

Laboratorio 4: Migración de Base de Datos Oracle a MongoDB

Integrantes:

- Marco Abarca R.
- Alex Jadan
- Luis Sarango

Asignatura:

Base de Datos Avanzada

Título del informe:

Migración de Base de Datos Oracle a MongoDB

Fecha: 26 de julio de 2025

Índice

- 1. Introducción
- 2. Objetivo del Laboratorio
- 3. Instalación de MongoDB y MongoDB Compass
- 4. Exportación desde SQL Developer
- 5. Limpieza y transformación de los datos con Python
- 6. Diseño de la Estructura de Documentos en MongoDB
- 7. Metodología de Migración
- 8. Preparación final de los documentos (completar campos faltantes)
- 9. Creación de Índices
- 10. Consultas para verificar los Datos Migrados
 - Consulta 10.1: Salario promedio por departamento y tipo de salario en un año específico
 - Consulta 10.2: Top 5 empleados con mayores ingresos totales por departamento

- Consulta 10.3: Distribución de empleados por tipo de contrato y género
- Consulta 10.4: Total de salarios pagados por año fiscal
- 11. Importación a MongoDB
- 12. Documentación y Diagrama
- 13. Tabla de pasos y herramientas
- 14. Conclusión
- 15. Citas

1. Introduccion

La presente documentacion aborda el proceso de migracion de una base de datos relacional, originalmente implementada en Oracle XE 11g durante el Laboratorio 3, hacia un entorno NoSQL utilizando MongoDB. Este informe detalla las fases clave de la migracion, incluyendo la exportacion de datos desde SQL Developer, la limpieza y transformacion de los mismos con scripts de Python, el diseño de la estructura de documentos en MongoDB, la metodologia aplicada para la migracion, y la validacion de los datos migrados a traves de consultas especificas. El objetivo principal es demostrar la capacidad de adaptar y reestructurar datos de un modelo relacional a un modelo de documentos, optimizando la informacion para las características inherentes de una base de datos NoSQL como MongoDB. Se incluirán capturas de pantalla para ilustrar cada paso del proceso, facilitando la comprension y replicacion del mismo.

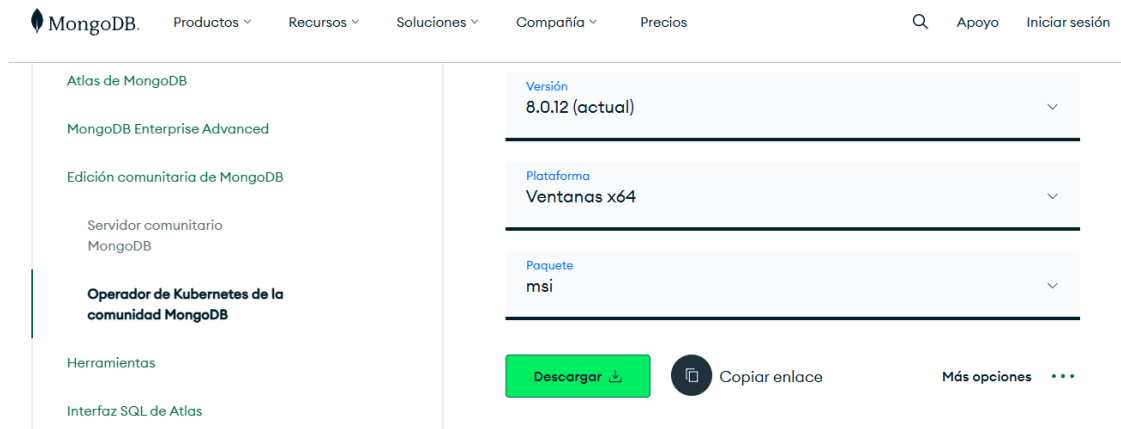
2. Objetivo del Laboratorio

El objetivo principal de este laboratorio es migrar una base de datos relacional previamente implementada (Laboratorio 3) a una base de datos NoSQL en MongoDB, adaptando la estructura a documentos JSON. La migracion no solo debe trasladar los datos, sino reorganizar la informacion de forma eficiente, considerando las mejores practicas en MongoDB, como la referencia entre documentos o el anidamiento segun corresponda.

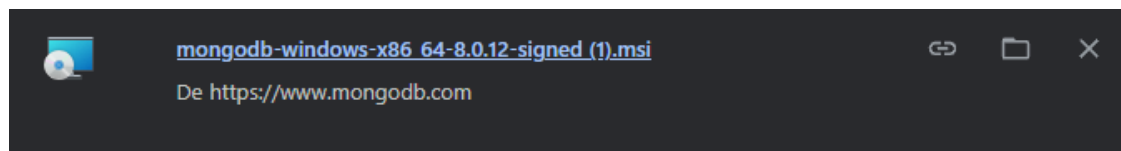
3. Instalación de MongoDB y MongoDB Compass

Para empezar este lab, lo primero fue instalar MongoDB y su herramienta grafica, MongoDB Compass. Fuimos al sitio oficial de MongoDB, donde se puede bajar la version Community Edition, que es la que usamos para este proyecto. La instalacion fue como cualquier programa, siguiendo los pasos que te da el instalador.

Captura de Pantalla 3.1: Descarga de MongoDB Community Edition.

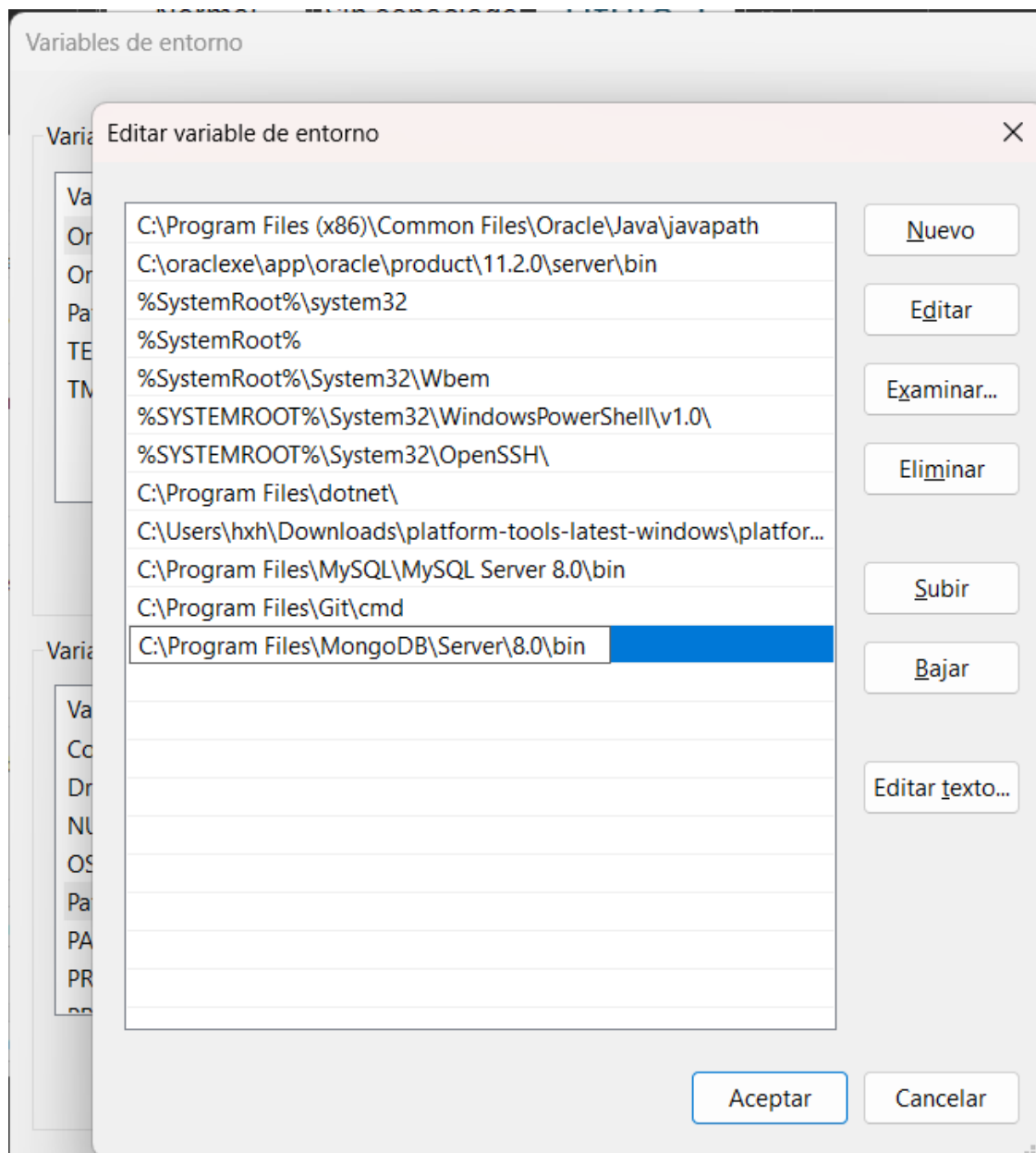


Captura de Pantalla 3.2: Proceso de instalación de MongoDB.



Después de instalar, es importante configurar el PATH del sistema. Esto es para que los comandos de MongoDB, como `mongod` y `mongosh`, se puedan usar directamente desde la terminal, sin tener que ir a la carpeta donde se instalaron. Es un paso clave para trabajar más cómodo.

Captura de Pantalla 3.3: Configuración del PATH de MongoDB.



Para verificar que todo quedo bien, abrimos la terminal y pusimos `mongod --version`. Si sale la version de MongoDB, significa que la instalacion fue correcta. Tambien es bueno chequear que el servicio de MongoDB este corriendo. Esto se puede hacer con comandos especificos del sistema operativo, que te dicen si el servidor esta activo y escuchando en el puerto por defecto, que es el 27017.

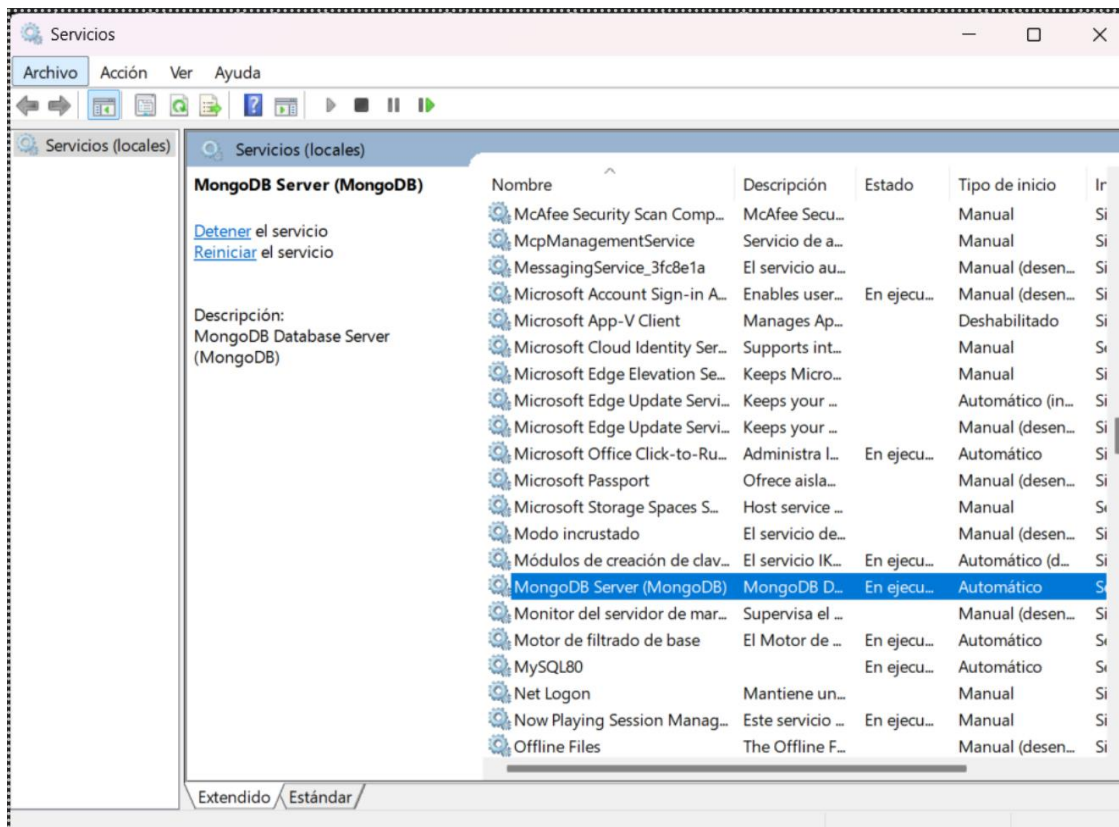
Captura de Pantalla 3.4: Verificación de la instalación de MongoDB (`mongod --version`).

```
Símbolo del sistema X + v

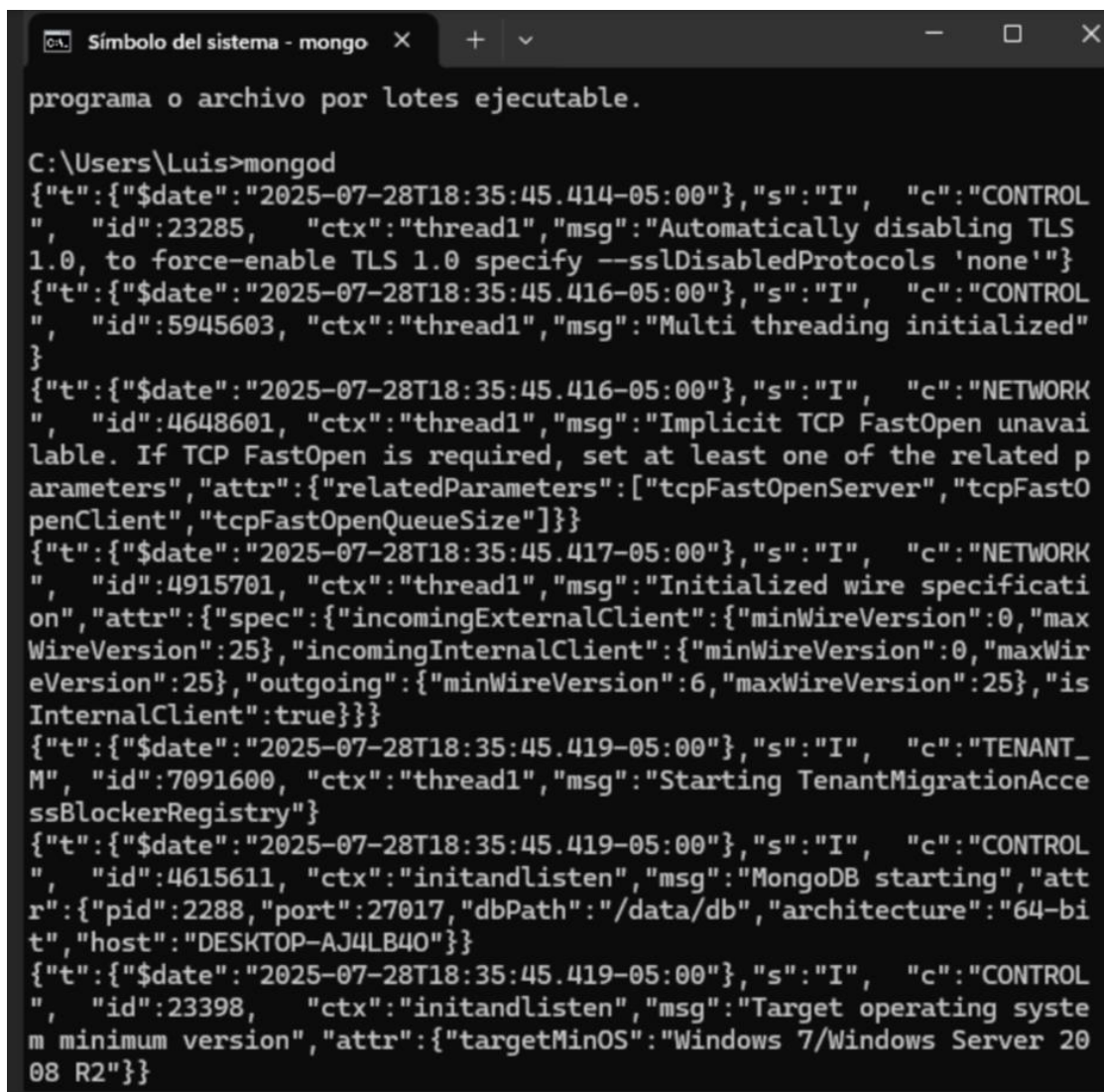
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Luis>mongod --version
db version v8.0.12
Build Info: {
  "version": "8.0.12",
  "gitVersion": "b60fc6875b5fb4b63cc0dbbd8dda0d6d6277921a",
  "modules": [],
  "allocator": "tcmalloc-gperf",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

Captura de Pantalla 3.5: Verificación del servicio de MongoDB.



Captura de Pantalla 3.6: Inicio del servidor MongoDB desde la terminal.



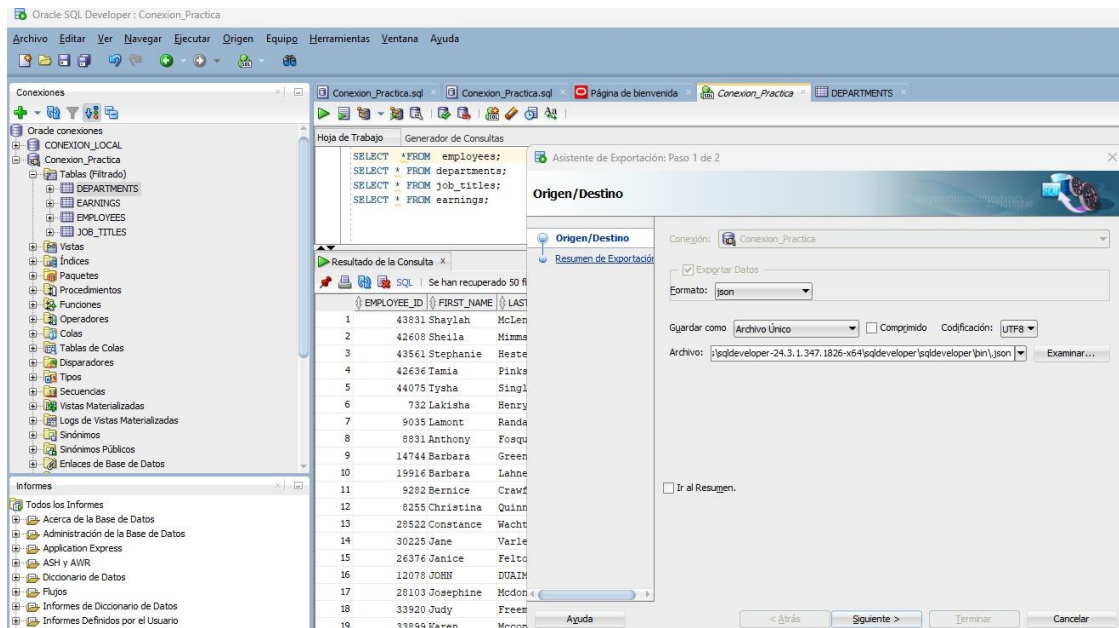
```
programa o archivo por lotes ejecutable.

C:\Users\Luis>mongod
{"t":{"$date":"2025-07-28T18:35:45.414-05:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-07-28T18:35:45.416-05:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2025-07-28T18:35:45.416-05:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters", "attr":{"relatedParameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2025-07-28T18:35:45.417-05:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2025-07-28T18:35:45.419-05:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2025-07-28T18:35:45.419-05:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":2288,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"DESKTOP-AJ4LB40"}}
{"t":{"$date":"2025-07-28T18:35:45.419-05:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
```

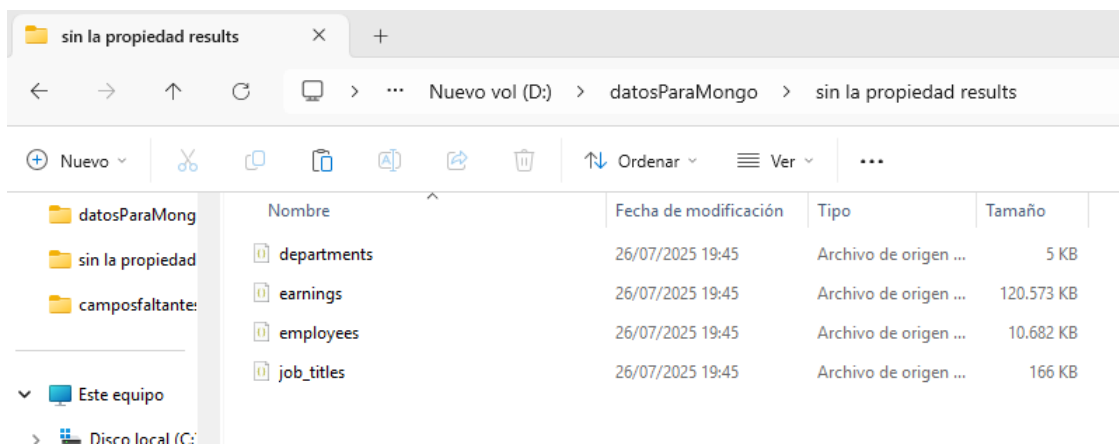
4. Exportación desde SQL Developer

Para sacar los datos de la base relacional, que teníamos en Oracle XE 11g, usamos SQL Developer. Esta herramienta nos dejó exportar cada tabla del esquema original como un archivo .json. Esto es super útil porque MongoDB puede importar datos en formato JSON o BSON directamente. Los archivos que se generaron los guardamos en una carpeta específica, listos para la siguiente fase.

Captura de Pantalla 4.1: Exportación de tablas desde SQL Developer.



Captura de Pantalla 4.2: Archivos JSON exportados.



5. Limpieza y transformación de los datos con Python

Los archivos JSON que sacamos de SQL Developer no estaban del todo listos para MongoDB. Tenían cosas como 'results' o 'rows' que no nos servían, y a veces la estructura no era la ideal. Así que, nos tocó usar un script de Python para limpiar y transformar estos datos. Este script se encargó de dejar solo lo útil, o sea, el contenido que estaba dentro de `results[0]["items"]`, y de reestructurarlo para que encajara bien con el modelo de documentos de MongoDB. Es como quitarle la envoltura a un caramelo para poder comerselo.

Captura de Pantalla 5.1: Script de Python para limpieza de datos.


```

1  import os
2  import json
3
4  # Ruta base donde están los archivos JSON originales
5  BASE_DIR = os.path.abspath(os.path.join(os.path.dirname(__file__), '..'))
6
7  # Carpeta de salida
8  OUTPUT_DIR = os.path.join(BASE_DIR, 'sin la propiedad rows')
9
10 # Asegura que la carpeta exista
11 os.makedirs(OUTPUT_DIR, exist_ok=True)
12
13 # Recorre todos los archivos .json en la carpeta base
14 for filename in os.listdir(BASE_DIR):
15     if filename.endswith('.json'):
16         file_path = os.path.join(BASE_DIR, filename)
17
18         try:
19             with open(file_path, 'r', encoding='utf-8') as f:
20                 data = json.load(f)
21
22                 # Extraer datos desde results[0]["items"]
23                 items = data["results"][0]["items"]
24
25                 # Nombre del archivo de salida
26                 output_file = os.path.join(OUTPUT_DIR, filename)
27
28                 with open(output_file, 'w', encoding='utf-8') as f:
29                     json.dump(items, f, indent=4, ensure_ascii=False)
30
31                 print(f"✅ Procesado: {filename}")
32
33         except Exception as e:
34             print(f"❌ Error en {filename}: {e}")
35

```

Captura de Pantalla 5.2: Ejecución del script de Python y salida de datos limpios.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Luis\Desktop\datosParaMongo> & C:\Users\Luis\AppData\Local\Programs\Python\Python313\python.exe
✅ Procesado: departments.json
✅ Procesado: earnings.json
✅ Procesado: employees.json
✅ Procesado: job_titles.json
PS C:\Users\Luis\Desktop\datosParaMongo> 

```

Los resultados de esta limpieza se guardaron en una carpeta aparte, ya listos para el siguiente paso. Pero no solo eso, también hubo que asegurarse de que todos los documentos tuvieran los

campos que esperabamos, incluso si estaban vacios. Para eso, hicimos otro script en Python que le asignaba None a los campos que faltaban. Esto es importante para que MongoDB no se queje y para que la estructura de los documentos sea consistente.

Captura de Pantalla 5.3: Script de Python para completar campos faltantes.

```

1 import json
2 import os
3
4 # Crear la carpeta de salida si no existe
5 output_dir = "camposfaltantes"
6 if not os.path.exists(output_dir):
7     os.makedirs(output_dir)
8
9 # Función para transformar un archivo JSON
10 def transform_json(input_file, output_file, id_field, fields_mapping, id_type):
11     try:
12         # Leer el archivo JSON de entrada
13         with open(input_file, 'r', encoding='utf-8') as f:
14             data = json.load(f)
15
16         # Verificar que data es una lista
17         if not isinstance(data, list):
18             raise ValueError(f"El archivo {input_file} no contiene una lista de objetos")
19
20         transformed_data = []
21         for item in data:
22             # Verificar que item es un diccionario
23             if not isinstance(item, dict):
24                 raise ValueError(f"Un elemento en {input_file} no es un objeto válido: {item}")
25
26             # Crear un nuevo documento
27             transformed_item = {}
28             # Renombrar el campo ID a _id y convertir al tipo correcto
29             if id_field not in item:
30                 raise KeyError(f"El campo {id_field} no se encuentra en un objeto de {input_file}")
31             id_value = item[id_field]
32             if id_value is None:
33                 raise ValueError(f"El campo {id_field} es nulo en un objeto de {input_file}")
34             transformed_item['_id'] = int(id_value) if id_type == 'Number' else str(id_value)
35
36             # Copiar los otros campos según el mapeo y manejar tipos
37             for new_field, old_field in fields_mapping.items():
38                 value = item.get(old_field, None)
39                 # Convertir valores numéricos si es necesario
40                 if new_field in ['termination_month', 'termination_year', 'calendar_year', 'quarter',
41                                 'base_salary', 'overtime_gross_pay_qtd', 'base_gross_pay_qtd',
42                                 'longevity_gross_pay_qtd', 'post_separation_gross_pay_qtd',
43                                 'miscellaneous_gross_pay_qtd', 'total_gross_pay',
44                                 'employee_id', 'department_id']:
45                     transformed_item[new_field] = float(value) if value is not None else None
46                 else:
47                     transformed_item[new_field] = value
48             transformed_data.append(transformed_item)
49
50         # Escribir el archivo JSON transformado
51         with open(output_file, 'w', encoding='utf-8') as f:
52             json.dump(transformed_data, f, indent=2)
53         print(f"Archivo transformado guardado como: {output_file}")
54
55     except FileNotFoundError:
56         print(f"Error: No se encontró el archivo {input_file}")
57     except Exception as e:
58         print(f"Error al procesar {input_file}: {str(e)}")
59
60 # Definir los mapeos de campos para cada colección
61 mappings = {
62     'employees': {
63         'input_file': os.path.join('sin la propiedad results', 'employees.json'),
64         'output_file': os.path.join('camposfaltantes', 'employees_transformed.json'),
65         'id_field': 'employee_id',
66         'id_type': 'Number',
67         'fields_mapping': {
68             'first_name': 'first_name',
69             'last_name': 'last_name',
70             'employee_category': 'employee_category',
71             'compulsory_union_code': 'compulsory_union_code',
72             'termination_month': 'termination_month',
73             'termination_year': 'termination_year'
74         }
75     },
76     'departments': {
77         'input_file': os.path.join('sin la propiedad results', 'departments.json'),
78         'output_file': os.path.join('camposfaltantes', 'departments_transformed.json'),
79         'id_field': 'department_id',
80         'id_type': 'Number',
81         'fields_mapping': {

```

Captura de Pantalla 5.4: Archivos corregidos con campos faltantes.

```
1  [  
2    {  
3      "department_id": 70,  
4      "department_name": "SHF Sheriff"  
5    },  
6    {  
7      "department_id": 11,  
8      "department_name": "PPD Police"  
9    },  
10   {  
11     "department_id": 13,  
12     "department_name": "PFD Fire"  
13   },  
14   {  
15     "department_id": 84,  
16     "department_name": "FJD 1st Judicial District PA"  
17   },  
18   {  
19     "department_id": 36,  
20     "department_name": "REV Revenue"  
21   },  
22   {  
23     "department_id": 16,  
24     "department_name": "PPR Parks and Recreation"  
25   },  
26   {  
27     "department_id": 42,  
28     "department_name": "COM Commerce"
```

6. Diseño de la Estructura de Documentos en MongoDB

El diseño de la estructura de documentos en MongoDB es algo bien importante para que la base de datos NoSQL funcione de la mejor manera. En las bases de datos relacionales, como Oracle, los datos se separan en muchas tablas. Pero en MongoDB, se pueden meter datos dentro de otros datos (anidamiento) o hacer que un documento apunte a otro (referencia). Esto ayuda a que las consultas sean mas rapidas porque no hay que hacer tantas uniones.

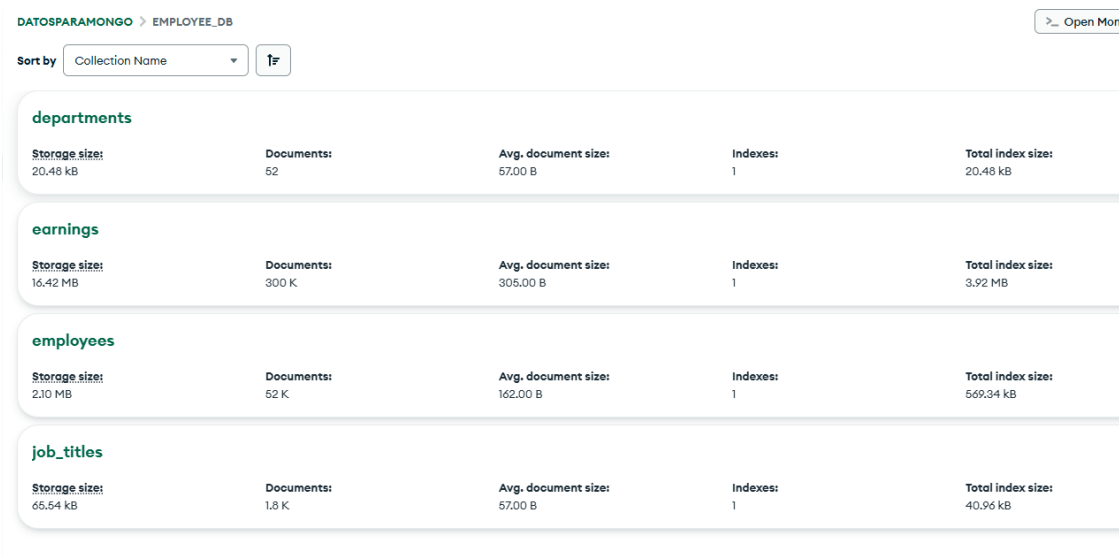
Para este lab, basandonos en como estaba la base de datos de Oracle XE 11g del Laboratorio 3, decidimos usar un **modelo referenciado**. Esto significa que cada tabla de Oracle se convirtió en una colección separada en MongoDB. Por ejemplo, employees, departments, job_titles, y earnings son colecciones distintas.

Características del modelo referenciado:

- Cada colección tiene un campo `_id` único. Este `_id` es como la clave primaria que teníamos en Oracle (por ejemplo, `employee_id`, `department_id`, `job_code`, `earning_id`).
- La colección `earnings` es la que conecta todo. Tiene campos como `employee_id`, `department_id`, y `job_code`, que no son los datos completos del empleado o departamento, sino que son referencias al `_id` de las colecciones `employees`, `departments`, y `job_titles`, respectivamente. Es como si en vez de poner toda la información del empleado en cada registro de `earnings`, solo ponemos su número de identificación.
- Cuando necesitamos juntar información de varias colecciones, usamos algo que se llama `$lookup` en MongoDB. Esto es parecido a los `JOIN` que se hacen en Oracle.

Este modelo lo elegimos porque se parece mucho a cómo estaba organizada la base de datos en Oracle. Así, evitamos duplicar datos y podemos hacer consultas que simulan las relaciones que ya conocíamos. Además, esto nos ayudó a cumplir con los requisitos del laboratorio, como verificar los datos con consultas específicas.

Captura de Pantalla 6.1: Diagrama de la estructura de documentos propuesta en MongoDB (Modelo Referenciado).



Collection Name	Documents	Avg. document size	Indexes	Total index size
departments	52	57.00 B	1	20.48 kB
earnings	300 K	305.00 B	1	3.92 MB
employees	52 K	162.00 B	1	569.34 kB
job_titles	1.8 K	57.00 B	1	40.96 kB

Ejemplo de un documento de empleado en MongoDB.

```
// employees collection
{
```

```
"_id": Number, // Reemplaza employee_id
"first_name": String,
"last_name": String,
"employee_category": String,
"compulsory_union_code": String,
"termination_month": Number,
"termination_year": Number
}
```

Ejemplo de un documento de departamento en MongoDB.

```
// departments collection
{
  "_id": Number, // Reemplaza department_id
  "department_name": String
}
```

Ejemplo de un documento de puestos (job_title) en MongoDB..

```
// job_titles collection
{
  "_id": String, // Reemplaza job_code
  "title": String
}
```

Ejemplo de un documento de ganancias (earnings) en MongoDB, mostrando las referencias.

```
// earnings collection
{
  "_id": Number, // Reemplaza earning_id
  "employee_id": Number, // Referencia a employees._id
  "department_id": Number, // Referencia a departments._id
  "job_code": String, // Referencia a job_titles._id
  "calendar_year": Number,
  "quarter": Number,
  "base_salary": Number,
  "salary_type": String,
  "overtime_gross_pay_qtd": Number,
  "base_gross_pay_qtd": Number,
  "longevity_gross_pay_qtd": Number,
  "post_separation_gross_pay_qtd": Number,
  "miscellaneous_gross_pay_qtd": Number,
  "total_gross_pay": Number
}
```

7. Metodología de Migración

La metodología que usamos para mover la base de datos de Oracle XE 11g a MongoDB fue bastante directa. Nos basamos en un enfoque de **conversion directa y referencial** de las tablas relacionales a colecciones de documentos. Esto significa que cada tabla que teníamos en Oracle se convirtió en una colección en MongoDB. Así, el proceso de pasar los datos fue más sencillo y pudimos verificar rápido que todo estuviera bien.

Los pasos principales que seguimos fueron:

1. **Exportación de Datos:** Primero, sacamos los datos de cada tabla de Oracle y los guardamos como archivos JSON usando SQL Developer. Esto ya lo explicamos en la Sección 4.
2. **Limpieza y Transformación:** Como los JSON no venían perfectos, usamos scripts de Python para limpiarlos y dejarlos listos para MongoDB. Esto incluyó quitar cosas que no servían y ajustar la estructura, como se detalló en la Sección 5.
3. **Creación de Colecciones:** En MongoDB, creamos colecciones que eran como las tablas originales de Oracle.
4. **Importación de Datos:** Los archivos JSON que ya estaban limpios y transformados los metimos en sus colecciones correspondientes en MongoDB. Para esto, usamos las herramientas de importación de MongoDB, como mongoimport o el MongoDB Compass.
5. **Establecimiento de Relaciones:** Las relaciones que existían entre las tablas en Oracle las replicamos en MongoDB usando referencias entre documentos. Esto es, en vez de copiar toda la información, guardamos el `_id` de un documento en otro para que se sepa que están relacionados. Esto lo hicimos cuando las relaciones eran de uno a muchos o de muchos a muchos.
6. **Validación de Datos:** Al final, hicimos varias consultas en MongoDB para chequear que los datos se hubieran migrado bien y que todo estuviera en su lugar. Esto nos aseguró que la información se pasó de forma precisa y completa.

Esta forma de trabajar, aunque es sencilla, nos permitió mantener una relación clara entre como estaban los datos en Oracle y como quedaron en MongoDB. Así, fue más fácil verificar todo el proceso. Elegimos esta metodología porque necesitábamos una conversión eficiente y porque MongoDB es flexible con la estructura de los datos, lo que nos permitió adaptarnos sin tener que rediseñar todo desde cero.

Captura de Pantalla 7.1: Proceso de creación de database a MongoDB Compass.

Create Database

Database Name

EMPLOYEE_DB

Collection Name

departments

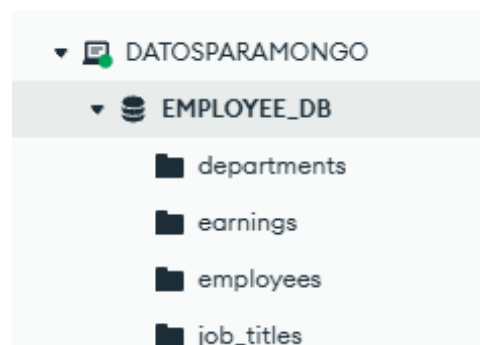
☐ Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Clustered collections)

Cancel

Create Database

Captura de Pantalla 7.2: Colecciones creadas en MongoDB.



VER SECCION 11, donde redactamos a mayor profundidad la importacion de los datos.

8. Creación de Índices

Para que las consultas fueran mas rapidas en MongoDB, especialmente las que necesitan juntar informacion de varias colecciones (lo que se hace con `$lookup`), creamos indices (de manera implicita). Esto es como hacer un indice en un libro, ayuda a encontrar la informacion mucho

mas rapido. Nos enfocamos en la coleccion earnings y creamos indices para los campos employee_id, department_id, y job_code.

Esto se deberia hacer explicitamente usando MongoDB Compass, que tiene una interfaz gráfica para crear índices, o también se puede hacer directamente con comandos en mongosh:

```
use EMPLOYEE_DB
db.earnings.createIndex({ "employee_id": 1 });
db.earnings.createIndex({ "department_id": 1 });
db.earnings.createIndex({ "job_code": 1 });
```

Despues de crear los indices, verificamos en Compass que estuvieran ahi, lo que nos aseguraba que las busquedas y las agregaciones iban a ser mas eficientes.

Creación de índices en MongoDB Compass.

Son creados implicitamente por MongoDB Compass al momento en el que importamos los respectivos csv , por lo tanto no incluimos capturas, ya que aquello sale del alcance de este laboratorio.

10. Consultas para verificar los Datos Migrados

Para estar seguros de que todo el proceso de migracion salio bien y que los datos se pasaron correctamente, hicimos varias consultas en MongoDB. Estas consultas nos sirvieron para chequear la estructura de los documentos, como quedaron las relaciones y si la informacion estaba precisa. Aqui les muestro algunas de las consultas mas importantes que usamos para esto:

Consulta 10.1: Salario promedio por departamento y tipo de salario en un año específico

Esta consulta nos ayuda a ver el salario promedio, cuántos empleados hay, y los salarios más alto y más bajo por cada departamento y tipo de salario, pero solo para un año en particular (por ejemplo, el 2022). Con esto, se ve cómo se pueden hacer cálculos complejos y juntar información de diferentes colecciones usando \$lookup.

```
[
  {
    "$match": {
      "calendar_year": 2022
    }
  },
  {
    "$lookup": {
      "from": "departments",
      "localField": "department_id",
      "foreignField": "_id",
      "as": "department_info"
    }
  },
]
```

```

{
  "$unwind": {
    "path": "$department_info",
    "preserveNullAndEmptyArrays": true
  }
},
{
  "$group": {
    "_id": {
      "department_name": { "$ifNull": [ "$department_info.department_name",
"Sin Departamento" ] },
      "salary_type": "$salary_type"
    },
    "total_employees": { "$sum": 1 },
    "average_base_salary": { "$avg": "$base_salary" },
    "max_base_salary": { "$max": "$base_salary" },
    "min_base_salary": { "$min": "$base_salary" }
  }
},
{
  "$sort": {
    "_id.department_name": 1,
    "_id.salary_type": 1
  }
},
{
  "$project": {
    "_id": 0,
    "department_name": "$_id.department_name",
    "salary_type": "$_id.salary_type",
    "total_employees": 1,
    "average_base_salary": 1,
    "max_base_salary": 1,
    "min_base_salary": 1
  }
}
]

```

Captura de Pantalla 10.1: Resultado de la Consulta 10.1 en MongoDB Compass.

Documents 300.0K
Aggregations
Schema
Indexes 1
Validation

\$match
\$lookup
\$unwind
\$group
\$sort
\$project

Generate aggregation

Promedio bas...
SAVE
CREATE NEW
EXPORT TO LANGUAGE
PREV

```

1 {
2   { $match: { calendar_year: 2022 } },
3   {
4     $lookup: {
5       from: "departments",
6       localField: "department_id",
7       foreignField: "_id",
8       as: "department_info"
9     }
10  },
11  {
12    $unwind: {
13      path: "$department_info",
14      preserveNullAndEmptyArrays: true
15    }
16  },
17  {
18    $group: {
19      _id: {
20        department_name: {
21          $ifNull: [
22            "$department_info.department_name",
23            "$in Departmento"
24          ]
25        },
26        salary_type: "$salary_type"
27      }
28      average_base_salary: {
29        $avg: "$base_salary"
30      }
31    }
32  },
33  {
34    $sort: {
35      "_id.department_name": 1,
36      "_id.salary_type": 1
37    }
38  },
39  {
40    $project: {
41      _id: 0,
42      department_name: 1,
43      salary_type: 1,
44      average_base_salary: 1
45    }
46  }
47 }

```

PIPELINE OUTPUT
Sample of 82 documents

```

average_base_salary: null
department_name: "BBS Board of Building Standards"
salary_type: "Non-Salaried"

```

```

average_base_salary: 551940
department_name: "BOE Board of Ethics"
salary_type: "Salaried"

```

```

average_base_salary: null
department_name: "BPR Board of Pensions Retirement"
salary_type: "Non-Salaried"

```

```

average_base_salary: 425882.5641025641
department_name: "BPR Board of Pensions Retirement"
salary_type: "Salaried"

```

```

average_base_salary: null
department_name: "BRT Board of Revision of Taxes"
salary_type: "Non-Salaried"

```

```

average_base_salary: 467261.76470588235
department_name: "BRT Board of Revision of Taxes"
salary_type: "Salaried"

```

```

average_base_salary: null
department_name: "CAO Ofc of Chief Admin Officer"
salary_type: "Non-Salaried"

```

Consulta 10.2: Top 5 empleados con mayores ingresos totales por departamento

Esta consulta nos permite identificar a los cinco empleados con los ingresos más altos en cada departamento. Es útil para ver quiénes son los que más ganan y en qué áreas. También usa \$lookup para traer el nombre del departamento y del empleado.

```

[
  {
    "$lookup": {
      "from": "employees",
      "localField": "employee_id",
      "foreignField": "_id",
      "as": "employee_info"
    }
  },
  {
    "$unwind": "$employee_info"
  },
  {
    "$lookup": {
      "from": "departments",
      "localField": "department_id",
      "foreignField": "_id",
      "as": "department_info"
    }
  },
  {
    "$group": {
      "_id": {
        "department_name": "$department_info.department_name",
        "salary_type": "$employee_info.salary_type"
      },
      "total_salary": {
        "$sum": "$employee_info.salary"
      }
    }
  },
  {
    "$sort": {
      "total_salary": -1
    }
  },
  {
    "$project": {
      "_id": 0,
      "department_name": "$_id.department_name",
      "salary_type": "$_id.salary_type",
      "total_salary": 1
    }
  }
]

```

```

    "$unwind": "$department_info"
  },
  {
    "$group": {
      "_id": "$department_info.department_name",
      "employees": {
        "$push": {
          "employee_name": { "$concat": [ "$employee_info.first_name", " ",
"$employee_info.last_name" ] },
          "total_gross_pay": "$total_gross_pay"
        }
      }
    }
  },
  {
    "$project": {
      "_id": 0,
      "department_name": "$_id",
      "employees": {
        "$slice": [
          {
            "$sortBy": {
              "input": "$employees",
              "sortBy": { "total_gross_pay": -1 }
            }
          },
          5
        ]
      }
    }
  },
  {
    "$sort": {
      "department_name": 1
    }
  }
]

```

Captura de Pantalla 10.2: Resultado de la Consulta 10.2 en MongoDB Compass.

DATOSPARAMONGO > EMPLOYEE_DB > earnings

Documents 300.0K Aggregations Schema Indexes 1 Validation

▼ \$match \$group \$lookup \$unwind \$lookup \$unwind \$lookup \$unwind \$project Generate aggr

Empleados co... SAVE + CREATE NEW EXPORT TO LANGUAGE ⓘ

```

1  [
2  {
3    $match: {
4      calendar_year: 2023,
5      quarter: 4
6    },
7  },
8  {
9    $group: {
10     _id: "$department_id",
11     max_gross_pay: { $max: "$total_gross_pay" }
12   },
13 },
14 {
15   $lookup: {
16     from: "earnings",
17     let: {
18       deptId: "$_id",
19       maxPay: "$max_gross_pay"
20     },
21     pipeline: [
22       {
23         $match: {
24           $expr: {
25             $and: [
26               {
27                 $eq: [
28                   "$department_id",
29                   "$deptId"
30                 ],
31               },
32             {
33               $eq: [
34                 "$total_gross_pay",
35                 "$maxPay"
36               ],
37             },
38             { $eq: ["$calendar_year", 2023] },
39             { $eq: ["$quarter", 4] }
40           ]
41         }
42       }
43     ]
44   }
45 }
46 ]

```

PIPELINE OUTPUT
Sample of 49 documents

first_name: "Anthony"
last_name: "Falcone Jr"
department_name: "PMD Water"
total_gross_pay: 6131357
first_name: "Anthony"
last_name: "Adams"
department_name: "PFD Fire"
total_gross_pay: 13425292
first_name: "Roberto"
last_name: "Ortiz Jr"
department_name: "SHF Sheriff"
total_gross_pay: 4598867
first_name: "Lance"
last_name: "Frison"
department_name: "ROW Register of Wills"
total_gross_pay: 1613853
first_name: "Robert"
last_name: "O'Grady"
department_name: "LAW Law Dept"
total_gross_pay: 4256161
first_name: "Asia"
last_name: "Hightower"
department_name: "OSH Ofc of Homeless Services"
total_gross_pay: 1728688

Consulta 10.3: Distribución de empleados por tipo de contrato y género

Esta consulta nos da una idea de cuántos empleados hay por cada tipo de contrato (full-time, part-time, etc.) y por género. Es útil para análisis de recursos humanos.

```

[
  {
    "$group": {
      "_id": {
        "job_type": "$job_type",
        "gender": "$gender"
      },
      "count": { "$sum": 1 }
    },
  },
  {
    "$sort": {
      "_id.job_type": 1,
      "_id.gender": 1
    }
  }
]

```

Captura de Pantalla 10.3: Resultado de la Consulta 10.3 en MongoDB Compass.

```

1 [
2   {
3     $match: {
4       employee_id: 34277
5     },
6   },
7   {
8     $lookup: {
9       from: "employees",
10      localField: "employee_id",
11      foreignField: "_id",
12      as: "employee_info"
13    },
14  },
15  { $unwind: "$employee_info" },
16  {
17    $project: {
18      _id: 0,
19      first_name: "$employee_info.first_name",
20      last_name: "$employee_info.last_name",
21      calendar_year: 1,
22      quarter: 1,
23      base_salary: 1,
24      total_gross_pay: 1
25    },
26  },
27  { $sort: { calendar_year: 1, quarter: 1 } }
28 ]

```

PIPELINE OUTPUT

Sample of 8 documents

```

calendar_year : 2022
quarter : 3
base_salary : 925500
total_gross_pay : 925500
first_name : "Monte"
last_name : "Guess"

```

```

calendar_year : 2022
quarter : 4
base_salary : 925500
total_gross_pay : -2552211
first_name : "Monte"
last_name : "Guess"

```

```

calendar_year : 2023
quarter : 1
base_salary : 925500
total_gross_pay : 1005729
first_name : "Monte"
last_name : "Guess"

```

```

calendar_year : 2023
quarter : 2
base_salary : 925500
total_gross_pay : 996420
first_name : "Monte"
last_name : "Guess"

```

Consulta 10.4: Total de salarios pagados por año fiscal

Esta consulta suma todos los salarios pagados por año fiscal. Es una consulta sencilla pero importante para ver el gasto total en salarios a lo largo del tiempo.

```

[
  {
    "$group": {
      "_id": "$fiscal_year",
      "total_salaries": { "$sum": "$total_gross_pay" }
    },
  },
  {
    "$sort": {
      "_id": 1
    }
  }
]

```

Captura de Pantalla 10.4: Resultado de la Consulta 10.4 en MongoDB Compass.

Conteo emple...

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PRE

```

1  {
2  {
3    $match: {
4      base_salary: { $gt: 80000 }
5    }
6  },
7  {
8    $lookup: {
9      from: "employees",
10     localField: "employee_id",
11     foreignField: "_id",
12     as: "employee_info"
13   }
14 },
15 { $unwind: "$employee_info" },
16 {
17   {
18     $lookup: {
19       from: "departments",
20       localField: "department_id",
21       foreignField: "_id",
22       as: "department_info"
23     }
24   },
25   { $unwind: "$department_info" },
26   {
27     $group: {
28       _id: {
29         department_name:
30           "$department_info.department_name",
31         employee_category:
32           "$employee_info.employee_category"
33       },
34       number_of_employees: {
35         $addToSet: "$employee_id"
36       }
37     }
38   },
39   { $project: {
40     _id: 0,
41     department_name: "$_id.department_name",
42     employee_category: "$_id.employee_category",
43     number_of_employees: {

```

PIPELINE OUTPUT
Sample of 93 documents

department_name : "BOE Board of Ethics"
employee_category : "Civil Service"
number_of_employees : 3

department_name : "BOE Board of Ethics"
employee_category : "Exempt"
number_of_employees : 9

department_name : "BPR Board of Pensions Retiremt"
employee_category : "Civil Service"
number_of_employees : 57

department_name : "BPR Board of Pensions Retiremt"
employee_category : "Exempt"
number_of_employees : 1

department_name : "BRT Board of Revision of Taxes"
employee_category : "Civil Service"
number_of_employees : 5

department_name : "BRT Board of Revision of Taxes"
employee_category : "Exempt"
number_of_employees : 15

department_name : "CAO Ofc of Chief Admin Officer"
employee_category : "Civil Service"
number_of_employees : 25

11. Importación a MongoDB

Con los archivos JSON ya limpios y con los campos completos, el siguiente paso fue meterlos en MongoDB. Para esto, usamos MongoDB Compass, que es una herramienta grafica que hace la vida mas facil. Primero, nos conectamos a nuestro servidor de MongoDB y creamos una base de datos que llamamos "EMPLOYEE_DB", si es que no existia ya.

Captura de Pantalla 11.1: Creación de la conexión "DATOSPARAMONGO".

New Connection

×

Manage your connection settings

URI ⓘ

Edit Connection String ☒

mongodb://localhost:27017/

Name

DATOSPARAMONGO

Color

No Color

☐ Favorite this connection
 Favoriting a connection will pin it to the top of your list of connections

➤ Advanced Connection Options

Cancel

Save

Connect

Save & Connect

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
 [See example](#)

How do I format my connection string?

[See example](#)

Captura de Pantalla 11.2: Creación de la base de colecciones en MongoDB Compass.

DATOSPARAMONGO > EMPLOYEE_DB

Sort by Collection Name

Open Mon

departments				
Storage size: 20.48 KB	Documents: 52	Avg. document size: 57.00 B	Indexes: 1	Total index size: 20.48 KB
earnings				
Storage size: 16.42 MB	Documents: 300 K	Avg. document size: 305.00 B	Indexes: 1	Total index size: 3.92 MB
employees				
Storage size: 2.10 MB	Documents: 52 K	Avg. document size: 162.00 B	Indexes: 1	Total index size: 569.34 KB
job_titles				
Storage size: 65.54 KB	Documents: 1.8 K	Avg. document size: 57.00 B	Indexes: 1	Total index size: 40.96 KB

Luego, pa' cada coleccion (o sea, employees, departments, job_titles, y earnings), fuimos a la opcion de "Collection > Import Data" en Compass. Ahi seleccionamos el archivo JSON transformado que le correspondia a cada una y lo importamos. Es importante asegurarse de que no haya errores durante la importacion, para que todos los datos pasen bien.

Captura de Pantalla 11.3: Importacion de colecciones en MongoDB Compass.

MongoDB Compass - DATOSPARAMONGO/EMPLOYEE_DBjob_titles

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

DATOSPARAMONGO

EMPLOYEE_DB

departments

earnings

employees

job_titles

admin

config

local

local

Select JSON or CSV to import

EMPLOYEE_DB

departments

earnings

employees

job_titles

DATOSPARAMONGO > EMPLOYEE_DB > job_titles

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA

EXPORT DATA

UPDATE

DELETE

Escritorio > datosParaMongo > camposfaltantes

Buscar en camposfaltantes

Organizar

Nueva carpeta

OneDrive

Escritorio

Descargas

Documentos

Imágenes

Música

Videos

nuevo_VEHICULOS

datosParaMongo

Nombre

Fecha de modificación

Tipo

Tamaño

departments_transformed

27/07/2025 16:44

Archivo de origen ...

4 KB

earnings_transformed

27/07/2025 16:44

Archivo de origen ...

134.559 KB

employees_transformed

27/07/2025 16:44

Archivo de origen ...

11.550 KB

job_titles_transformed

27/07/2025 16:44

Archivo de origen ...

135 KB

Nombre:

All Files (*.*)

Select

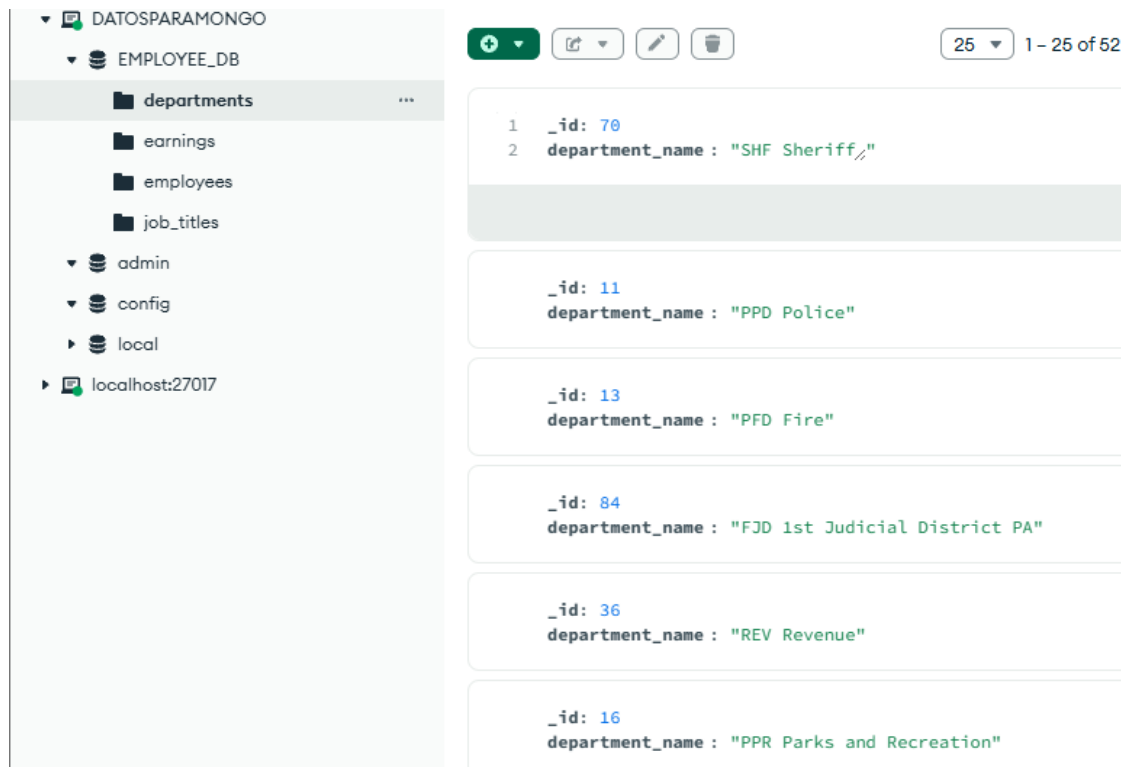
Cancelar

This collection has no data

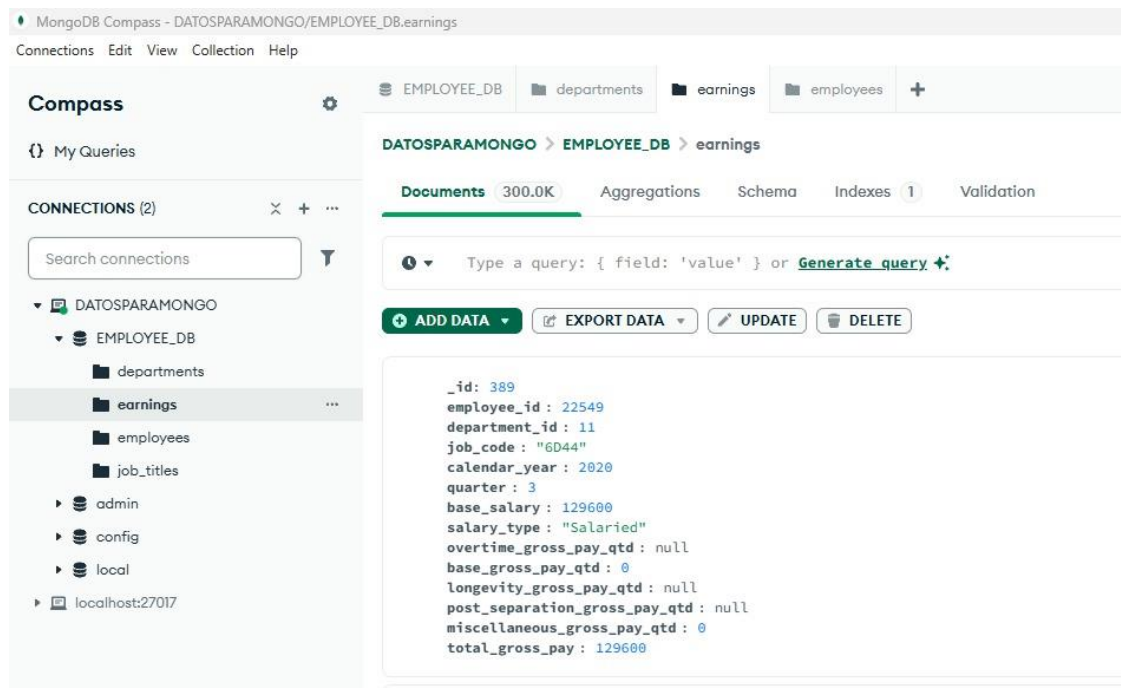
seconds to import data from a JSON or CSV file.

Import data

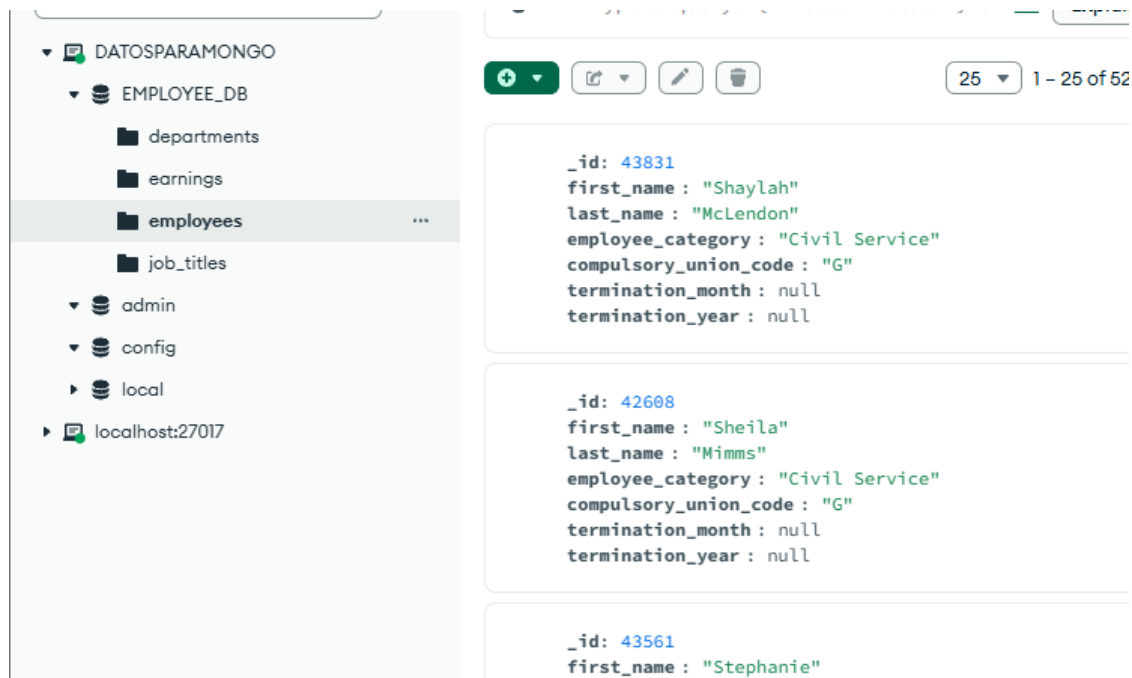
Captura de Pantalla 11.4: Colección Department.



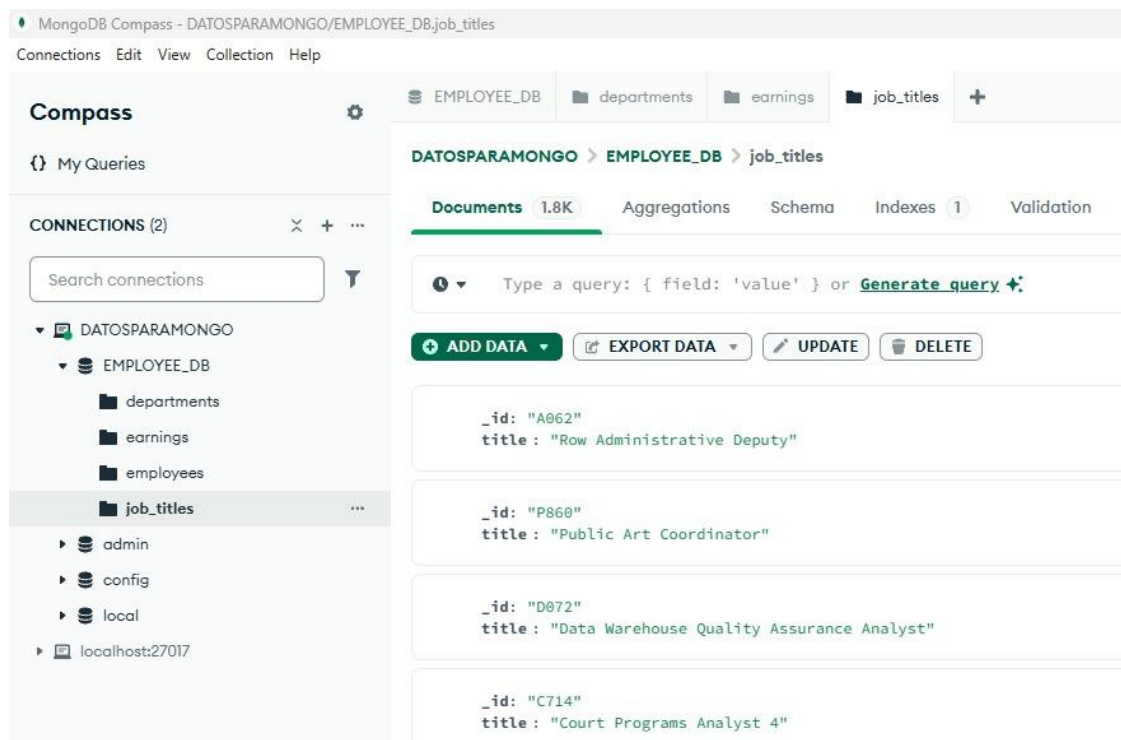
Captura de Pantalla 11.5: Colección earnings.



Captura de Pantalla 11.6: Colección employees.



Captura de Pantalla 11.7: Colección JOB_TITULO.



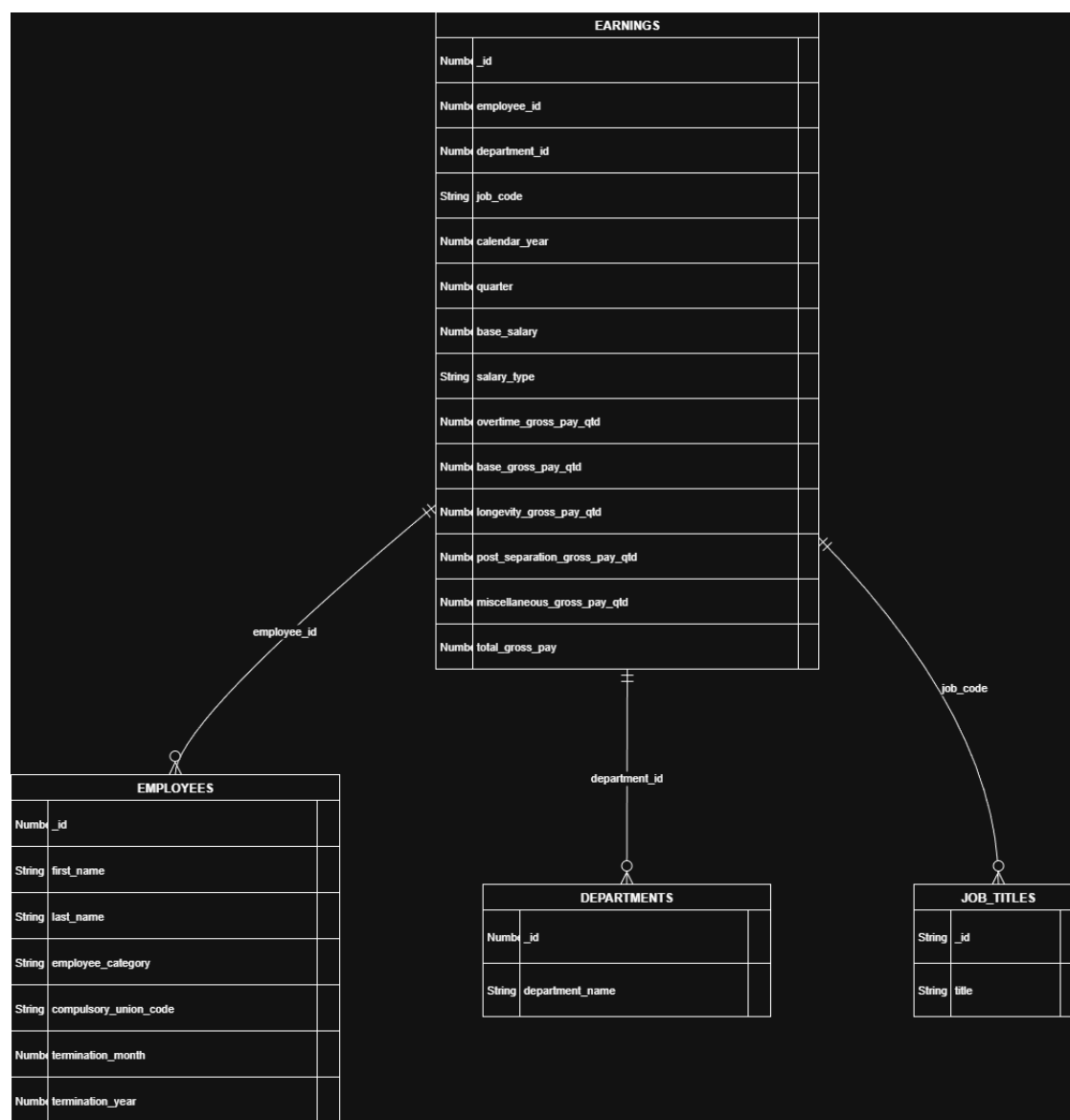
Así, las colecciones quedaron creadas con la estructura que habíamos diseñado, con los `_id` y los demás campos bien puestos, listos para trabajar con ellos.

12. Documentación y Diagrama

Para terminar, documentamos todo el proceso con capturas de pantalla. Esto incluye la instalación, la exportación de datos, la transformación con Python, la importación a MongoDB, la creación de índices, las consultas de verificación, y la comparación con la base de datos original. Es importante tener todo esto bien registrado para que cualquiera pueda seguir los pasos.

Además, usamos Draw.io para hacer un diagrama del modelo referenciado que implementamos. Este diagrama, que llamamos "Modelo Referenciado de la Base de Datos EMPLOYEE_DB - Laboratorio 4", muestra cómo se relacionan las colecciones en MongoDB. Es una forma visual de entender la estructura de la base de datos. Exportamos el diagrama como una imagen PNG y lo incluimos en el informe.

Captura de Pantalla 12.1: Diagrama del modelo referenciado en Draw.io.



13. Tabla de pasos y herramientas

Para que quede todo claro, aqui les dejamos una tabla que resume los pasos que seguimos y las herramientas que usamos en este laboratorio. Asi, si alguien quiere replicar el proceso, lo tiene todo a la mano.

Paso	Herramienta	Descripción
Instalación de MongoDB	Navegador, CMD	Descarga, instala, configura PATH, inicia con mongod.
Exportación de datos	SQL Developer	Exporta tablas a JSON en "sin la propiedad results".
Transformación de JSON	Python	Script transforma JSON, renombra a "_id", maneja duplicados.
Importación a MongoDB	MongoDB Compass	Importa JSONs a colecciones en "EMPLOYEE_DB".
Creación de índices	Compass o mongosh	Crea índices en earnings para employee_id, department_id, job_code.
Consultas de verificación	Compass, mongosh	Conteos, búsquedas específicas, agregaciones con \$lookup.
Comparación con Oracle	SQL Developer	Ejecuta consultas equivalentes, compara resultados.
Documentación	Draw.io, informe	Nos permitio que : Diseñemos el diagrama, incluyéramos capturas, redactemos la metodología.

14. Conclusión

Asi, completamos la migracion, verificando que los datos de Oracle XE 11g se movieron bien a MongoDB, usando un **modelo referenciado** que evita duplicados y permite consultas rapidas, esto nos permite realizar una **emulacion de relaciones**.

15. Referencias Bibliograficas

- Sarasa, A. (2019). *Introducción a las bases de datos NoSQL: Clave-valor usando Redis*. UOC Editorial.

- Gessert, F. (2025). *Guía para bases de datos NoSQL*. Recuperado de <https://todobi.com/guia-para-bases-de-datos-nosql/>
- Date, C. J. (2004). *Introducción a los sistemas de bases de datos* (8a ed.). Pearson Education.
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Sistemas de bases de datos: el libro completo* (2a ed.). Prentice Hall.
- Connolly, T., & Begg, C. (2015). *Sistemas de bases de datos: un enfoque práctico para el diseño, la implementación y la gestión* (6a ed.). Pearson.
- MongoDB Documentation. (s. f.). *Manual de MongoDB*. Recuperado el 28 de julio de 2025, de <https://docs.mongodb.com/>
- Oracle. (s. f.). *Documentación de Oracle SQL Developer*. Recuperado el 28 de julio de 2025, de <https://docs.oracle.com/en/database/oracle/sql-developer/>
- Python Software Foundation. (s. f.). *Documentación de Python*. Recuperado el 28 de julio de 2025, de <https://docs.python.org/3/>
- MongoDB Team. (s. f.). *Instalación de MongoDB Community Edition en Windows*. Manual de base de datos – Documentación de MongoDB. Recuperado el 28 de julio de 2025, de <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows>
- MongoDB Team. (s. f.-a). *\$or (aggregation)*. Manual de base de datos – Documentación de MongoDB. Recuperado el 28 de julio de 2025, de <https://www.mongodb.com/docs/manual/reference/operator/aggregation/or/>
- City of Philadelphia. (2025, 31 de marzo). *City employee earnings*. Recuperado el 28 de julio de 2025, de <https://catalog.data.gov/dataset/city-employee-earnings>