Arquitectura lógica propuesta para RadarPremios

Arranquemos desde lo honesto: quieres un sistema que no se engañe, que haga lo que dice, y que te lo demuestre con trazas, métricas y reportes. La propuesta refuerza lo que ya tienes, cierra los huecos típicos (datos, reproducibilidad, drift, sobreadaptación) y te da palancas para mejorar sin romper.

Puntos de atención y riesgos clave

- **Calidad de datos:** scrapers con fallas intermitentes, formatos cambiantes, duplicados y lag de publicación.
- **Reproducibilidad real:** semillas, PRNG, versiones de señales y modelo deben fijar resultados byte a byte.
- **Sobreadaptación y señales frágiles:** hot/cold y reglas de patrón pueden "lucir bien" y no generalizar.
- **Dependencias temporales:** calendario heterogéneo (días/horas por juego), T+1, festivos y TZ local.
- **Idempotencia y concurrencia:** evitar dobles ejecuciones por día/juego; re-ejecuciones seguras.
- **Deriva y estabilidad:** distribución de dígitos/signos cambia; señales pierden potencia con el tiempo.
- **Observabilidad y auditoría:** necesitas ver, explicar y reconstruir cada decisión con mínimos clicks.
- **Mensajería responsable:** separar hechos vs hipótesis; avisos claros de no-garantía en todos los reportes.

Arquitectura por capas

1) Ingesta y saneamiento de datos

- **Validación cruda:**
    - Esquema mínimo por juego; rechazo suave a filas inválidas a "data/crudo/rejected" con razón.
    - Deduplicación por llave canónica:
        - 4D/Astro: juego, fecha_sorteo, número, signo (si aplica).
        - Baloto/Revancha: juego, fecha_sorteo, conjunto ordenado.
- **Depuración y normalización:**
    - Canonización de formatos (cadenas a enteros, padding de 4 dígitos, mapeo de signos).
    - Control de calidad con tests de uniformidad básicos (chi-cuadrado/KS) y "runs test" por ventana.
- **Contratos y SLAs:**

    - **Label:** status por sorteo
        - ok, partial, failed, late; logs con timestamps, URL, latencia y reintentos.
    - **Label:** idempotencia
        - constraint único por (juego, fecha_sorteo, fuente, versión_parser).

2) Capa de señales y fundamentos (constructiva y medible)

- **Catálogo de señales versionado:**
    - Hot/Cold por posición (suavizado exponencial), last-seen, Markov posicional (orden 1; opcional ajuste por AIC/BIC para orden>1), patrón (repetidos, consecutivos, suma, módulo 9, paridad), complejidad (RLE/LZ proxy).
    - Astro: + Markov de signo y co-ocurrencia dígito/signo.
    - Baloto/Revancha: frecuencias marginales/conjuntas, presencia/ausencia (Markov binario), pares/ternas, equilibrio par/impar y bajo/alto, penalización picks humanos (1–31).
- **Normalización y mezcla:**
    - Escalado a [0,1] por juego/ventana; fusión por media ponderada configurable (momentum vs estabilidad).
- **Montecarlo y controles LLN:**
    - Baseline azar para lift; intervalos por bootstrap con semilla fija.
    - Reporte de p-values y divergencia KL semanal para drift.

3) Motor de elección y reproducibilidad

- **PRNG y semilla:**
    - PCG64 con semilla determinista: make_seed(ns | juego_id | fecha_iso | model_version | signals_version).
    - Persistir: seed, ns, PRNG, hash de features, vector de pesos.
- **Ranking y muestreo:**
    - Orden total: score desc, tiebreak lexicográfico (0000…9999) y orden fijo de signo.
    - Muestreo ponderado SRSWOR (Efraimidis–Spirakis) sobre el top prefiltrado para balance exploración/explotación.
- **Coberturas y filtros:**
    - No-duplicados recientes, cuotas por paridad/alto-bajo, límite por jugador, exclusiones de políticas (ej. 0000 si penaliza).
- **Portafolio por presupuesto (opcional):**
    - Solver simple de cobertura: maximizar diversidad marginal en bandas de señales, sujeto a K y cobertura mínima.

4) Evaluación T+1, ajuste y “online learning” seguro

- **Métricas base:**
    - Hit exacto, aciertos parciales (4/3/2/1), top-k recall, lift vs azar por juego/ventana.
- **Ajuste de pesos con freno:**

    - Subir señales con lift>1 y estabilidad>umbral; bajar/desactivar las nulas/negativas.
    - Rate limit ±10%/día, consolidación semanal y rollback por versión.
- **Bandits/A-B (opcional, fase avanzada):**
    - Temperatura softmax (λ) con A/B y test de superioridad (métrica: top-k recall semanal).
    - Thompson Sampling entre perfiles de mezcla de ventanas (corto vs largo).

5) Reportes y transparencia

- **Diario (T+1 07:10):**
    - Tabla por juego: candidatos vs resultado, métricas, lift, “qué funcionó/qué falló”.
    - Tabla de señales: peso actual vs propuesto, semáforos y notas de confianza.
    - Trazabilidad: seed, PRNG, versions, features_hash, snapshot topN.
- **Semanal (Dom 08:00):**
    - Tendencias, estabilidad, drift KL, cambios aplicados, plan de la semana.
- **Mensajería responsable:**
    - Leyendas visibles: análisis estadístico, sin garantías, juego responsable, contacto de ayuda.

6) Orquestación, operación y observabilidad

- **Programación:**
    - GEN en el mismo día por juego (ventanas: 12:00/18:00 según agenda), EVAL 07:00, REPORT 07:10, semanal 08:00.
    - Idempotencia: antes de correr, consultar rp_runs; si existe (status in {ok, partial}) para (juego, fecha, kind, versions) -> no-op.
- **Concurrencia:**
    - Paralelizar por juego con colas locales; límites de CPU/RAM por job; timeout y backoff.
- **Logs y métricas:**
    - Logging estructurado JSON por etapa (context: run_id, juego, fecha, kind, versions).
    - Métricas: tiempos, tasas de éxito, lag de datos, tasa de duplicados, p95 de latencia, incidencias.
- **Alertas:**
    - Falla de ingesta por juego/día, lift < 1 sostenido 7 días, drift alto, repetidos altos en candidatos, ausencia de reportes.
- **Backups y migraciones:**
    - Backup posterior a cada ciclo completo. Migraciones versionadas (schema_version) con dry-run + verificación.

Diseño de datos y contratos reforzados

- **rp_runs:**
    - **Clave única:** (juego, target_fecha, kind, model_version, signals_version).

    - **Campos extra:** prng, seed_ns, features_hash, config_hash, started_ts, ended_ts, status_detail.
- **run_candidates / rp_predictions:**
    - **Label:** integridad
        - FK a rp_runs; signals_snapshot JSON compacto (solo señales usadas y ventana).
    - **Índices:** (juego, target_fecha, rank), (juego, combo, target_fecha).
- **rp_eval:**
    - hits_parciales JSON normalizado (por posición y conteo), recall@k, lift_baseline, pvalues (chi/KS), drift_kl.
- **signals_config / signals_history:**
    - version, window, weight, enabled, rationale (texto breve), author (auto si automático), change_ts.
- **rp_daily_report:**
    - path_html, path_csv, checksum, generated_ts, run_id_eval.

Controles de calidad y anti-fragilidad

- **Data tests automáticos:**
    - Uniformidad (chi/KS) por ventana, runs test, duplicados, valores fuera de dominio, lag de publicación.
- **Model sanity checks:**
    - Correlación score–acierto por bandas (esperable positivo), varianza de score (evitar colapso), cobertura de portafolio (no degenerado).
- **Selección constructiva:**
    - Orden total + PRNG reproducible + muestreo medible. Sin decisiones “a dedo”.
- **Validación fuera de muestra:**
    - Split por tiempo (backtesting rolling) para señales nuevas; criterio de promoción a producción con umbrales de lift y estabilidad.

Roadmap de implementación

- **Semana 1–2 (Fase 1):**
    - **Orquestación:** orchestrate_today con idempotencia y logs JSON.
    - **Señales base:** hot/cold, last-seen, patrones; generación Top-K; reporte diario.
    - **Datos:** contratos reforzados, unique keys, backup tras ciclo.
- **Semana 3–4 (Fase 2):**
    - **Markov posicional y de signo;** baseline Montecarlo; métricas de drift y p-values en reportes.
    - **Observabilidad:** métricas y alertas mínimas.
- **Semana 5–6 (Fase 3):**
    - **Ajuste de pesos online** con rate limit, versionado y rollback.
    - **Semanal:** tendencias, estabilidad, plan.
- **Semana 7+ (Fase 4):**

    - **Portafolio por presupuesto,** A/B de temperatura softmax, bandits para mezcla de ventanas.
    - **Dashboard** (lectura): métricas, KPIs, auditoría de runs.

KPIs y criterios de aceptación

- **Disponibilidad:** % sorteos con candidatos generados ≥ 99%.
- **Trazabilidad:** % runs con seed, versions, features_hash y snapshot ≥ 99.9%.
- **Calidad:** tasa de duplicados crudos < 0.5%; validaciones de uniformidad con p-values estables (sin alarmas falsas).
- **Lift:** lift vs azar > 1.0 (7 y 30 días) por juego; estabilidad de señales (r score–acierto > 0).
- **Tiempo:** GEN < 5 min/juego; EVAL+REPORT < 5 min totales.
- **Drift:** alertas de KL resueltas < 48h.

Qué te va a dar esto en la práctica

- **Confianza operacional:** cada decisión es re-ejecutable y auditable.
- **Mejora sostenida:** ajustes pequeños, medidos, con rollback.
- **Transparencia real:** reportes que explican por qué y cuánto aportó cada señal.
- **Resiliencia:** si una pieza falla, el resto no se cae.