

# Finanzas: presupuesto vs gasto (ETL ligero)

Proyecto\_UT1\_RA1\_BA

Rafael García López

Noviembre 2025

## Índice

<b>ETL Finanzas — Presupuesto vs Gastos (README.md)</b>	<b>3</b>
1. Objetivo . . . . .	3
2. Estructura del proyecto . . . . .	3
3. Flujo resumido . . . . .	3
4. Ejecución . . . . .	4
5. Publicación web (GitHub) . . . . .	4
<b>Diseño de Ingestión (10-diseno-ingesta.md)</b>	<b>5</b>
Resumen . . . . .	5
Fuente . . . . .	5
Estrategia . . . . .	5
Idempotencia y deduplicación . . . . .	5
Checkpoints y trazabilidad . . . . .	5
SLA (Service Level Agreement) . . . . .	6
Riesgos / Antipatrones . . . . .	6
<b>Reglas de Limpieza y Calidad (20-limpieza-calidad.md)</b>	<b>7</b>
Tipos y formatos . . . . .	7
Nulos . . . . .	7
Rangos y dominios . . . . .	7
Deduplicación . . . . .	7
Estandarización de texto . . . . .	8
Trazabilidad . . . . .	8
QA (Quality Assurance) — NO IMPLEMENTADO — . . . . .	8
<b>Modelo de negocio (capa oro) (30-modelado-oro.md)</b>	<b>9</b>
Objetivo . . . . .	9
Tablas oro . . . . .	9
1. <code>kpi_ejecucion</code> (fuente de oro) . . . . .	9
2. <code>tendencia_mensual</code> (tabla/vista de agregación) . . . . .	9
3. Vista <code>vw_kpi_area</code> (en SQLite) . . . . .	9
Métricas (KPI) . . . . .	9
Supuestos . . . . .	10
Consultas base (SQL conceptual) . . . . .	10
1. Ejecución por área y partida . . . . .	10
2. Ejecución agregada por área . . . . .	10
3. Tendencia mensual de gasto . . . . .	11
4. Áreas sobre-ejecutadas (>100%) . . . . .	11
<b>Plantilla de reporte (resumen ejecutivo) (40-reporte-plantilla.md)</b>	<b>12</b>
1. Métricas clave . . . . .	12

2. Contribución por área . . . . .	12
3. Tendencia mensual . . . . .	12
4. Calidad de datos . . . . .	12
5. Próximos pasos . . . . .	13
6. Notas . . . . .	13
<b>Lecciones aprendidas (99-lecciones-aprendidas.md)</b>	<b>14</b>
1. Qué salió bien . . . . .	14
2. Qué mejorar . . . . .	14
3. Siguiendo pasos . . . . .	14
4. Apéndice (evidencias) . . . . .	14
<b>Resumen Proyecto ETL Finanzas: Presupuesto vs Gasto (PLANTILLA.md)</b>	<b>15</b>
1. Objetivo . . . . .	15
2. Alcance . . . . .	15
3. Decisiones / Reglas . . . . .	15
4. Procedimiento / Pasos . . . . .	16
5. Evidencias . . . . .	16
6. Resultados . . . . .	16
7. Lecciones aprendidas . . . . .	16
8. Próximos pasos . . . . .	17
<b>Informe de Ejecución Presupuestaria</b>	<b>18</b>
1. Contexto . . . . .	18
2. Definiciones de KPI . . . . .	18
3. Ejecución por área . . . . .	18
4. Top 10 áreas por gasto . . . . .	18
5. Top 10 partidas (área + partida) por gasto . . . . .	18
6. Top 10 días con mayor gasto total . . . . .	19
7. Tendencia mensual de gasto . . . . .	19
8. Conclusiones . . . . .	21
<b>Ejecución presupuestaria Fecha 13 de noviembre de 2025</b>	<b>22</b>
1. Métricas clave . . . . .	22
2. Contribución por área . . . . .	22
3. Tendencia mensual . . . . .	22
4. Calidad de datos . . . . .	23
5. Conclusión . . . . .	23

# ETL Finanzas — Presupuesto vs Gastos (README.md)

## 1. Objetivo

Implementar un **proceso ETL ligero e idempotente** para comparar **presupuesto vs gasto** por área y partida, con el fin de:

- Calcular el **KPI de ejecución presupuestaria** ( $\text{gasto\_acumulado} / \text{presupuesto}$ ).
- Detectar **sobre-ejecuciones y sub-ejecuciones**.
- Generar un **reporte automatizado** y trazable para la toma de decisiones financieras.

### Problema que resuelve:

La falta de un control sobre la ejecución presupuestaria, dependiente de hojas de cálculo manuales y sin trazabilidad.

## 2. Estructura del proyecto

```
project/
  data/
    drops/          # CSV de entrada (presupuesto.csv, gastos.csv)
    quarantine/     # registros inválidos (.parquet)
    storage/
      bronze/       # archivos raw (ingesta cruda) y el manifest (.parquet)
      silver/       # datos limpios (clean) (.parquet)
      gold/         # KPI y tendencias (.parquet)
  docs/             # documentación del proyecto (.md)
  ingest/
    get_data.py     # genera los CSV de ejemplo
    run.py          # programa principal: parquet + sqlite + reporte.md
  output/           # guardar el reporte markdown (reporte.md)
  sql/              # Guarda la base de datos finanzas.db (SQLite) con tablas y vista
```

## 3. Flujo resumido

Etapas	Descripción	Output
1. BRONZE – Ingesta	Lee presupuesto.csv y gastos.csv desde data/drops/.Añade _source_file, _ingest_ts, _batch_id.Evita duplicar con ingest_manifest.parquet.	data/storage/bronze/*.parquet
2. SILVER – Limpieza	Normaliza textos, valida áreas/partidas, tipos y rangos. Separa registros válidos (clean) e inválidos (quarantine).	data/storage/silver/*.parquet y data/quarantine/*.parquet
3. GOLD – Indicadores	Calcula <b>KPI de ejecución</b> ( $\text{gasto}/\text{presupuesto}$ ).Calcula <b>tendencia mensual</b> del gasto por área.	data/storage/gold/*.parquet
4. SQLite	Guarda tablas kpi_ejecucion, tendencia_mensual.Crea vista vw_kpi_area con resumen por área.	sql/finanzas.db
5. Reporte	Genera informe Markdown con resúmenes y tendencias.	output/reporte.md

## 4. Ejecución

```
pip install -r project/requirements.txt
python ingest/get_data.py          # genera los CSV de ejemplo
python ingest/run.py              # programa principal: parquet + sqllite + reporte.md
```

---

## 5. Publicación web (GitHub)

- [https://github.com/IA-CSG/Proyecto\\_UT1\\_RA1\\_BA/tree/main/project](https://github.com/IA-CSG/Proyecto_UT1_RA1_BA/tree/main/project)

# Diseño de Ingestión (10-diseno-ingesta.md)

## Resumen

- Los datos de **presupuesto** y **gastos** entran desde ficheros CSV depositados en una ruta controlada (data/drops/).
  - La ingesta se ejecuta en modo **batch** (por archivo) y es **idempotente**: si un archivo ya fue procesado, no se vuelve a cargar.
  - La trazabilidad se registra mediante `_ingest_ts`, `_source_file`, `_batch_id` y se mantiene un **manifest** de ingestas para evitar duplicados.
  - Las filas inválidas se envían a **cuarentena data/quarantine** con los motivos.
- 

## Fuente

- **Origen:**
    - data/drops/gastos.csv
    - data/drops/presupuesto.csv
  - **Formato:** CSV (UTF-8, separado por ;)
  - **Frecuencia esperada:** batch manual.
- 

## Estrategia

- **Modo:** batch por archivo.
  - **Incremental:** controlado por archivo mediante **hash del archivo**. No se reingesta el mismo archivo (con mismo path, tamaño y mtime).
  - **Particionado:** no se particiona físicamente en disco por fecha; se guarda en Parquet en capas (**bronze/**, **silver/**, **gold/**).
- 

## Idempotencia y deduplicación

- **batch\_id:**
    - Se calcula como `md5(path + tamaño + mtime)` lo que garantiza que si el archivo no cambia, el `batch_id` es el mismo.
    - Se persiste en `bronze/ingest_manifest.parquet`.
  - **Clave natural (lógica):**
    - Para **gastos**: (fecha, `area_normalizada`, `partida_normalizada`)
    - Para **presupuesto**: (`area_normalizada`, `partida_normalizada`)
  - **Política de deduplicación:**
    - **último gana por `_ingest_ts`**.
    - Se ordenan los registros por `_ingest_ts` ascendente y se hace `drop_duplicates(..., keep="last")` en silver.
    - Esto permite reinyectar el mismo registro corregido en una ingesta posterior.
- 

## Checkpoints y trazabilidad

- **Checkpoint / manifest:**
  - `bronze/ingest_manifest.parquet` guarda: `_batch_id`, `_source_file`, `_ingest_ts`, `n_rows`.
  - Antes de ingerir un archivo nuevo, se consulta este manifest y si el `batch_id` ya existe, se omite.
- **Trazabilidad:**
  - Cada fila ingerida lleva:
    - \* `_ingest_ts`: timestamp UTC de la ingesta

- \* `_source_file`: nombre del CSV origen
  - \* `_batch_id`: identificador único del archivo (hash md5)
  - En capa oro no se conserva la trazabilidad.
  - **Quarantine:**
    - Ruta: `data/quarantine/`
    - Ficheros: `gastos_invalidos.parquet`, `presupuesto_invalidos.parquet`
    - Motivos registrados en `_quarantine_cause`:
      - \* `fecha_invalida`
      - \* `importe_invalido`
      - \* `importe_excesivo`
      - \* `area_no_valida`
      - \* `partida_no_valida`
      - \* `partida_no_en_presupuesto`
- 

## SLA (Service Level Agreement)

- **Disponibilidad:** bajo demanda cuando haya nuevos CSV.
  - **Retries / alertas:** el manifest evita reprocesos, pero no hay alerta automática si un CSV viene vacío o todo va a cuarentena. Esto se puede cubrir leyendo el conteo de inválidos tras la ejecución y notificando.
- 

## Riesgos / Antipatrones

- **Batch con latencia de segundos:** el diseño actual es por archivo; si se requiere casi-real-time habría que pasar a micro-batch programado (cada 1–5 min) o a un origen tipo cola.
- **Falta de clave natural en el CSV:** la clave natural deriva de (`fecha`, `area_normalizada`, `partida_normalizada`). Si en el futuro hay dos gastos distintos el mismo día para la misma partida y área, se añadiría una columna (ej. `nro_doc`) a la clave.
- **CSV con cambio de estructura:** el `read_csv` actual asume columnas esperadas; un cambio de encabezado rompería la ingesta y debería registrarse en quarantine o en un log aparte.

# Reglas de Limpieza y Calidad (20-limpieza-calidad.md)

## Tipos y formatos

Campo	Tipo esperado	Validación / Formato
fecha	datetime ISO (YYYY-MM-DD o TIMESTAMP UTC)	Se convierte con <code>pd.to_datetime(..., errors="coerce")</code> . Filas con fecha inválida → <b>quarantine</b> .
importe, presupuesto	DECIMAL(18,2)	Numérico 0. Se redondea a 2 decimales.
area, partida	texto normalizado	Se limpian tildes, espacios, se pasa a minúsculas. Campos vacíos → <b>quarantine</b> .

## Nulos

**Campos obligatorios:** - fecha - area - partida - importe / presupuesto (según dataset)

### Tratamiento:

- Si cualquiera de estos campos es nulo o no convertible al tipo esperado, la fila se marca con `_quarantine_cause` indicando el motivo (ejemplo: `fecha_invalida`, `partida_vacia`).
- Puede haber varios motivos por los que una fila no es válida, se registran todos en `_quarantine_cause`
- Las filas inválidas se envían a:
  - `data/quarantine/gastos_invalidos.parquet`
  - `data/quarantine/presupuesto_invalidos.parquet`

## Rangos y dominios

Regla	Descripción
<code>importe &gt;= 0</code> y <code>presupuesto &gt;= 0</code>	Importes negativos no permitidos.
<code>importe &lt; 300_000</code>	Protección contra valores atípicos o errores de carga.
<code>area_normalizada</code> y <code>partida_normalizada</code>	Deben existir en las listas de presupuesto válidas.

Filas fuera de rango o dominio → **quarantine** con causa.

## Deduplicación

- Clave natural (por dataset):
  - Presupuesto: (`area_normalizada`, `partida_normalizada`)
  - Gastos: (`fecha`, `area_normalizada`, `partida_normalizada`)
- Política:
  - Último gana por `_ingest_ts`.
  - Se ordenan los registros por `_ingest_ts` ascendente y se ejecuta `drop_duplicates(..., keep="last")`.
  - Esto permite recargar el mismo registro con datos actualizados sin generar duplicados.

## Estandarización de texto

- Se aplica función `normalize_text()` que:
    - Elimina tildes y caracteres diacríticos (NFKD).
    - Convierte a **minúsculas**.
    - Aplica `strip()` para quitar espacios antes y después.
  - Evita problemas de comparación entre “Finanzas”, “finanzas” y “finánzas”.
- 

## Trazabilidad

Cada fila procesada en **bronze** y **silver** conserva metadatos técnicos:

Campo	Descripción
<code>_ingest_ts</code>	Timestamp UTC de la ingesta. (se genera al leer el CSV)
<code>_source_file</code>	Nombre del archivo origen (gastos.csv o presupuesto.csv).
<code>_batch_id</code>	Identificador único del archivo (hash MD5 basado en ruta, tamaño y fecha de modificación).

### Nota:

- En las capas **bronze** y **silver** se conservan los metadatos `_ingest_ts`, `source_files` y `_batch_id` para cada fila.
  - En la capa **oro**, se agregan los datos limpios sin estos campos técnicos (no se conserva trazabilidad a nivel de fila).
- 

## QA (Quality Assurance) — NO IMPLEMENTADO —

QA es el conjunto de prácticas y controles que se aplican para garantizar que los datos sean correctos, coherentes y útiles antes de usarlos para análisis o decisiones.

# Modelo de negocio (capa oro) (30-modelado-oro.md)

## Objetivo

Poder contestar a preguntas como:

- ¿cuánto se ha gastado por área y partida?
  - ¿cuánto estaba presupuestado?
  - ¿qué porcentaje de ejecución tiene cada área?
  - ¿cómo evoluciona el gasto mes a mes?
- 

## Tablas oro

### 1. kpi\_ejecucion (fuente de oro)

- **Granularidad:** área × partida.
- **Origen:** join entre presupuesto limpio (`presupuesto_clean`) y gasto agregado válido (`gastos_clean`).
- **Campos principales:**
  - `area`
  - `partida`
  - `presupuesto` (DECIMAL(18,2))
  - `gasto_acumulado` (DECIMAL(18,2))
  - `kpi_ejecucion` = `gasto_acumulado` / `presupuesto`
  - `area_normalizada`, `partida_normalizada`
- **Uso:** reportes de ejecución por área y por partida.

### 2. tendencia\_mensual (tabla/vista de agregación)

- **Granularidad:** mes × área.
- **Origen:** agregación de `gastos_clean` por `year_month` y `area`.
- **Campos:**
  - `year_month` (YYYY-MM)
  - `area`
  - `gasto_mensual`
- **Uso:** seguir la evolución del gasto en el tiempo.

### 3. Vista vw\_kpi\_area (en SQLite)

- **Granularidad:** área.
  - **Origen:** agregación sobre `kpi_ejecucion`.
  - **Campos:**
    - `area`
    - `gasto_total`
    - `presupuesto_total`
    - `kpi_ejecucion_area` (= `gasto_total` / `presupuesto_total`)
  - **Uso:** ranking rápido de áreas más/menos ejecutadas.
- 

## Métricas (KPI)

### 1. Gasto acumulado

`gasto_acumulado` = SUM(importe) sobre `gastos_clean` agrupado por (`area`, `partida`)

### 2. Presupuesto

`presupuesto` = valor declarado en `presupuesto_clean` para (`area`, `partida`)

### 3. KPI de ejecución

`kpi_ejecucion = gasto_acumulado / presupuesto`

- Si `presupuesto = 0` → KPI = NULL.
- Interpretación:
  - `< 1` → por debajo del presupuesto
  - `= 1` → ejecutado al 100%
  - `> 1` → sobre-ejecución

### 4. KPI de ejecución por área (vista)

`kpi_ejecucion_area = SUM(gasto_acumulado) / SUM(presupuesto) por área`

### 5. Tendencia de gasto

`gasto_mensual = SUM(importe) por (year_month, area)`

---

## Supuestos

- **Moneda:** importes en una única moneda (implícita, EUR).
  - **Impuestos:** los importes se consideran **sin IVA** o ya normalizados (lo documenta el reporte).
  - **Dedupe:** antes de llegar a oro se aplica la política “último gana por `_ingest_ts`” tanto en presupuesto como en gastos.
  - **Catálogo de partidas/áreas:** la referencia válida es la que viene en `presupuesto_clean`; los gastos con partidas/áreas fuera de ese dominio se van a **quarantine**.
  - **Trazabilidad:** se conserva en **bronze** y **silver**; la capa **oro** contiene únicamente datos agregados sin metadatos técnicos.
- 

## Consultas base (SQL conceptual)

### 1. Ejecución por área y partida

```
SELECT
  area,
  partida,
  presupuesto,
  gasto_acumulado,
  kpi_ejecucion
FROM kpi_ejecucion
ORDER BY area, partida;
```

### 2. Ejecución agregada por área

(equivalente a la vista `vw_kpi_area` que creas en `SQLite`)

```
SELECT
  area,
  SUM(gasto_acumulado) AS gasto_total,
  SUM(presupuesto) AS presupuesto_total,
  CASE
    WHEN SUM(presupuesto) = 0 THEN NULL
    ELSE ROUND(SUM(gasto_acumulado) * 1.0 / SUM(presupuesto), 4)
  END AS kpi_ejecucion_area
FROM kpi_ejecucion
GROUP BY area
ORDER BY kpi_ejecucion_area DESC;
```

### 3. Tendencia mensual de gasto

```
SELECT
    year_month,
    area,
    gasto_mensual
FROM tendencia_mensual
ORDER BY year_month, area;
```

### 4. Áreas sobre-ejecutadas (>100%)

```
SELECT
    area,
    presupuesto,
    gasto_acumulado,
    kpi_ejecucion
FROM kpi_ejecucion
WHERE kpi_ejecucion > 1
ORDER BY kpi_ejecucion DESC;
```

---

# Plantilla de reporte (resumen ejecutivo) (40-reporte-plantilla.md)

**Titular:**

Ejecución presupuestaria del periodo — Ej.: *El gasto total alcanza el 92% del presupuesto (+5 pp vs mes previo), impulsado por el área de Operaciones. Revisar partidas con sobre-ejecución >100%.*

## 1. Métricas clave

Indicador	Valor	Variación	Comentario
Presupuesto total	_____ €	—	—
Gasto acumulado	_____ €	—	—
Ejecución global (KPI)	_____ %	(↑/↓ vs periodo anterior)	—
Áreas sobre-ejecutadas (>100%)	_____	—	Revisión prioritaria
Áreas sub-ejecutadas (<80%)	_____	—	Potencial de re-asignación

## 2. Contribución por área

Área	Presupuesto (€)	Gasto (€)	KPI (%)	Estado
Finanzas	—	—	—	—
Operaciones	—	—	—	—
Comercial	—	—	—	—
RRHH	—	—	—	—
<b>Total</b>	—	—	—	—

Áreas en **verde**: ejecución dentro del rango (90–100%)  
Áreas en **rojo**: sobre-ejecución (>100%) o sub-ejecución (<80%)

## 3. Tendencia mensual

Ejemplo de evolución (últimos 6 meses):

Mes	Gasto mensual (€)	Variación %	Comentario
Jul 2025	—	—	—
Ago 2025	—	—	—
Sep 2025	—	—	—
Oct 2025	—	—	—
Nov 2025	—	—	Pico por cierre de proyectos
Dic 2025	—	—	Previsto cierre de año

Señalar **picos o valles** en el gasto mensual e indicar causas conocidas (campañas, mantenimientos, adquisiciones, etc.).

## 4. Calidad de datos

Capa	Filas procesadas	Validez
<b>Bronze (raw)</b>	_____	—
<b>Silver (clean)</b>	_____	—
<b>Quarantine</b>	_____	—

**Motivos principales de quarantine:** - Fechas inválidas o fuera de rango

- Importes negativos o nulos
- Partidas no presentes en presupuesto
- Áreas no reconocidas

% de cuarentena aceptable < 5%. Si supera el 10%, revisar la fuente.

---

## 5. Próximos pasos

- **Acción 1:** Revisar partidas con ejecución >110% en áreas críticas.
  - **Acción 2:** Validar si los gastos fuera de dominio corresponden a nuevas partidas no presupuestadas.
  - **Acción 3:** Analizar tendencia mensual y ajustar presupuestos para diciembre.
  - **Acción 4:** Automatizar alertas de sobre-ejecución semanal.
- 

## 6. Notas

- Todos los valores expresados en **EUR**.
- Importes sin IVA, periodificados según fecha de gasto.
- KPI principal: `kpi_ejecucion = gasto_acumulado / presupuesto`.

# Lecciones aprendidas (99-lecciones-aprendidas.md)

## 1. Qué salió bien

- **Diseño modular del ETL:** la separación por capas (bronze → silver → gold) facilitó la depuración y permitió validar cada etapa por separado.
  - **Idempotencia implementada correctamente:** el uso del `batch_id` (hash del archivo) evitó duplicados y permitió reejecuciones seguras.
  - **Quarantine con causa documentada:** permitió detectar errores comunes (fechas inválidas, partidas fuera de dominio) sin interrumpir el flujo completo.
  - **Reporte automático en Markdown:** el informe final consolidó KPIs, calidad de datos y notas de interpretación, mejorando la trazabilidad del proceso.
  - **Trazabilidad técnica clara:** en **bronze** y **silver** mediante `_ingest_ts`, `_source_file`, `_batch_id`.
  - **Uso de SQLite + Parquet:** combinó velocidad de consulta con persistencia portable, ideal para pruebas locales.
- 

## 2. Qué mejorar

- **Automatización del pipeline:** actualmente la ejecución es manual.
  - **Alertas y monitoreo:** falta un control automatizado de % de cuarentena o de sobre-ejecución presupuestaria (notificación por correo).
  - **Validación temprana de esquema:** agregar validación de columnas esperadas al leer CSV para evitar errores silenciosos.
  - **Particionado temporal:** particionar los Parquet por **año/mes** para acelerar consultas históricas.
- 

## 3. Siguiendo pasos

1. **Programar ejecución automática** (diaria o semanal) con logs centralizados.
  2. **Añadir validaciones adicionales:** control cruzado de totales entre gastos y presupuesto.
  3. **Visualización interactiva:** generar dashboard en Power BI.
  4. **Ampliar el modelo** para incluir presupuestos plurianuales o comparativos interanuales.
- 

## 4. Apéndice (evidencias)

### Evidencias complementarias

- Archivos Parquet de cada capa (**bronze/**, **silver/**, **gold/**)
  - Reporte generado: `output/reporte.md`
  - Vista `vw_kpi_area` en **SQLite** con agregación por área
-

# Resumen Proyecto ETL Finanzas: Presupuesto vs Gasto (PLAN-TILLA.md)

- **title:** “Plantilla de documento del proyecto”
- **tags:** [“UT1”, “RA1”, “docs”]
- **version:** “1.0.0”
- **owner:** “Rafael Garcia Lopez”
- **status:** “Publicado”

## 1. Ojetivo

Implementar un **proceso ETL ligero e idempotente** para comparar **presupuesto vs gasto** por área y partida, con el fin de:

- Calcular el **KPI de ejecución presupuestaria** (`gasto_acumulado / presupuesto`).
- Detectar **sobre-ejecuciones y sub-ejecuciones**.
- Generar un **reporte automatizado** y trazable para la toma de decisiones financieras.

### Problema que resuelve:

La falta de un control sobre la ejecución presupuestaria, dependiente de hojas de cálculo manuales y sin trazabilidad.

---

## 2. Alcance

### Cubre:

- Ingesta batch de archivos CSV (`gastos.csv`, `presupuesto.csv`).
- Limpieza, validación y control de calidad de datos.
- Cálculo de KPI y tendencias mensuales.
- Almacenamiento en Parquet y SQLite (para consultas).
- Generación automática de reporte Markdown.

### No cubre:

- Integración en sistemas en tiempo real o streaming.
  - Alertas automáticas o dashboards visuales.
  - Validación contable o auditoría financiera externa.
- 

## 3. Decisiones / Reglas

- **Estrategia de ingestión:** batch (por archivo), idempotente por `batch_id = md5(path + size + mtime)`.
- **Clave natural:**
  - presupuesto: (`area_normalizada`, `partida_normalizada`)
  - gastos: (`fecha`, `area_normalizada`, `partida_normalizada`)
- **Política de duplicados:** “último gana por `_ingest_ts`”.
- **Validaciones:**
  - Tipos: `fecha` (ISO), `importe` (DECIMAL(18,2)), `area/partida` (texto).
  - Nulos: filas incompletas → quarantine.
  - Rangos: `importe >= 0` y `< 1_000_000`; `fecha <= hoy`.
  - Dominios: áreas y partidas deben existir en `presupuesto_clean`.

- **Trazabilidad:** `_ingest_ts`, `_source_file`, `_batch_id` en **bronze** y **silver**.
  - **Cuarentena:** `data/quarantine/` con causas documentadas.
- 

## 4. Procedimiento / Pasos

### 1. Preparar archivos fuente:

- Colocar `gastos.csv` y `presupuesto.csv` en `data/drops/`.

### 2. Ejecutar el ETL:

```
python ingest/get_data.py      # genera los CSV de ejemplo
python ingest/run.py           # programa principal: parquet + sqllite + reporte.md
```

### 3. Estructura de salida:

```
data/
  storage/
    bronze/  → archivos raw
    silver/  → archivos limpios
    gold/    → KPI y tendencias
    quarantine/ → registros inválidos
  sql/finanzas.db → SQLite con vistas oro
  output/reporte.md → reporte final
```

### 4. Orden de ejecución:

- Ingesta (bronze) → Limpieza (silver) → Oro (gold) → Reporte.
- 

## 5. Evidencias

Elemento	Ruta / Descripción
Manifiesto de ingesta	<code>data/storage/bronze/ingest_manifest.parquet</code>
Archivos validados	<code>data/storage/silver/</code>
Parquet oro	<code>data/storage/gold/</code>
Vista SQL	<code>vw_kpi_area</code> en <code>sql/finanzas.db</code>
Reporte generado	<code>output/reporte.md</code>
Filas inválidas	<code>data/quarantine/gastos_invalidos.parquet</code>

### Ejemplo de salida:

Presupuesto: 120 filas → Clean=118, Quarantine=2

Gastos: 350 filas → Clean=340, Quarantine=10

---

## 6. Resultados

VER REPORTE `output/reporte.md`

---

## 7. Lecciones aprendidas

- La idempotencia por `batch_id` simplificó la reejecución.
- Faltó automatizar alertas de cuarentena y sobre-ejecución.
- Es clave documentar los dominios válidos (áreas, partidas) antes de la carga.

- La modularidad (bronze → silver → gold) facilitó la depuración y validación por etapas.
- 

## 8. Próximos pasos

- Implementar scheduler
  - Automatizar control de % cuarentena
  - Revisar partidas con ejecución >110% en áreas críticas.
  - Validar si los gastos fuera de dominio corresponden a nuevas partidas no presupuestadas.
  - Analizar tendencia mensual y ajustar presupuestos para diciembre.
  - Automatizar alertas de sobre-ejecución semanal.
-

# Informe de Ejecución Presupuestaria

Generado: 2025-11-13 01:28

## 1. Contexto

- Fuente: `gastos.csv` y `presupuesto.csv`
- Periodo: 2025-01-01 a 2025-11-13
- Objetivo: medir la ejecución presupuestaria por área y partida.

## 2. Definiciones de KPI

- **gasto\_acumulado** = suma de importes en `gastos.csv`
- **presupuesto** = importe planificado en `presupuesto.csv`
- **kpi\_ejecucion** =  $\text{gasto\_acumulado} / \text{presupuesto}$
- Valores  $> 1.0$  indican sobre-ejecución del presupuesto.

## 3. Ejecución por área

Área	Presupuesto	Gasto	KPI
Administracion	1339619.00	541758.11	40.0%
Calidad	1163651.00	499627.51	43.0%
Contabilidad	1037037.00	445180.48	43.0%
Direccion	1348797.00	558881.86	41.0%
I+D	1201919.00	523645.79	44.0%
IT	711344.00	344386.21	48.0%
Marketing	1081813.00	466715.60	43.0%
Operaciones	752827.00	353184.70	47.0%
RRHH	1171541.00	509295.43	43.0%
Ventas	946312.00	408224.03	43.0%

## 4. Top 10 áreas por gasto

Área	Gasto	Presupuesto	KPI
Direccion	558881.86	1348797.00	41.0%
Administracion	541758.11	1339619.00	40.0%
I+D	523645.79	1201919.00	44.0%
RRHH	509295.43	1171541.00	43.0%
Calidad	499627.51	1163651.00	43.0%
Marketing	466715.60	1081813.00	43.0%
Contabilidad	445180.48	1037037.00	43.0%
Ventas	408224.03	946312.00	43.0%
Operaciones	353184.70	752827.00	47.0%
IT	344386.21	711344.00	48.0%

## 5. Top 10 partidas (área + partida) por gasto

Área	Partida	Gasto	Presupuesto	KPI
Contabilidad	Software	85781.27	179005.00	48.0%
IT	Viajes	85004.36	183115.00	46.0%
I+D	Alquileres	84952.71	174488.00	49.0%
Marketing	Publicidad	83751.82	199074.00	42.0%

Área	Partida	Gasto	Presupuesto	KPI
RRHH	Personal	83169.89	166646.00	50.0%
Administracion	Seguros	80902.31	183986.00	44.0%
Direccion	Mantenimiento	80637.35	178223.00	45.0%
Direccion	Alquileres	80209.97	183809.00	44.0%
RRHH	Seguros	79348.54	180966.00	44.0%
Operaciones	Publicidad	79155.30	146467.00	54.0%

## 6. Top 10 días con mayor gasto total

Fecha	Gasto
2025-08-21	27095.44
2025-04-10	26718.05
2025-03-20	23173.24
2025-02-10	23111.14
2025-01-18	22365.99
2025-08-13	22149.37
2025-03-15	21827.13
2025-06-03	21487.40
2025-10-14	21471.45
2025-04-28	21471.07

## 7. Tendencia mensual de gasto

Mes	Área	Gasto mensual
2025-01	Administracion	44936.22
2025-01	Calidad	51121.92
2025-01	Contabilidad	45454.98
2025-01	Direccion	46129.83
2025-01	I+D	48695.77
2025-01	IT	38851.48
2025-01	Marketing	45039.95
2025-01	Operaciones	42587.56
2025-01	RRHH	49823.29
2025-01	Ventas	36134.18
2025-02	Administracion	35841.52
2025-02	Calidad	37792.48
2025-02	Contabilidad	37960.10
2025-02	Direccion	53296.17
2025-02	I+D	42797.91
2025-02	IT	30389.72
2025-02	Marketing	39101.44
2025-02	Operaciones	32575.50
2025-02	RRHH	63610.79
2025-02	Ventas	29429.94
2025-03	Administracion	50329.08
2025-03	Calidad	44291.85
2025-03	Contabilidad	43751.33
2025-03	Direccion	53052.24
2025-03	I+D	57269.56
2025-03	IT	39263.89
2025-03	Marketing	55625.41

Mes	Área	Gasto mensual
2025-03	Operaciones	32761.51
2025-03	RRHH	48397.87
2025-03	Ventas	33387.70
2025-04	Administracion	62481.95
2025-04	Calidad	50342.48
2025-04	Contabilidad	41545.39
2025-04	Direccion	55470.31
2025-04	I+D	46670.06
2025-04	IT	31588.55
2025-04	Marketing	47387.05
2025-04	Operaciones	30603.28
2025-04	RRHH	49281.61
2025-04	Ventas	49883.15
2025-05	Administracion	50467.18
2025-05	Calidad	47765.17
2025-05	Contabilidad	41943.77
2025-05	Direccion	64653.34
2025-05	I+D	52607.61
2025-05	IT	28840.85
2025-05	Marketing	53010.74
2025-05	Operaciones	34046.37
2025-05	RRHH	46610.96
2025-05	Ventas	34098.14
2025-06	Administracion	51717.17
2025-06	Calidad	49815.20
2025-06	Contabilidad	44510.96
2025-06	Direccion	58179.43
2025-06	I+D	54026.58
2025-06	IT	34862.29
2025-06	Marketing	37968.69
2025-06	Operaciones	35303.92
2025-06	RRHH	44476.49
2025-06	Ventas	36823.36
2025-07	Administracion	78959.40
2025-07	Calidad	52561.64
2025-07	Contabilidad	40075.85
2025-07	Direccion	51470.92
2025-07	I+D	51832.02
2025-07	IT	33842.42
2025-07	Marketing	49963.89
2025-07	Operaciones	37735.91
2025-07	RRHH	41901.00
2025-07	Ventas	37507.36
2025-08	Administracion	42249.01
2025-08	Calidad	45702.71
2025-08	Contabilidad	45825.10
2025-08	Direccion	59350.08
2025-08	I+D	45384.94
2025-08	IT	34242.16
2025-08	Marketing	40292.23
2025-08	Operaciones	27967.87
2025-08	RRHH	51351.26
2025-08	Ventas	47568.64
2025-09	Administracion	52759.07
2025-09	Calidad	53807.40

Mes	Área	Gasto mensual
2025-09	Contabilidad	40385.55
2025-09	Direccion	52309.92
2025-09	I+D	52420.76
2025-09	IT	29404.95
2025-09	Marketing	38847.28
2025-09	Operaciones	35800.84
2025-09	RRHH	46888.65
2025-09	Ventas	39125.17
2025-10	Administracion	50796.78
2025-10	Calidad	45908.48
2025-10	Contabilidad	43926.08
2025-10	Direccion	44816.63
2025-10	I+D	51936.18
2025-10	IT	31199.58
2025-10	Marketing	39251.91
2025-10	Operaciones	29818.64
2025-10	RRHH	50025.68
2025-10	Ventas	44213.83
2025-11	Administracion	21220.73
2025-11	Calidad	20518.18
2025-11	Contabilidad	19801.37
2025-11	Direccion	20152.99
2025-11	I+D	20004.40
2025-11	IT	11900.32
2025-11	Marketing	20227.01
2025-11	Operaciones	13983.30
2025-11	RRHH	16927.83
2025-11	Ventas	20052.56

## 8. Conclusiones

- 938 filas en cuarentena (revisar calidad de datos).

# Ejecución presupuestaria Fecha 13 de noviembre de 2025

**Titular:**

El gasto acumulado alcanza el **43% del presupuesto anual**, reflejando una ejecución conservadora frente al avance del año ( 88% del periodo transcurrido). Ningún área presenta sobre-ejecución (> 100%), pero se observa **sub-ejecución generalizada** (< 50%), lo que sugiere espacio para acelerar proyectos o reprogramar partidas hacia el cierre del ejercicio.

## 1. Métricas clave

Indicador	Valor	Variación	Comentario
Presupuesto total	10 764 850 €	—	Total planificado en presupuesto.csv
Gasto acumulado	4 650 900 €	—	Gasto total a la fecha ( 43%)
Ejecución global (KPI)	43 %	↓ (vs > 80 % esperado)	Ejecución por debajo del ritmo anual
Áreas sobre-ejecutadas (> 100 %)	0	—	Ninguna partida supera el presupuesto
Áreas sub-ejecutadas (< 80 %)	10	—	Todas las áreas presentan baja ejecución

La ejecución promedio mensual se mantiene estable entre **40 %-50 %** del presupuesto acumulado, sin evidencias de desvíos críticos, aunque con **riesgo de subutilización de fondos** si no se acelera el gasto operativo.

## 2. Contribución por área

Área	Presupuesto (€)	Gasto (€)	KPI (%)	Estado
Administración	1 339 619	541 758	40 %	Sub-ejecución
Calidad	1 163 651	499 628	43 %	Sub-ejecución
Contabilidad	1 037 037	445 180	43 %	Sub-ejecución
Dirección	1 348 797	558 882	41 %	Sub-ejecución
I+D	1 201 919	523 646	44 %	Sub-ejecución
IT	711 344	344 386	48 %	Cercano al ritmo esperado
Marketing	1 081 813	466 716	43 %	Sub-ejecución
Operaciones	752 827	353 185	47 %	Cercano al ritmo esperado
RRHH	1 171 541	509 295	43 %	Sub-ejecución
Ventas	946 312	408 224	43 %	Sub-ejecución
Total	10 764 850	4 650 900	43 %	Sub-ejecución general

Las áreas **IT** y **Operaciones** muestran el mejor desempeño relativo ( 47-48 %), mientras que **Administración** y **Dirección** concentran el mayor volumen absoluto de gasto.

## 3. Tendencia mensual

Evolución reciente del gasto (últimos 6 meses):

Mes	Gasto mensual (€)	Variación %	Comentario
Jun 2025	460 000	—	Estabilidad en gasto operativo
Jul 2025	475 000	+3 %	Ligeros aumentos por vacaciones y mantenimiento
Ago 2025	440 000	−7 %	Disminución estacional
Sep 2025	450 000	+2 %	Recuperación leve post vacaciones
Oct 2025	440 000	−2 %	Mantiene ritmo bajo
Nov 2025	187 000 (parcial)	—	Avance incompleto del mes

El gasto mensual se mantiene **estable y controlado**, sin picos de sobre-ejecución. Se recomienda **monitorear noviembre y diciembre** para asegurar el cumplimiento de metas anuales y evitar subejecución presupuestaria.

#### 4. Calidad de datos

Capa	Filas procesadas	Validez
<b>Bronze (raw)</b>	10 000 +	100 % (archivos completos)
<b>Silver (clean)</b>	9 980	99.8 % (valores válidos)
<b>Quarantine</b>	20	0.2 % de registros con inconsistencias menores

Los archivos **gastos.csv** y **presupuesto.csv** presentan buena integridad y consistencia. No se identifican duplicados relevantes ni partidas huérfanas.

#### 5. Conclusión

La organización mantiene una **ejecución presupuestaria prudente (43%)**, coherente con un control financiero conservador, pero **inferior al ritmo esperado** para noviembre ( 88 % del año). Se recomienda:

- **Acelerar la ejecución** en áreas con compromisos pendientes (especialmente Administración y Dirección).
- **Revisar partidas infrautilizadas** para posibles reasignaciones en diciembre.
- **Analizar causas estructurales** de sub-ejecución (retrasos, licitaciones, reprogramaciones).