



INFORME DE PRUEBA PRÁCTICA

I. PORTADA

Tema:	Prueba Practica Primer Parcial: Fragmentación vertical + horizontal (mixta) con vista global
Unidad de Organización Curricular:	<u>PROFESIONAL</u>
Nivel y Paralelo:	5to “A”
Alumnos participantes:	Cholota Guaman Carlos Sebastian Mazabanda Pilamunga Diego Abraham Tixilema Puaquiza Kevin Alexander Tubon Chipantiza Danilo Alexander
Asignatura:	Sistemas de Bases De Datos Distribuidos
Docente:	Ing. Caiza Caizabuan Jose Ruben

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Diseñar e implementar, desde cero, una BD que aplique fragmentación mixta sobre la tabla **Alumnos**, simulando varios sitios en una sola instancia, y exponer una vista global que reconstruya los datos de forma transparente

Específicos:

- Crear una Base de Datos desde cero incluyendo tablas y datos de prueba.
- Crear fragmentaciones horizontales y verticales
- Exponer una vista global

2.2 Modalidad

Presencial

2.3 Tiempo de duración

Presenciales: 3

No presenciales: 0

2.4 Instrucciones

Escenario

Simula tres “sedes” como esquemas: HUACHI, INGAHURCO, QUEROCHACA.

Tabla global conceptual: Alumnos (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Ciudad, Sede, Carrera, FechaIngreso).

- Fragmentación horizontal por Sede → cada sede almacena solo sus alumnos.
 - Fragmentación vertical dentro de cada sede:
 - o Fragmento V1 (identificación/contacto): AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
 - o Fragmento V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso
- (Ambos fragmentos incluyen AlumnoID para permitir JOIN).

Entregables del estudiante (un solo PDF + .sql)

1. Diseño de fragmentación (1 página): diagrama simple que muestre horizontal por sede y vertical en V1/V2 (justifica por qué tu fragmentación preserva reconstrucción con JOIN y UNION ALL).
2. Script .sql con todo lo ejecutado (creación BD, esquemas, tablas, datos, vistas).
3. Evidencias (capturas):
 - o Esquemas y tablas creadas.
 - o SELECT COUNT(*) y consultas por sede/carrera sobre dbo.Alumnos.
 - o Demostración de transparencia: una consulta global que no “sepa” de los fragmentos y devuelva datos unificados.



2.5 Conclusiones: 5–7 líneas sobre ventajas/desafíos de la fragmentación y cómo este diseño se podría replicar (opcional) para alta disponibilidad.

2.6 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

- SQL Server Management
- Editor de texto (Word)
- Internet

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☒ Plataformas educativas
- ☒ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ☐ Recursos audiovisuales
- ☐ Gamificación
- ☐ Inteligencia Artificial

Otros (Especifique): _____

2.7 Actividades desarrolladas

Instrucciones

Escenario

Simula tres “sedes” como esquemas: HUACHI, INGAHURCO, QUEROCHACA.

Tabla global conceptual: Alumnos (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Ciudad, Sede, Carrera, FechaIngreso).

- Fragmentación horizontal por Sede → cada sede almacena solo sus alumnos.
 - Fragmentación vertical dentro de cada sede:
 - o Fragmento V1 (identificación/contacto): AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
 - o Fragmento V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso
- (Ambos fragmentos incluyen AlumnoID para permitir JOIN).

Entregables del estudiante

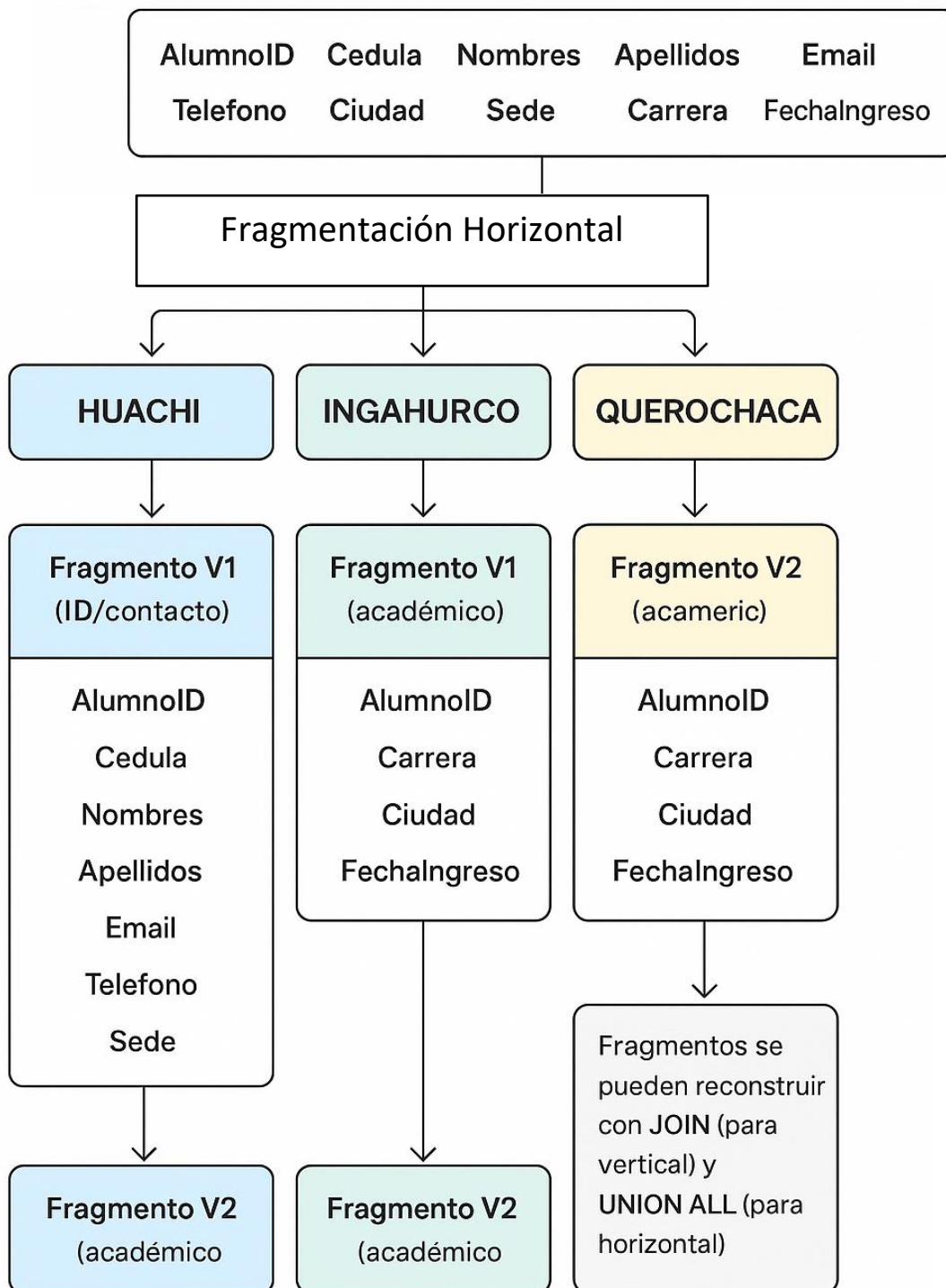
1. Diseño de fragmentación: diagrama simple que muestre horizontal por sede y vertical en V1/V2 (justifica por qué tu fragmentación preserva reconstrucción con JOIN y UNION ALL).
2. Script .sql con todo lo ejecutado (creación BD, esquemas, tablas, datos, vistas).
3. Evidencias (capturas):
 - o Esquemas y tablas creadas.
 - o SELECT COUNT(*) y consultas por sede/carrera sobre dbo.Alumnos.
 - o Demostración de transparencia: una consulta global que no “sepa” de los fragmentos y devuelva datos unificados.
4. Conclusiones: 5–7 líneas sobre ventajas/desafíos de la fragmentación y cómo este diseño se podría replicar para alta disponibilidad.



2.7.1 Resultados obtenidos

2.7.1.1 Diseño de fragmentación en diagrama

**Tabla global
Alumnos**





2.7.2 Desarrollo en SQL Server Management

2.7.2.1 Creación de la Base de Datos

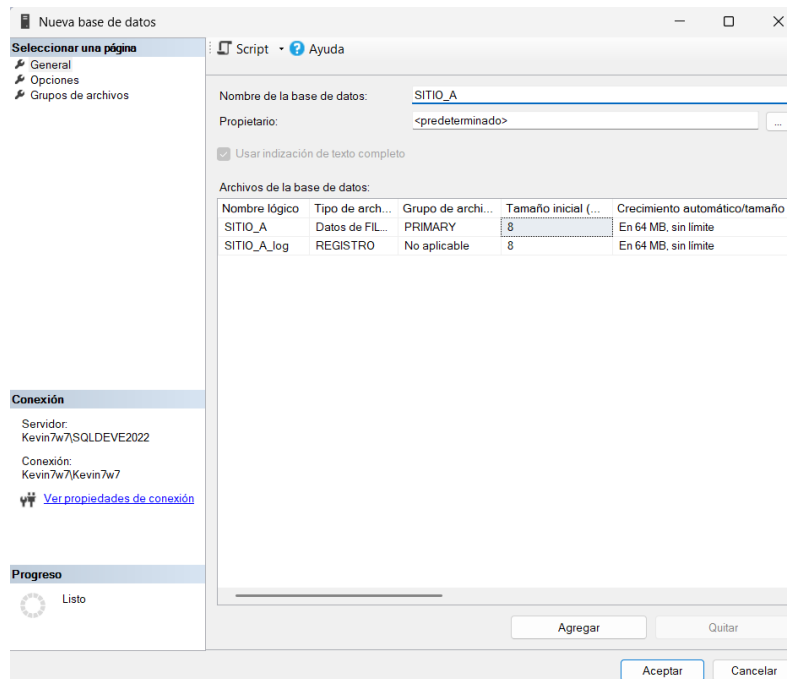


Ilustración 1. Creación de instancia "SITIO_A"

2.7.2.2 Crear la en SQL de la Base de datos

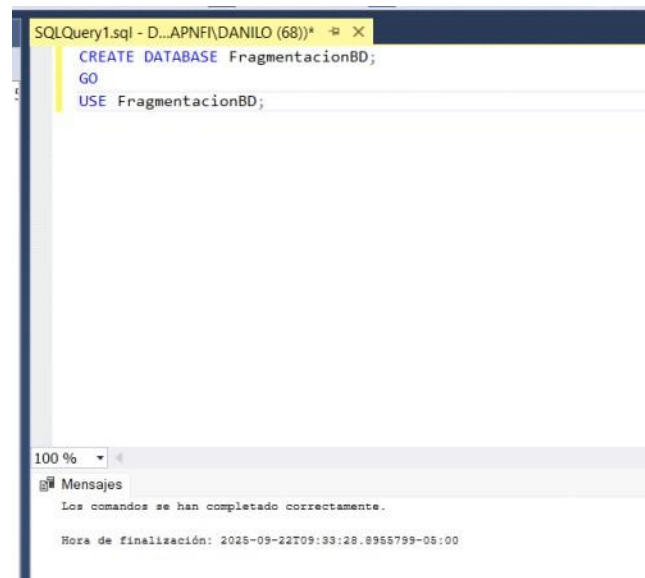
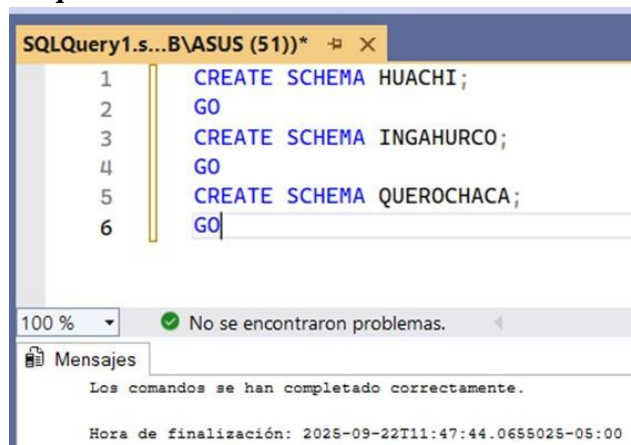


Ilustración 2. Creación en SQL Server

2.7.2.3 Creación de esquemas



```
SQLQuery1.s...B\ASUS (51))*
1 CREATE SCHEMA HUACHI;
2 GO
3 CREATE SCHEMA INGAHURCO;
4 GO
5 CREATE SCHEMA QUEROCHACA;
6 GO
```

100 % No se encontraron problemas.

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-09-22T11:47:44.0655025-05:00

Ilustración 3. Creación de esquemas

2.7.2.4 Crea las tablas simulando la fragmentacion mixta



```
SQLQuery1.sql - D...APNFI\DANILO (68))*
-- HUACHI
CREATE TABLE dbo.HUACHI_Alumnos_V1 (
    AlumnoID INT PRIMARY KEY,
    Cedula VARCHAR(10),
    Nombres VARCHAR(50),
    Apellidos VARCHAR(50),
    Email VARCHAR(100),
    Telefono VARCHAR(15),
    Sede VARCHAR(20)
);

CREATE TABLE dbo.HUACHI_Alumnos_V2 (
    AlumnoID INT PRIMARY KEY,
    Carrera VARCHAR(50),
    Ciudad VARCHAR(50),
    FechaIngreso DATE
);

-- INGAHURCO
CREATE TABLE dbo.INGAHURCO_Alumnos_V1 (
    AlumnoID INT PRIMARY KEY,
    Cedula VARCHAR(10),
    Nombres VARCHAR(50),
    Apellidos VARCHAR(50),
    Email VARCHAR(100),
    Telefono VARCHAR(15),
    Sede VARCHAR(20)
);

CREATE TABLE dbo.INGAHURCO_Alumnos_V2 (
    AlumnoID INT PRIMARY KEY,
    Carrera VARCHAR(50),
    Ciudad VARCHAR(50),
    FechaIngreso DATE
);
```

100 % Los comandos se han completado correctamente.

Mensajes

Hora de finalización: 2025-09-22T09:45:26.4083189-05:00

100 % Consulta ejecutada correctamente.

Ilustración 4. Creacion de tablas

En este paso se crean las tablas que representan la fragmentación mixta del esquema original. La fragmentación horizontal se aplica separando los datos por sede (HUACHI, INGAHURCO y QUEROCHACA), mientras que la fragmentación vertical divide los atributos en dos grupos: datos personales (Alumnos_V1) y datos académicos (Alumnos_V2). Cada sede cuenta con sus propias tablas fragmentadas, y todas comparten la clave primaria AlumnoID, lo que permite reconstruir los registros completos mediante operaciones de unión (JOIN).



2.7.2.5 Verificación de diagrama

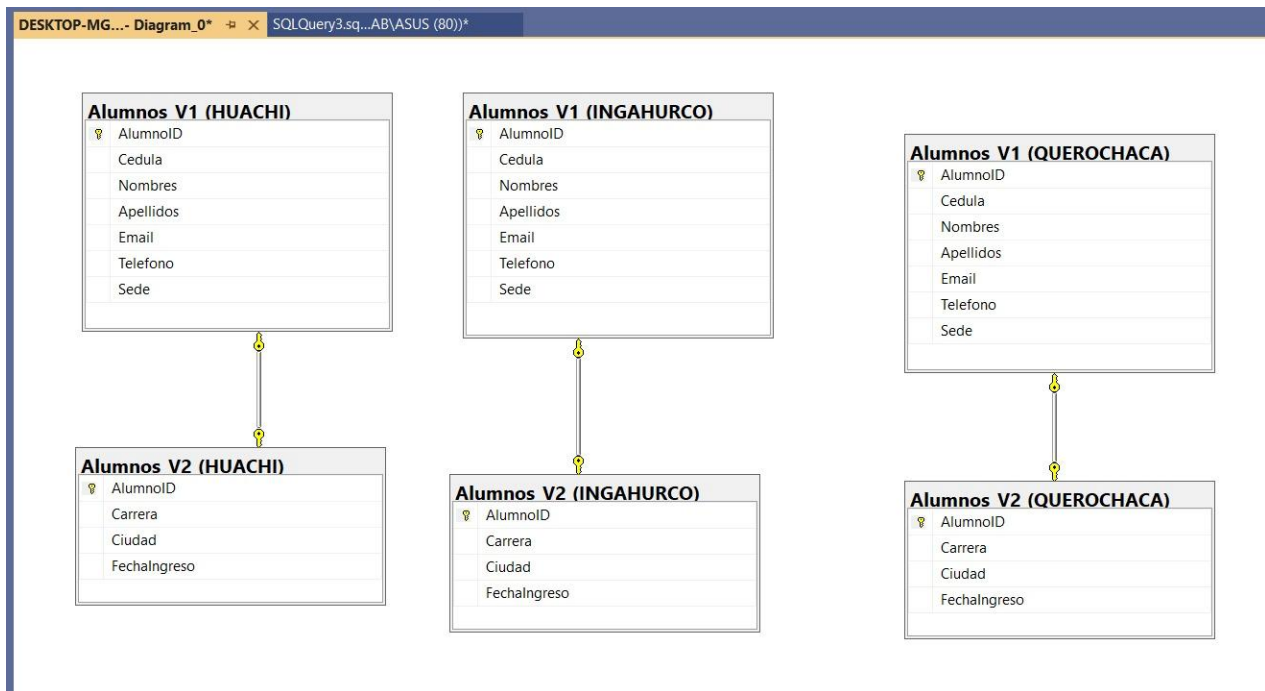


Ilustración 5. Diagrama

2.7.2.6 Insertar datos de ejemplo

```
SQLQuery1.sql - D:\APNFI\DAÑILO (61) (*)
-- HUACHI
INSERT INTO dbo.HUACHI_Alumnos_V1 VALUES (1, '0102030405', 'Luis', 'Pérez', 'luis@ejemplo.com', '0991234567', 'HUACHI');
INSERT INTO dbo.HUACHI_Alumnos_V2 VALUES (1, 'Ingeniería', 'Ambato', '2022-09-01');

-- INGAHURCO
INSERT INTO dbo.INGAHURCO_Alumnos_V1 VALUES (2, '0203040506', 'Ana', 'Gómez', 'ana@ejemplo.com', '0987654321', 'INGAHURCO');
INSERT INTO dbo.INGAHURCO_Alumnos_V2 VALUES (2, 'Contabilidad', 'Ambato', '2021-03-15');

-- QUEROCHACA
INSERT INTO dbo.QUEROCHACA_Alumnos_V1 VALUES (3, '0304050607', 'Carlos', 'Rosa', 'carlos@ejemplo.com', '0976543210', 'QUEROCHACA');
INSERT INTO dbo.QUEROCHACA_Alumnos_V2 VALUES (3, 'Agronomía', 'Quero', '2020-07-10');

100 %
Mensajes
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
(1 fila afectada)
Hora de finalización: 2025-09-22T09:55:08.3318228-05:00
```

Ilustración 6. Inserccion de datos



```
SQLQuery1.sql - D:\APNFI\DANILO (61))* -# X
-- HUACHI
INSERT INTO dbo.HUACHI_Alumnos_V1 VALUES (4, '1102030405', 'Maria', 'Salazar', 'maria@huachi.edu.ec', '0998887766', 'HUACHI');
INSERT INTO dbo.HUACHI_Alumnos_V2 VALUES (4, 'Administración', 'Ambato', '2023-01-15');

INSERT INTO dbo.HUACHI_Alumnos_V1 VALUES (5, '1103040506', 'David', 'Vega', 'david@huachi.edu.ec', '0997776655', 'HUACHI');
INSERT INTO dbo.HUACHI_Alumnos_V2 VALUES (5, 'Turismo', 'Ambato', '2022-05-10');

-- INGAHURCO
INSERT INTO dbo.INGAHURCO_Alumnos_V1 VALUES (6, '1202030405', 'Sofía', 'Paredes', 'sofia@ingahurco.edu.ec', '0988887766', 'INGAHURCO');
INSERT INTO dbo.INGAHURCO_Alumnos_V2 VALUES (6, 'Sistemas', 'Ambato', '2021-11-20');

INSERT INTO dbo.INGAHURCO_Alumnos_V1 VALUES (7, '1203040506', 'Jorge', 'Cabrera', 'jorge@ingahurco.edu.ec', '0987776655', 'INGAHURCO');
INSERT INTO dbo.INGAHURCO_Alumnos_V2 VALUES (7, 'Educación', 'Ambato', '2020-03-05');

-- QUEROCHACA
INSERT INTO dbo.QUEROCHACA_Alumnos_V1 VALUES (8, '1302030405', 'Valeria', 'Mendoza', 'valeria@querochaca.edu.ec', '0978887766', 'QUEROCHACA');
INSERT INTO dbo.QUEROCHACA_Alumnos_V2 VALUES (8, 'Veterinaria', 'Quero', '2022-08-12');

INSERT INTO dbo.QUEROCHACA_Alumnos_V1 VALUES (9, '1303040506', 'Esteban', 'Luna', 'esteban@querochaca.edu.ec', '0977776655', 'QUEROCHACA');
INSERT INTO dbo.QUEROCHACA_Alumnos_V2 VALUES (9, 'Mecánica', 'Quero', '2023-02-28');
```

Ilustración 4.1 Inserción de datos

Se insertaron registros únicos en las tablas Alumnos_V1 y Alumnos_V2 de cada sede para simular la distribución real de alumnos en un entorno fragmentado.

2.7.2.7 Creamos la vista global

```
SQLQuery1.sql - D:\APNFI\DANILO (61))* -# X
CREATE VIEW dbo.Alumnos AS
SELECT
    V1.AlumnoID,
    V1.Cedula,
    V1.Nombres,
    V1.Apellidos,
    V1.Email,
    V1.Telefono,
    V1.Sede,
    V2.Carrera,
    V2.Ciudad,
    V2.FechaIngreso
FROM dbo.HUACHI_Alumnos_V1 V1
JOIN dbo.HUACHI_Alumnos_V2 V2 ON V1.AlumnoID = V2.AlumnoID

UNION ALL

SELECT
    V1.AlumnoID,
    V1.Cedula,
    V1.Nombres,
    V1.Apellidos,
    V1.Email,
    V1.Telefono,
    V1.Sede,
    V2.Carrera,
    V2.Ciudad,
    V2.FechaIngreso
FROM dbo.INGAHURCO_Alumnos_V1 V1
JOIN dbo.INGAHURCO_Alumnos_V2 V2 ON V1.AlumnoID = V2.AlumnoID
```

100 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-09-22T10:09:57.5975541-05:00

Ilustración 7. Creación de vista

Para evitar errores de duplicidad al crear la vista global, se optó por seleccionar las columnas de forma explícita en lugar de utilizar SELECT *. Esto se debe a que, al realizar un JOIN entre fragmentos verticales, ambas tablas contienen campos con el mismo nombre (como AlumnoID), lo que genera conflictos al momento de definir la estructura de la vista.



2.7.2.8 Prueba de la vista global creada

SQLQuery1.sql - D:\APNFI\DANILO (61))*

```
USE FragmentacionBD;  
GO  
SELECT * FROM dbo.Alumnos;
```

100 %

Resultados Mensajes

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede	Carrera	Ciudad	FechaIngreso
1	1	0102030405	Luis	Pérez	luis@ejemplo.com	0991234567	HUACHI	Ingeniería	Ambato	2022-09-01
2	4	1102030405	María	Salazar	maria@huachi.edu.ec	0998887766	HUACHI	Administración	Ambato	2023-01-15
3	5	1103040506	David	Vega	david@huachi.edu.ec	0997776655	HUACHI	Turismo	Ambato	2022-05-10
4	2	0203040506	Ana	Gómez	ana@ejemplo.com	0987654321	INGAHURCO	Contabilidad	Ambato	2021-03-15
5	6	1202030405	Sofía	Paredes	sofia@ingahurco.edu.ec	0988887766	INGAHURCO	Sistemas	Ambato	2021-11-20
6	7	1203040506	Jorge	Cabrera	jorge@ingahurco.edu.ec	0987776655	INGAHURCO	Educación	Ambato	2020-03-05
7	3	0304050607	Carlos	Mora	carlos@ejemplo.com	0976543210	QUEROCH...	Agronomía	Quero	2020-07-10
8	8	1302030405	Valeria	Mendo...	valeria@querochaca.e...	0978887766	QUEROCH...	Veterinaria	Quero	2022-08-12
9	9	1303040506	Esteban	Luna	esteban@querochaca...	0977776655	QUEROCH...	Mecánica	Quero	2023-02-28

Ilustración 8. Comprobacion de vista

Se ejecutó una consulta sobre la vista `dbo.Alumnos` para verificar que los datos fragmentados por sede y por tipo de información se integran correctamente. El resultado muestra registros completos de alumnos provenientes de distintas sedes, lo que confirma que la reconstrucción de la tabla original mediante JOIN y UNION ALL fue exitosa.

2.7.2.9 Conteo por sede

SQLQuery1.sql - D:\APNFI\DANILO (61))*

```
SELECT Sede, COUNT(*) AS TotalPorSede  
FROM dbo.Alumnos  
GROUP BY Sede;
```

100 %

Resultados Mensajes

	Sede	TotalPorSede
1	HUACHI	3
2	INGAHURCO	3
3	QUEROCHACA	3

Ilustración 9. Conteo de sede total

SQLQuery1.sql - D:\APNFI\DANILO (61))*

```
SELECT Carrera, COUNT(*) AS TotalPorCarrera  
FROM dbo.Alumnos  
GROUP BY Carrera;
```

100 %

Resultados Mensajes

	Carrera	TotalPorCarrera
1	Administración	1
2	Agronomía	1
3	Contabilidad	1
4	Educación	1
5	Ingeniería	1
6	Mecánica	1
7	Sistemas	1
8	Turismo	1
9	Veterinaria	1

Ilustración 10. Conteo por carrera



2.7.2.10 Fragmentación horizontal por Sede → cada sede almacena solo sus alumnos.

SQLQuery1.sql - D:\APNFI\DANILO (61))

```
-- Ver alumnos de la sede HUACHI
SELECT * FROM dbo.HUACHI_Alumnos_V1;

-- Ver alumnos de la sede INGAHURCO
SELECT * FROM dbo.INGAHURCO_Alumnos_V1;

-- Ver alumnos de la sede QUEROCHACA
SELECT * FROM dbo.QUEROCHACA_Alumnos_V1;
```

100 %

Resultados Mensajes

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	1	0102030405	Luis	Pérez	luis@ejemplo.com	0991234567	HUACHI
2	4	1102030405	Maria	Salazar	maria@huachi.edu.ec	0998887766	HUACHI
3	5	1103040506	David	Vega	david@huachi.edu.ec	0997776655	HUACHI

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	2	0203040506	Ana	Gómez	ana@ejemplo.com	0987654321	INGAHURCO
2	6	1202030405	Sofia	Paredes	sofia@ingahurco.edu.ec	0988887766	INGAHURCO
3	7	1203040506	Jorge	Cabrera	jorge@ingahurco.edu.ec	0987776655	INGAHURCO

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	3	0304050607	Carlos	Mora	carlos@ejemplo.com	0976543210	QUEROCHACA
2	8	1302030405	Valeria	Mendoza	valeria@querochaca.edu.ec	0978887766	QUEROCHACA
3	9	1303040506	Esteban	Luna	esteban@querochaca.edu.ec	0977776655	QUEROCHACA

Ilustración 11. Fragmentación de cada sede

2.7.2.11 Fragmentación vertical dentro de cada sede:

Fragmento V1 (identificación/contacto): AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede

SQLQuery1.sql - D:\APNFI\DANILO (61))

```
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
FROM dbo.HUACHI_Alumnos_V1;

-- INGAHURCO: Fragmento V1 (identificación/contacto)
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
FROM dbo.INGAHURCO_Alumnos_V1;

-- QUEROCHACA: Fragmento V1 (identificación/contacto)
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
FROM dbo.QUEROCHACA_Alumnos_V1;
```

100 %

Resultados Mensajes

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	1	0102030405	Luis	Pérez	luis@ejemplo.com	0991234567	HUACHI
2	4	1102030405	Maria	Salazar	maria@huachi.edu.ec	0998887766	HUACHI
3	5	1103040506	David	Vega	david@huachi.edu.ec	0997776655	HUACHI

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	2	0203040506	Ana	Gómez	ana@ejemplo.com	0987654321	INGAHURCO
2	6	1202030405	Sofia	Paredes	sofia@ingahurco.edu.ec	0988887766	INGAHURCO
3	7	1203040506	Jorge	Cabrera	jorge@ingahurco.edu.ec	0987776655	INGAHURCO

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	3	0304050607	Carlos	Mora	carlos@ejemplo.com	0976543210	QUEROCHACA
2	8	1302030405	Valeria	Mendoza	valeria@querochaca.edu.ec	0978887766	QUEROCHACA
3	9	1303040506	Esteban	Luna	esteban@querochaca.edu.ec	0977776655	QUEROCHACA



2.7.2.12 Fragmentación vertical dentro de cada sede:

Fragmento V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso (Ambos fragmentos incluyen AlumnoID para permitir JOIN).

AlumnoID	Carrera	Ciudad	FechaIngreso
1	Ingeniería	Ambato	2022-09-01
2	Administración	Ambato	2023-01-15
3	Turismo	Ambato	2022-05-10

AlumnoID	Carrera	Ciudad	FechaIngreso
1	Contabilidad	Ambato	2021-03-15
2	Sistemas	Ambato	2021-11-20
3	Educación	Ambato	2020-03-05

AlumnoID	Carrera	Ciudad	FechaIngreso
1	Agronomía	Quero	2020-07-10
2	Veterinaria	Quero	2022-08-12
3	Mecánica	Quero	2023-02-28

Ilustración 13. Fragmento V2

2.7.2.13 Demostración de transparencia: una consulta global que no “sepa” de los fragmentos y devuelva datos unificados

SQLQuery1.sql - D...APNF\IDANILO (61))*

SELECT * FROM dbo.Alumnos;

100 %

Resultados Mensajes

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede	Carrera	Ciudad	FechaIngreso
1	1	0102030405	Luis	Pérez	luis@ejemplo.com	0991234567	HUACHI	Ingeniería	Ambato	2022-09-01
2	4	1102030405	Maria	Salazar	maria@huachi.edu.ec	0998887766	HUACHI	Administración	Ambato	2023-01-15
3	5	1103040506	David	Vega	david@huachi.edu.ec	0997776655	HUACHI	Turismo	Ambato	2022-05-10
4	2	0203040506	Ana	Gómez	ana@ejemplo.com	0987654321	INGAHURCO	Contabilidad	Ambato	2021-03-15
5	6	1202030405	Sofia	Paredes	sofia@ingahurco.edu.ec	0988887766	INGAHURCO	Sistemas	Ambato	2021-11-20
6	7	1203040506	Jorge	Cabrera	jorge@ingahurco.edu.ec	0987776655	INGAHURCO	Educación	Ambato	2020-03-05
7	3	0304050607	Carlos	Mora	carlos@ejemplo.com	0976543210	QUEROCHACA	Agronomía	Quero	2020-07-10
8	8	1302030405	Valeria	Mendoza	valeria@querochaca.edu.ec	0978887766	QUEROCHACA	Veterinaria	Quero	2022-08-12
9	9	1303040506	Esteban	Luna	esteban@querochaca.edu.ec	0977776655	QUEROCHACA	Mecánica	Quero	2023-02-28

Ilustración 14. Transparencia



2.7.3 SCRIPT SQL

```
CREATE DATABASE FragmentacionBD;
GO

USE FragmentacionBD;
GO

CREACIÓN DE LAS TABLAS SIMULANDO LA FRAGMENTACION MIXTA
-- HUACHI
CREATE TABLE dbo.HUACHI_Alumnos_V1 (
    AlumnoID INT PRIMARY KEY,
    Cedula VARCHAR(10),
    Nombres VARCHAR(50),
    Apellidos VARCHAR(50),
    Email VARCHAR(100),
    Telefono VARCHAR(15),
    Sede VARCHAR(20)
);

CREATE TABLE dbo.HUACHI_Alumnos_V2 (
    AlumnoID INT PRIMARY KEY,
    Carrera VARCHAR(50),
    Ciudad VARCHAR(50),
    FechaIngreso DATE
);

-- INGAHURCO
CREATE TABLE dbo.INGAHURCO_Alumnos_V1 (
    AlumnoID INT PRIMARY KEY,
    Cedula VARCHAR(10),
    Nombres VARCHAR(50),
    Apellidos VARCHAR(50),
    Email VARCHAR(100),
    Telefono VARCHAR(15),
    Sede VARCHAR(20)
);

CREATE TABLE dbo.INGAHURCO_Alumnos_V2 (
    AlumnoID INT PRIMARY KEY,
    Carrera VARCHAR(50),
    Ciudad VARCHAR(50),
    FechaIngreso DATE
);

-- QUEROCHACA
CREATE TABLE dbo.QUEROCHACA_Alumnos_V1 (
    AlumnoID INT PRIMARY KEY,
    Cedula VARCHAR(10),
    Nombres VARCHAR(50),
    Apellidos VARCHAR(50),
    Email VARCHAR(100),
    Telefono VARCHAR(15),
    Sede VARCHAR(20));
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



```
CREATE TABLE dbo.QUEROCHACA_Alumnos_V2 (  
    AlumnoID INT PRIMARY KEY,  
    Carrera VARCHAR(50),  
    Ciudad VARCHAR(50),  
    FechaIngreso DATE  
);
```

INGRESO DE DATOS

-- HUACHI

```
INSERT INTO dbo.HUACHI_Alumnos_V1 VALUES (4, '1102030405',  
'María', 'Salazar', 'maria@huachi.edu.ec', '0998887766',  
'HUACHI');
```

```
INSERT INTO dbo.HUACHI_Alumnos_V2 VALUES (4,  
'Administración', 'Ambato', '2023-01-15');
```

```
INSERT INTO dbo.HUACHI_Alumnos_V1 VALUES (5, '1103040506',  
'David', 'Vega', 'david@huachi.edu.ec', '0997776655',  
'HUACHI');
```

```
INSERT INTO dbo.HUACHI_Alumnos_V2 VALUES (5, 'Turismo',  
'Ambato', '2022-05-10');
```

-- INGAHURCO

```
INSERT INTO dbo.INGAHURCO_Alumnos_V1 VALUES (6,  
'1202030405', 'Sofía', 'Paredes', 'sofia@ingahurco.edu.ec',  
'0988887766', 'INGAHURCO');
```

```
INSERT INTO dbo.INGAHURCO_Alumnos_V2 VALUES (6, 'Sistemas',  
'Ambato', '2021-11-20');
```

```
INSERT INTO dbo.INGAHURCO_Alumnos_V1 VALUES (7,  
'1203040506', 'Jorge', 'Cabrera', 'jorge@ingahurco.edu.ec',  
'0987776655', 'INGAHURCO');
```

```
INSERT INTO dbo.INGAHURCO_Alumnos_V2 VALUES (7,  
'Educación', 'Ambato', '2020-03-05');
```

-- QUEROCHACA

```
INSERT INTO dbo.QUEROCHACA_Alumnos_V1 VALUES (8,  
'1302030405', 'Valeria', 'Mendoza',  
'valeria@querochaca.edu.ec', '0978887766', 'QUEROCHACA');
```

```
INSERT INTO dbo.QUEROCHACA_Alumnos_V2 VALUES (8,  
'Veterinaria', 'Quero', '2022-08-12');
```

```
INSERT INTO dbo.QUEROCHACA_Alumnos_V1 VALUES (9,  
'1303040506', 'Esteban', 'Luna',  
'esteban@querochaca.edu.ec', '0977776655', 'QUEROCHACA');
```

```
INSERT INTO dbo.QUEROCHACA_Alumnos_V2 VALUES (9,  
'Mecánica', 'Quero', '2023-02-28');
```

CREACION DE LA VISTA GLOBAL CREADA

```
CREATE VIEW dbo.Alumnos AS
```

```
SELECT
```

```
    V1.AlumnoID,
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



V1.Nombres,
V1.Apellidos,
V1.Email,
V1.Telefono,
V1.Sede,
V2.Carrera,
V2.Ciudad,
V2.FechaIngreso

```
FROM dbo.HUACHI_Alumnos_V1 V1  
JOIN dbo.HUACHI_Alumnos_V2 V2 ON V1.AlumnoID = V2.AlumnoID
```

UNION ALL

SELECT

V1.AlumnoID,
V1.Nombres,
V1.Apellidos,
V1.Email,
V1.Telefono,
V1.Sede,
V2.Carrera,
V2.Ciudad,
V2.FechaIngreso

```
FROM dbo.INGAHURCO_Alumnos_V1 V1  
JOIN  dbo.INGAHURCO_Alumnos_V2  V2  ON  V1.AlumnoID  =  
V2.AlumnoID
```

UNION ALL

SELECT

V1.AlumnoID,
V1.Nombres,
V1.Apellidos,
V1.Email,
V1.Telefono,
V1.Sede,
V2.Carrera,
V2.Ciudad,
V2.FechaIngreso

```
FROM dbo.QUEROCHACA_Alumnos_V1 V1  
JOIN  dbo.QUEROCHACA_Alumnos_V2  V2  ON  V1.AlumnoID  =  
V2.AlumnoID;
```

PRUEBA DE LA VISTA GLOBAL CREADA
USE FragmentacionBD;
GO

```
SELECT * FROM dbo.Alumnos;
```

```
CONTEO POR SEDE  
SELECT Sede, COUNT(*) AS TotalPorSede  
FROM dbo.Alumnos  
GROUP BY Sede;
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



CONTEO POR CARRERA

```
SELECT Carrera, COUNT(*) AS TotalPorCarrera
FROM dbo.Alumnos
GROUP BY Carrera;
```

Fragmentación horizontal por Sede → cada sede almacena solo sus alumnos.

-- HUACHI

```
SELECT * FROM dbo.HUACHI_Alumnos_V1;
```

-- INGAHURCO

```
SELECT * FROM dbo.INGAHURCO_Alumnos_V1;
```

-- QUEROCHACA

```
SELECT * FROM dbo.QUEROCHACA_Alumnos_V1;
```

Fragmentación vertical dentro de cada sede:

Fragmento V1 (identificación/contacto): AlumnoID, Cedula, Nombres,

Apellidos, Email, Telefono, Sede

-- HUACHI

```
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email,
Telefono, Sede
FROM dbo.HUACHI_Alumnos_V1;
```

-- INGAHURCO

```
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email,
Telefono, Sede
FROM dbo.INGAHURCO_Alumnos_V1;
```

-- QUEROCHACA

```
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email,
Telefono, Sede
FROM dbo.QUEROCHACA_Alumnos_V1;
```

Fragmentación vertical dentro de cada sede:

Fragmento V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso

(Ambos fragmentos incluyen AlumnoID para permitir JOIN)

-- HUACHI

```
SELECT AlumnoID, Carrera, Ciudad, FechaIngreso
FROM dbo.HUACHI_Alumnos_V2;
```

-- INGAHURCO

```
SELECT AlumnoID, Carrera, Ciudad, FechaIngreso
FROM dbo.INGAHURCO_Alumnos_V2;
```

-- QUEROCHACA

```
SELECT AlumnoID, Carrera, Ciudad, FechaIngreso
FROM dbo.QUEROCHACA_Alumnos_V2;
```

demostración de transparencia: una consulta global que no “sepa” de los fragmentos y devuelva datos unificado

```
SELECT * FROM dbo.Alumnos;
```




2.8 Habilidades blandas empleadas en la práctica

- ☒ Liderazgo
- ☒ Trabajo en equipo
- ☐ Comunicación asertiva
- ☐ La empatía
- ☐ Pensamiento crítico
- ☐ Flexibilidad
- ☐ La resolución de conflictos
- ☐ Adaptabilidad
- ☐ Responsabilidad

2.9 Conclusiones

La fragmentación mixta permite distribuir datos eficientemente según criterios lógicos y funcionales. La fragmentación horizontal por sede mejora la localización de datos, mientras que la vertical facilita el acceso segmentado por tipo de información. Este diseño garantiza reconstrucción completa mediante JOIN y UNION ALL, y puede replicarse en entornos distribuidos reales para mejorar disponibilidad, escalabilidad y rendimiento. El principal desafío es mantener la integridad y sincronización entre fragmentos.

- La creación desde cero permitió comprender el ciclo completo de diseño e implementación de una base de datos. Al definir tablas, atributos y poblarlas con datos de prueba, se establecieron las bases para aplicar fragmentación y simular entornos distribuidos de forma controlada y funcional.
- La aplicación de fragmentación mixta demostró cómo dividir los datos estratégicamente mejora el rendimiento, la organización y la seguridad. La fragmentación horizontal por sede localiza los datos, mientras que la vertical permite separar información sensible o especializada
- La vista global logró abstraer la complejidad del sistema distribuido, permitiendo consultas unificadas sin que el usuario conozca los fragmentos subyacentes. Esto garantiza transparencia, integridad y facilidad de acceso, replicando el comportamiento de una base de datos centralizada sobre una arquitectura distribuida.



2.10 Recomendaciones

Automatizar la inserción y actualización de datos en los fragmentos para garantizar que cualquier cambio en un sitio se replique de forma inmediata y consistente en la vista global.

Implementar mecanismos de control de integridad (constraints, triggers o procedimientos almacenados) que aseguren que las claves primarias y foráneas no se pierdan al trabajar con fragmentos verticales y horizontales.

Simular entornos distribuidos reales en distintas instancias o servidores, de manera que se prueben aspectos como latencia, replicación y concurrencia entre sitios.

Documentar claramente las reglas de fragmentación y mantener un catálogo global de metadatos que describa qué información se almacena en cada fragmento, para facilitar administración y auditoría.

Optimizar el diseño de vistas globales incluyendo filtros o particiones que reduzcan costos de consulta cuando el volumen de datos aumente.

Extender el modelo a otras tablas relacionadas, aplicando fragmentación coherente para mantener consistencia en todo el sistema distribuido.

Probar casos de recuperación ante fallos (por ejemplo, si una sede no responde), verificando que la vista global pueda seguir funcionando con datos parciales o replicados.

2.11 Referencias bibliográficas

2.12 Anexos