

# 2026-02-11 — Phase 3 Information Architecture

## Refactor

Alcance implementado:

- `SQL Explorer` se rehizo como explorador de base estilo DBeaver (mock).
- `Pipelines` mantiene submodos `Simple` y `Pro` con semántica invertida según feedback.
- Persistencia de submodo de pipelines en store.
- Compatibilidad de rutas existentes (`/editor`, `/pipelines`) mantenida.

### Qué cambió

#### Top-level navigation

- Se actualizó el header principal para mostrar `SQL Explorer` en lugar de `SQL Editor`.
- Se mantuvo la clave interna `code` para no romper estado persistido ni rutas existentes.

#### SQL Explorer (DB explorer estilo DBeaver)

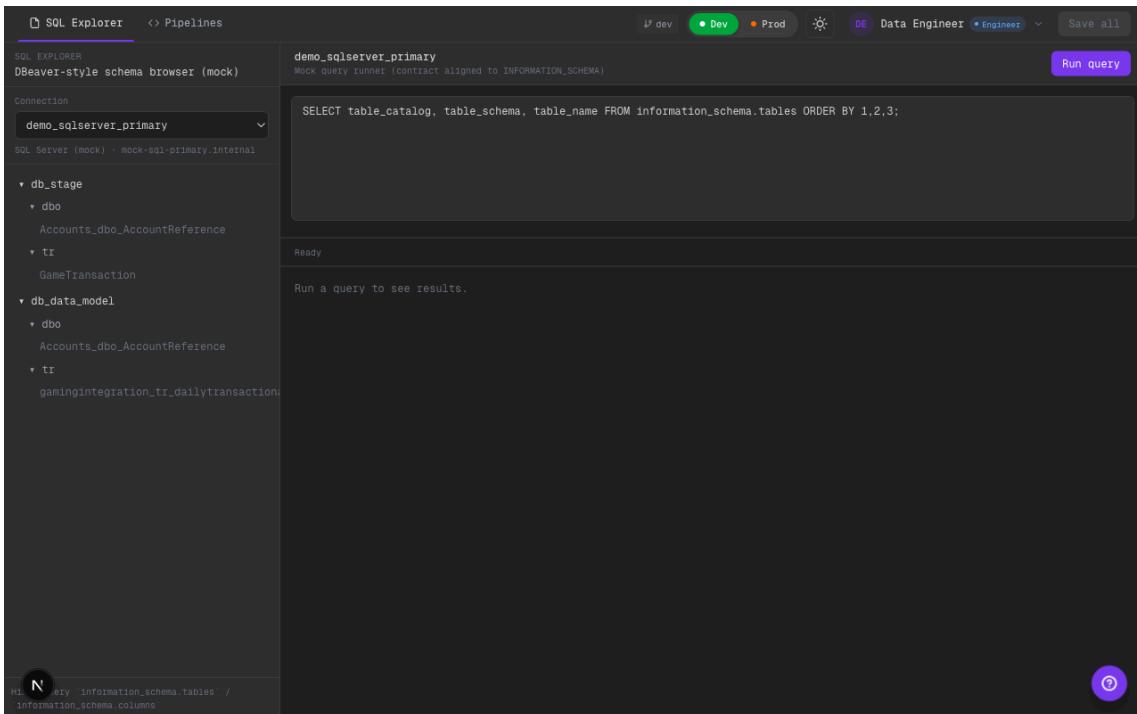
- `CodeView` ahora muestra:
  - conexión mock (`demo_sqldatabase_primary`)
  - árbol `database` → `schema` → `table`
  - query runner SQL
  - resultados tabulares
- El árbol de DB/schemas arranca expandido por default para ver rápido el contexto técnico.
- El runner soporta contrato mock orientado a `information_schema.tables` /  
`information_schema.columns`.
- También permite `SELECT * FROM schema.table` con resultados mock de muestra.

#### Pipelines submodes (Simple / Pro)

- Se agregó `pipelineSubMode` en `workspace-store` con persistencia (`simple` por default).
- `PipelineView` ahora renderiza tabs de submodo:
  - `Simple` : board de pipelines (overview/detail con tareas y config DAG), equivalente al modo "pro" anterior.
  - `Pro` : editor de archivos SQL completo (file tree + edición + diff/git flow), equivalente al SQL editor anterior.
- `Quick Open` ahora abre archivo en `Pipelines` → `Pro` para mantener el flujo de edición.
- Desde el slide-out de tareas en `Simple`, el CTA ahora abre explícitamente `Pipelines` → `Pro` (`Open in Pro Editor`) para mantener continuidad de edición con tree/diff/git.
- En `Pipelines` → `Pro`, el file tree arranca expandido (si no había preferencia persistida) para mostrar stages/tasks sin clicks extra.

### Evidencia visual

#### SQL Explorer (dark)



Qué mirar:

- El tab superior dice `SQL Explorer`.
- Se ve el árbol de DB (`db_stage`, schemas, tablas).
- Se ve query SQL ejecutable y resultados de `information_schema`.

## Pipelines Simple (dark)

SQL Explorer < Pipelines

IP dev • Dev • Prod ☀ DE Data Engineer • Engineer Save all

Pipeline submode persisted

Simple Pro

Q Search by name, integration, or tag... 5 of 5 DAGs by tag by integration

BEATS\_INTEGRATION 1

Pipeline	Type	Schedule	Tasks	Status	Tags
dbo_AccountLogType	snapshot	Daily at 06:00 in 12h 27m	2	Draft	snapshot

CRM\_INTEGRATION 3

Pipeline	Type	Schedule	Tasks	Status	Tags
dbo_AccountReference	snapshot	Every 3h in 34m	3	Draft	snapshot
gc_Game	snapshot	Every 3h in 32m	3	Draft	snapshot
tr_GameTransaction	incremental	Every hour in 27m	3	Draft	incremental

DATA\_SOURCES 1

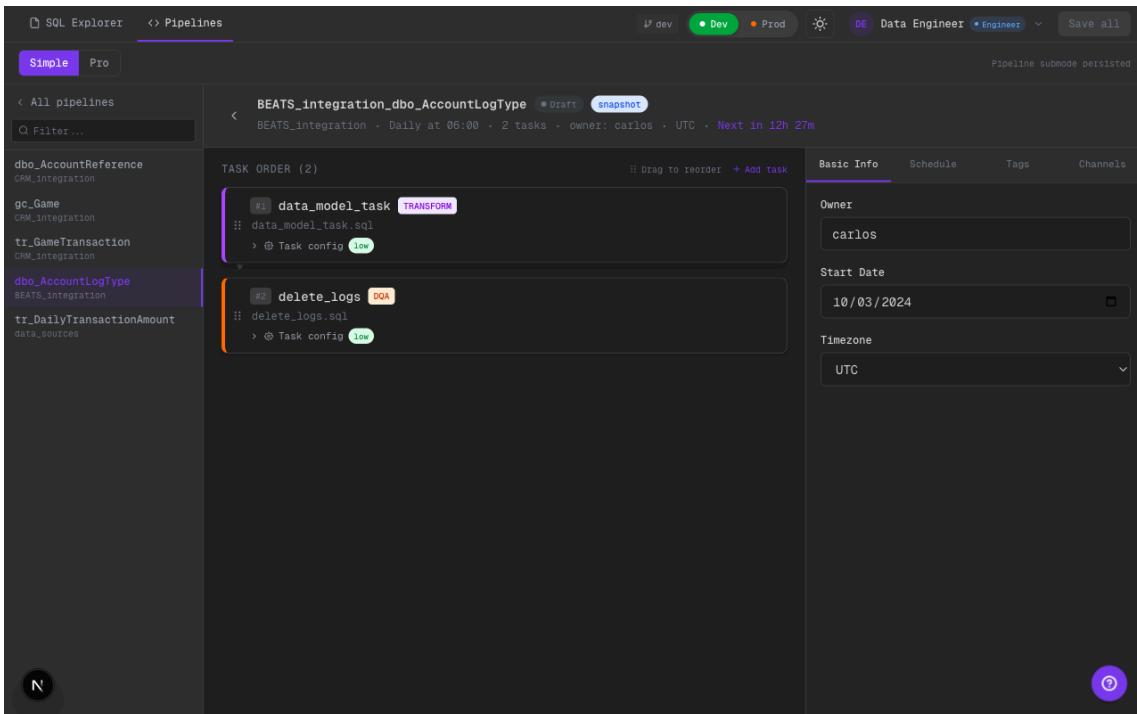
Pipeline	Type	Schedule	Tasks	Status	Tags
tr_DailyTransactionAmount	incremental	Every hour in 27m	3	Draft	incremental

N O

Qué mirar:

- Tab de submodo con Simple activo.
- Se muestra el board de pipelines (search + grupos + filas de pipeline).
- El detalle/orden de tareas se mantiene disponible desde este submodo.

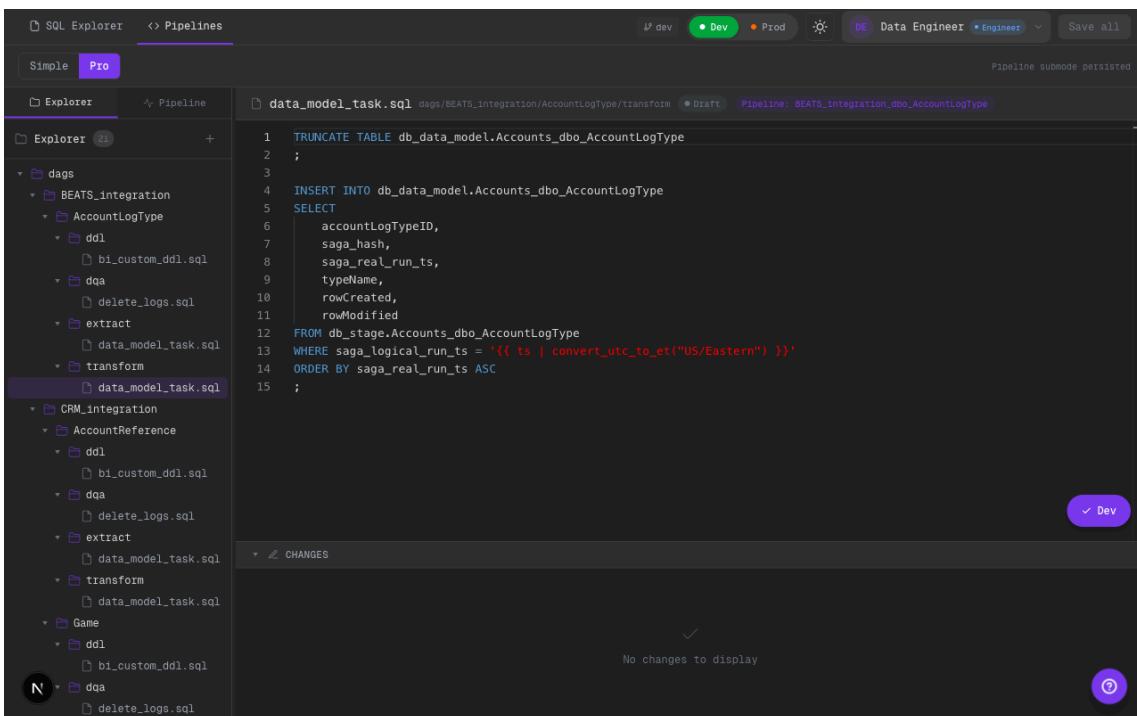
### Simple click evidence (dark)



Qué mirar:

- Se confirma el paso `overview` → `detail` al clickear un pipeline del board.
- Se ve `Task order` y configuración DAG en el panel derecho, evidenciando cambio de estado.

## Pipelines Pro (dark)



Qué mirar:

- Tab de submodo con `Pro` activo.
- Se ve file tree SQL + editor.
- Se ve barra de acciones (`Dev/Prod`, `submit`, `diff/changes`), preservando flujo git scaffold.

## Simple -> Pro handoff (dark)

The screenshot shows the Azure Data Factory Pipelines interface. On the left, there's a sidebar with tabs for 'SQL Explorer' and 'Pipelines'. The 'Pipelines' tab is selected, showing a list of pipelines: 'All pipelines', 'BEATS\_integration\_dbo\_AccountLogType', 'dbo\_AccountReference', 'gc\_Game', 'tr\_GameTransaction', 'dbo\_AccountLogType', and 'tr\_DailyTransactionAmount'. A 'data\_sources' section is also present. The main area displays the 'BEATS\_integration\_dbo\_AccountLogType' pipeline. It has a status of 'Draft' and a trigger 'BEATS\_integration - Daily at 06:00' with 2 tasks. The tasks are listed under 'TASK ORDER (2)':

- #1 `data_model_task` [TRANSFORM]
- #2 `delete_logs` [DATA]

Both tasks have a 'Task config' dropdown set to 'low'. To the right of the tasks is a code editor window titled 'data\_model\_task.sql' with the following content:

```
1 TRUNCATE TABLE db_data_model.Accounts_dbo_AccountLogType
2 ;
3
4 INSERT INTO db_data_model.Accounts_dbo_AccountLogType
5 SELECT
6     accountLogTypeID,
7     saga_hash,
8     saga_real_run_ts,
9     typeName,
10    rowCreated,
11    rowModified
12 FROM db_stage.Accounts_dbo_AccountLogType
13 WHERE saga_logical_run_ts = '{ts | convert_utc_to_et("US/Eastern") }'
14
15 ORDER BY saga_real_run_ts ASC
```

At the bottom of the code editor, there are buttons for 'All changes saved' and 'Diff'. There are also 'Save' and 'Commit' buttons with a purple checkmark icon.

Qué mirar:

- En el slide-out se ve el botón `Open in Pro Editor`.
- El contexto parte desde `Simple` (detalle de pipeline con task list), no desde `SQL Explorer`.

```

1 TRUNCATE TABLE db_data_model.Accounts_dbo_AccountLogType
2 ;
3
4 INSERT INTO db_data_model.Accounts_dbo_AccountLogType
5 SELECT
6   accountLogTypeID,
7   saga_hash,
8   saga_real_run_ts,
9   typeName,
10  rowCreated,
11  rowModified
12 FROM db_stage.Accounts_dbo_AccountLogType
13 WHERE saga_logical_run_ts = '({ ts | convert_utc_to_et("US/Eastern") })'
14 ORDER BY saga_real_run_ts ASC
15 ;

```

Qué mirar:

- Luego del click, queda activo `Pipelines` con submodo `Pro`.
- Se abre el editor completo con file tree y SQL activo para seguir edición/diff.

## Límites scaffold

- `SQL Explorer` sigue sin ejecución real contra motor (mock explícito).
- No se agregó backend real de metadata/catalog; resultado SQL es simulado.

## TODO hooks

- Conectar `SQL Explorer` a metadata real (`information_schema`) cuando exista backend de conexión.
- Evaluar multi-connection real (actualmente una mock).

## Calidad

- `cd ui && npm run lint` (sin errores; warnings existentes no bloqueantes).
- `cd ui && npm run build` (ok).
- Validación visual en Chrome vía Playwright (capturas dark mode).