

Calidad y Documentación

Tests aplicados

Se definieron tests de calidad en dbt para asegurar integridad, consistencia y valores razonables en los datos de cada capa:

Capa Silver (stg_taxi_zones y stg_taxi_trips)

- **unique y not_null**
 - zone_id en **stg_taxi_zones** → garantiza que cada zona TLC sea única y no nula.
 - pickup_datetime y dropoff_datetime en **stg_taxi_trips** → asegura que todos los viajes tengan fechas válidas.
 - pu_location_id y do_location_id en **stg_taxi_trips** → no pueden ser nulos.
- **accepted_values**
 - service_type en **stg_taxi_trips** → restringido a valores ['yellow', 'green'].
- **relationships (integridad referencial)**
 - pu_location_id y do_location_id en **stg_taxi_trips** deben existir en stg_taxi_zones.zone_id.

```
models:
- name: stg_taxi_zones
  description: "Dimensión de zonas de taxi de NYC con borough y zone."
  columns:
    - name: zone_id
      description: "ID de zona según TLC."
      tests:
        - unique
        - not_null
    - name: borough
      description: "Borough (Manhattan, Brooklyn, etc.)"
    - name: zone
      description: "Nombre de la zona TLC."

- name: stg_taxi_trips
  description: "Tabla unificada de viajes Yellow y Green con calidad mínima y enriquecimiento de zonas."
  columns:
    - name: service_type
      description: "Tipo de servicio (yellow o green)."
      tests:
        - accepted_values:
            values: ['yellow', 'green']

    - name: pickup_datetime
      description: "Fecha y hora de recogida."
      tests:
        - not_null
```

Capa Gold (dimensiones y fct_trips)

- **unique y not_null**
 - date_sk en **dim_date**.
 - zone_sk en **dim_zone**.
 - vendor_sk en **dim_vendor**.
 - rate_code_sk en **dim_rate_code**.
 - payment_type_sk en **dim_payment_type**.
 - service_type_sk en **dim_service_type**.

- trip_type_sk en **dim_trip_type**.
 - trip_id en **fct_trips** como clave primaria.
- **relationships** (*integridad referencial entre hechos y dimensiones*)
 - pickup_date_sk y dropoff_date_sk → deben existir en dim_date.date_sk.
 - pu_zone_sk y do_zone_sk → deben existir en dim_zone.zone_sk.
 - vendor_sk → debe existir en dim_vendor.vendor_sk.
 - rate_code_sk → debe existir en dim_rate_code.rate_code_sk.
 - payment_type_sk → debe existir en dim_payment_type.payment_type_sk.
 - service_type_sk → debe existir en dim_service_type.service_type_sk.
 - trip_type_sk → debe existir en dim_trip_type.trip_type_sk.

```
- name: dim_date
  description: "Dimensión de fechas con claves sustitutas y atributos de calendario."
  columns:
    - name: date_sk
      description: "Clave sustituta única para cada fecha."
      tests:
        - unique
        - not_null
    - name: full_date
      description: "Fecha completa (YYYY-MM-DD)."
      tests:
        - not_null
    - name: year
      description: "Año de la fecha."
    - name: month
      description: "Mes de la fecha."
    - name: day
      description: "Día del mes."
    - name: day_of_week
      description: "Número del día de la semana (0=domingo)."

- name: dim_zone
  description: "Dimensión de zonas TLC con borough y nombre de zona."
  columns:
    - name: zone_sk
      description: "Clave sustituta (usa el TLC LocationID)."
      tests:
        - unique
```

Diccionario de datos

Capa Raw: datos originales descargados del TLC de NYC (Yellow y Green Taxi Trips + Taxi Zone Lookup).

Capa Silver:

- stg_taxi_trips unifica y limpia datos de viajes (elimina registros inválidos, estandariza service_type).

- stg_taxi_zones provee el catálogo de zonas TLC con borough y nombre.

Capa Gold:

- Dimensiones maestras: dim_date, dim_zone, dim_vendor, dim_rate_code, dim_payment_type, dim_service_type, dim_trip_type.
- Tabla de hechos: fct_trips con granularidad **1 fila = 1 viaje**.
- Todas las dimensiones se unen a través de llaves sustitutas (*_sk).

8 Tables





NAME ↑	TYPE	CLASSIFICATI...	OWNER	ROWS
 DIM_DATE	Table	—	 ROLE_MARTIN...	4.0K
 DIM_PAYMENT_T...	Table	—	 ROLE_MARTIN...	6
 DIM_RATE_CODE	Table	—	 ROLE_MARTIN...	7
 DIM_SERVICE_TY...	Table	—	 ROLE_MARTIN...	2
 DIM_TRIP_TYPE	Table	—	 ROLE_MARTIN...	2
 DIM_VENDOR	Table	—	 ROLE_MARTIN...	7
 DIM_ZONE	Table	—	 ROLE_MARTIN...	265
 FCT_TRIPS	Table	—	 ROLE_MARTIN...	187.2M

Tabla	Descripción	Clave primaria	Campos relevantes
STG_TAXI_TRIPS	Datos de viajes Yellow/Green unificados y limpiados, con filtros de calidad iniciales.	<i>(no aplica)</i>	pickup_datetime, dropoff_datetime, service_type, pu_location_id, do_location_id, trip_distance, payment_type_desc
STG_TAXI_ZONES	Taxi zones con borough y nombre.	zone_id	borough, zone
DIM_DATE	Dimensión de fechas con atributos calendario.	date_sk	full_date, year, month, day, day_of_week
DIM_ZONE	Dimensión de zonas TLC (pickup y dropoff).	zone_sk	borough, zone
DIM_VENDOR	Catálogo de vendedores TLC.	vendor_sk	vendor_id, vendor_name
DIM_RATE_CODE	Catálogo de códigos de tarifa.	rate_code_sk	rate_code_id, rate_code_desc
DIM_PAYMENT_TYPE	Catálogo de tipos de pago.	payment_type_sk	payment_type_desc
DIM_SERVICE_TYPE	Catálogo de tipos de servicio (yellow/green).	service_type_sk	service_type
DIM_TRIP_TYPE	Catálogo de tipos de viaje (street-hail, dispatch, etc.).	trip_type_sk	trip_type, trip_type_desc
FCT_TRIPS	Tabla de hechos de viajes, una fila por viaje con métricas de negocio y llaves a dimensiones.	trip_id	pickup_date_sk, dropoff_date_sk, pu_zone_sk, do_zone_sk, vendor_sk, rate_code_sk, payment_type_sk, service_type_sk, trip_type_sk, métricas (trip_distance, fare_amount, tip_amount, total_amount, trip_duration_min, mph)

Auditoría de cargas y calidad de datos

Durante la construcción de la tabla fct_trips, se aplicaron filtros para eliminar viajes inválidos:

```
WHERE 1=1
-- Filtros de calidad
AND DATEDIFF('minute', pickup_datetime, dropoff_datetime) > 0
AND DATEDIFF('minute', pickup_datetime, dropoff_datetime) < 1440
AND trip_distance > 0
AND trip_distance < 100
AND fare_amount ≥ 0
AND fare_amount < 1000

{% if is_incremental() %}
AND pickup_datetime > (SELECT COALESCE(MAX(pickup_datetime), '2000-01-01'::TIMESTAMP) FROM {{ table }})
{% endif %}

{% if var('process_month', none) %}
AND DATE_TRUNC('month', pickup_datetime) = '{{ var("process_month") }}'::DATE
{% endif %}
```