

## **ENTREGA 1 y 2 - Proyecto Base de Datos: “Agentes de Gestión UX”**

Santiago Cianciolo

**Fecha:** 07/10/2025 (entrega 1)

**Archivo:** Entrega1SQL\_Cianciolo.sql

**Tema:** Base de datos para gestión de agentes de experiencia de usuario

---

### **1. Introducción**

El proyecto propone una base de datos destinada a registrar información sobre los agentes de atención al usuario de una empresa, incluyendo el volumen de tickets atendidos, el cumplimiento del SLA y la favorabilidad obtenida en la gestión.

El objetivo es centralizar los datos del equipo para facilitar reportes de rendimiento y niveles de satisfacción de los clientes.

---

### **2. Objetivo**

Crear una base de datos funcional que permita almacenar y relacionar información de los agentes, la cantidad de tickets gestionados, el cumplimiento de los SLA y la favorabilidad del servicio, de manera que se puedan analizar métricas clave del desempeño del equipo.

---

### **3. Situación problemática**

Actualmente, la información se encuentra dispersa en hojas de cálculo y registros manuales, lo que dificulta analizar el desempeño de los agentes y la eficiencia de atención y genera:

- Dificultad para analizar tendencias históricas
- Errores en la consolidación de datos
- Tiempo excesivo en la generación de reportes
- Imposibilidad de realizar análisis cruzados en tiempo real

Con este modelo, se busca centralizar la información, reducir errores y simplificar el análisis de métricas e indicadores.

---

#### 4. Modelo de negocio

La empresa “UX SA” cuenta con un equipo de agentes que gestionan consultas de usuarios a través de un chatbot que genera tickets con ciertas métricas.

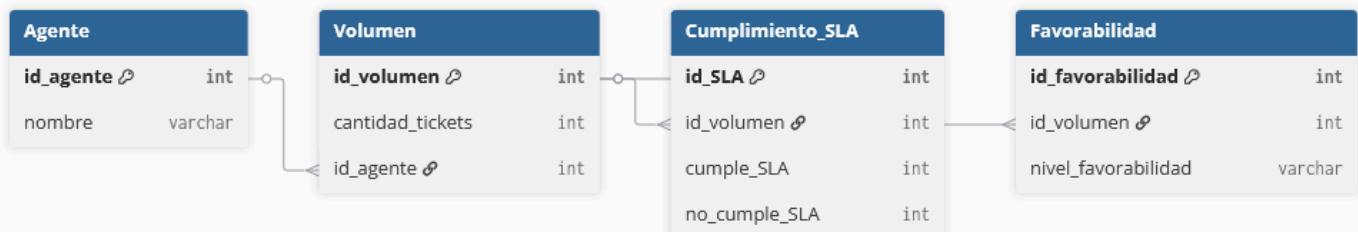
Cada agente gestiona un volumen de tickets, los cuales se evalúan en términos de cumplimiento del SLA y favorabilidad (satisfacción). Esto permite conocer el rendimiento de cada agente y la satisfacción asociada a su gestión.

---

#### 5. Diagrama Entidad-Relación (adaptado)

Agente (1) ---- (N) Volumen (1) ---- (1) Cumplimiento\_SLA (1) ---- (1) Favorabilidad

- **Agente** → registra datos personales de cada agente.
- **Volumen** → cantidad de tickets gestionados por cada agente.
- **Cumplimiento\_SLA** → relaciona el volumen con indicadores de cumplimiento.
- **Favorabilidad** → mide el nivel de favorabilidad de un volumen de tickets.



## 6. Listado de tablas

Tabla	Descripción	Campos principales	Claves
<b>Agente</b>	Datos de cada agente UX	id_agente, nombre	PK: id_agente
<b>Volumen</b>	Cantidad de tickets gestionados por cada agente	id_volumen, id_agente, cantidad_tickets	PK: id_volumen, FK: id_agente
<b>Cumplimiento_SLA</b>	Cumplimiento del SLA por volumen	id_SLA, id_volumen, cumple_SLA, no_cumple_SLA	PK: id_SLA, FK: id_volumen
<b>Favorabilidad</b>	Nivel de favorabilidad por volumen	id_favorabilidad, id_volumen, nivel_favorabilidad	PK: id_favorabilidad, FK: id_volumen

## 7. Explicación general del modelo

El modelo permite:

- Identificar qué agente generó cada volumen de tickets.
- Conocer el cumplimiento del SLA asociado a ese volumen.
- Medir la favorabilidad alcanzada en cada caso.

Es un modelo que busca simplificar el análisis inicial del desempeño del equipo.

## 8. Script SQL

GitHub: <https://github.com/sciadev/Entrega-1-SQL---Cianciolo>

## **ENTREGA 2 - Proyecto Base de Datos: “Agentes de Gestión UX”**

Santiago Cianciolo

**Fecha:** 04/11/2025

**Archivos:**

- Entrega2SQL\_Objetos\_Cianciolo.sql
- Entrega2SQL\_Datos\_Cianciolo.sql

### **Vistas**

#### **1) Vista: vw\_resumen\_agentes**

**Descripción:** Esta vista consolida información clave de cada agente incluyendo su nombre, total de tickets gestionados y promedio de favorabilidad.

**Objetivo:** Facilitar consultas rápidas sobre el rendimiento general de cada agente sin necesidad de realizar JOINs complejos.

**Tablas que la componen:** Agente, Volumen, Favorabilidad

#### **2) Vista: vw\_cumplimiento\_detallado**

**Descripción:** Proporciona un detalle completo del cumplimiento de SLA por agente y período.

**Objetivo:** Generar reportes de cumplimiento para la gestión y análisis de mejoras.

**Tablas que la componen:** Agente, Volumen, Cumplimiento\_SLA

### **Funciones**

#### **1) Función: fn\_porcentaje\_cumplimiento**

**Descripción:** Calcula el porcentaje de cumplimiento de SLA para un agente específico.

**Objetivo:** Estandarizar el cálculo de porcentajes de cumplimiento y evitar errores de cálculo manual.

**Datos/Tablas que manipula:** Recibe id\_agente como parámetro y consulta las tablas Volumen y Cumplimiento\_SLA.

#### **2) Función: fn\_favorabilidad\_a\_numero**

**Descripción:** Convierte valores textuales de favorabilidad (Excelente, Bueno, Regular) en valores numéricos para cálculos.

**Objetivo:** Facilitar el procesamiento matemático de las clasificaciones de favorabilidad para reportes estadísticos.

**Datos/Tablas que manipula:** Recibe nivel\_favorabilidad como parámetro textual y retorna valor numérico equivalente.

## **Stored Procedures**

### **1) Procedimiento: sp\_insertar\_volumen\_completo**

**Descripción:** Inserta un nuevo registro de volumen y automáticamente crea los registros relacionados de SLA y favorabilidad.

**Objetivo/Beneficio:** Garantiza la integridad referencial y evita la inserción manual en múltiples tablas, reduciendo errores.

**Tablas que componen/interactúa:** Volumen, Cumplimiento\_SLA, Favorabilidad

### **2) Procedimiento: sp\_reporte\_agente**

**Descripción:** Genera un reporte completo de un agente específico con todas sus métricas consolidadas.

**Objetivo/Beneficio:** Automatiza la generación de reportes individuales con información detallada y resumen estadístico.

**Tablas que componen/interactúa:** Agente, Volumen, Cumplimiento\_SLA, Favorabilidad

## **Triggers**

### **1) Trigger: tr\_auditoria\_volumen**

**Descripción:** Registra automáticamente cambios en la tabla Volumen para mantener un historial de auditoría.

**Objetivo:** Mantener trazabilidad de modificaciones para cumplimiento normativo y análisis histórico.

**Tabla/Situaciones de activación:** Se activa en UPDATE y DELETE sobre la tabla Volumen, registra en tabla auditoria\_volumen.

## **2) Trigger: tr\_validar\_favorabilidad**

**Descripción:** Valida que los valores de favorabilidad sean válidos (Excelente, Bueno, Regular) antes de la inserción.

**Objetivo:** Garantizar la calidad de datos evitando valores inválidos en métricas críticas.

**Tabla/Situaciones de activación:** Se activa en INSERT sobre la tabla Favorabilidad cuando nivel\_favorabilidad no es un valor permitido.

## **Explicación General del Modelo**

El proyecto permite identificar qué agente generó cada volumen de tickets, conocer el cumplimiento del SLA asociado a ese volumen y medir la favorabilidad alcanzada en cada caso. Es un modelo que busca simplificar el análisis inicial del desempeño del equipo y que con la implementación de vistas, funciones, procedimientos y triggers, proporciona una solución automatizada para la gestión de métricas UX.

Los objetos de base de datos implementados en esta segunda entrega mejoran la funcionalidad del sistema, proporcionando herramientas para consultas optimizadas,, procesos automatizados y validaciones de integridad de datos.

## **Scripts SQL**

GitHub:

<https://github.com/sciadev/Entrega-1-SQL---Cianciolo>

<https://github.com/sciadev/Entrega-2-SQL--Cianciolo>