

# Diseño, limpieza y carga de datos para un sistema de consulta

## Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Corona Correa, S. L. (2023). Diseño, limpieza y carga de datos para un sistema de consulta. *Cuadernos Técnicos Universitarios de la DGTIC*, 1 (1), páginas (111 - 125).

<https://doi.org/10.22201/dgtic.ctud.2023.1.1.23>

**Susana Laura Corona Correa**

Dirección General de Cómputo y de  
Tecnologías de Información y Comunicación,  
Universidad Nacional Autónoma de México

[susanacc@unam.mx](mailto:susanacc@unam.mx)

ORCID: 0009-0003-0560-1745

## Resumen:

La existencia de nuevas necesidades de consultas y reportes dio origen al desarrollo del Sistema de Consulta de Información de un cuerpo colegiado de la Universidad Nacional Autónoma de México, lo que implicó el diseño de su base de datos utilizando prácticas de metodologías ágiles. Se presentan las actividades realizadas en el análisis de requerimientos de los datos, el entendimiento de la problemática actual, el análisis exploratorio de la información, el diseño de la base de datos, la identificación y validación de los catálogos, así como el proceso de limpieza de datos, carga y verificación de la información inicial para su operación. El resultado principal fue contar con un nuevo diseño de base de datos, poblado con datos revisados, que cumpliera con las necesidades del cuerpo colegiado para facilitar su consulta y aprovechamiento.

## Palabras clave:

Limpieza de datos, carga de datos, diseño de bases de datos.

## 1. INTRODUCCIÓN

Se solicitó a la Dirección General de Cómputo y de Tecnologías de Información y Comunicación realizar un proyecto conjunto con un cuerpo colegiado de la Universidad Nacional Autónoma de México para actualizar su sistema para consultar datos de sus expedientes y miembros. Para ello se realizó el diseño de una nueva base de datos y se hizo la limpieza y carga de los datos relevantes de los expedientes a consultar, tales como: fecha, lugar, tipo de proceso atendido, participantes de un proceso, persona designada, asistencia, entre otros.

Asimismo, mediante un proceso de análisis se identificaron oportunidades de mejora en la calidad de los datos<sup>1</sup> de la información histórica del registro de los expedientes de este cuerpo colegiado.

## 2. OBJETIVO

Mejorar la calidad de los datos de los expedientes y sus procesos, así como de los datos históricos, a través del diseño de la base de datos para cumplir con las necesidades establecidas por el usuario sobre la consulta, registro y generación de reportes de los expedientes de las sesiones de un cuerpo colegiado de la Universidad Nacional Autónoma de México.

## 3. ANTECEDENTES

La documentación de las sesiones de un cuerpo colegiado de la Universidad Nacional Autónoma de México (UNAM) se concentra en expedientes generados a partir de su instauración hasta la fecha, lo cual se ha hecho en carpetas físicas estructuradas en uno o varios tomos. Los expedientes contienen principalmente el orden del día, la lista de asistencia, el acta de la sesión y sus anexos.

Se digitalizaron los expedientes históricos a partir de una iniciativa previa con la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), con lo que la oficina de este cuerpo colegiado incorporó la digitalización de su contenido como un proceso al finalizar la integración de cada uno de ellos.

El control de los datos relevantes de los expedientes de las sesiones ordinarias y extraordinarias de este cuerpo colegiado se ha realizado por medio de su registro en una hoja de cálculo; posteriormente este archivo era importado de forma periódica por personal técnico en la base de datos del sistema anterior.

Debido a que los usuarios del sistema anterior no contaban con la posibilidad de agregar su propia información, los datos que podían consultar estaban incompletos y desactualizados, lo que ha dificultado su uso de forma oportuna.

Adicionalmente, se han requerido búsquedas y salidas de información para cubrir peticiones de información, debido a que los filtros del sistema anterior sólo permiten buscar por número de expediente, año, tipo de sesión, proceso, cargo y entidad.

---

1 La calidad de los datos se refiere al grado en el cual los datos satisfacen “los requisitos definidos por la organización a la que pertenece el producto” de acuerdo con la norma ISO 25012:2008 (ISO 25000, s.f.)

Adicionalmente, la información obtenida en los listados no cumplía con el formato, calidad y datos necesarios para resolver las consultas y reportes solicitados a este cuerpo colegiado, debido a que no permitía su exportación en archivos de hoja de cálculo; únicamente se mostraba el resultado vía web, por lo que el usuario del sistema no podía usarlo fácilmente para generar sus reportes.

Otras oportunidades de mejora del diseño de la base de datos anterior surgen a partir de la falta de catálogos, ya que se tenían datos sin normalizar y los datos importados de hojas de cálculo con los expedientes presentaban diversos errores, tales como: datos incompletos, errores ortográficos, valores incorrectos, entre otros.

Lo anterior ha ocasionado que los usuarios del sistema usaran su archivo de hoja de cálculo y la carpeta física del expediente para poder dar respuesta a las solicitudes de información y reportes, debido a que el sistema no cubría sus necesidades.

Por lo anterior, se tomó la decisión de desarrollar una aplicación y diseño de la base de datos nuevos, para lo cual se acordaron diversas actividades con la oficina del cuerpo colegiado para alcanzar los objetivos y considerar filtros adicionales.

## 4. PROCESO REALIZADO PARA EL DISEÑO DE LA BASE DE DATOS, LA LIMPIEZA Y CARGA DE DATOS

### 4.1 ANÁLISIS DE REQUERIMIENTOS Y ENTENDIMIENTO DE LA PROBLEMÁTICA ACTUAL

En esta etapa hubo sesiones de trabajo acerca de los datos de los expedientes para entender las necesidades de información del área, la problemática para la consulta y generación de reportes, así como el uso de la información y los objetivos a alcanzar respecto de los datos.

Estas sesiones permitieron comprender la estructura de los datos, el flujo de información que sigue el usuario desde su registro hasta su consulta y la periodicidad con la que se realizan estas actividades. De esta manera se contribuyó a definir mejoras en el diseño de la base de datos para el nuevo sistema, así como para definir validaciones y consideraciones en la limpieza de su información.

Durante todo el proyecto se incorporaron prácticas de metodologías ágiles como la generación de un *backlog* o lista detallada de actividades, participación en reuniones diarias (*dailys*), retrospectivas de los periodos de trabajo (*sprints*) para identificar mejoras en cada ciclo de desarrollo, entre otras.

### 4.2 ANÁLISIS EXPLORATORIO DE LA INFORMACIÓN

En esta etapa se revisó el archivo de hoja de cálculo que contenía 1,200 filas con los datos de expedientes y procesos atendidos por el cuerpo colegiado, desde su instalación. Dado que el proceso se ejecutaría sólo una vez antes de iniciar la operación del sistema, se eligió crear fórmulas directamente en una copia del archivo de hoja de cálculo del usuario del sistema para generar comandos INSERT con el fin de agregar los datos en una tabla en la base de datos con la finalidad de facilitar su análisis como se muestra en el Anexo A. Las tablas usadas para el análisis fueron eliminadas al final del proceso de revisión.

En este caso se utilizó Microsoft Excel para generar las fórmulas y PgAdmin como cliente para crear las tablas en donde se va a analizar la información y ejecutar los comandos INSERT; se muestra un ejemplo en el Anexo B.

Tras la creación y población de la tabla para el análisis de datos, se revisaron duplicados y la longitud máxima de los datos, se identificaron los valores que podían ser homogeneizados en catálogos, valores que no tenían una correspondencia exacta con los nombres de los miembros del cuerpo colegiado, fechas que no tenían un formato adecuado, datos inconsistentes de acuerdo con el contexto, entre otros; por ejemplo, se ubicaron campos vacíos que debían contener algún valor, fechas inconsistentes, etc. En el Anexo C se muestra un ejemplo de la consulta utilizada en este paso del proceso.

Los resultados de este análisis se comunicaron a la oficina del cuerpo colegiado y se acordaron los cambios que se aplicaron en DGTIC, tras considerar los catálogos validados y las indicaciones obtenidas del usuario del sistema. Por su parte la oficina realizó la revisión de los datos contrastando la información faltante o incorrecta con los expedientes físicos.

### 4.3 IDENTIFICACIÓN Y VALIDACIÓN DE CATÁLOGOS

Los datos históricos contenían algunos datos sucios, es decir, datos que “tienen una estructura incorrecta, numerosas imprecisiones o simplemente están incompletos” (Tableau, s.f.), los cuales pudieron ser generados por errores en la captura, cambio de requerimientos, cambio de políticas para almacenar los datos, entre otros.

Como resultado del análisis exploratorio se identificó que la información era de captura libre, sin considerar catálogos o validaciones, lo cual dificultaba la generación de reportes por medio de filtros, por ejemplo, tenía diferentes direcciones para la misma ubicación como se puede ver en la figura 1.

#### Figura 1

*Ejemplo de nombres de lugar diferentes para la misma ubicación*

|  |
|--|
| OFICINAS DE LA RECTORÍA DE LA UNAM                     |
| OFICINAS DE RECTORÍA                                   |
| OFICINAS DE RECTORIA                                   |
| RECTORÍA DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO |

En esta etapa se identificaron los datos que eran candidatos a ser catálogos, los cuales se revisaron, modificaron y validaron con la ayuda y experiencia de los usuarios del sistema.

Se elaboró la equivalencia que tenía el dato original en el archivo de hoja de cálculo con el valor final en la base de datos aprobado por la oficina del cuerpo colegiado, con la finalidad de homologar los datos y facilitar la generación de estadísticas y reportes. Un ejemplo de esto se puede ver en la figura 2.

## Figura 2

*Ejemplo de correspondencia de nombres de lugar diferentes y la ubicación aprobada por el usuario*

|   |                                    |
|---|------------------------------------|
| OFICINAS DE LA RECTORÍA DE LA UNAM                        | OFICINAS DE LA RECTORÍA DE LA UNAM |
| OFICINAS DE RECTORÍA                                      | OFICINAS DE LA RECTORÍA DE LA UNAM |
| OFICINAS DE RECTORIA                                      | OFICINAS DE LA RECTORÍA DE LA UNAM |
| RECTORÍA DE LA UNIVERSIDAD NACIONAL<br>AUTÓNOMA DE MÉXICO | OFICINAS DE LA RECTORÍA DE LA UNAM |

Se insertaron los registros del catálogo de lugares con los valores validados usando el cliente de bases de datos, así como los datos equivalentes entre el valor actual del archivo de hoja de cálculo y el valor revisado en una tabla creada en la base de datos para tal fin, a través del cliente PGAdmin.

Tanto las equivalencias de los lugares como el catálogo final fueron utilizados en la consulta para agregar los datos de las sesiones con los lugares en los que se llevaron a cabo, usando consultas de tipo INSERT as SELECT en scripts de SQL; un ejemplo de esta consulta se puede ver en el Anexo D.

Del mismo modo se identificaron 12 catálogos más, mediante consultas para obtener los valores distintos de los datos capturados de los expedientes y en tres de ellos con las equivalencias revisadas manualmente y validadas por el usuario del sistema. Estos catálogos tienen en su mayoría menos de 10 registros, a excepción del catálogo de entidades con 96 registros.

Los scripts mencionados se generaron y ejecutaron con el cliente PGAdmin:

- 10 *scripts* de SQL para crear las tablas usadas para almacenar temporalmente los datos obtenidos de los archivos de hoja de cálculo de expedientes, notas, personas, miembros, comunicados, entre otros, y poblarlas.
- 3 *script* con los valores equivalentes de los catálogos.
- 17 *scripts* de SQL para poblar las tablas principales del sistema como expedientes, procesos, personas, miembros del cuerpo colegiado, entre otras.

Con base en este contexto se realizó un nuevo diseño de bases de datos para el sistema de consulta, así como un proceso de limpieza y carga de los registros.

## 4.4 DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos consistió en la identificación de las tablas necesarias para almacenar los datos de los expedientes, sus procesos, personas relacionadas con los procesos, miembros del cuerpo colegiado, sesiones y registros de asistencia, así como los catálogos y otras tablas necesarias para el funcionamiento del sistema.

Posteriormente se determinaron las relaciones entre las tablas identificadas y se consideraron los requerimientos de los usuarios del sistema acerca del registro de la información, consultas y reportes; se

consideró también el conocimiento obtenido de las características de los datos en el análisis exploratorio y la aplicación de buenas prácticas de diseño de bases de datos, por ejemplo:

- Todas las tablas del diseño de la base de datos tienen llave primaria<sup>2</sup>.
- La integridad relacional se fortaleció usando constraints o restricciones de llave foránea<sup>3</sup>.
- Identificación de los datos obligatorios y establecimiento de restricciones de tipo NOT NULL.
- Identificación de los valores por defecto necesarios.
- Uso de nombres cortos y descriptivos para las tablas y campos.
- Uso de la nomenclatura elegida por el equipo de desarrollo en el nombrado de los objetos de la base de datos consistente en usar nombres cortos, sin usar ñ ni acentos, nombre de la tabla en singular, llave primaria usando id\_nombre\_tabla para identificar el registro; lo anterior se muestra en la figura 3.

### Figura 3

*Ejemplo de la estructura de la tabla lugar y la nomenclatura usada en los nombres del objeto y campos*



- Incorporación de la descripción de cada campo y tabla como parte de su estructura usando el comando COMMENT de postgresQL.
- Normalización<sup>4</sup> de la información que lo requería.
- Identificación de las características de los datos para establecer tipos de datos adecuados al contenido; algunos casos fueron limpiados antes de poder ser agregados, debido a que no tenían valores adecuados al formato o contenido esperados, por ejemplo fechas erróneas, tales como "19 DE ABRIL AL 12 DE NOVIEMBRE DE 1999", lo que no corresponde a una fecha sino a un periodo.
- Documentación del diseño en un diagrama entidad relación, el cual se incorporó a la documentación técnica.

2 En términos generales, la llave primaria (PK) es un atributo (o identificación de atributos) que de manera única identifica a cualquier renglón dado (Coronel et. al., 2011).

3 Una llave foránea (FK) es un atributo cuyos valores corresponden con los valores de la llave primaria de la tabla relacionada (Coronel et. al., 2011).

4 La normalización es un proceso para evaluar y corregir estructuras de tablas a fin de minimizar redundancias de datos, con lo cual se reduce la probabilidad de anomalías de datos (Coronel et. al., 2011).

Se hizo la actualización de la última versión del diagrama y la generación de su diccionario de datos a partir de la estructura de la base de datos. Este proceso, conocido como ingeniería en reversa, incluye el uso de algunas herramientas como DBeaver, Navicat, MySQL Workbench, entre otros. En la herramienta se indican los datos de conexión a la base de datos para generar el diagrama entidad relación o el diccionario de datos a partir de los objetos de la base de datos a la cual se conectó. La obtención del diccionario de datos también se puede realizar por medio de consultas, como se muestra en Anexo E.

Se obtuvo la retroalimentación del equipo de trabajo acerca del diseño de la base de datos, y se incorporaron las mejoras detectadas por el equipo, por ejemplo: relacionar directamente los estudios académicos con los datos generales de la persona, agregar el campo que permite relacionar un miembro con su antecesor para poder generar el reporte cronológico solicitado por el usuario del sistema, entre otros. Finalmente se generó el *script* de SQL para la creación de los objetos como tablas, secuencias y vistas.

## 4.5 REVISIÓN Y VALIDACIÓN DE DATOS DE PERSONAS

Durante el análisis exploratorio se identificó que había oportunidades de mejora en los nombres de las personas que se encontraban en los datos de los expedientes con diferentes roles; por ejemplo, en algunos se tenía el nombre completo, en otros casos se presentaba el nombre con abreviaturas o bien con errores de ortografía. Con la ayuda de la oficina del cuerpo colegiado se determinó cuál era el valor correcto que debía estar en el registro del expediente en el sistema, y con la equivalencia indicada se corrigieron los datos.

## 4.6 PROCESO DE CARGA

Se hizo una relación de los datos que proporcionó el usuario con la tabla y campo en la base de datos, se estableció el orden para ingresar los datos, se identificaron las correcciones a realizar por medio de scripts de SQL, se realizaron pruebas del proceso de carga y limpieza de datos, y finalmente se ejecutó el proceso en el ambiente de producción usando el comando `psql` para ejecutar los scripts.

## 4.7 VERIFICACIÓN DE LOS DATOS

Durante todo el proceso se realizaron revisiones de los datos almacenados en las tablas para verificar:

- Los valores válidos de acuerdo con el formato y características establecidas,
- el número de registros respecto de la información proporcionada por el usuario,
- los valores de texto libre modificados por catálogos aprobados.

Adicionalmente a la revisión de los datos por medio de consultas, el proceso de ingeniería de software incluyó pruebas funcionales; durante este proceso los probadores reportaron incidencias relacionadas con los datos, las cuales fueron analizadas y atendidas. Respecto a las incidencias reportadas se determinó la acción o acciones a seguir, tales como:

- El dato requería alguna modificación o corrección,

- la aplicación tenía que ser actualizada por el desarrollador del módulo reportado para que funcionara adecuadamente con los datos ingresados, o bien,
- era necesaria más información para confirmar los valores adecuados en la base de datos.

## 5. RESULTADOS

Como resultado del proceso de diseño, análisis, limpieza y carga se logró mejorar la calidad de los datos del registro de los expedientes y los datos históricos de los miembros del cuerpo colegiado.

El nuevo diseño de la base de datos está compuesto por 48 tablas. Las tablas principales almacenan los datos generales de los expedientes. Se agregaron los datos de 982 expedientes, 1,203 procesos y la asistencia a 1,195 sesiones. Se verificó, como indicador de éxito de la carga de cada tabla, que la cantidad de registros de la fuente de datos fuera la misma que el número de registros en el nuevo modelo; por ejemplo, si en el archivo de hoja de cálculo se tenían 982 expedientes, esa misma cantidad tenía que estar en la base de datos final. Otro indicador verificado fue que el número de registros en la tabla final en producción debía corresponder con un catálogo y tener la misma cantidad de registros que los valores iniciales sin catálogo.

### Figura 4

*Ejemplo de datos de expediente con valores en entidad, comparado con la tabla final proceso con valores del catálogo de entidades*

```
SELECT count(*) FROM proceso where id_entidad is not null;  
-- 1083  
SELECT count(*) FROM z_expediente_nuevo where trim(entidad) !='';  
-- 1083
```

De acuerdo con los valores verificados, se homologaron y limpiaron los nombres de 1,686 personas relacionadas con el proceso en uno o más roles. Por ejemplo, si el primer apellido de una misma persona estaba escrito de formas distintas, la oficina del cuerpo colegiado revisaba manualmente los datos para indicar cuál era el valor adecuado. Con este dato verificado se limpiaron los datos relacionados con esa persona mediante *scripts* de SQL.

Se identificaron 14 catálogos, entre ellos: lugares, tipos de proceso y entidades. Con apoyo de estos catálogos se asignaron los valores correspondientes para sustituir y estandarizar los datos capturados de forma libre en la base de datos final.

## 6. CONCLUSIONES

Ante la problemática inicial se decidió realizar un nuevo diseño de bases de datos que permitiera realizar las consultas y reportes que le son solicitados al cuerpo colegiado, y llevar a cabo los procesos necesarios para contar con datos poblados de acuerdo con la nueva estructura y con criterios de calidad de datos.



Las mejoras implementadas en el diseño de la base de datos contribuyeron a fortalecer aspectos de calidad de los datos tales como la precisión<sup>5</sup>, al detectar y corregir datos inconsistentes con los valores reales o que no tuvieran el formato adecuado.

El proceso de carga se fortaleció usando algunas prácticas de limpieza de datos, con el fin de mejorar aspectos como la consistencia de la información, el uso de valores correctos mediante la implementación de catálogos, la completitud al evitar la omisión de valores obligatorios, entre otros.

El nuevo diseño y los procesos de limpieza de datos ejecutados en procesos de migración o en carga de datos contribuyeron a que la información obtenida por los usuarios del sistema para hacer consultas, generar reportes y estadísticas se genere a partir de datos más precisos. Por ejemplo, para obtener estadísticas acerca de en cuántas sesiones había participado un miembro del cuerpo colegiado, y dado que sus nombres no estaban estandarizados en las listas de asistencia, se podía considerar como personas distintas a una misma persona. Estos datos fueron estandarizados mediante consultas en *scripts* de SQL a partir del catálogo de miembros y la equivalencia revisada con el usuario del sistema.

Es recomendable revisar los atributos de calidad de datos mencionados en el estándar ISO 25012:2008, identificar los datos relevantes en el proyecto y definir las mejoras necesarias de los datos para cumplir con los atributos de calidad. Adicionalmente, se sugiere incluir procesos de pruebas de los sistemas con los datos con la finalidad de confirmar que funcionan adecuadamente, o en su caso establecer las acciones correctivas necesarias.

Finalmente, respecto al diseño de bases de datos es recomendable establecer lineamientos generales y una nomenclatura para nombrar los objetos en conjunto con el equipo de desarrollo con la finalidad de estandarizar el diseño.

## REFERENCIAS BIBLIOGRÁFICAS

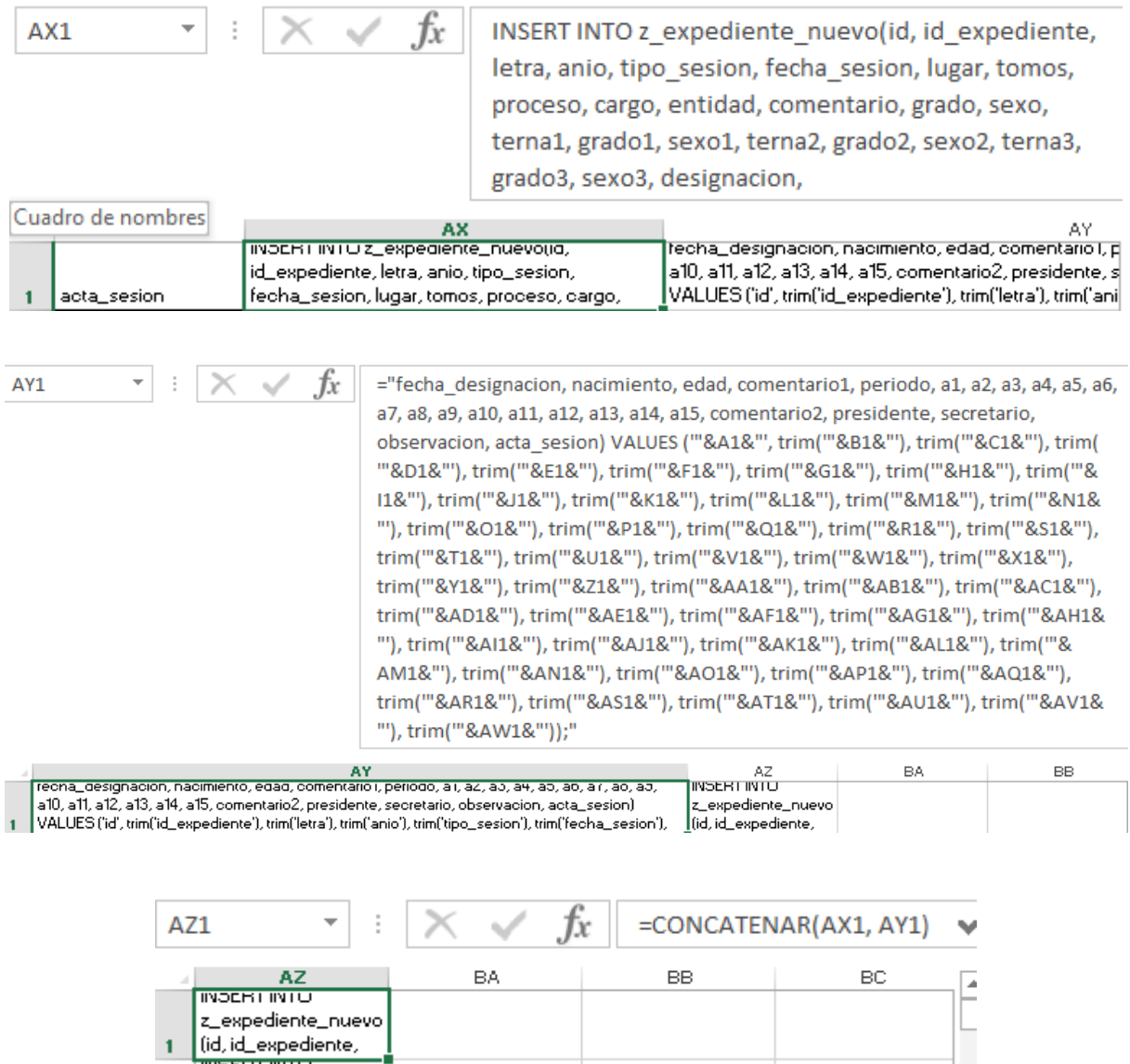
- Coronel, C. et. al. (2011). *Bases de datos. Diseño, implementación y administración*. México, Cengage Learning.
- González, M.F. (2013). Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación. *Revista Facultad de Ingeniería, UPTC*, julio - diciembre de 2013, Vol. 22, No. 35. Revisado el 6 de septiembre de 2023. Obtenido de: <https://revistas.uptc.edu.co/index.php/ingenieria/article/view/2519/2374>
- ISO 25000 (s.f.). *ISO 25000 calidad de software y datos*. Recuperado el 6 de septiembre de 2023 de <https://iso25000.com/index.php/normas-iso-25000/iso-25012>
- Tableau (s.f.). *Los datos "sucios" tienen consecuencias: Cómo solucionar los problemas más comunes de preparación de datos*. Recuperado el 6 de septiembre de 2023 de: <https://www.tableau.com/es-es/learn/whitepapers/costs-of-dirty-data>

5 La precisión es el grado en el cual el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso (González, 2013).

## ANEXO A. EJEMPLO DE LAS FÓRMULAS DE EXCEL

**Figura 5**

Ejemplo de las fórmulas usadas en el archivo de Excel con los expedientes



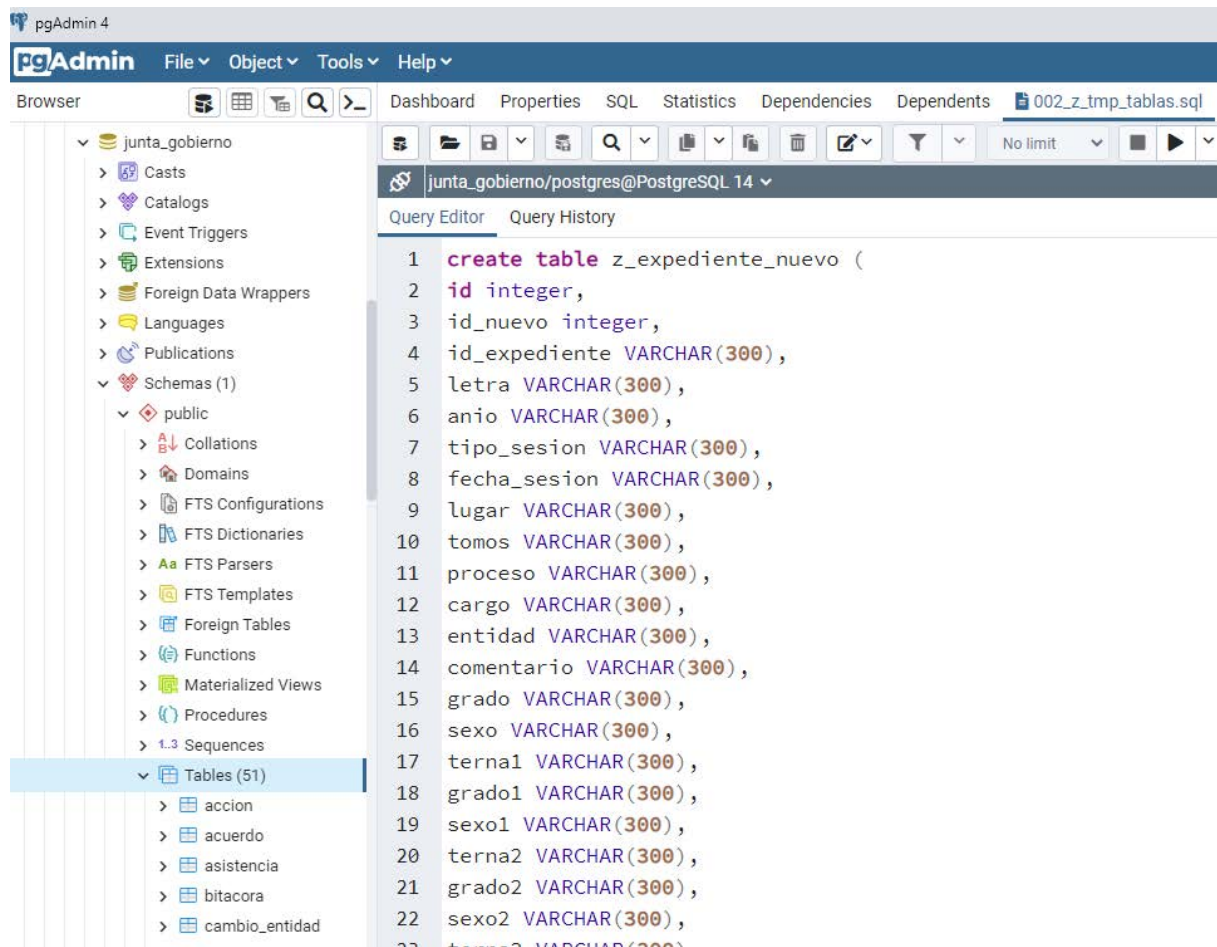
The image shows three examples of Excel formula bars and cell contents:

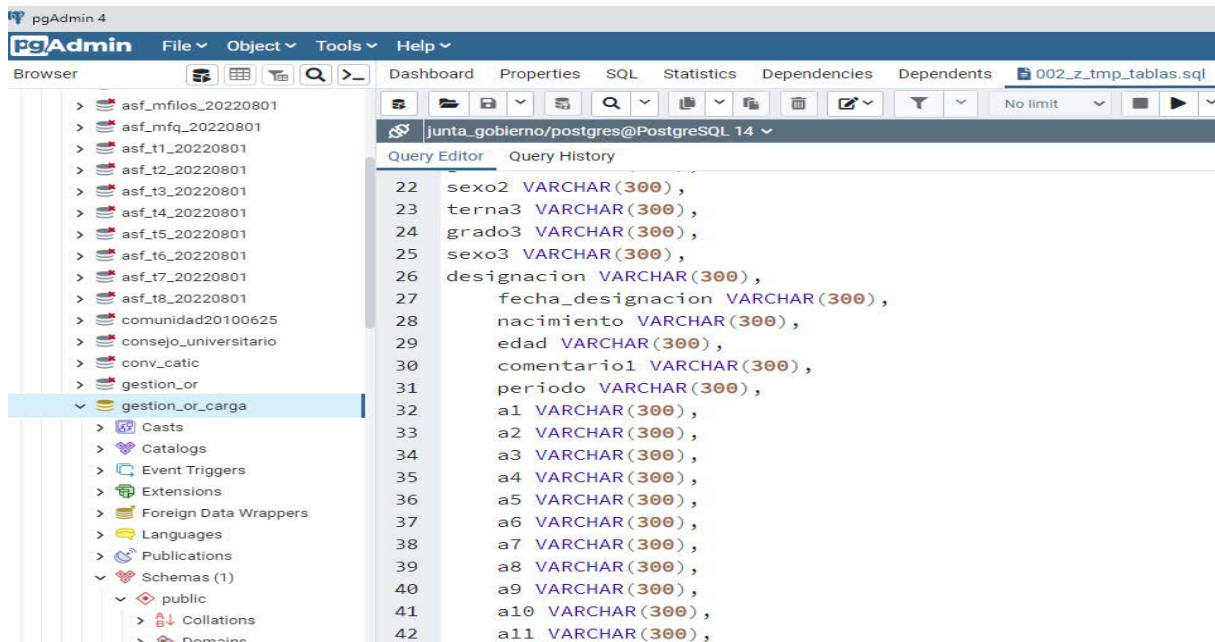
- AX1:** The formula bar contains the text: `INSERT INTO z_expediente_nuevo(id, id_expediente, letra, anio, tipo_sesion, fecha_sesion, lugar, tomos, proceso, cargo, entidad, comentario, grado, sexo, terna1, grado1, sexo1, terna2, grado2, sexo2, terna3, grado3, sexo3, designacion,`
- AY1:** The formula bar contains a long concatenation formula: `= "fecha_designacion, nacimiento, edad, comentario1, periodo, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, comentario2, presidente, secretario, observacion, acta_sesion) VALUES (" & A1 & ", trim(" & B1 & "), trim(" & C1 & "), trim(" & D1 & "), trim(" & E1 & "), trim(" & F1 & "), trim(" & G1 & "), trim(" & H1 & "), trim(" & I1 & "), trim(" & J1 & "), trim(" & K1 & "), trim(" & L1 & "), trim(" & M1 & "), trim(" & N1 & "), trim(" & O1 & "), trim(" & P1 & "), trim(" & Q1 & "), trim(" & R1 & "), trim(" & S1 & "), trim(" & T1 & "), trim(" & U1 & "), trim(" & V1 & "), trim(" & W1 & "), trim(" & X1 & "), trim(" & Y1 & "), trim(" & Z1 & "), trim(" & AA1 & "), trim(" & AB1 & "), trim(" & AC1 & "), trim(" & AD1 & "), trim(" & AE1 & "), trim(" & AF1 & "), trim(" & AG1 & "), trim(" & AH1 & "), trim(" & AI1 & "), trim(" & AJ1 & "), trim(" & AK1 & "), trim(" & AL1 & "), trim(" & AM1 & "), trim(" & AN1 & "), trim(" & AO1 & "), trim(" & AP1 & "), trim(" & AQ1 & "), trim(" & AR1 & "), trim(" & AS1 & "), trim(" & AT1 & "), trim(" & AU1 & "), trim(" & AV1 & "), trim(" & AW1 & ");"`
- AZ1:** The formula bar contains the formula: `=CONCATENAR(AX1, AY1)`. The cell content shows the start of the concatenated text: `INSERT INTO z_expediente_nuevo (id, id_expediente,`

## ANEXO B. EJEMPLO DE USO DE PGADMIN

**Figura 6**

Uso del cliente PGAdmin para ejecutar el script de SQL para crear la tabla que fue usada para insertar posteriormente los datos del archivo de Excel de expedientes para su revisión

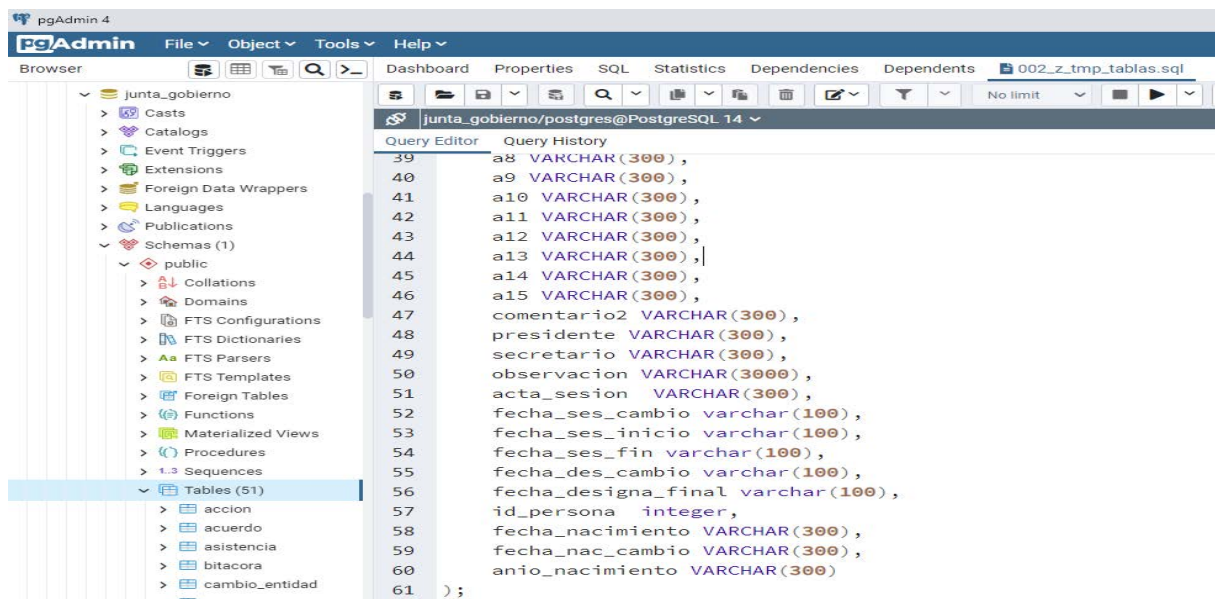




```

22 sexo2 VARCHAR(300),
23 terna3 VARCHAR(300),
24 grado3 VARCHAR(300),
25 sexo3 VARCHAR(300),
26 designacion VARCHAR(300),
27 fecha_designacion VARCHAR(300),
28 nacimiento VARCHAR(300),
29 edad VARCHAR(300),
30 comentario1 VARCHAR(300),
31 periodo VARCHAR(300),
32 a1 VARCHAR(300),
33 a2 VARCHAR(300),
34 a3 VARCHAR(300),
35 a4 VARCHAR(300),
36 a5 VARCHAR(300),
37 a6 VARCHAR(300),
38 a7 VARCHAR(300),
39 a8 VARCHAR(300),
40 a9 VARCHAR(300),
41 a10 VARCHAR(300),
42 a11 VARCHAR(300),

```



```

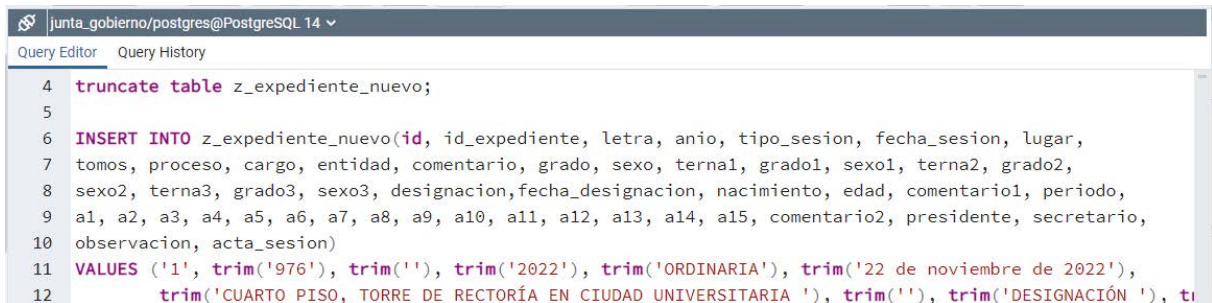
39 a8 VARCHAR(300),
40 a9 VARCHAR(300),
41 a10 VARCHAR(300),
42 a11 VARCHAR(300),
43 a12 VARCHAR(300),
44 a13 VARCHAR(300),
45 a14 VARCHAR(300),
46 a15 VARCHAR(300),
47 comentario2 VARCHAR(300),
48 presidente VARCHAR(300),
49 secretario VARCHAR(300),
50 observacion VARCHAR(3000),
51 acta_sesion VARCHAR(300),
52 fecha_ses_cambio varchar(100),
53 fecha_ses_inicio varchar(100),
54 fecha_ses_fin varchar(100),
55 fecha_des_cambio varchar(100),
56 fecha_designa_final varchar(100),
57 id_persona integer,
58 fecha_nacimiento VARCHAR(300),
59 fecha_nac_cambio VARCHAR(300),
60 anio_nacimiento VARCHAR(300)
61 );

```

*Nota:* La instalación de postgresQL incluye el cliente PGAdmin y se puede descargar del sitio <https://www.postgresql.org/download/>

## Figura 7

Uso del cliente PGAdmin para ejecutar scripts de SQL para insertar datos



```
4 truncate table z_expediente_nuevo;
5
6 INSERT INTO z_expediente_nuevo(id, id_expediente, letra, anio, tipo_sesion, fecha_sesion, lugar,
7 tomos, proceso, cargo, entidad, comentario, grado, sexo, terna1, grado1, sexo1, terna2, grado2,
8 sexo2, terna3, grado3, sexo3, designacion, fecha_designacion, nacimiento, edad, comentario1, periodo,
9 a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, comentario2, presidente, secretario,
10 observacion, acta_sesion)
11 VALUES ('1', trim('976'), trim(''), trim('2022'), trim('ORDINARIA'), trim('22 de noviembre de 2022'),
12 trim('CUARTO PISO, TORRE DE RECTORÍA EN CIUDAD UNIVERSITARIA '), trim(''), trim('DESIGNACIÓN '), tr
```

## ANEXO C. EJEMPLO DE CONSULTA PARA INSERTAR SESIONES

### Figura 8

Ejemplo de la consulta usada para insertar los datos de las sesiones

```
INSERT INTO sesion(id_expediente, id_lugar, fecha)
SELECT DISTINCT e.id_expediente, id_lugar, ze.fecha_ses_inicio::date
FROM expediente e JOIN z_expediente_nuevo ze on (e.id_expediente = ze.id_nuevo)
LEFT JOIN
z_tmp_eq_lugar zl ON
(REPLACE(TRIM(LOWER(zl.lugar_s)), ' ', '') = REPLACE(TRIM(LOWER(ze.lugar)), ' ', ''))
LEFT JOIN
lugar l on (REPLACE(TRIM(LOWER(zl.lugar_c)), ' ', '') = REPLACE(TRIM(LOWER(l.nombre) ), ' ', ''))
ORDER BY id_expediente
;
```

## ANEXO D. EJEMPLO DE CONSULTA PARA REVISAR FECHAS

### Figura 9

*Ejemplo de la consulta usada para revisar las fechas de designación y término de un miembro con sus fechas de asistencia mínima y máxima en los campos 1 y 2 de la lista de asistencia*

```
SELECT '1', min(fecha_ses_inicio::date) as fec_minima, max(fecha_ses_fin::date) as fec_maxima,
m.id_miembro , p.nombre, p.primer_apellido, p.segundo_apellido, m.fecha_designacion,
    m.toma_protesta, m.fecha_termino
FROM z_expediente_nuevo ze JOIN z_tmp_eq_asistente_miembro za
    ON (trim(ze.a1)= za.asistente) JOIN miembro m ON (za.id_miembro = m.id_miembro) JOIN
    sesion s ON (ze.id = s.id_sesion) JOIN persona p ON (p.id_persona = m.id_persona)
    group by m.id_miembro , m.fecha_designacion, m.toma_protesta, m.fecha_termino,
    p.nombre, p.primer_apellido, p.segundo_apellido

UNION

SELECT '2', min(fecha_ses_inicio::date) as fec_minima, max(fecha_ses_fin::date) as fec_maxima,
m.id_miembro , p.nombre, p.primer_apellido, p.segundo_apellido, m.fecha_designacion,
    m.toma_protesta, m.fecha_termino
FROM z_expediente_nuevo ze JOIN z_tmp_eq_asistente_miembro za
    ON (trim(ze.a2)= za.asistente) JOIN miembro m ON (za.id_miembro = m.id_miembro) JOIN
    sesion s ON (ze.id = s.id_sesion) JOIN persona p ON (p.id_persona = m.id_persona)
    group by m.id_miembro , m.fecha_designacion, m.toma_protesta, m.fecha_termino,
    p.nombre, p.primer_apellido, p.segundo_apellido
```

## ANEXO E. EJEMPLO DE CONSULTA PARA DICCIONARIO DE DATOS

**Figura 10**

*Ejemplo de la consulta y descripción de los campos obtenida de la base de datos en postgresQL excluyendo las tablas de configuración de Laravel*

```
select t1.table_name as tabla_nombre ,
       t1.COLUMN_NAME AS columna_nombre,
       t1.COLUMN_DEFAULT AS columna_defecto,
       t1.IS_NULLABLE AS columna_nulo,
       t1.DATA_TYPE AS columna_tipo_dato,
       COALESCE(t1.NUMERIC_PRECISION,
               t1.CHARACTER_MAXIMUM_LENGTH) AS columna_longitud,
       PG_CATALOG.COL_DESCRIPTION(t2.OID,|
       t1.DTD_IDENTIFIER::int) AS columna_descripcion
FROM
  INFORMATION_SCHEMA.COLUMNS t1
  INNER JOIN PG_CLASS t2 ON (t2.RELNAME = t1.TABLE_NAME)
WHERE
  t1.TABLE_SCHEMA = 'public' AND t1.table_name NOT IN
  ('failed_jobs', 'role_has_permissions', 'roles', 'sessions',
   'flyway_schema_history', 'jobs, migrations', 'model_has_permissions',
   'model_has_roles', 'password_resets', 'permissions', 'personal_access_tokens',
   'jobs', 'migrations')
ORDER by
  t1.table_name ,
  t1.ORDINAL_POSITION;
```

| tabla_nombre | columna_nombre | columna_defec | columna_nu | columna_tipo_dato           | columna_longitud | columna_descripcion  |
|--------------|----------------|---------------|------------|-----------------------------|------------------|--|
| lugar        | id_lugar       |               | NO         | integer                     |                  | Identificador del lugar.   |
| lugar        | nombre         |               | NO         | character varying           | 200              | Nombre del lugar donde se realizó la sesión de la Junta de Gobierno. |
| lugar        | created_at     | now()         | NO         | timestamp without time zone |                  | Fecha y hora de creación del registro.                               |
| lugar        | updated_at     |               | YES        | timestamp without time zone |                  | Fecha y hora de última modificación del registro.                    |