Exactitud, abstención y confiabilidad de LLM de bajo costo para la interpretación de mensajes financieros en español rioplatense

Anónimo
Universidad / Institución
correo@institucion.ar

***Resumen*—Un asistente financiero conversacional debe transformar lenguaje cotidiano en registros estructurados sin corromper el historial del usuario. Este trabajo evalúa experimentalmente tres modelos orientados a bajo costo y latencia en sus catálogos respectivos (Gemini 3.1 Flash Lite, Gemini 3.5 Flash Lite y Mistral Small 4) sobre 140 mensajes en español rioplatense, usando el núcleo conversacional de LUKA como caso de aplicación. El diseño es pareado $140 \times k$ con un baseline determinista, Wilson IC95 %, McNemar exacto con corrección Holm-Bonferroni, y ambas condiciones evaluadas contra un *mismo* target canónico. Los resultados muestran una tensión central: los campos críticos (CFA = type∧amount∧fecha∧currency) alcanzan 0.79/0.78/0.61, pero el registro completo que exige categoría cae a 0.52/0.62/0.34. Contrario a lo esperado, inyectar la taxonomía cerrada en el prompt no produce mejoras significativas cuando la evaluación normaliza semánticamente ambas condiciones ($p \geq 0{,}17$): la brecha previa se explica por el criterio de evaluación, no por el prompt. Un experimento pareado nuevo muestra que las referencias temporales relativas degradan la resolución de fecha frente a fechas explícitas de contenido idéntico en el proveedor más económico (0.29 vs. 0.86, $p = 0{,}008$), mientras Gemini la resuelve sin degradación significativa. La estrategia de aclaración presenta alta cobertura (recall $\geq 0{,}96$) y cero persistencias inseguras en LLM, a costa de falsos positivos en Mistral (precisión 0.71). El costo calibrado con tokens reales es $1.09–2.54 por 1000 mensajes. Concluimos que estos modelos operan como capa de interpretación dentro de un pipeline con gate operacional de validación, no como autoridad final sobre el dato persistido.**

***Index Terms*—LLM, extracción estructurada, finanzas personales, español rioplatense, abstención, evaluación pareada, confiabilidad**

I. Introducción

Registrar gastos en planillas exige traducir el habla cotidiana —"gasté 8 lucas en el súper", "ayer pagué la luz"— a campos rígidos (monto, fecha, categoría). Esa fricción desalienta el registro. Los LLM permiten una interfaz lenguaje→estructura sobre canales existentes como WhatsApp [1], [2], pero en finanzas un error no es trivial: un monto o fecha mal anclados contamina el historial y decisiones posteriores.

Falta evidencia controlada sobre si modelos de bajo costo pueden sostener esa transformación en español rioplatense. Identificamos una cobertura limitada de la intersección que este uso exige: *extracción estructurada* ∩ *dominio financiero* ∩ *español rioplatense coloquial* ∩ *abstención ante datos faltantes* ∩ *múltiples operaciones por mensaje*. La literatura cubre sectores de ese espacio —extracción estructurada [6], [7], agentes conversacionales [4], [5], abstención [10], [11], NLP financiero [9]— pero no el cruce completo con validación pre-persistencia.

Usamos el término *bajo costo* en sentido operativo, no arquitectónico: los tres modelos son las opciones de menor costo/latencia de sus respectivos catálogos API comerciales (Flash-Lite en Gemini; Mistral Small 4, de 119B parámetros totales y 6.5B activos, en Mistral). No asumimos comparabilidad arquitectónica entre ellos.

Preguntas de investigación. RQ1: ¿con qué exactitud por atributo (type, amount, currency, category, fecha) opera el LLM? RQ2: ¿cuándo se abstiene correctamente ante ambigüedad y qué riesgo de persistencia insegura implica? RQ3: ¿cómo afectan la temporalidad relativa y la multioperación? RQ4: ¿cuál es el trade-off exactitud–latencia–costo?

Hipótesis. H1: `movement_type` y `amount` superan a `category`. H2: las referencias temporales *relativas* degradan la resolución de fecha frente a fechas explícitas de igual contenido (testeada con pares pareados; §V-D). H3: los mensajes multioperación presentan menor exactitud de campos críticos que los de movimiento único (testeada con Fisher exact sobre definiciones equivalentes sin categoría). H4: proveer la taxonomía cerrada en el prompt mejora categoría y registro completo (A/B).

Contribuciones. (1) Un protocolo de evaluación que separa campos críticos (CFA) del registro completo y evalúa todas las condiciones contra un mismo target canónico, evitando el confusor metodológico de cambiar prompt y criterio de evaluación simultáneamente. (2) Evidencia de que la mejora aparente por inyectar taxonomía es en gran parte un artefacto de evaluación, y de que la temporalidad relativa es un estresor real medido con pares pareados. (3) Métricas orientadas al riesgo operativo: tasa de persistencia insegura global, separación invención real vs. default de política, y consistencia semántica vs. superficial entre réplicas.

II. Antecedentes

II-A. Extracción estructurada con LLM

La salida JSON conforme a esquema se induce por prompting y se fuerza vía parámetros del provider o entrenamiento específico. Lu *et al.* proponen SchemaRL [6] para generar JSON válido; Singh *et al.* introducen el Structured Output Benchmark (SOB) [7], que separa explícitamente cumplimiento de esquema de exactitud de valores —conformidad estructural

casi perfecta con exactitud semántica sustancialmente menor—, hallazgo que replica nuestro análisis de errores. Wang *et al.* proponen STED y scoring de consistencia entre réplicas [8], base de nuestro análisis de estabilidad.

II-B. LLM en finanzas y en español

Guo *et al.* evalúan si ChatGPT es experto financiero sobre tareas de NLP financiero [9]. En español rioplatense, la coloquialidad ("lucas"=miles, "palo"=millón, elisión de moneda) y la morfología verbal añaden ambigüedad que los benchmarks en inglés no capturan.

II-C. Interfaces conversacionales y despliegue en mensajería

Los agentes conversacionales tool-augmented operan sobre WhatsApp y combinan multi-turno con invocación de herramientas [4]. Roller *et al.* [5] sistematizan problemas abiertos; Eltigani *et al.* documentan WaLLM [1]; Brito *et al.* un sistema multi-agente para educación financiera [2]. La tesis de este trabajo es que, aun con esos avances, la persistencia financiera requiere garantías adicionales.

II-D. Abstención y calibración epistémica

Baidya propone PassiveQA, marco de tres acciones (responder/preguntar/abstenerse) [10]; Baan *et al.* estudian cuándo *clarificar, abstenerse o responder* [11]. Adoptamos esa matriz: el modelo debe dejar en `null` lo faltante y preguntar en `reply_text` sin inventar.

III. METODOLOGÍA

III-A. Definición de la tarea

Entrada: texto libre en español rioplatense enviado por WhatsApp. Salida: objeto JSON con `intent`, `movement_type` (ingreso/egreso/null), `amount` (float), `currency` (ISO 4217), `category`, `description`, `fecha` (YYYY-MM-DD o null), `reply_text` y, para cardinalidad > 1, `movements[]`. La fecha relativa ("ayer", "el martes") debe resolverse contra la fecha provista en contexto.

III-B. Dataset v1.2

140 mensajes balanceados: 5 tipologías $\times$ 28 (simple, coloquial, temporal, ambiguo, múltiples operaciones). Ejemplos: "Gasté $5000 en comida" (S001), "Me dejé 15 lucas en el súper" (C001), "Ayer pagué 30 mil de luz" (T001 $\rightarrow$2026-08-17), "Gasté en la farmacia" (A002), "Hoy cobré 50 mil de un trabajo y gasté 10 mil en transporte" (P001). Moneda controlada a ARS. `base_date`=2026-08-18, timezone de Argentina implícito. Cambios de versión: (i) v1.1 corrigió `expects_clarification` de A001 (con monto y tipo no debe pedir aclaración); (ii) v1.2 reescribió A004/A007/A021 porque sus textos aparecían *literalmente* entre los ejemplos de `prompt.md` (contaminación benchmark/prompt detectada por búsqueda de prefijos); las reescrituras preservan el GT semántico y una verificación posterior no encuentra solapamientos (0/140). La construcción fue por autor único sin doble anotación; declaramos explícitamente el sesgo potencial de *benchmark diseñado por el autor del prompt* (§IX).

III-C. Sub-dataset de pares temporales (H2)

El dataset principal no contiene mensajes con fecha explícita, por lo que H2 no era testeable sobre él. Creamos `dataset_temporal_pairs.json`: 14 pares con contenido semántico idéntico (misma entidad, monto, categoría y GT completo) donde la única diferencia es la forma de la referencia temporal —relativa ("ayer", "anteayer", "el martes", "la semana pasada") vs. explícita ("el 17 de agosto")—. GT de fecha resuelta contra la misma `base_date`; categorías restringidas al vocabulario mapeable.

III-D. Ground truth y taxonomía: un solo target

Cada mensaje no-multiop tiene GT con `movement_type`, `amount`, `currency`, `category`, `description`, `fecha`, `expects_clarification`; los multiop exponen `movements[]`. La taxonomía canónica tiene 9 clases (Comida, Servicios, Transporte, Ocio, Vivienda, Salud, Ingresos, Educación, Ropa). **Ambas condiciones experimentales se evalúan contra el mismo target canónico**: el GT literal (17 sinónimos de dominio, ej. "nafta"$\rightarrow$Transporte, "luz"$\rightarrow$Servicios, "regalo"$\rightarrow$Ocio) y las predicciones del modelo se normalizan mediante un mapa congelado *a priori* antes de comparar (`taxonomy_map.json` v2). Las predicciones sin entrada en el mapa cuentan como incorrectas (política conservadora declarada; cubre salidas ambiguas como "transferencias" o "otros"). Los mapas `trabajo`/`venta`$\rightarrow$Ingresos y `regalo`$\rightarrow$Ocio son juicios semánticos *a priori* documentados como amenaza de constructo. El vocabulario de sinónimos se construyó inspeccionando las salidas observadas de la condición A: es un grado de libertad que favorece a A y cuantificamos su sensibilidad con la política conservadora (§V-C).

III-E. Modelos y artefactos reproducibles

Evaluamos `gemini-3.1-flash-lite`, `gemini-3.5-flash-lite` y `mistral-small-2603` (Mistral Small 4, versión congelada; el alias mutable `latest` fue reemplazado y sus corridas archivadas), más `rule-baseline` determinista (Algoritmo 1, §XII). Versiones, temperatura (0.1), mecanismos JSON (`responseMimeType` en Gemini; `response_format={type: json_object}` en Mistral), fecha de corrida y prompt versionado (`paper/prompt_luka.md`, commit LUKA `72d48716`, 23479 bytes, sha256 `8ac9e2ba`) se registran en `results/model_versions.json`. La implementación completa está en el repositorio del artículo. Precios oficiales verificados el 2026-08-31: 0.25/1.50, 0.30/2.50 y 0.15/0.60 USD por 1M tokens input/output.

III-F. Protocolo

El harness invoca `LLMService.process_message(text, context)` (parsing LLM puro, sin DB) del núcleo de LUKA [3]. Diseño pareado $140 \times k$: cada mensaje atraviesa todos los sistemas bajo el mismo GT. Condición A: contexto solo con `FECHA ACTUAL`. Condición B: mismo prompt $+$

taxonomía de 9 canónicas en contexto. Para consistencia, $R = 3$ réplicas por modelo sobre los 140 mensajes en condición A (las réplicas de Gemini se distribuyeron en días distintos por cuota free-tier, ver §V-G y §IX).

IV. Métricas

Sea GT el ground truth y PRED la salida del modelo (sobre 112 no-multiop salvo indicación).

CFA y FullRecord (definiciones *a priori*):

$$\text{CFA} := \texttt{movement\_type} \wedge \texttt{amount} \wedge \texttt{fecha} \wedge \texttt{currency} \quad (1)$$

$$\text{FullRecord} := \text{CFA} \wedge \texttt{category} \quad (2)$$

`fecha`: si gt es null, correcto por asignación determinista post-LLM del pipeline; si no, match exacto. Distinguimos dos cantidades con nombres inequívocos: *Relative Date Resolution* (solo los 28 mensajes con referencia temporal, capacidad del LLM) y *Effective Date Assignment* (los 112, incluyendo el post-proceso). `description` se reporta aparte (heurística auxiliar, excluida de FullRecord).

Abstención y riesgo operativo. Matriz sobre 112: GT `expects_clarification` (23) vs. "modelo pregunta" (regex sobre `reply_text`; el detector se declara como heurístico, §IX). Reportamos recall/precision/F1 y `false_clarification_rate`. Además, dos métricas de riesgo global: *UnsafePersistRate* = registros persistibles (con amount y type, sin pedir aclaración) cuyo amount o type es incorrecto / total persistible, medido sobre los 112 (no solo los ambiguos); e *invención* separada en crítica (completar amount/type con GT null) y *default de política* (completar currency←ARS por decisión de sistema, que no es alucinación).

Multioperación. Matching greedy bipartito por (type, mínimo $|\Delta\text{amount}|$): `movement_accuracy` (56 movimientos) y `exact_multi_record` (28 mensajes exactos).

H2 y H3. H2 usa los 14 pares pareados: McNemar exacto por par (relativa ok & explícita mal vs. viceversa). H3 compara CFA en single (112) vs. `exact_multi` (28) — ambas definiciones *sin* categoría, para no sesgar — con Fisher exact (muestras no pareadas).

Schema. Separamos *validez sintáctica JSON* (`parse_ok`) de *cumplimiento de contrato* (`intent`/`reply_text` presentes, sin campos extra): ambas pueden diferir, como documenta SOB [7].

Inferencia estadística. Wilson IC95 %; McNemar exacto pareado por mensaje con corrección Holm-Bonferroni por familia de comparaciones; Fisher exact para H3.

Latencia y costo. mean/median/std/p95 (ms, $n = 140$). Costo con tokens *calibrados*: un probe de llamadas directas que replica el armado de LUKA midió el ratio real caracteres/token (3.40 Gemini, 3.53 Mistral, std ≈0.001; salida ≈81–117 tokens/mensaje, sin thinking tokens), eliminando la aproximación chars/4.

Consistencia. Entre réplicas distinguimos *consistencia semántica* (igualdad por atributo y de CFA/FullRecord, excluye `reply_text`) de *consistencia superficial* (JSON completo idéntico, incluida la frase de respuesta).

V. Resultados

Todas las cifras provienen de `results/analysis_summary.json` (v3) y `results/h2_pairs.json`, generados por `analysis/metrics.py` y `analysis/h2_pairs.py`.

V-A. Exactitud global (condición A)

La Tabla I muestra el patrón central: `movement_type` y `currency` son techo, `amount` 0.74–0.80, y la brecha crítico/completo es la tensión dominante. CFA alcanza 0.786 [0.701,0.852] (3.1, 88/112), 0.777 [0.691,0.844] (3.5, 87/112) y 0.607 [0.515,0.693] (Mistral, 68/112); FullRecord cae a 0.518 [0.426,0.608] (58/112), 0.616 [0.524,0.701] (69/112) y 0.339 [0.258,0.431] (38/112). El baseline determinista es competitivo en CFA (0.634) pero débil en FullRecord (0.366), en categoría canónica (0.679) y en resolución de fechas relativas (0.607): el LLM aporta allí donde las reglas no llegan, aunque el análisis no aísla cuánta parte de esa ventaja proviene de cada atributo.

V-B. Desglose por tipología

FullRecord por tipología ($n = 28$): simple 0.71/0.75/0.57 (3.1/3.5/Mistral), coloquial 0.54/0.86/0.32, temporal 0.64/0.71/0.43, ambiguo 0.18/0.14/0.04. Los IC por celda son anchos ($\pm \approx 0{,}18$): diferencias intra-tipología deben leerse con cautela. El colapso de ambiguo es esperable: la mitad de esos mensajes no tiene registro completo por diseño (falta el dato), y lo relevante allí es la conducta de abstención (§V-E).

V-C. Efecto de la taxonomía en el prompt (A/B)

Con el mismo target canónico para ambas condiciones, el efecto de inyectar la taxonomía en el prompt *no es significativo* (Tabla II). FullRecord: 3.1 $0{,}518 \rightarrow 0{,}589$ ($+7.1$ pp, $p = 0{,}215$), 3.5 $0{,}616 \rightarrow 0{,}634$ ($+1.8$ pp, $p = 0{,}832$), Mistral $0{,}339 \rightarrow 0{,}411$ ($+7.2$ pp, $p = 0{,}169$); en los tres casos $p_{\text{Holm}} \geq 0{,}99$ sobre la familia de comparaciones. Categoría: $0{,}723 \rightarrow 0{,}750$, $0{,}839 \rightarrow 0{,}795$ y $0{,}634 \rightarrow 0{,}759$; en 3.5 la condición B es levemente *peor*. CFA no mejora y en Mistral degrada ($0{,}607 \rightarrow 0{,}536$, $p = 0{,}021$, ns tras Holm), asociado a una caída de resolución temporal en B (0.875→0.804).

La lectura metodológica es el hallazgo principal: cuando el criterio de evaluación se mantiene constante, la mayor parte de la mejora aparente de la taxonomía desaparece, porque la normalización semántica *a posteriori* de las salidas libres recupera la mayoría de los aciertos. Bajo la política conservadora sin sinónimos de predicción, la categoría de A caería a 0.566/0.684/0.487 y los Δ de categoría subirían a $+18{,}4/+11{,}1/+27{,}2$ pp: el efecto real del prompt en el prompt está entre ambos escenarios, y ninguno justifica la magnitud $+21$–32 pp que se obtendría cambiando simultáneamente la función de evaluación. La implicancia de diseño es directa: normalizar la salida del modelo contra la taxonomía del sistema es tan importante como pedírsela en el contexto.

Tabla I
CONDICIÓN A ($n = 112$ NO-MULTIOP SALVO LATENCIA $n = 140$). AMBAS CONDICIONES Y EL BASELINE SE EVALÚAN CONTRA EL MISMO TARGET CANÓNICO. CFA=MOVEMENT_TYPE∧AMOUNT∧FECHA∧CURRENCY; FULL=CFA∧CATEGORY. IC WILSON 95 %.

Métrica	3.1 Flash Lite	3.5 Flash Lite	Mistral Small 4	rule-baseline
`movement_type`	1.000	0.982	0.973	0.929
`amount`	0.795	0.795	0.741	0.786
`currency`	1.000	1.000	1.000	1.000
`category` (canónica)	0.723 [0.634,0.798]	0.839 [0.760,0.896]	0.634 [0.542,0.717]	0.679 [0.587,0.758]
`description` (aux.)	0.821 [0.740,0.881]	0.848 [0.770,0.903]	0.857 [0.781,0.910]	0.321 [0.242,0.413]
Relative Date Resolution ($n = 28$)	0.964	0.929	0.500	0.607
Effective Date Assignment ($n = 112$)	0.991	0.982	0.875	0.902
CFA	0.786 [0.701,0.852]	0.777 [0.691,0.844]	0.607 [0.515,0.693]	0.634 [0.542,0.717]
FullRecord	0.518 [0.426,0.608]	0.616 [0.524,0.701]	0.339 [0.258,0.431]	0.366 [0.283,0.458]
Multi `movement_acc.`	1.000 (56/56)	1.000 (56/56)	0.786 (44/56)	0.750 (42/56)
Multi `exact_multi`	1.000 (28/28)	1.000 (28/28)	0.750 (21/28)	0.500 (14/28)
UnsafePersistRate (persistidos)	0/89	0/87	0/80	1/83 (0.012)

Tabla II
A/B TAXONOMÍA ($n = 112$), AMBAS CONDICIONES CONTRA TARGET CANÓNICO. p=MCNEMAR EXACTO (FULL); p_H=HOLM-BONFERRONI.

Modelo	Full A→B	Δ	p (p_H)
3.1	0.518→0.589	+7.1 pp	0.215 (1.00)
3.5	0.616→0.634	+1.8 pp	0.832 (1.00)
Mistral 4	0.339→0.411	+7.2 pp	0.169 (1.00)

Tabla III
H2: PARES PAREADOS ($n = 14$), FECHA EXACTA CORRECTA POR VARIANTE, COND. A.

	Relative Date Res.		McNemar pareado	
Modelo	Relativa	Explícita	b / c	p
3.1	0.929 (13/14)	1.000 (14/14)	0 / 1	1.00
3.5	0.857 (12/14)	1.000 (14/14)	0 / 2	0.50
Mistral 4	0.286 (4/14)	0.857 (12/14)	0 / 8	0.008

V-D. H2: temporalidad relativa con pares pareados

La Tabla III evalúa los 14 pares de contenido idéntico que solo difieren en la forma de la fecha. Mistral Small 4 resuelve la fecha en 85.7 % de los mensajes con fecha explícita pero solo en 28.6 % de sus equivalentes relativos (McNemar pareado $b = 0$, $c = 8$, $p = 0{,}008$). El mismo patrón arrastra el CFA (0.857 vs. 0.286): ante "la semana pasada gasté 12 mil" el modelo registra con fecha incorrecta o inexistente, mientras "el 12 de agosto gasté 12 mil" se registra correctamente. Los modelos Gemini no muestran esa degradación: 3.1 resuelve 0.929 (relativa) vs. 1.000 (explícita) y 3.5 0.857 vs. 1.000, diferencias no significativas con $n = 14$ ($p = 1{,}00$ y $p = 0{,}50$, en ambos casos $b = 0$: la fecha explícita nunca es peor). H2 queda confirmada solo para Mistral Small 4; el contraste entre proveedores es coherente con la Relative Date Resolution del dataset principal (0.93–0.96 en Gemini, 0.50 en Mistral) y delimita dónde se necesita refuerzo de fecha: proveedor, no tarea.

Tabla IV
ABSTENCIÓN CONDICIÓN A ($n = 112$, 23 REQUIEREN ACLARACIÓN). PRED "PREGUNTA" POR REGEX SOBRE `REPLY_TEXT`.

Métrica	3.1	3.5	Mistral 4	Baseline
TP / FP / FN / TN	23/0/0/89	22/1/1/88	22/9/1/80	23/6/0/83
Recall	1.000	0.957	0.957	1.000
Precision	1.000	0.956	0.710	0.793
F1	1.000	0.956	0.815	0.885
false_clar_rate	0.000	0.011	0.101	0.067
Invención crítica /23	0	0	0	1
Default moneda ARS /23	1	1	1	0

V-E. Abstención y persistencia insegura

La Tabla IV muestra la matriz de abstención. La estrategia de aclaración presenta alta cobertura (recall $\geq$0.96) y *cero* persistencias inseguras entre los 23 ambiguos en todos los sistemas; con 0/23, el límite superior binomial 95 % de la tasa poblacional es $\approx$13 %, por lo que reportamos "no se observaron aceptaciones inseguras", no "el riesgo es cero". El trade-off está en los falsos positivos: 3.1 no sobre-pregunta (precisión 1.00), 3.5 casi no (0.96) y Mistral dispara 9 falsas aclaraciones (precisión 0.71, `false_clar` 0.10), lo que degrada la UX sin ganar seguridad. La invención crítica (completar monto o tipo ausentes) es 0/23 en los tres LLM; el único caso *invented* residual en LLM es el default de moneda ARS, que es política de sistema y no alucinación, y por eso se reporta separado. El baseline sí comete una invención crítica y una persistencia insegura global (1/83): persiste un registro con error crítico que los LLM evitan.

V-F. Latencia, costo y schema

La Tabla V muestra el trade-off operativo. Mistral Small 4 es el más rápido (mean 2.8 s, p95 5.4 s) frente a Gemini 3.1 (11.4 s, p95 53.7 s) y 3.5 (12.6 s, p95 54.3 s), con dispersión alta en Gemini (std 17–20 s) que lo excluye de UX síncrona. El costo calibrado con tokens reales de la API es $2.00 (3.1), $2.53 (3.5) y $1.09 (Mistral) por 1000 mensajes: la brecha de costo real entre Mistral y Gemini es menor que la que sugería la estimación por caracteres, y el costo dominante

Tabla V
LATENCIA (MS, $n = 140$) Y COSTO CALIBRADO COND. A. TOKENS REALES VIA PROBE API (INPUT ≈6.4–6.7K TOK/MSG).

Modelo	mean	median	p95	std	$/1000 msgs
3.1 Flash Lite	11372	3981	53682	17455	2.00
3.5 Flash Lite	12562	2587	54296	19849	2.53
Mistral Small 4	2798	2223	5373	2708	1.09
rule-baseline	0	0	0	0	0

Tabla VI
CONSISTENCIA INTER-RÉPLICA COND. A, $R = 3$ EN LOS TRES MODELOS (PROP. DE MENSAJES IDÉNTICOS ENTRE RUNS). SEM. = POR ATRIBUTO; SUP. = JSON COMPLETO.

	3.1	3.5	Mistral 4
`movement_type/amount`	0.814	0.807	0.879
`currency`	0.814	0.807	0.907
`category`	0.800	0.736	0.750
`fecha`	0.243	0.257	0.843
CFA	0.243	0.257	0.807
FullRecord	0.243	0.236	0.657
Multi exact	0.071	0.036	0.536
Sup. (JSON compl.)	0.229	0.171	0.500

es el prompt fijo (≈6,4–6,7k tokens por mensaje, 23.5KB de prompt), lo que hace relevante el caching de contexto en producción. Schema: `parse_ok`=140/140 en todos los LLM (validez sintáctica perfecta), pero el cumplimiento del contrato no es total: `missing_required` 1 (3.1), 27–28 (3.5) y 7 (Mistral) en condición A, concentrados en `reply_text` vacío en multiop. El baseline se excluye de esta tabla por diseño (no fue construido como generador JSON).

V-G. Consistencia entre réplicas

Con $R = 3$ completo para los tres modelos (Tabla VI), Mistral Small 4 presenta mayor estabilidad observada: CFA 0.807 (113/140), FullRecord 0.657 (92/140), atributos críticos 0.879. En Gemini la inconsistencia se concentra en un atributo: tipo, monto y moneda se mantienen (0.81) pero la *fecha* colapsa (0.24–0.26), arrastrando CFA (0.24–0.26), FullRecord (0.24) y multi (0.04–0.07). Dos lecturas: (i) las réplicas de Gemini se ejecutaron en días distintos por cuota, por lo que parte de esa inestabilidad temporal puede ser *drift* del proveedor entre llamadas, no solo muestreo a temperatura 0.1 — un riesgo operativo propio de depender de APIs gestionadas; (ii) aun descontando fecha, la consistencia superficial (JSON completo idéntico) es baja en todos (0.17–0.50): respuestas con registro idéntico difieren en la frase de respuesta, de modo que medir consistencia sobre el JSON completo subestima la estabilidad semántica. Conclusión operativa: temperatura baja no equivale a determinismo, y la capa de persistencia debe tolerar o reducir variación (reintento/voto) en atributos críticos.

V-H. Comparaciones pareadas entre modelos

Tras Holm-Bonferroni, la ventaja de CFA de Gemini sobre Mistral es robusta ($p = 2\times10^{-5}$ y 2×10^{-5}; $p_H \leq 3{,}1\times10^{-4}$), y Gemini supera al baseline en CFA ($p_H \leq 4{,}0 \times 10^{-3}$). En FullRecord, 3.5 supera al baseline ($p_H = 2{,}4 \times 10^{-4}$) y 3.1 marginalmente ($p = 0{,}012$, $p_H = 0{,}127$, ns tras corrección). Mistral empata con el baseline en CFA ($p = 0{,}70$) y FullRecord ($p = 0{,}76$): sin la taxonomía inyectada, un extractor de reglas compite con el modelo más barato en campos críticos, aunque no en resolución temporal ni en conducta de abstención. Entre Geminis no hay diferencia en CFA ($p = 1{,}00$).

VI. H3: MULTIOPERACIÓN VS. MOVIMIENTO ÚNICO

Con definiciones equivalentes sin categoría (CFA single vs. `exact_multi`), H3 se *refuta* para los LLM: Gemini es *mejor* en multi (1.000 vs. 0.786/0.777; Fisher $p \approx 0{,}004$) y Mistral no muestra degradación significativa (0.750 vs. 0.607; $p = 0{,}192$). Solo el baseline cae en multi (0.500 vs. 0.634; $p = 0{,}203$, ns). La explicación es estructural: los mensajes multiop del dataset no contienen fechas relativas ni ambigüedad, justo los estresores donde los LLM fallan; la dificultad del multi no está en los campos críticos sino en producir *todos* los movimientos con conteo exacto (exact_multi 0.75–1.00) y en el cumplimiento del contrato (`movements[]`). La formulación original de H3 ("multi presenta más error") solo describiría al baseline.

VII. ANÁLISIS DE ERRORES

Monto coloquial. Los errores de `amount` (20–26 %) se concentran en coloquialismos con elisión: "Me sacaron 50 lucas del cajero" (C005), "Gasté 2 palos" (C002), "palo y medio" (C024 →1 500 000). Los modelos aciertan el tipo pero fallan el factor o confunden ingreso/egreso en extracciones.

Categoría. Contra el target canónico, el error dominante ya no es vocabulario (la normalización semántica lo absorbe) sino asignación forzada donde no hay clase natural: ante `category`=null del GT o casos como `regalo` (mapeado a Ocio por decisión *a priori*), el modelo elige una canónica plausible en lugar de abstenerse. Una política que exigiera `category`=null + aclaración en casos taxonómicamente irresolubles no está integrada hoy a la métrica de abstención, y la marcamos como mejora de diseño.

Temporalidad. Gemini resuelve referencias relativas con alta precisión (0.93–0.96) y Mistral no (0.50 en A; 0.29 en el experimento pareado). La diferencia entre Relative Date Resolution y FullRecord temporal indica que, incluso con fecha correcta, la categoría explica parte de la pérdida de registro completo; el análisis no aísla causalmente cada atributo.

Ambigüedad y sobre-pregunta. Los FP de Mistral (9) ocurren en mensajes con GT completo (ej. "Pagué 20"). No generan aceptación insegura, pero degradan UX. El detector regex de "pregunta" es heurístico: una arquitectura que exponga `requires_clarification` como campo estructurado eliminaría esta fuente de ruido.

VIII. DISCUSIÓN

El gate correcto no es el oracle. Durante la evaluación sabemos si CFA es correcto porque tenemos GT; en producción no. La política de persistencia debe apoyarse en validaciones *observables*: validez de esquema, consistencia determinista (amount>0, type∈{ingreso,egreso}, fecha válida), estimación

de confianza y la conducta de aclaración del modelo, informadas por la tasa de error crítico medida aquí (12–22 % de CFA erróneo en A incluso para el mejor modelo). `CFA` es un oracle de benchmark, no una regla implementable.

Política formal de persistencia. Los resultados sustentan un policy engine de tres estados: *Accept* si esquema válido $\wedge$ validación determinista $\wedge$ confianza alta; *Clarify* si falta campo requerido $\wedge$ confianza baja $\wedge$ conflicto semántico; *Reject* si esquema inválido u operación no soportada. La categoría nunca bloquea la persistencia (CFA no la incluye) pero se registra como sugerencia confirmable.

Sobre las hipótesis. H1 se confirma (campos léxico-numéricos techo, categoría limitante). H2 se confirma con diseño pareado solo para Mistral; en Gemini la degradación relativa no es significativa (0.86–0.93 vs. 1.00, $n = 14$). H3 se refuta para los LLM: la multioperación no degrada los campos críticos. H4 se refuta en su forma fuerte: inyectar la taxonomía no mejora significativamente cuando se evalúa correctamente; lo que importa es normalizar la salida contra la taxonomía.

Frente a SOB [7] y SchemaRL [6], este trabajo mide esquema *y* semántica separados y agrega el eje de persistencia segura. Frente a STED [8], mostramos que la consistencia superficial subestima la semántica. Frente a PassiveQA/Clarify [10], [11], cuantificamos que la aclaración con alta cobertura y baja aceptación insegura es alcanzable, a costa de falsos positivos que varían por modelo.

IX. Threats to Validity

Validez interna. Una corrida principal por condición (temperatura 0.1). El post-proceso de fecha asigna null→hoy, por eso reportamos Relative Date Resolution aparte. El detector de "pregunta" es regex, no juicio humano. El mapa de sinónimos de predicciones se construyó inspeccionando salidas de A: grado de libertad que favorece a A; cuantificamos la política conservadora alternativa en §V-C. Las réplicas $R = 3$ de Gemini se ejecutaron en días distintos por cuota: parte de su inconsistencia de fecha (0.24–0.26) puede ser *drift* temporal del proveedor y no solo variabilidad por temperatura, por lo que la comparación de consistencia entre proveedores debe leerse con esa salvedad.

Validez externa. 140 mensajes construidos, no observaciones de producción; español rioplatense y ARS único; sin multiturno ni voz. El experimento H2 usa $n = 14$ pares: diferencias pequeñas (como las de Gemini, $\leq$14 pp) no tienen potencia para resultar significativas. No generalizamos a "el español rioplatense" sino que aportamos evidencia sobre él.

Validez de constructo. Autor único para dataset, GT y errores (sin κ): sesgo potencial de benchmark diseñado por el autor del prompt; verificamos contaminación literal y la declaramos. `trabajo`/`venta`/`regalo`→canónica son juicios *a priori*. "Invención" se define operacionalmente y no captura invención de categoría.

Reproducibilidad. Mistral se congeló (`mistral-small-2603`); los alias Gemini se registran como versión con probe de fecha, pero pueden divergir en re-ejecuciones. La cuota free-tier (500 req/día por modelo) obligó a distribuir las réplicas de Gemini en días distintos; el protocolo se completó íntegro (patches, pares H2 y $R = 3$). El costo usa tokens reales de un probe de muestra (cpt input constante; el ratio de salida difiere $\approx$3 % del costo total). Precios verificados a la fecha; pueden cambiar.

X. Aplicación a LUKA

LUKA [3] es el sistema de aplicación: asistente de finanzas personales por WhatsApp que convierte texto en movimientos vía LLM con salida JSON forzada. No se afirma despliegue en producción sin riesgo. La arquitectura derivada de los hallazgos es un pipeline con gate operacional: mensaje → LLM con contexto (`FECHA ACTUAL` + taxonomía del usuario) → validación de esquema → validaciones deterministas (montos positivos, tipos válidos, fecha bien formada, coherencia) → policy engine *Accept*/*Clarify*/*Reject* → persistencia solo tras el gate. La observación experimental informa el calibrado del gate (tasa de error crítico por modelo, conducta de aclaración), pero el gate nunca depende del GT: en producción el oracle no existe. El LLM interpreta; el sistema verifica; el usuario confirma cuando el gate lo exige; la base de datos persiste solo después.

XI. Conclusiones y trabajo futuro

RQ1: los campos críticos alcanzan CFA 0.78–0.79 (Gemini) y 0.61 (Mistral Small 4); el registro completo con categoría cae a 0.34–0.62: la brecha crítico/completo es el hallazgo central. RQ2: la estrategia de aclaración tiene alta cobertura (recall $\geq 0{,}96$) y cero aceptaciones inseguras observadas, con falsos positivos que varían por modelo (precisión 0.71–1.00). RQ3: la temporalidad relativa degrada la resolución de fecha con contenido controlado solo en Mistral (0.29 vs. 0.86, $p = 0{,}008$; en Gemini la diferencia no es significativa), y la multioperación *no* degrada los campos críticos de los LLM. RQ4: no hay dominancia: Gemini gana exactitud, Mistral 4 gana latencia (2.8 s vs. 11.4–12.6 s) y costo ($1.09 vs. $2.00–2.53 por 1000 mensajes) con consistencia observada mayor.

La implicancia central: estos modelos son una capa de interpretación viable si operan dentro de un pipeline con validación de esquema, validaciones deterministas y política formal de aclaración; *no* son una autoridad de persistencia directa del registro completo.

Trabajo futuro. (i) Dataset 300–500 mensajes con factores independientes (monto×temporalidad×cardinalidad) y doble anotación con κ; (ii) campo estructurado `requires_clarification` en lugar del detector regex, integrado a la métrica de abstención; (iii) modelo fuerte de referencia; (iv) evaluación multidivisa; (v) validación en producción real con métricas de fricción; (vi) investigar el mecanismo del drift de fecha en Gemini entre días (caching, versionado silencioso del proveedor).

XII. Apéndice: artefactos y especificación del baseline

Artefactos del repositorio. `data/dataset.json` (v1.2, con changelog),

`data/dataset_temporal_pairs.json` (H2), `data/taxonomy_map.json` (v2, mapas congelados y fecha de congelamiento), `paper/prompt_luka.md` (prompt versionado, sha256), `results/model_versions.json` (versiones, temperatura, mecanismos JSON, precios y fuentes), `results/usage_calibration.json` (probe de tokens), `results/raw/*.jsonl` (salidas crudas por corrida), `results/analysis_summary.json` y `results/h2_pairs.json`, `analysis/metrics.py`, `analysis/h2_pairs.py`, `experiments/run_experiment.py`, `experiments/rule_baseline.py`, `experiments/usage_probe.py`.

Baseline rule-based (spec). El Algoritmo 1 formaliza el extractor determinista. Monto: regex de importes con separadores de miles tolerados y sufijos coloquiales con multiplicador (10^3/10^6). Tipo: vocabulario de superficie (ingreso: "cobré", "me pagaron", "transferencia", "recibí", "venta", "depositaron", "sobró"…; egreso: "gasté", "pagué", "compré", "me dejé", "aboné", "débito", "saqué"…). Categoría: diccionario fijo de 18 sinónimos $\rightarrow$ 9 canónicas, idéntico al mapa del paper. Multioperación: fragmentación por conector "y"/"e" con fecha compartida; si alguna cláusula carece de monto o tipo válidos, se degrada a movimiento único. Negación y modismos no se manejan; es un piso de ingeniería, no un sistema optimizado.

Algoritmo 1 Baseline determinista basado en reglas (`rule-baseline`)

Require: texto t en español rioplatense, fecha base b, diccionario $\mathcal{T}$: 18 sinónimos $\rightarrow$ 9 canónicas
Ensure: registro estructurado r

$M \leftarrow \{\, m : m$ coincide con la regex de monto en $t \,\}$
if t contiene "y"/"e" $\wedge\ |M| \geq 2$ **then**
  $F \leftarrow$ fragmentos de t separados por el conector
  $\phi \leftarrow \textsc{Fecha}(t, b)$ ▷ compartida por todos los movimientos
  $L \leftarrow \emptyset$
  for all $f \in F$ **do**
    $L \leftarrow L \cup \{\, (\textsc{Tipo}(f),\ \textsc{Monto}(f),\ \mathcal{T}(f)) \,\}$
  end for
  if todo $(\tau, v, c) \in L$ cumple $\tau \neq$ null $\wedge\ v \neq$ null **then**
    r.movements $\leftarrow \{(\tau, v, \text{ARS}, c, \phi)\}_{(\tau,v,c)\in L}$ ▷ el primero se replica a nivel mensaje
    return r
  end if
end if
$v \leftarrow \textsc{Monto}(t) \cdot \text{mult}(s)$ ▷ s = sufijo coloquial: lucas, mil, k $\rightarrow \times 10^3$; palo(s) $\rightarrow \times 10^6$
$\tau \leftarrow \textsc{Tipo}(t)$ ▷ vocabulario de superficie; ingreso $\wedge$ egreso $\rightarrow$ null
$c \leftarrow \mathcal{T}(t)$ ▷ primera palabra clave contenida en t; null si no hay
$\phi \leftarrow \textsc{Fecha}(t, b)$ ▷ "ayer" $\rightarrow b - 1$; día $d \rightarrow b - \delta$ con $\delta = ((\text{wd}(b) - \text{wd}(d)) \bmod 7)$, $0 \rightarrow 7$; si no, null
if $v =$ null **then**
  `reply_text` $\leftarrow$ "necesito el monto"
else if $\tau =$ null **then**
  `reply_text` $\leftarrow$ "aclaré si es ingreso o egreso"
else
  `reply_text` $\leftarrow$ "estoy procesando el movimiento"
end if
moneda $\leftarrow$ ARS si $v \neq$ null $\vee\ \tau \neq$ null ▷ default de política
return r con (tipo $= \tau$, monto $= v$, moneda, categoría $= c$, fecha $= \phi$)

Referencias

[1] H. Eltigani, R. Haroon, A. Kocak, A. Bin Faisal, N. Martin y F. Dogar, "WaLLM — Insights from an LLM-Powered Chatbot deployment via WhatsApp," arXiv:2505.08894, 2025. [En línea]. Disponible: https://arxiv.org/abs/2505.08894

[2] P. C. O. Brito, J. Honório, J. A. B. Moura, C. Jones y U. Terton, "Multi-Agent System with Generative AI and Deep Reinforcement Learning for Adaptive Financial Education," en *Proc. CSEDU*, vol. 2, pp. 1313–1324, 2026. DOI 10.5220/0015042800004021.

[3] LUKA: agente financiero conversacional para WhatsApp. Repositorio oficial. [En línea]. Disponible: https://github.com/blob1618/luka

[4] E. C. Acikgoz *et al.*, "Can a Single Model Master Both Multi-turn Conversations and Tool Use? CoALM: A Unified Conversational Agentic Language Model," arXiv:2502.08820, 2025. [En línea]. Disponible: https://arxiv.org/abs/2502.08820

[5] S. Roller *et al.*, "Open-Domain Conversational Agents: Current Progress, Open Problems, and Future Directions," arXiv:2006.12442, 2020. [En línea]. Disponible: https://arxiv.org/abs/2006.12442

[6] Y. Lu *et al.*, "Learning to Generate Structured Output with Schema Reinforcement Learning," en *Proc. ACL 2025*, pp. 4905–4918, 2025. [En línea]. Disponible: https://aclanthology.org/2025.acl-long.243/

[7] A. K. Singh *et al.*, "The Structured Output Benchmark: A Multi-Source Benchmark for Evaluating Structured Output Quality in Large Language Models," arXiv:2604.25359, 2026. [En línea]. Disponible: https://arxiv.org/abs/2604.25359

[8] G. Wang *et al.*, "STED and Consistency Scoring: A Framework for Evaluating LLM Structured Output Reliability," arXiv:2512.23712, 2025. [En línea]. Disponible: https://arxiv.org/abs/2512.23712

[9] Y. Guo, Z. Xu y Y. Yang, "Is ChatGPT a Financial Expert? Evaluating Language Models on Financial Natural Language Processing," en *Findings of EMNLP 2023*, 2023. [En línea]. Disponible: https://arxiv.org/abs/2310.12664

[10] M. S. Baidya, "PassiveQA: A Three-Action Framework for Epistemically Calibrated Question Answering via Supervised Finetuning," arXiv:2604.04565, 2026. [En línea]. Disponible: https://arxiv.org/abs/2604.04565

[11] J. Baan, W. Aziz, B. Plank y R. Fernández, "Clarify, Abstain or Answer? Strategising in Conversation with Belief-Augmented Generation," arXiv:2605.25831, 2026. [En línea]. Disponible: https://arxiv.org/abs/2605.25831