

Universidad ORT Uruguay

Facultado de Ingeniería

Escuela de Tecnología

**OBLIGATORIO BASE DE DATOS 2
DOCUMENTO DE ANÁLISIS**



Gian Gatto – 248529



Thiago García da Rosa - 317133

N3E - PRESENCIAL

Docente: Mariano Sendic

Analista en T. I.

13/05/2024

Contenido

Creación de Base de Datos y de tablas	3
Ingreso de Inserts	4
Creación de Índices	6
CONSULTAS	7
Ejercicio 3 A	7
Ejercicio 3 B	7
Ejercicio 3 C	7
Ejercicio 3 D	7
Ejercicio 3 E	8
Ejercicio 3 F	8
Ejercicios con TSQL	9
Ejercicio 4 A	9
Ejercicio 4 B	9
Ejercicio 4 C	9
DISPARADORES	10
Ejercicio 5 A	10
Ejercicio 5 B	10
VISTA	11
Ejercicio 6	11
MONGODB	12
Creación Base de datos e ingreso de Inserts	12
Consulta A versión 1	15
Consulta A versión 2	15
Consulta B	16
Consulta C	17
Consultas Extras	17

Creación de Base de Datos y de tablas

```
CREATE DATABASE FoodInspections
GO
USE FoodInspections
GO
/*
Creacion de las tablas
*/
CREATE TABLE Establecimientos(estNumero int identity not null,
                                estNombre varchar(40) not null,
                                estDireccion varchar(60) not
null,
                                estTelefono varchar(50),
                                estLatitud money,
                                estLongitud money,
                                constraint pk_Estab primary
key(estNumero))
GO
CREATE TABLE Licencias(licNumero int identity not null,
                        estNumero int not null,
                        licFchEmision date,
                        licFchVto date,
                        licStatus character(3) not null,
                        constraint pk_Licencia primary
key(licNumero),
                        constraint fk_EstLic foreign key(estNumero)
references Establecimientos(estNumero),
                        constraint ck_StatusLic check(licStatus in
('APR', 'REV')))
GO
CREATE TABLE TipoViolacion(violCodigo int identity not null,
                             violDescrip varchar(30) not null,
                             constraint pk_TipoViol primary
key(violCodigo))
GO
CREATE TABLE Inspecciones(inspID int identity not null,
                            inspFecha datetime,
                            estNumero int not null,
                            inspRiesgo varchar(5) not null,
                            inspResultado varchar(25) not null,
                            violCodigo int not null,
                            inspComents varchar(100),
                            constraint pk_Inspect primary
key(inspID),
                            constraint fk_EstabInsp foreign
key(estNumero) references Establecimientos(estNumero),
                            constraint fk_ViolInsp foreign
key(violCodigo) references TipoViolacion(violCodigo),
                            constraint ck_Riesgo check(inspRiesgo
IN('Bajo', 'Medio', 'Alto')),
                            constraint ck_Result
check(inspResultado IN('Pasa', 'Falla', 'Pasa con condiciones', 'Oficina no
encontrada'))))
GO
```

Ingreso de Inserts

```
-- Inserts para Establecimientos
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Restaurante El Buen Sabor', 'Av. Libertador 123', '123-456-
789', -34.901234, -56.123456);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Pizzería Bella Napoli', 'Calle Italia 456', '987-654-321',
-34.912345, -56.134567);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Cafetería Aroma Fresco', 'Av. Brasil 789', '456-789-123', -
34.923456, -56.145678);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Bar El Encuentro', 'Calle Uruguay 1011', '321-654-987', -
34.934567, -56.156789);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Comedor Las Delicias', 'Av. España 1213', '789-123-456', -
34.945678, -56.167890);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Restaurante Mar y Tierra', 'Av. Costanera 1415', '159-357-
258', -34.956789, -56.178901);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Hamburguesería Don Manolo', 'Av. Italia 1617', '753-951-
456', -34.967890, -56.189012);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Parrillada El Fogón', 'Calle Argentina 1819', '246-802-
468', -34.978901, -56.190123);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Café Expresso', 'Av. del Libertador 2021', '357-913-579', -
34.989012, -56.201234);
INSERT INTO Establecimientos (estNombre, estDireccion, estTelefono, estLatitud,
estLongitud) VALUES ('Pizzería Rápida', 'Calle Bolivia 2223', '468-024-680', -
34.990123, -56.212345);
```

```
select * from Establecimientos
```

```
-- Inserts para Licencias
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (1,
'2023-01-15', '2024-01-15', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (2,
'2023-02-20', '2024-02-20', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (3,
'2023-03-10', '2024-03-10', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (4,
'2023-04-25', '2024-04-25', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (5,
'2023-05-05', '2024-05-05', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (6,
'2023-06-18', '2024-06-18', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (7,
'2023-07-29', '2024-07-29', 'APR');

INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (8,
'2023-07-29', '2023-09-15', 'APR');
INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus) VALUES (9,
'2023-07-29', '2023-07-30', 'APR');
```

```
select * from Licencias
```

```
-- Inserts para TipoViolacion
```

```

INSERT INTO TipoViolacion (violDescrip) VALUES ('Insectos en la cocina');
INSERT INTO TipoViolacion (violDescrip) VALUES ('Mal manejo de alimentos');
INSERT INTO TipoViolacion (violDescrip) VALUES ('Mala Higiene');
INSERT INTO TipoViolacion (violDescrip) VALUES ('Mal estado Instalaciones');
INSERT INTO TipoViolacion (violDescrip) VALUES ('Personal no capacitado');

select * from TipoViolacion

-- Inserts para Inspecciones
INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-01-20', 1, 'Bajo', 'Pasa', 2, 'Inspección
de rutina.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-02-25', 2, 'Medio', 'Pasa con
condiciones', 3, 'Se requiere mejorar la limpieza en la cocina.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-03-30', 3, 'Alto', 'Falla', 1, 'Presencia
de cucarachas en el área de almacenamiento.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-04-05', 4, 'Bajo', 'Pasa', 5, 'Todo en
orden, personal capacitado.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-05-10', 5, 'Medio', 'Pasa con
condiciones', 2, 'Se detectaron algunas deficiencias en el manejo de
alimentos.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-06-15', 6, 'Alto', 'Oficina no
encontrada', 4, 'El establecimiento se encontraba cerrado durante el horario de
la inspección.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-07-20', 7, 'Bajo', 'Pasa', 4, 'Se realizó
mantenimiento preventivo de las instalaciones.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-08-25', 8, 'Medio', 'Falla', 3, 'Se
encontraron múltiples violaciones a las normas de higiene.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-09-30', 9, 'Alto', 'Pasa con condiciones',
2, 'Se requiere mejorar la limpieza en las áreas de preparación de alimentos.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-10-05', 10, 'Bajo', 'Pasa', 3, 'Todo en
orden, personal capacitado.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-11-10', 1, 'Medio', 'Pasa con
condiciones', 1, 'Se detectó la presencia de moscas en el área de preparación.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2023-12-15', 2, 'Alto', 'Falla', 4, 'Se
detectaron fugas de gas en la cocina.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-01-20', 3, 'Bajo', 'Pasa', 5, 'Todo en
orden, personal capacitado.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-02-25', 4, 'Medio', 'Pasa con
condiciones', 3, 'Se requiere mejorar la higiene en las áreas comunes.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-03-30', 5, 'Alto', 'Oficina no
encontrada', 1, 'El establecimiento se encontraba cerrado durante el horario de
la inspección.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-04-05', 6, 'Bajo', 'Pasa', 1, 'Se realizó
fumigación para eliminar insectos.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-05-10', 7, 'Medio', 'Falla', 2, 'Se
encontró un mal manejo de alimentos.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-06-15', 8, 'Alto', 'Pasa con condiciones',
3, 'Se requiere mejorar la limpieza en las áreas de preparación de alimentos.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-07-20', 9, 'Bajo', 'Pasa', 5, 'Todo en
orden, personal capacitado.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-08-25', 10, 'Medio', 'Pasa con
condiciones', 4, 'Se detectaron problemas de mantenimiento en las
instalaciones.');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-05-10', 3, 'Alto', 'Falla', 2, 'No saben
cocinar pollo al curry');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-03-23', 3, 'Alto', 'Falla', 3, 'Cocinas
totalmente sucias');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-09-12', 3, 'Alto', 'Falla', 4, 'Cocinas no
funcionan correctamente');
```

```

INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents) VALUES ('2024-12-12', 3, 'Alto', 'Falla', 5, 'Personal
no esta para nada capacitado.');
```

```
select * from Inspecciones
```

Creación de Índices

```

CREATE INDEX idx_estNumero ON Establecimientos(estNumero);
CREATE INDEX idx_estNombre ON Establecimientos(estNombre);
CREATE INDEX idx_estDireccion ON Establecimientos(estDireccion);

CREATE INDEX idx_lic_estNumero_LicStatus ON Licencias(estNumero, LicStatus);
CREATE INDEX idx_lic_LicFchVto ON Licencias(LicFchVto);

CREATE INDEX idx_violCodigo ON TipoViolacion(violCodigo);

CREATE INDEX idx_insp_estNumero_inspFecha ON Inspecciones(estNumero, inspFecha);
CREATE INDEX idx_insp_inspRiesgo ON Inspecciones(inspRiesgo);
CREATE INDEX idx_insp_inspResultado ON Inspecciones(inspResultado);
CREATE INDEX idx_insp_violCodigo ON Inspecciones(violCodigo);
```

CONSULTAS

Ejercicio 3 A

```
select e.estNombre, e.estTelefono, e.estDireccion, i.inspFecha
from Establecimientos e, Inspecciones i
where i.estNumero = e.estNumero and
      i.inspFecha = (select MAX(inspFecha) from Inspecciones i
                     where i.inspResultado LIKE 'falla');
```

Ejercicio 3 B

```
select top 5 t.violCodigo, t.violDescrip, count(i.inspID) as cantidadInspeccion
from TipoViolacion t, Inspecciones i
where t.violCodigo = i.violCodigo and
      YEAR(i.inspFecha) = '2024'
group by t.violCodigo, t.violDescrip
```

Ejercicio 3 C

```
SELECT e.estNumero, e.estNombre
FROM Establecimientos e
WHERE (
    SELECT COUNT(DISTINCT violCodigo)
    FROM TipoViolacion
) = (
    SELECT COUNT(DISTINCT i.violCodigo)
    FROM Inspecciones i
    WHERE i.estNumero = e.estNumero
);
```

Ejercicio 3 D

```
SELECT
    e.estNombre,
    e.estNumero,
    COUNT(i.inspID) AS total_inspecciones,
    SUM(CASE WHEN i.inspResultado IN ('Falla', 'Pasa con condiciones') THEN 1
    ELSE 0 END) AS inspecciones_reprobadas,
    ROUND(
        (SUM(CASE WHEN i.inspResultado IN ('Falla', 'Pasa con condiciones') THEN
1 ELSE 0 END) * 100.0) /
        COUNT(i.inspID),
        2
    ) AS porcentaje_reprobadas
FROM
    Establecimientos e
LEFT JOIN
    Inspecciones i ON e.estNumero = i.estNumero
GROUP BY
    e.estNumero, e.estNombre
ORDER BY
    porcentaje_reprobadas DESC;
```

Ejercicio 3 E

```
SELECT
    e.estNumero,
    e.estNombre,
    COUNT(i.inspID) AS total_inspecciones,
    SUM(CASE WHEN i.inspResultado = 'Pasa' THEN 1 ELSE 0 END) AS total_aprobadas,
    ROUND(
        (SUM(CASE WHEN i.inspResultado = 'Pasa' THEN 1 ELSE 0 END) * 100.0) /
COUNT(i.inspID),
        2
    ) AS porcentaje_aprobadas,
    SUM(CASE WHEN i.inspResultado IN ('Falla', 'Pasa con condiciones') THEN 1
ELSE 0 END) AS total_reprobadas,
    ROUND(
        (SUM(CASE WHEN i.inspResultado IN ('Falla', 'Pasa con condiciones') THEN
1 ELSE 0 END) * 100.0) / COUNT(i.inspID),
        2
    ) AS porcentaje_reprobadas
FROM
    Establecimientos e
JOIN
    Licencias l ON e.estNumero = l.estNumero
LEFT JOIN
    Inspecciones i ON e.estNumero = i.estNumero
WHERE
    l.LicStatus = 'APR'
GROUP BY
    e.estNumero, e.estNombre
ORDER BY
    porcentaje_aprobadas DESC, total_inspecciones DESC;
```

Ejercicio 3 F

```
SELECT
    e.estNombre,
    e.estDireccion,
    DATEDIFF(day, l.licFchEmision, l.licFchVto) AS DiasRenovacion
FROM
    Establecimientos e
JOIN
    Licencias l ON e.estNumero = l.estNumero;
```


Ejercicios con TSQL

Ejercicio 4 A

```
CREATE PROCEDURE ObtenerViolacionesPorRiesgo
    @Riesgo NVARCHAR(5)
AS
BEGIN
    SET NOCOUNT ON;

    SELECT DISTINCT
        tv.violCodigo,
        tv.violDescrip
    FROM
        Inspecciones i
    JOIN
        TipoViolacion tv ON i.violCodigo = tv.violCodigo
    WHERE
        i.inspRiesgo = @Riesgo
    ORDER BY
        tv.violCodigo;
END;
```

Ejercicio 4 B

```
CREATE FUNCTION CantidadEstablecimientosConViolacionLicenciaVencida
(
    @violCodigo INT
)
RETURNS INT
AS
BEGIN
    DECLARE @cantidad INT;

    SELECT @cantidad = COUNT(DISTINCT e.estNumero)
    FROM Establecimientos e
    JOIN Inspecciones i ON e.estNumero = i.estNumero
    JOIN Licencias l ON e.estNumero = l.estNumero
    WHERE i.violCodigo = @violCodigo
        AND l.LicFchVto < GETDATE()
        AND l.LicNumero NOT IN (
            SELECT LicNumero
            FROM Licencias l2
            WHERE l2.estNumero = e.estNumero
                AND l2.LicFchEmission > l.LicFchVto
        );

    RETURN @cantidad;
END;
```

Ejercicio 4 C

```
CREATE PROCEDURE ObtenerInspeccionesPorFecha
(
    @FechaInicio DATE,
    @FechaFin DATE,
    @CantidadOficinaNoEncontrada INT OUTPUT,
```

```

        @CantidadSinComentarios INT OUTPUT
    )
AS
BEGIN
    SET NOCOUNT ON;

    SELECT @CantidadOficinaNoEncontrada = COUNT(*)
    FROM Inspecciones
    WHERE inspFecha BETWEEN @FechaInicio AND @FechaFin
        AND inspResultado = 'Oficina no encontrada';

    SELECT @CantidadSinComentarios = COUNT(*)
    FROM Inspecciones
    WHERE inspFecha BETWEEN @FechaInicio AND @FechaFin
        AND (inspComents IS NULL OR inspComents = '');

END;

```

DISPARADORES

Ejercicio 5 A

```

GO
CREATE TRIGGER trg_AfterInsertEstablecimiento
ON Establecimientos
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO Licencias (estNumero, LicFchEmision, LicFchVto, LicStatus)
    SELECT
        i.estNumero,
        GETDATE() AS LicFchEmision,
        DATEADD(DAY, 90, GETDATE()) AS LicFchVto,
        'APR' AS LicStatus
    FROM
        INSERTED i;

END;

```

Ejercicio 5 B

```

GO
CREATE TRIGGER trg_BeforeInsertInspecciones
ON Inspecciones
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @FechaActual DATE = GETDATE();

    IF EXISTS (
        SELECT 1
        FROM INSERTED i
        JOIN Licencias l ON i.estNumero = l.estNumero
        WHERE l.LicFchVto BETWEEN @FechaActual AND DATEADD(DAY, 5, @FechaActual)
            AND l.LicStatus = 'APR'
    )

```

```

    )
    BEGIN
        RAISERROR ('No se puede ingresar inspecciones para establecimientos con
licencias próximas a vencer en los próximos 5 días.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    INSERT INTO Inspecciones (inspFecha, estNumero, inspRiesgo, inspResultado,
violCodigo, inspComents)
    SELECT inspFecha, estNumero, inspRiesgo, inspResultado, violCodigo,
inspComents
    FROM INSERTED;
END;

```

VISTA

Ejercicio 6

```

GO
CREATE VIEW LicenciasVigentes
AS
SELECT
    LicNumero,
    estNumero,
    LicFchEmision,
    LicFchVto,
    LicStatus,
    DATEDIFF(DAY, GETDATE(), LicFchVto) AS DiasParaVencimiento
FROM
    Licencias
WHERE
    LicFchVto >= GETDATE() AND LicStatus = 'APR';

```

MONGODB

Creación Base de datos e ingreso de Inserts

use obligatorio

```
db.createCollection("Establecimientos")
```

```
db.createCollection("Licencias")
```

```
db.createCollection("TipoViolacion")
```

```
db.createCollection("Inspecciones")
```

```
db.createCollection("Conversaciones")
```

```
db.Establecimientos.insertMany([
```

```
  { estNumero: 1, estNombre: 'Restaurante El Buen Sabor', estDireccion: 'Av. Libertador 123',  
    estTelefono: '123-456-789', estLatitud: -34.901234, estLongitud: -56.123456 },
```

```
  { estNumero: 2, estNombre: 'Pizzería Bella Napoli', estDireccion: 'Calle Italia 456', estTelefono: '987-  
654-321', estLatitud: -34.912345, estLongitud: -56.134567 },
```

```
  { estNumero: 3, estNombre: 'Cafetería Aroma Fresco', estDireccion: 'Av. Brasil 789', estTelefono: '456-  
789-123', estLatitud: -34.923456, estLongitud: -56.145678 },
```

```
  { estNumero: 4, estNombre: 'Bar El Encuentro', estDireccion: 'Calle Uruguay 1011', estTelefono: '321-  
654-987', estLatitud: -34.934567, estLongitud: -56.156789 },
```

```
  { estNumero: 5, estNombre: 'Comedor Las Delicias', estDireccion: 'Av. España 1213', estTelefono: '789-  
123-456', estLatitud: -34.945678, estLongitud: -56.167890 },
```

```
  { estNumero: 6, estNombre: 'Restaurante Mar y Tierra', estDireccion: 'Av. Costanera 1415', estTelefono:  
'159-357-258', estLatitud: -34.956789, estLongitud: -56.178901 },
```

```
  { estNumero: 7, estNombre: 'Hamburguesería Don Manolo', estDireccion: 'Av. Italia 1617', estTelefono:  
'753-951-456', estLatitud: -34.967890, estLongitud: -56.189012 },
```

```
  { estNumero: 8, estNombre: 'Parrillada El Fogón', estDireccion: 'Calle Argentina 1819', estTelefono:  
'246-802-468', estLatitud: -34.978901, estLongitud: -56.190123 },
```

```
  { estNumero: 9, estNombre: 'Café Espresso', estDireccion: 'Av. del Libertador 2021', estTelefono: '357-  
913-579', estLatitud: -34.989012, estLongitud: -56.201234 },
```

```
  { estNumero: 10, estNombre: 'Pizzería Rápida', estDireccion: 'Calle Bolivia 2223', estTelefono: '468-  
024-680', estLatitud: -34.990123, estLongitud: -56.212345 }
```

```
])
```

```
db.Licencias.insertMany([
```

```
  { licNumero: 1, estNumero: 1, licFchEmision: new Date('2023-01-15'), licFchVto: new Date('2024-01-  
15'), licStatus: 'APR' },
```

```
  { licNumero: 2, estNumero: 2, licFchEmision: new Date('2023-02-20'), licFchVto: new Date('2024-02-  
20'), licStatus: 'APR' },
```

```

    { licNumero: 3, estNumero: 3, licFchEmision: new Date('2023-03-10'), licFchVto: new Date('2024-03-10'), licStatus: 'APR' },

    { licNumero: 4, estNumero: 4, licFchEmision: new Date('2023-04-25'), licFchVto: new Date('2024-04-25'), licStatus: 'APR' },

    { licNumero: 5, estNumero: 5, licFchEmision: new Date('2023-05-05'), licFchVto: new Date('2024-05-05'), licStatus: 'APR' },

    { licNumero: 6, estNumero: 6, licFchEmision: new Date('2023-06-18'), licFchVto: new Date('2024-06-18'), licStatus: 'APR' },

    { licNumero: 7, estNumero: 7, licFchEmision: new Date('2023-07-29'), licFchVto: new Date('2024-07-29'), licStatus: 'APR' }

  ])

```

```

db.TipoViolacion.insertMany([

  { violCodigo: 1, violDescrip: 'Insectos en la cocina' },

  { violCodigo: 2, violDescrip: 'Mal manejo de alimentos' },

  { violCodigo: 3, violDescrip: 'Mala Higiene' },

  { violCodigo: 4, violDescrip: 'Mal estado Instalaciones' },

  { violCodigo: 5, violDescrip: 'Personal no capacitado' }

])

```

```

db.Inspecciones.insertMany([

  { inspID: 1, inspFecha: new Date('2023-01-20'), estNumero: 1, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 2, inspComents: 'Inspección de rutina.' },

  { inspID: 2, inspFecha: new Date('2023-02-25'), estNumero: 2, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 3, inspComents: 'Se requiere mejorar la limpieza en la cocina.' },

  { inspID: 3, inspFecha: new Date('2023-03-30'), estNumero: 3, inspRiesgo: 'Alto', inspResultado: 'Falla', violCodigo: 1, inspComents: 'Presencia de cucarachas en el área de almacenamiento.' },

  { inspID: 4, inspFecha: new Date('2023-05-04'), estNumero: 4, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 5, inspComents: 'Todo en orden, personal capacitado.' },

  { inspID: 5, inspFecha: new Date('2023-10-05'), estNumero: 5, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 2, inspComents: 'Se detectaron algunas deficiencias en el manejo de alimentos.' },

  { inspID: 6, inspFecha: new Date('2023-06-15'), estNumero: 6, inspRiesgo: 'Alto', inspResultado: 'Oficina no encontrada', violCodigo: 4, inspComents: 'El establecimiento se encontraba cerrado durante el horario de la inspección.' },

  { inspID: 7, inspFecha: new Date('2023-07-20'), estNumero: 7, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 4, inspComents: 'Se realizó mantenimiento preventivo de las instalaciones.' },

  { inspID: 8, inspFecha: new Date('2023-08-25'), estNumero: 8, inspRiesgo: 'Medio', inspResultado: 'Falla', violCodigo: 3, inspComents: 'Se encontraron múltiples violaciones a las normas de higiene.' },

])

```

```
{ inspID: 9, inspFecha: new Date('2023-09-30'), estNumero: 9, inspRiesgo: 'Alto', inspResultado: 'Pasa con condiciones', violCodigo: 2, inspComents: 'Se requiere mejorar la limpieza en las áreas de preparación de alimentos.' },
```

```
{ inspID: 10, inspFecha: new Date('2023-05-10'), estNumero: 10, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 3, inspComents: 'Todo en orden, personal capacitado.' },
```

```
{ inspID: 11, inspFecha: new Date('2023-10-11'), estNumero: 1, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 1, inspComents: 'Se detectó la presencia de moscas en el área de preparación.' },
```

```
{ inspID: 12, inspFecha: new Date('2023-12-15'), estNumero: 2, inspRiesgo: 'Alto', inspResultado: 'Falla', violCodigo: 4, inspComents: 'Se detectaron fugas de gas en la cocina.' }
```

```
])
```

```
db.Inspecciones.insertMany([
```

```
{ inspID: 13, inspFecha: new Date('2024-06-10'), estNumero: 3, inspRiesgo: 'Alto', inspResultado: 'Pasa con condiciones', violCodigo: 2, inspComents: 'Se requiere mejorar la limpieza en las áreas de preparación de alimentos.' },
```

```
{ inspID: 14, inspFecha: new Date('2024-07-15'), estNumero: 4, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 5, inspComents: 'Inspección rutinaria sin problemas destacables.' },
```

```
{ inspID: 15, inspFecha: new Date('2024-08-20'), estNumero: 5, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 1, inspComents: 'Se observaron algunas deficiencias en la limpieza del área de almacenamiento.' },
```

```
{ inspID: 16, inspFecha: new Date('2024-09-25'), estNumero: 6, inspRiesgo: 'Alto', inspResultado: 'Pasa con condiciones', violCodigo: 4, inspComents: 'Se detectaron problemas en el sistema de ventilación de la cocina.' },
```

```
{ inspID: 17, inspFecha: new Date('2024-10-30'), estNumero: 7, inspRiesgo: 'Bajo', inspResultado: 'Pasa', violCodigo: 4, inspComents: 'El establecimiento realizó mantenimiento preventivo de forma adecuada.' },
```

```
{ inspID: 18, inspFecha: new Date('2024-11-15'), estNumero: 8, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 3, inspComents: 'Se observaron áreas que requieren mejoras en la higiene.' }
```

```
])
```

```
db.Inspecciones.insertOne(
```

```
{ inspID: 19, inspFecha: new Date('2024-12-15'), estNumero: 4, inspRiesgo: 'Medio', inspResultado: 'Pasa con condiciones', violCodigo: 3, inspComents: 'Se observaron áreas que requieren mejoras en la higiene.' }
```

```
)
```

```
db.Conversaciones.insertMany([
```

```
{fecha: new Date('2023-01-20'),mensaje: 'Se realizó una inspección de rutina en Restaurante El Buen Sabor.',inspID: 1,inspResultado: 'Pasa'},
```

```
{fecha: new Date('2023-02-25'),mensaje: 'Se requiere mejorar la limpieza en la cocina de Pizzería Bella Napoli.',inspID: 2,inspResultado: 'Pasa con condiciones'},
```

```

    {fecha: new Date('2023-05-04'),mensaje: 'Todo en orden, personal capacitado en Bar El Encuentro.',inspID: 4,inspResultado: 'Pasa'},

    {fecha: new Date('2023-10-05'),mensaje: 'Se detectaron algunas deficiencias en el manejo de alimentos en Comedor Las Delicias.',inspID: 5,inspResultado: 'Pasa con condiciones'},

    {fecha: new Date('2023-07-20'),mensaje: 'Se realizó mantenimiento preventivo de las instalaciones en Parrillada El Fogón.',inspID: 7,inspResultado: 'Pasa'},

    {fecha: new Date('2024-07-15'),mensaje: 'Inspección rutinaria sin problemas destacables en Comedor Las Delicias.',inspID: 14,inspResultado: 'Pasa'},

    {fecha: new Date('2023-03-30'),mensaje: 'Presencia de cucarachas en el área de almacenamiento de Cafetería Aroma Fresco.',inspID: 3,inspResultado: 'Falla'},

    {fecha: new Date('2023-08-25'),mensaje: 'Se encontraron múltiples violaciones a las normas de higiene en Café Espresso.',inspID: 8,inspResultado: 'Falla'},

    {fecha: new Date('2024-08-20'),mensaje: 'Se observaron algunas deficiencias en la limpieza del área de almacenamiento en Restaurante Mar y Tierra.',inspID: 15,inspResultado: 'Pasa con condiciones'},

    {fecha: new Date('2024-09-25'),mensaje: 'Se detectaron problemas en el sistema de ventilación de la cocina en Hamburguesería Don Manolo.',inspID: 16,inspResultado: 'Pasa con condiciones'},

    {fecha: new Date('2024-11-15'),mensaje: 'Se observaron áreas que requieren mejoras en la higiene en Parrillada El Fogón.',inspID: 18,inspResultado: 'Pasa con condiciones'},

    {fecha: new Date('2024-12-15'),mensaje: 'Se observaron áreas que requieren mejoras en la higiene en Comedor Las Delicias.',inspID: 19,inspResultado: 'Pasa con condiciones'}
  ])

```

```

db.Conversaciones.insertMany([

  {fecha: new Date('2023-03-30'),mensaje: 'Presencia de cucarachas en el área de almacenamiento de Cafetería Aroma Fresco.',inspID: 3,inspResultado: 'Falla'},

  {fecha: new Date('2023-08-25'),mensaje: 'Se encontraron múltiples violaciones a las normas de higiene en Café Espresso.',inspID: 8,inspResultado: 'Falla'},

  {fecha: new Date('2023-12-15'),mensaje: 'Se detectaron fugas de gas en la cocina en Pizzería Bella Napoli.',inspID: 12,inspResultado: 'Falla'},

  {fecha: new Date('2024-08-20'),mensaje: 'Se observaron algunas deficiencias en la limpieza del área de almacenamiento en Restaurante Mar y Tierra.',inspID: 15,inspResultado: 'Falla'}

])

```

Consulta A versión 1

```
db.Conversaciones.countDocuments({ inspResultado: "Falla" })
```

Consulta A versión 2

```

db.Conversaciones.aggregate([

  {

    $group: {

```

```

        _id: "$violCodigo"
    }
},
{
    $count: "uniqueViolations"
}
])

```

Consulta B

```

db.Inspecciones.aggregate([
    { $match: { inspResultado: /Pasa/i } },

    {
        $group: {
            _id: "$estNumero",
            totalAprobadas: { $sum: 1 }
        }
    },

    { $sort: { totalAprobadas: -1 } },

    { $limit: 3 },

    // utilizando el lookup, obtenemos más detalles del establecimiento
    {
        $lookup: {
            from: "Establecimientos",
            localField: "_id",
            foreignField: "estNumero",
            as: "establecimiento"
        }
    },

```



```
// utilizando project, proyectamos los campos que quieres utilizar de establecimientos
{
  $project: {
    _id: 0,
    estNumero: "$_id",
    estNombre: "$establecimiento.estNombre",
    estDireccion: "$establecimiento.estDireccion",
    totalAprobadas: 1
  }
}
])
```

Consulta C

```
db.Conversaciones.updateMany(
  { "inspResultado": "Falla" },
  { $addToSet: { etiquetas: "IMPORTANTE" } }
);

db.Conversaciones.find()
```

Consultas Extras

Consulta número total de inspecciones por establecimiento

```
db.Inspecciones.aggregate([ { $group: { _id: "$estNumero", totalInspecciones: { $sum: 1 } } } ])
```

Consultar las inspecciones agrupadas por riesgo

```
db.Inspecciones.aggregate([ { $group: { _id: "$inspRiesgo", totalInspecciones: { $sum: 1 } } } ])
```

Buscar establecimientos que han tenido inspecciones con resultado de "Falla"

```
db.Inspecciones.find({ inspResultado: "Falla" }).forEach(function(inspeccion) {
  db.Establecimientos.findOne({ estNumero: inspeccion.estNumero })
})
```

Buscar inspecciones que han sido marcadas como "IMPORTANTE"

```
db.Conversaciones.find({ etiquetas: "IMPORTANTE" })
```