



DATA ENGINEERING

# Data Contracts

*Sin caer en  
burocracia documental*

---

Cuándo usarlos · Qué contienen · Cómo validarlos · Cómo no ahogarlos



**LuisRai** · Data Architect

[linkedin.com/in/lraigosov](https://www.linkedin.com/in/lraigosov)

@LuisRai

# Un contrato, no un documento

*"Un data contract es un acuerdo formal, versionado y ejecutable entre el productor de datos y sus consumidores."*

## NO es

- Un wiki que nadie lee
- Un Excel de columnas
- Una reunión de alineación
- Un README olvidado

## SÍ es

- Código o YAML versionado
- Tests automatizados
- SLA explícito con dueño
- Ejecutable en CI/CD



**La clave: el contrato vive en código, se versiona, y se valida automáticamente.**

Ref: Andrew Jones · *Driving Data Quality with Data Contracts* (O'Reilly, 2023)

# Usarlos cuando el dolor ya existe

*No son para todos los datos. Empieza donde duele.*



## Pipelines rotos

Cambios upstream sin aviso que rompen downstream



## Múltiples consumidores

Más de un equipo depende del mismo dataset



## Datos críticos de negocio

KPIs, ML features, reportes regulatorios



## SLAs incumplidos

Frescura o completitud indefinida o ignorada



## Datos externos / 3eros

APIs, partners, proveedores de datos



## Data Mesh / dominios

Cada dominio publica productos de datos



Evita contratos para tablas internas sin consumidores externos — solo añaden fricción.

Ref: Atlan · Data Contracts Explained (2025) | Soda · Definitive Guide to Data Contracts (2026)

# Anatomía de un Data Contract

01

## Schema

Campos, tipos, nullability. En YAML/Avro versionado.

02

## Calidad

Reglas de completitud, unicidad, rangos válidos, frescura.

03

## SLA

Frecuencia de actualización, latencia máxima, uptime.

04

## Dueño y Dominio

Equipo responsable, canal de soporte, canal de cambios.

05

## Semántica

Qué significa cada campo para el negocio, no solo su tipo.

# Así luce en código

```
orders_contract.yaml

contractName      : orders_v2
version           : 2.1.0
owner             : commerce-domain
schema           :
  - name          : order_id
    type          : string  not_null: true pattern: ^ORD-\d{8}$
  - name          : amount
    type          : decimal  min: 0.01
quality          :
  freshness       : maxDelay: 15m
  completeness:   order_id: 0.999
sla              : uptime: 99.9%  support: #data-on-call
```

↑ *Mínimo viable. Sin secciones que nunca se actualizan.*

Ref: ODCS v3.0.0 · Bitol / Linux Foundation (Oct 2024)

# Validación en 3 capas

*Si no está automatizado, no está validado.*



CAPA 1

## Schema / Estructura

- Tipos correctos · nullability · patrones regex
- Herramientas: Great Expectations, dbt tests, Pydantic



CAPA 2

## Calidad de Datos

- Completitud · unicidad · rangos · frescura
- Herramientas: Soda Core, Monte Carlo, dbt-expectations



CAPA 3

## SLA & Operación

- Latencia · uptime · alertas en canal acordado
- Herramientas: Observabilidad + on-call runbooks

# Del borrador al contrato vivo

1

## Draft en YAML/JSON

Producer y consumer acuerdan campos mínimos + SLA

2

## PR + Code Review

Revisión técnica + revisión de negocio en un mismo PR

3

## Tests en CI

Schema checks + quality rules corren en cada merge

4

## Monitoreo continuo

Alertas automáticas si el contrato se rompe en prod

5

## Versionado semántico

Cambios breaking = major version + notificación a consumers

# Por qué mueren los contracts

Señales de que tu contrato es decorativo, no operativo:

 <p>El contrato vive en Confluence</p>	 <p>Pásalo a código. YAML en Git, PR para cambios.</p>
 <p>Nadie sabe quién es el owner</p>	 <p>Owner = equipo con on-call. Sin dueño, sin contrato.</p>
 <p>Se creó y nadie lo ejecuta</p>	 <p>Sin tests automáticos = sin contrato. Solo es un README.</p>
 <p>Tiene 40 campos opcionales</p>	 <p>Mínimo viable: schema + 3 reglas de calidad + SLA.</p>
 <p>Se actualiza 1 vez al año</p>	 <p>Si el dato cambia y el contrato no, ya está roto.</p>

# ¿Tu contrato está vivo?

Responde estas 7 preguntas. Si hay algún NO, actúa.



¿Está en Git con historial de cambios?

1



¿Tiene un owner con nombre y canal de contacto?

2



¿Los tests corren en CI en cada deploy?

3



¿Hay alertas si se rompe en producción?

4



¿Los consumidores lo conocen y lo usan?

5



¿Se actualiza cuando cambia el schema?

6



¿Tiene menos de 20 campos/secciones?

7

“

**Un data contract sin tests automáticos es solo una promesa.**

— Principio #1, ODCS Community

💬 **¿Ya tienes data contracts en tu equipo?**

Cuéntame en los comentarios ↓ o conecta conmigo si quieres profundizar.



¿Te fue útil? [Comparte](#) · [Guarda](#) · [Comenta](#)