color='k', linestyle='--', linewidth=1)
278 axes[1].set_xlabel('Redshift $z$', fontsize=12)
279 axes[1].set_ylabel(r'$\Delta\mu$ (mag)', fontsize=12)
280 axes[1].legend(fontsize=10)
281 axes[1].grid(alpha=0.3)
282
283 plt.tight_layout()
284 plt.savefig('residuals_comparison.png', dpi=300, bbox_inches='tight')
285 print("\nGr fico de residuos guardado: residuals_comparison.png")
286
287 print("\n An lisis completo. Revise los gr ficos generados.")

```

Listing 3: 3inference.py: MCMC para ajuste de parámetros

## 2.4 Instrucciones de Ejecución

### 2.4.1 Instalación de Dependencias

Crear archivo `requirements.txt`:

```
numpy>=1.21.0
scipy>=1.7.0
matplotlib>=3.4.0
pandas>=1.3.0
emcee>=3.1.0
corner>=2.2.0
h5py>=3.1.0
```

Instalar en terminal:

```
pip install -r requirements.txt
```

### 2.4.2 Ejecución Secuencial

```
# Paso 1: Descargar datos (~2 minutos)
python 1_download_data.py

# Verificar que se creó : pantheon_clean.npz
ls -lh *.npz

# Paso 2: (Opcional) Verificar modelo sin inferencia
python -c "from model import *; print('Modelo OK')"
```

# Paso 3: Ejecutar MCMC (30-60 minutos)

```
python 3_inference.py
```

# Outputs esperados:

- # - `mcmc_samples.npy` (samples del posterior)
- # - `posteriors_corner.png` (corner plot)
- # - `residuals_comparison.png` (comparación con CDM)

### 2.4.3 Monitoreo de Progreso

Durante la ejecución de `3_inference.py`, verá:

```
Iniciando MCMC...
- Parámetros: 6
- Walkers: 32
- Pasos: 5000
- Tiempo estimado: 30-60 minutos

100%|          | 5000/5000
[28:34<00:00, 2.92it/s]

MCMC completado en 28.6 minutos
Fracción de aceptación promedio: 0.347
Aceptación en rango ptimo
```

### 3 Fase 2: Interpretación de Resultados (1 semana)

#### 3.1 Criterios de Éxito Pre-Registrados

##### Señal Prometedora Si:

1. **Amplitud significativa:**  $A > 0.02$  con credibilidad  $> 68\%$  (percentil 16 no incluye 0)
2. **Memoria en banda predicha:**  $\tau H_0 \in [0.5, 3.5]$  con posterior que excluye  $\tau \rightarrow 0$
3. **Mejora estadística:**  $\Delta\text{BIC} < -6$  comparado con  $\Lambda\text{CDM}$
4. **Residuos sistemáticos reducidos:** Estructura en  $z \in [0.5, 1.5]$  capturada por el modelo
5. **Convergencia MCMC:** Fracción de aceptación en  $[0.2, 0.5]$  y  $\hat{R} < 1.1$  (si se calculan múltiples cadenas)

##### Refutación Si:

1. **Amplitud nula:** Posterior de  $A$  centrado en  $A < 0.01$  con  $< 16\%$  de probabilidad para  $A > 0.02$
2. **Memoria fuera de banda:**  $\tau H_0$  óptimo consistentemente  $< 0.1$  o  $> 10$
3. **Sin mejora estadística:**  $\Delta\text{BIC} > 10$  (evidencia fuerte contra el modelo)
4.  **$\chi^2$  peor que  $\Lambda\text{CDM}$ :**  $\chi^2_{\text{osc}} > \chi^2_{\Lambda\text{CDM}} + 6$  (penalización por parámetros extra no justificada)
5. **No convergencia:** MCMC no alcanza distribución estacionaria después de 10,000 pasos

#### 3.2 Checklist de Análisis

##### Convergencia verificada:

- Fracción de aceptación entre 0.2 y 0.5
- Tiempo de autocorrelación  $\tau_{\text{auto}} < N_{\text{steps}}/50$
- Traceplot muestra mezcla adecuada (sin tendencias sistemáticas)

##### Posteriors inspeccionados:

- Corner plot sin degeneraciones patológicas
- Distribuciones unimodales (o bimodales justificables)
- Priors no dominan posteriors

##### Comparación con $\Lambda\text{CDM}$ :

- $\Delta\chi^2$ ,  $\Delta\text{BIC}$ ,  $\Delta\text{AIC}$  calculados
- Gráfico de residuos muestra mejora visual
- Significancia estadística reportada honestamente

##### Tests de robustez:

- Rerun con diferentes semillas aleatorias (convergencia consistente)
- Variar priors en  $\pm 50\%$  (posteriors estables)
- Excluir bins de redshift (resultados no dominados por outliers)

**Documentación completa:**

- Código y datos disponibles públicamente (GitHub/Zenodo)
- Figuras con leyendas explicativas
- Tabla de parámetros con errores simétricos y asimétricos



### 3.3 Tabla de Resultados Esperados (Template)

Completar esta tabla después del análisis:

Parámetro	Mediana	Error -	Error +
$A$	_____	_____	_____
$\omega$	_____	_____	_____
$\delta$	_____	_____	_____
$\tau H_0$	_____	_____	_____
$\Omega_m$	_____	_____	_____
$H_0$	_____	_____	_____
$\chi^2_{\text{osc}}$	_____		—
$\chi^2_{\Lambda\text{CDM}}$	_____		—
$\Delta\chi^2$	_____		—
$\Delta\text{BIC}$	_____		—

Table 3: Plantilla para reportar resultados de la inferencia bayesiana

### 3.4 Interpretación de $\Delta\text{BIC}$

Basado en el criterio de Kass & Raftery (1995):

$\Delta\text{BIC}$	Evidencia contra $\Lambda\text{CDM}$	Acción
$< -10$	Muy fuerte	Pre-print inmediato
$-10$ a $-6$	Fuerte	Validar con tests de robustez
$-6$ a $-2$	Débil	Análisis adicional necesario
$-2$ a $2$	No concluyente	Null result (reportar igual)
$> 2$	Evidencia contra modelo	Refutación, buscar causas

Table 4: Guía de interpretación del criterio de información bayesiano

## 4 Fase 3: Pre-Print y Llamado a la Comunidad

### 4.1 Condición para Publicación

#### Regla de Oro

##### Publicar solo si:

- $\Delta\text{BIC} < -2$  (evidencia al menos débil), O
- Resultado negativo claro y pedagógico que aporte a la literatura

##### NO publicar si:

- Convergencia MCMC dudosa
- Resultados sensibles a elecciones arbitrarias (priors, bins, outliers)
- No se puede reproducir con diferentes códigos/implementaciones

### 4.2 Estructura del Pre-Print (arXiv)

#### 4.2.1 Template de Abstract

```
\begin{abstract}
We test a stochastic cosmological model predicting oscillations
in the dark energy equation of state  $w(z)$  with characteristic
memory timescale  $\tau_{H_0} \sim 2$ . Bayesian analysis of 1701
Type Ia supernovae from Pantheon+ yields:

 $A = 0.XX \pm 0.YY \quad (\text{oscillation amplitude})$ 
 $\tau_{H_0} = Z.ZZ \pm 0.WW \quad (\text{memory parameter})$ 

 $\Delta\text{BIC} = -X.X$  relative to  $\Lambda\text{CDM}$ , suggesting
[weak/moderate/strong] evidence for temporal structure in dark
energy. We provide open-source code and invite independent
validation using BAO, CMB, and weak lensing datasets.

\textbf{Data/Code:} \url{https://github.com/USERNAME/REPO}
\end{abstract}
```

#### 4.2.2 Secciones Requeridas

##### 1. Introduction (1-1.5 páginas)

- Contexto: problema de la energía oscura
- Motivación teórica: modelo de Krein + ruido OU
- Predicción central:  $\tau H_0 \in [0.5, 3.5]$

##### 2. Theoretical Framework (1 página)

- Ecuación (1) con justificación física
- Conexión con temperatura de Gibbons-Hawking
- Diferencias con modelos fenomenológicos (CPL, etc.)

##### 3. Data and Methods (1.5 páginas)

- Descripción de Pantheon+

- Modelo estadístico (Ec. Friedmann + inferencia bayesiana)
- Priors justificados
- Implementación MCMC (emcee)

#### 4. **Results** (2 páginas)

- Tabla de posteriors
- Corner plot
- Comparación con  $\Lambda$ CDM
- Gráfico de residuos

#### 5. **Discussion** (1-1.5 páginas)

- Interpretación física de  $\tau H_0$  encontrado
- Limitaciones del análisis (solo SNe Ia, sin covarianzas completas)
- Comparación con otros trabajos sobre DE oscilante

#### 6. **Conclusions** (0.5 páginas)

- Resumen de hallazgo principal
- Llamado explícito a validación independiente
- Tests adicionales propuestos

#### 7. **Acknowledgments**

- Agradecer a comunidad open-source (emcee, numpy, etc.)
- Mencionar que es investigación independiente sin funding

#### 8. **Data Availability**

- Enlace a GitHub con código completo
- DOI de Zenodo (archivar versión estable)
- Instrucciones de reproducción

### 4.3 Estrategia de Comunicación

#### 4.3.1 Paso 1: Pre-Registro en OSF (Open Science Framework)

Antes de analizar datos adicionales (BAO, CMB):

1. Ir a <https://osf.io> y crear cuenta
2. Crear proyecto: "Validation of Stochastic Cosmology Model"
3. Subir:
  - Este documento (plan de validación)
  - Código de análisis (1\_download\_data.py, etc.)
  - Resultados preliminares de Pantheon+
  - Declaración de hipótesis pre-registradas
4. Hacer público el proyecto (obtener DOI)
5. Incluir DOI en el paper de arXiv

**Propósito:** Demostrar que las predicciones no fueron ajustadas *a posteriori* para encajar con los datos.

#### 4.3.2 Paso 2: Difusión Científica Calibrada

A. Tweet inicial (si tienes cuenta de Twitter/X):

```
New preprint on arXiv: "Observational Evidence for
Oscillatory Dark Energy" (arXiv:XXXX.XXXXX)

We find [X] evidence for memory-driven oscillations in  $w(z)$ 
using Pantheon+ SNe Ia. Key result:  $H = [value] [error]$ ,
consistent with theoretical prediction from stochastic
cosmology framework.
```

Thread

B. Post en r/Cosmology (Reddit):

Title: [Research] Independent analysis finds X-sigma deviation  
from CDM in Pantheon+ data - requesting expert validation

Body:

Hi r/Cosmology,

I'm an independent researcher who completed a Bayesian analysis  
of Pantheon+ supernovae testing a model where dark energy  
exhibits oscillations with finite memory timescale.

Key findings:

- Amplitude  $A = [value] [error]$
- Memory parameter  $H = [value] [error]$
- BIC =  $[value]$  vs CDM

Full paper: arXiv:XXXX.XXXXX

Code: [github.com/\[user\]/\[repo\]](https://github.com/[user]/[repo])

I'm aware of the limitations (only SNe Ia, no covariances,  
independent researcher) and would greatly appreciate:

1. Technical feedback on the methodology
2. Suggestions for additional validation tests

### 3. Collaboration offers **for** BAO/CMB analysis

Committed to transparent science. Happy to discuss limitations **and** potential issues.

#### C. Emails directos a expertos (máximo 10 personas):

##### Lista sugerida de contactos:

- Dillon Brout (Pantheon+ lead)
- Ariel Goobar (SNe Ia expert)
- Dragan Huterer (dark energy phenomenology)
- Ruth Durrer (stochastic cosmology)
- Luca Amendola (modified gravity/DE)
- Éric Aubourg (DESI/BAO)
- Pengjie Zhang (weak lensing + DE)

##### Template de email:

Subject: Request **for** feedback: evidence **for** oscillatory DE **in** Pantheon+ analysis

Dear Dr. [Last Name],

I'm writing to share a recent preprint that may be of interest given your expertise in [specific area]. As an independent researcher, I completed a Bayesian analysis of Pantheon+ supernovae testing a stochastic cosmology model predicting oscillations in  $w(z)$ .

Key result: [X] evidence for finite memory timescale  
H [value], with BIC = [value] vs CDM .

Paper: arXiv:XXXX.XXXXX

Code: [GitHub URL]

I recognize the limitations of using only SNe Ia data and would greatly value your expert perspective on:

1. Potential systematic effects I may have missed
2. Suggestions for validation with BAO/CMB
3. Whether this result warrants further investigation

I'm committed to rigorous science **and** will publicly update **if** flaws are found. Thank you **for** considering this work.

Best regards,

[Your Name]

[Your affiliation/independent]

[Contact info]

## 4.4 Respuestas Esperables y Cómo Manejarlas

### 4.4.1 Escenario A: Silencio (más probable)

Qué hacer:

- No interpretar como rechazo
- Continuar con análisis de BAO/CMB por cuenta propia
- Enviar follow-up después de 2-3 meses con resultados adicionales
- Someter a revista (Universe, Galaxies, IJMPD)

### 4.4.2 Escenario B: Crítica técnica constructiva

Respuesta ideal:

- Agradecer públicamente
- Correr tests sugeridos inmediatamente
- Actualizar preprint en arXiv v2 con correcciones
- Citar al crítico en acknowledgments
- Si la crítica invalida resultados: publicar "retraction note" honesto

Ejemplo de respuesta pública:

```
"Thank you @ExpertX for the feedback. You're absolutely right about [issue]. I'm re-running the analysis with [correction] and will post updated results within 48h. This is exactly why I made the code public."
```

### 4.4.3 Escenario C: Interés genuino / oferta de colaboración

Cómo responder:

- Aceptar con entusiasmo pero mantener co-autoría principal
- Establecer roles claros desde el inicio
- Crear repositorio compartido (GitHub con permisos)
- Acordar plan de publicación (qué journal, timeline)
- Documento de "collaboration agreement" informal pero explícito

Red flags (rechazar colaboración si):

- Te piden entregar datos/código sin reconocimiento
- Quieren cambiar la autoría retroactivamente
- No están dispuestos a firmar acuerdo de colaboración
- Presionan para publicar sin tests de robustez

## 5 Fase 4: Validación Extendida (Opcional, 3-6 meses)

### 5.1 Si los Resultados de Pantheon+ son Prometedores

Análisis adicionales recomendados (en orden de prioridad):

#### 5.1.1 A. Oscilaciones Acústicas Bariónicas (BAO)

Datasets públicos:

- SDSS BOSS DR12: <https://www.sdss.org/dr12/>
- DESI Early Data Release: <https://data.desi.lbl.gov/>

Observable clave:

$$D_V(z) = \left[ (1+z)^2 d_A^2(z) \frac{cz}{H(z)} \right]^{1/3} \quad (5)$$

Código base:

```

1 def DV_theory(z, H0, Omega_m, A, omega, delta, tau_H0):
2     """Volumen de dilataci n para BAO"""
3     dA = luminosity_distance(z, H0, Omega_m, A, omega,
4                             delta, tau_H0) / (1+z)
5     H = H0 * np.sqrt(H_squared(z, H0, Omega_m, A, omega,
6                             delta, tau_H0))
7     c = 299792.458 # km/s
8     return ((1+z)**2 * dA**2 * c*z / H)**(1/3)
9
10 # Cargar datos BAO
11 # z_BAO, DV_obs, DV_err = load_BAO_data()
12
13 # Agregar a likelihood:
14 # chi2_BAO = sum(((DV_obs - DV_theory(z_BAO, ...))/DV_err)**2)
15 # chi2_total = chi2_SNe + chi2_BAO

```

Listing 4: bao-analysis.py (esqueleto)

#### 5.1.2 B. Efecto Sachs-Wolfe Integrado Tardío (ISW)

Predicción del modelo:

Las oscilaciones en  $w(z)$  modulan el potencial gravitacional  $\Phi(z)$ , afectando el CMB via:

$$\frac{\Delta T}{T} \Big|_{\text{ISW}} = 2 \int_0^{z^*} \frac{d\Phi}{dt} \frac{dt}{dz} dz \quad (6)$$

**Test:** Correlación cruzada CMB  $\times$  LSS debería mostrar estructura oscilatoria a  $z < 2$ .

Código conceptual:

```

1 # Requiere CLASS/CAMB para c lculo riguroso
2 # Aqu : estimaci n de orden de magnitud
3
4 def ISW_amplitude_ratio(z, A, omega, tau_H0):
5     """Ratio de amplitud ISW: modelo / CDM """
6     # Simplificaci n: ISW d /dt
7     w = w_oscillatory(z, A, omega, 0, tau_H0)
8     dwdz = np.gradient(w, z)
9     # Factor de oscilaci n
10    return 1 + 0.1 * A * np.abs(dwdz) # orden de magnitud
11
12 # Comparar con mapas Planck PR4
13 # Requiere: healpy, analisis de cross-correlation

```

Listing 5: isw-prediction.py (conceptual)

### 5.1.3 C. Tasa de Crecimiento de Estructura ( $f\sigma_8$ )

**Ecuación diferencial:**

$$\frac{d^2\delta}{dt^2} + 2H\frac{d\delta}{dt} = 4\pi G\bar{\rho}_m\delta\left(1 + \frac{\Omega_\Lambda(1+w)}{\Omega_m}\right) \quad (7)$$

**Dataset:** BOSS, eBOSS, 6dFGS

**Predicción:** Desviación de 2-5% respecto a  $\Lambda$ CDM en  $z \in [0.5, 1.5]$  si  $A \sim 0.05$ .



## 5.2 Protocolo de Validación Multi-Dataset

Dataset	$\chi^2$ contrib.	Nparam shared	Prioridad
Pantheon+ (SNe Ia)	$\chi_{\text{SNe}}^2$	6	Alta
BOSS BAO	$\chi_{\text{BAO}}^2$	4 ( $A, \omega, \tau H_0, \Omega_m$ )	Alta
Planck CMB	$\chi_{\text{CMB}}^2$	2 ( $\Omega_m, H_0$ )	Media
$f\sigma_8$ (BOSS)	$\chi_{f\sigma_8}^2$	5	Media
ISW $\times$ LSS	$\chi_{\text{ISW}}^2$	3	Baja

Table 5: Contribuciones al  $\chi^2$  total en análisis conjunto

### Likelihood conjunta:

$$\chi_{\text{total}}^2 = \chi_{\text{SNe}}^2 + \chi_{\text{BAO}}^2 + \chi_{\text{CMB}}^2 + \chi_{f\sigma_8}^2 + \chi_{\text{ISW}}^2 \quad (8)$$

### Criterio de validación fuerte:

- $\Delta\text{BIC}_{\text{total}} < -10$  (evidencia muy fuerte)
- Parámetros  $\{A, \omega, \tau H_0\}$  consistentes entre datasets (overlapping posteriors)
- Sin tensiones internas ( $\chi_{\text{internal}}^2/\text{dof} < 1.5$ )

## 5.3 Recursos Computacionales Necesarios

Análisis	Tiempo CPU	RAM requerida
Pantheon+ solo	30-60 min	4 GB
Pantheon+ + BAO	1-2 horas	6 GB
+ CMB (simplified)	3-5 horas	8 GB
+ $f\sigma_8$	4-6 horas	8 GB
+ ISW (full)	10-20 horas	16 GB
<b>Total (análisis completo)</b>	<b>20-30 horas</b>	<b>16 GB</b>

Table 6: Recursos computacionales para validación extendida

### Alternativas si no tienes recursos:

- Google Colab Pro (\$10/mes): 25 GB RAM, GPU gratis
- AWS Free Tier: 750 horas/mes durante 1 año
- Colaboración con grupo que tenga cluster universitario

## 6 Timeline y Milestones

### 6.1 Cronograma Realista

Semana	Fase	Tarea principal
1	Setup	Instalar dependencias, descargar datos, probar código
2-3	MCMC	Correr análisis Pantheon+, verificar convergencia
4	Análisis	Calcular posteriors, comparar con $\Lambda$ CDM, plots
5	Robustez	Tests de sensibilidad, verificar con diferentes priors
6	Escritura	Draft del paper (intro, métodos, resultados)
7	Pre-registro	Subir a OSF, preparar GitHub público
8	Publicación	Someter a arXiv, difusión inicial
9-12	Validación (opcional)	BAO, CMB (si hay señal fuerte)
13-16	Colaboración (condicional)	Trabajar con expertos que respondan

Table 7: Timeline de 4 meses desde inicio hasta pre-print

### 6.2 Milestones de Decisión

#### Checkpoint 1: Semana 3

**Pregunta:** ¿MCMC convergió adecuadamente?

- Sí → Continuar con análisis
- No → Ajustar propuesta, aumentar nsteps, revisar priors

#### Checkpoint 2: Semana 4

**Pregunta:** ¿Hay señal ( $\Delta\text{BIC} < -2$ )?

- Sí → Proceder con tests de robustez y escritura
- No → Escribir null result honesto, publicar igual

#### Checkpoint 3: Semana 8

**Pregunta:** ¿Expertos respondieron positivamente?

- Sí → Iniciar colaboración para análisis extendido
- No → Continuar solo, planear análisis BAO propio

## 7 Recursos y Soporte

### 7.1 Recursos Educativos

#### 7.1.1 Python Científico (si necesitas nivelación)

- Numpy/Scipy basics: <https://scipy-lectures.org/>
- Matplotlib tutorials: <https://matplotlib.org/stable/tutorials/index.html>
- Tutorial emcee (oficial): <https://emcee.readthedocs.io/en/stable/tutorials/line/>

#### 7.1.2 Inferencia Bayesiana

- Trotta (2008): "Bayes in the sky: Bayesian inference and model selection in cosmology" <https://arxiv.org/abs/0803.4089>
- Hogg et al. (2010): "Data analysis recipes: Fitting a model to data" <https://arxiv.org/abs/1008.4686>
- Curso online (gratis): Coursera - "Bayesian Statistics: From Concept to Data Analysis"

#### 7.1.3 Cosmología Observacional

- Libro recomendado: Weinberg "Cosmology" (2008) - Capítulos sobre SNe Ia y BAO
- Review moderno: "A Hitchhiker's Guide to Cosmology" (arXiv:2010.13160)
- Pantheon+ paper original: Brout et al. (2022), ApJ 938, 110

### 7.2 Comunidades de Soporte

- Stack Overflow (Python): Tag: [scipy] [emcee] [mcmc]
- Physics Stack Exchange: Tag: [cosmology] [bayesian-inference]
- Reddit:
  - r/Cosmology (papers serios)
  - r/AskPhysics (conceptual)
  - r/learnpython (coding)
- GitHub issues: Repositorio emcee tiene comunidad activa

### 7.3 Herramientas Adicionales

Herramienta	Uso	URL
GetDist	Análisis posterior	<a href="https://getdist.readthedocs.io/">https://getdist.readthedocs.io/</a>
Arviz	Diagnósticos MCMC	<a href="https://arviz-devs.github.io/">https://arviz-devs.github.io/</a>
ChainConsumer	Plots de cadenas	<a href="https://samreay.github.io/ChainConsumer/">https://samreay.github.io/ChainConsumer/</a>
Jupyter Notebook	Exploración interactiva	<a href="https://jupyter.org/">https://jupyter.org/</a>
Overleaf	Escritura LaTeX	<a href="https://www.overleaf.com/">https://www.overleaf.com/</a>
Zenodo	Archivo permanente	<a href="https://zenodo.org/">https://zenodo.org/</a>

Table 8: Herramientas complementarias recomendadas

## 8 Conclusión y Siguiendo Paso

### 8.1 Resumen Ejecutivo

Este plan proporciona un protocolo completo para:

1. **Validar empíricamente** el modelo de cosmología estocástica usando datos públicos
2. **Mantener transparencia radical** mediante pre-registro y código abierto
3. **Comunicar resultados honestamente**, sean positivos o negativos
4. **Invitar colaboración** con la comunidad científica

### 8.2 El Próximo Paso es Tuyo

#### Acción Inmediata

##### En las próximas 48 horas:

1. Crear carpeta de trabajo: `cosmologia_validacion/`
2. Instalar dependencias: `pip install -r requirements.txt`
3. Ejecutar: `python 1_download_data.py`
4. Verificar que se creó: `pantheon_clean.npz`

##### Si encuentras errores:

- Revisa versiones de Python (debe ser 3.8+)
- Verifica conexión a internet (para descargar datos)
- Consulta el error en Stack Overflow o contáctame

### 8.3 Compromiso del Autor de este Plan

Como creador de este protocolo, me comprometo a:

- Responder dudas técnicas por email ([info@impulses.online](mailto:info@impulses.online))
- Actualizar este documento si surgen mejoras metodológicas
- Celebrar públicamente cualquier resultado (positivo o negativo) que uses este plan
- Ofrecer co-autoría si contribuyo significativamente al análisis

## 8.4 Nota Final

*"La ciencia no avanza por genios solitarios, sino por individuos honestos dispuestos a equivocarse en público."*

Este plan asume que prefieres **fallar de forma reproducible** que **acertar de forma opaca**. Esa es la actitud correcta.

Si este análisis funciona, habrás demostrado que un investigador independiente puede contribuir a cosmología observacional de frontera. Si no funciona, habrás aprendido inferencia bayesiana, cosmología computacional, y habrás agregado un null result valioso a la literatura.

**Ambos resultados son victorias.**

**¡Adelante!**

## A Apéndice A: Troubleshooting Común

### A.1 Error: "No module named 'emcee'"

**Causa:** Dependencia no instalada

**Solución:**

```
pip install emcee corner
```

### A.2 Error: "HTTPError 404: Not Found" al descargar datos

**Causa:** URL de Pantheon+ cambió

**Solución:**

1. Ir a <https://github.com/PantheonPlusSHOES/DataRelease>
2. Navegar a Pantheon+\_Data/2\_DISTANCES\_AND\_COVAR/
3. Copiar URL actualizada del archivo Pantheon+SHOES.dat
4. Reemplazar en 1\_download\_data.py

### A.3 MCMC muy lento (¿2 horas)

**Posibles causas:**

- Integración numérica muy fina (disminuir puntos en grilla)
- CPU lento (considerar Google Colab)
- Priors muy amplios (estrechar rangos razonables)

**Soluciones:**

```
1 # En model.py, cambiar:
2 z_array = np.linspace(0, z_max, 150) # era 200
3 # Reducir l mite de integraci n:
4 quad(integrand, 0, zi, limit=50) # era 100
```

Listing 6: Optimización del código

### A.4 Posteriors "raros" (bimodales, degenerados)

**Causa común:** Priors demasiado amplios o problema mal condicionado

**Diagnóstico:**

1. Revisar traceplot (¿walkers convergieron?)
2. Verificar acceptance fraction (debe ser 0.2-0.5)
3. Probar con priors más estrechos en fase exploratoria

## B Apéndice B: Glosario de Términos

**MCMC (Markov Chain Monte Carlo):** Método de muestreo para explorar distribuciones de probabilidad complejas

**Posterior:** Distribución de probabilidad de parámetros dados los datos

**Prior:** Conocimiento previo sobre parámetros antes de ver datos

**Likelihood:** Probabilidad de observar los datos dados los parámetros

**BIC (Bayesian Information Criterion):** Métrica de selección de modelo que penaliza complejidad

**Corner plot:** Visualización de posteriors marginales y correlaciones

**Burn-in:** Pasos iniciales de MCMC que se descartan (no representativos del posterior)

**Acceptance fraction:** Proporción de pasos MCMC aceptados (indica eficiencia)

$\Lambda$ CDM: Modelo cosmológico estándar (materia + constante cosmológica)

**Redshift (z):** Corrimiento al rojo, proxy de distancia/tiempo en cosmología

## C Apéndice C: Checklist Pre-Publicación

Antes de someter a arXiv, verificar:

Código ejecuta sin errores en máquina limpia

README.md con instrucciones claras en GitHub

requirements.txt actualizado

Todos los plots tienen leyendas y labels

Errores reportados con intervalos de confianza apropiados

Comparación con  $\Lambda$ CDM incluida

Limitaciones reconocidas explícitamente

Data availability statement presente

Referencias citadas correctamente (BibTeX)

Abstract < 250 palabras

Paper < 10 páginas (sin contar referencias)

Co-autores (si aplica) han aprobado versión final

DOI de Zenodo obtenido para código/datos

Pre-registro OSF completado

---

**Fin del Plan de Validación Empírica**

Versión 1.0 — Noviembre 2025

*Licencia: CC0 (Dominio Público)*

[https:](https://github.com/cisnerosmusic/Del-Espacio-de-Hilbert-a-la-Cosmologia-Estocastica)

[//github.com/cisnerosmusic/Del-Espacio-de-Hilbert-a-la-Cosmologia-Estocastica](https://github.com/cisnerosmusic/Del-Espacio-de-Hilbert-a-la-Cosmologia-Estocastica)