



Instituto Tecnológico
de Buenos Aires

Trabajo Práctico Obligatorio

“Sistema de gestión de aseguradoras”

Base de Datos II

Grupo 15

Segundo cuatrimestre de 2025

Integrantes:

Bassi, Santiago	Legajo: 64643
Hillar, Conrado	Legajo: 64633
Maruottolo, Ignacio	Legajo: 64611
Testoni, Ezequiel	Legajo: 64709

Fecha de entrega: Jueves 13 de Noviembre de 2025

TABLA DE CONTENIDOS

1. Justificación de la Elección de Bases de Datos	3
1.1. MongoDB: Base de datos de documentos	3
1.2. Redis: Capa de caché en memoria	3
2. Cómo ejecutar los servicios y las consultas	4
3. Resultados esperados	5

1. Justificación de la Elección de Bases de Datos

La arquitectura de este proyecto se basa en una estrategia de persistencia políglota, utilizando MongoDB como base de datos principal y Redis como una capa de caché. Esta combinación fue elegida para aprovechar las virtudes individuales de cada sistema teniendo en cuenta fundamentalmente los patrones de acceso de las consultas requeridas.

1.1. MongoDB: Base de datos de documentos

MongoDB fue elegido como el sistema de persistencia principal por su flexibilidad y su modelo de datos orientado a documentos, que se alinea naturalmente con las entidades del negocio de la aseguradora.

A diferencia de un modelo relacional estricto, MongoDB permite anidar información. En este proyecto, se aprovecha esta capacidad de varias maneras. Por ejemplo, la lista de vehículos de un cliente se embebe directamente en el documento del cliente. Esto es ideal para consultas que requieren obtener el perfil completo de un cliente, ya que se evita la necesidad de realizar operaciones costosas, como JOINs entre tablas. También se desnormaliza un resumen de las pólizas en cada documento de cliente. Esto acelera drásticamente las consultas que listan clientes junto con el estado de sus pólizas, ya que la información ya está pre-agregada. Este mismo patrón de embebido se replica en otros documentos, como los de pólizas y siniestros.

Por otra parte, el motor de agregación de MongoDB (aggregate) es potente para realizar cálculos y transformaciones sobre los datos. En el script de carga (data_loader.py), se utilizan agregaciones con \$merge para calcular y persistir estadísticas complejas, como el número de pólizas y siniestros por agente, optimizando futuras consultas sobre estos datos.

1.2. Redis: Capa de caché en memoria

Redis se implementa como una capa de caché para reducir la latencia en consultas frecuentes y disminuir la carga sobre la base principal MongoDB. El dispatcher ('query_dispatcher.py') implementa una estrategia de caché bastante simple: cuando se solicita una consulta (ej. q1_...), primero revisa si el resultado ya existe en Redis usando el nombre de la consulta como clave. Si se encuentra, el resultado se devuelve directamente desde Redis (que opera en memoria y es extremadamente rápido). Si no se encuentra, la consulta se ejecuta contra MongoDB. El resultado se almacena en Redis con un tiempo de expiración (1 hora) y luego se devuelve al usuario. Para las operaciones de escritura (ABM de clientes, alta de siniestros y pólizas), el sistema invalida las consultas cacheadas que podrían haberse vuelto obsoletas. Para eso, en el archivo 'cache_tags.py' se implementa una estructura que mapea a través de etiquetas las consultas que se ven afectadas por cambios en determinadas colecciones.

Esta arquitectura asegura que las lecturas repetidas de datos que no cambian con frecuencia sean casi instantáneas.

2. Cómo ejecutar los servicios y las consultas

El proyecto está contenedorizado con Docker, lo que simplifica enormemente el despliegue. Los servicios (la aplicación, MongoDB y Redis) se gestionan con docker-compose.

Primero deben descargar el proyecto de la siguiente forma:

```
git clone -b master git@github.com:santiagobassi/TPO-BDII.git
```

Luego, para levantar el entorno, desde la raíz del proyecto, ejecutar el siguiente comando para construir las imágenes e iniciar los contenedores en segundo plano:

```
docker-compose up -d --build
```

Esto iniciará los contenedores mongo, redis y app.

Una vez que los servicios están en funcionamiento, se deben cargar los datos iniciales de los archivos .csv en MongoDB, para luego interactuar con la aplicación. El script data_loader.py está diseñado para esto. Primero, ejecutar un shell interactivo dentro del contenedor de la aplicación:

```
docker exec -it app bash
```

Luego, ejecutar el script de carga, pasándole la ruta a los datos.

```
python data_loader.py /app/data
```

Una vez cargados los datos, se debe ejecutar la aplicación principal.

```
python main.py
```

La aplicación presentará un menú de opciones. Ingresar el número de la consulta que se desea ejecutar. Para salir, ingresar 0. Dichos números respetan el orden de las consultas según se indican en la consigna del trabajo. A continuación se muestra un recorte de la salida esperada una vez ejecutada de la aplicación principal:

Bienvenido al sistema de gestión de seguros.

Seleccione una consulta para ejecutar:

1. Clientes activos con pólizas vigentes
2. Siniestros abiertos con tipo, monto y cliente afectado
- ...
15. Emisión de nuevas pólizas

Ingrese el número de la consulta a ejecutar (0 para salir):

3. Resultados esperados

Una aclaración sobre la salida al ejecutar las consultas: se dejó una línea adicional que indica si la respuesta se generó en MongoDB (“From Mongo”) o se trajo cacheada de Redis (“From Redis”) para facilitar la visualización de este mecanismo, y en particular, de la metodología de invalidación. Esto no estaría incluido en la salida esperada del sistema en un caso de uso ordinario, es un agregado para la validación del trabajo.

A continuación se muestran, en el orden en que se indican en la consigna, los resultados esperados para cada consulta. Se mostrará, cuando sea pertinente, simplemente un fragmento ilustrativo del formato de la salida. La mayoría de estos resultados se pueden extender por cientos o miles.

Query 1

```
From Mongo: q1_clientes_activos_polizas_vigentes
[{"nombre": "Laura",
  "apellido": "Gómez",
  "dni": "32456789",
  "email": "laura@gmail.com",
  "telefono": "1145678901",
  "direccion": "Av. Rivadavia 1234",
  "ciudad": "Buenos Aires",
  "provincia": "Buenos Aires",
  "polizas": [{"nro_poliza": "POL1001",
    "tipo": "Auto",
    "fecha_fin": "2026-01-15",
    "estado": "Activa",
    "cobertura_total": 2000000.0,
    "cant_siniestros": 2}]]
```

Query 2

```
From Mongo: q2_siniestros_abiertos_cliente
[{'Tipo': 'Accidente', 'Monto': 500000.0, 'cliente': 'Laura Gómez'},
 {'Tipo': 'Danio', 'Monto': 150000.0, 'cliente': 'Andrés Fernández'},
 {'Tipo': 'Accidente', 'Monto': 500100.0, 'cliente': 'Andrés Fernández'},
 {'Tipo': 'Danio', 'Monto': 150100.0, 'cliente': 'Felicidad Ferrer'},
```

Query 3

```
From Mongo: q3_vehiculos_con_cliente_poliza
[{'Patente': 'AE123CD',
 'Marca': 'Toyota',
 'Modelo': 'Corolla',
 'Cliente': 'Laura Gómez',
 'Polizas_Auto': [{['nro_poliza': 'POL1001',
      'tipo': 'Auto',
      'fecha_fin': '2026-01-15',
      'estado': 'Activa',
      'cobertura_total': 2000000.0,
      'cant_siniestros': 2}]},
 {'Patente': 'AD456FG',
 'Marca': 'Ford',
 'Modelo': 'Fiesta',
 'Cliente': 'Martín Pérez',
 'Polizas_Auto': []}],
```

En esta consulta se muestran algunos resultados en los que la póliza del vehículo se encuentra vacía. Esto es porque el vehículo estaba marcado como “asegurado”, pero no había una póliza asociada.

Query 4

```
From Mongo: q4_clientes_sin_polizas_activas
[{'dni': '40321876',
 'email': 'sofia@gmail.com',
 'telefono': '1123456789',
 'direccion': 'Av. Santa Fe 2200',
 'ciudad': 'Córdoba',
 'provincia': 'Córdoba',
 'nombre': 'Sofía López'},
```

Query 5

```
From Mongo: q5_agentes_activos_cant_polizas
[{'Agente': 'Juan Torres', 'Cantidad Polizas': 23},
 {'Agente': 'María Martínez', 'Cantidad Polizas': 44},
 {'Agente': 'Pablo Ramos', 'Cantidad Polizas': 22},
 {'Agente': 'Carlos García', 'Cantidad Polizas': 22},
 {'Agente': 'Víctor Rolluedo', 'Cantidad Polizas': 22}]
```

Query 6

```
From Mongo: q6_polizas_vencidas_con_cliente
[{'Poliza': 'POL1003',
 'Fecha de Inicio': '10-12-2023',
 'Fecha de Vencimiento': '10-12-2024',
 'prima mensual': 12000.0,
 'cobertura total': 800000.0,
 'estado': 'Vencida',
 'Cliente': 'Martín Pérez'},
```

Query 7

```
From Mongo: q7_top10_clientes_cobertura_total
[{'CoberturaTotal': 2500000.0, 'Cliente': 'Ainara Santana'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Odalis Tenorio'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Olegario Aramburu'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Pacífica Barreda'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'María Carmen Delgado'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Herminio Soto'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'África Herrero'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Diana Cepeda'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Miguel Ángel Farré'},
 {'CoberturaTotal': 2500000.0, 'Cliente': 'Loida Ibañez'}]
```

Query 8

```
From Mongo: q8_siniestros_accidente_ultimo_anio
[{'Siniestro': 9001,
 'Fecha': '2025-03-20',
 'Monto': 500000.0,
 'Descripcion': 'Colisión frontal en autopista',
 'Estado': 'Abierto'},
```

Query 9

```
From Mongo: q9_polizas_activas_ordenadas
[{'Poliza': 'POL1113', 'FechaInicio': '2024-01-01', 'FechaFin': '2025-01-01'},
 {'Poliza': 'POL1118', 'FechaInicio': '2024-01-02', 'FechaFin': '2025-01-02'},
 {'Poliza': 'POL1123', 'FechaInicio': '2024-01-03', 'FechaFin': '2025-01-03'},
 {'Poliza': 'POL1128', 'FechaInicio': '2024-01-04', 'FechaFin': '2025-01-04'},
 {'Poliza': 'POL1133', 'FechaInicio': '2024-01-05', 'FechaFin': '2025-01-05'},
```

Query 10

```
From Mongo: q10_polizas_suspendidas_estado_cliente
[{'Poliza': 'POL1005', 'Cliente': 'Carla Ruiz', 'ClienteActivo': True},
 {'Poliza': 'POL1013', 'Cliente': 'Cándida Adán', 'ClienteActivo': True},
 {'Poliza': 'POL1021', 'Cliente': 'Josefina Arcos', 'ClienteActivo': True},
 {'Poliza': 'POL1029', 'Cliente': 'Vasco Blazquez', 'ClienteActivo': True},
 {'Poliza': 'POL1037', 'Cliente': 'Cornelio Huguet', 'ClienteActivo': True},
```

Query 11

```
From Mongo: q11_clientes_multiples_vehiculos
[]
```

Query 12

```
From Mongo: q12_agentes_cant_siniestros
[{'CantidadSiniestros': 22, 'Agente': 'María Martínez'},
 {'CantidadSiniestros': 12, 'Agente': 'Pablo Ramos'},
 {'CantidadSiniestros': 12, 'Agente': 'Lucía Suárez'},
 {'CantidadSiniestros': 11, 'Agente': 'Carlos García'},
 {'CantidadSiniestros': 10, 'Agente': 'Víctor Rolluedo'},
 {'CantidadSiniestros': 10, 'Agente': 'Juan Torres'}]
```

A continuación se detallan las acciones de las consultas 13, 14 y 15. Las salidas se muestran como un conjunto, pero los datos se leen, uno a uno de entrada estándar:

Query 13

```
Ingrese opción (A=Alta, B=Baja, M=Modificación): a
--- Alta de nuevo cliente ---
Nombre: Juan Pérez
Apellido: González
DNI: 12345678
Email: juancito@gmail.com
Teléfono: 1134683667
Dirección: Av. San Martín 202
Ciudad: Rosario
Provincia: Santa Fé

Cliente Juan Pérez González agregado con ID 206.
[{'nombre': 'Juan Pérez',
 'apellido': 'González',
 'dni': '12345678',
 'email': 'juancito@gmail.com',
 'telefono': '1134683667',
 'direccion': 'Av. San Martín 202',
 'ciudad': 'Rosario',
 'provincia': 'Santa Fé',
 'activo': True,
 'vehiculos': [],
 'polizas_resumen': [],
 '_id': 206}]
```

Query 14

```
--- Alta de nuevo siniestro ---
Ingrese número de póliza asociada (ej: POL1001): POL1001
Tipo de siniestro (Accidente, Robo, Incendio, etc.): Robo
Descripción breve: Me robaron la bicicleta a la salida del trabajo
Monto estimado: 40000

Siniestro creado correctamente con ID 9095 para la póliza POL1001.
[{'_id': 9095,
 'fecha': '2025-11-13',
 'tipo': 'Robo',
 'monto_estimado': 40000.0,
 'descripcion': 'Me robaron la bicicleta a la salida del trabajo',
 'estado': 'Abierto',
 'nro_poliza': 'POL1001',
 'id_cliente': 1,
 'id_agente': 101,
 'cliente_afectado': {'nombre_completo': 'Laura Gómez',
                      'telefono': '1145678901'}]]
```

Query 15

```
--- Emisión de nueva póliza ---
Ingrese DNI del cliente: 29543210
Ingrese la matrícula del agente: MAT001
¿La póliza está asociada a un vehículo? (s/n): s
Ingrese la patente del vehículo: AD456FG
Tipo de póliza (Auto, Vida, Hogar, etc.): Auto
Prima mensual: 800
Cobertura total: 50000
```

Póliza POL1161 creada correctamente para Martín Pérez.

```
[{'_id': 'POL1161',
  'fecha_inicio': '2025-11-13',
  'fecha_fin': '2026-11-13',
  'tipo': 'Auto',
  'prima_mensual': 800.0,
  'cobertura_total': 50000.0,
  'estado': 'Activa',
  'id_cliente': 2,
  'id_agente': 101,
  'id_vehiculo': 5002,
  'cliente_info': {'nombre_completo': 'Martín Pérez', 'activo': True},
  'agente_info': {'nombre_completo': 'Juan'},
  'vehiculo_info': {'marca': 'Ford',
                    'modelo': 'Fiesta',
                    'anio': 2019,
                    'patente': 'AD456FG'}]]
```