



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE: INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE: Tecnologías de la Información

SISTEMAS DE BASES DE DATOS DISTRIBUIDOS

DOCENTE:

JOSE RUBEN CAIZA CAIZABUANO

AGOSTO 2025 - ENERO 2026

Tabla de contenido

[I. PORTADA 3](#_Toc209983680)

[II. INFORME DE GUÍA PRÁCTICA 3](#_Toc209983681)

[2.1 Objetivos 3](#_Toc209983682)

[2.2 Modalidad 3](#_Toc209983683)

[2.3 Tiempo de duración 3](#_Toc209983684)

[2.4 Instrucciones 3](#_Toc209983685)

[2.5 Listado de equipos, materiales y recursos 4](#_Toc209983686)

[2.6 Actividades desarrolladas 5](#_Toc209983687)

[2.6.1 CREACIÓN DE UNA NUEVA INSTANCIA DENTRO DE SQL SERVER 5](#_Toc209983688)

[2.6.2 DISEÑO DE FRAGMENTACIÓN MIXTA (H+V) correctamente justificado (incluye PK en fragmentos verticales) 8](#_Toc209983689)

[2.6.3 IMPLEMENTACIÓN TÉCNICA (BD, esquemas, tablas, constraints, datos) 8](#_Toc209983690)

[2.6.4 VISTAS LOCALES + VISTA GLOBAL que reconstruyen sin pérdida 12](#_Toc209983691)

[2.6.5 VERIFICACIÓN (consultas, conteos, filtros por sede/carrera) 14](#_Toc209983692)

[2.7 Resultados obtenidos 17](#_Toc209983693)

[2.8 Habilidades blandas empleadas en la práctica 17](#_Toc209983694)

[2.9 Conclusiones 17](#_Toc209983695)

[2.10 Recomendaciones 17](#_Toc209983696)

**INFORME DE** **EVALUACIÓN PRÁCTICA PRIMER PARCIAL**

# PORTADA

Tema: Fragmentación vertical + horizontal (mixta) con vista

global

Unidad de Organización Curricular: PROFESIONAL

Nivel y Paralelo: Quinto “A”

Alumnos participantes: Aldas Jordan Wellington Ismael

Caguasango Bayas Alex Patricio

Gómez Llerena Luis Fernando

Paredes Barrera Luis Enrique

Asignatura: Sistemas de Base de Datos Distribuidas

Docente: Ing. José Caiza, Mg.

# INFORME DE GUÍA PRÁCTICA

1. **PP**
2. **YY**

## Objetivos

1. Objetivo General:

Diseñar e implementar, desde cero, una BD que aplique fragmentación mixta (horizontal + vertical) sobre la tabla Alumnos, simulando varios sitios en una sola instancia, y exponer una vista global que reconstruya los datos de forma transparente (union de fragmentos horizontales y join de fragmentos verticales).

1. Objetivos Específicos:

* Diseñar la fragmentación horizontal por sede y vertical en fragmentos de identificación/contacto y académico, asegurando la preservación de la información
* Implementar en SQL Server la base de datos con sus esquemas, tablas, fragmentos y vista global que reconstruya los datos de forma transparente.
* Verificar el correcto funcionamiento mediante consultas, conteos y validación de la reconstrucción de la tabla original en la vista global.

## Modalidad

Presencial

## Tiempo de duración

**Presenciales:** 3

**No presenciales:** 0

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

Apellidos, Email, Telefono, Sede

* + 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):

* Esquemas y tablas creadas.
* SELECT COUNT(\*) y consultas por sede/carrera sobre dbo.Alumnos.
* 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 (opcional) para alta disponibilidad.

**Rúbrica (100 pts)**

• Diseño de fragmentación mixta (H + V) correctamente justificado (incluye PK en

fragmentos verticales) – 30 pts.

• Implementación técnica (BD, esquemas, tablas, constraints, datos) – 30 pts.

• Vistas locales + vista global que reconstruyen sin pérdida – 20 pts.

• Verificación (consultas, conteos, filtros por sede/carrera) – 10 pts.

• Informe (PDF) claro con capturas y conclusiones – 10 pts

## Listado de equipos, materiales y recursos

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

* Laptop
* SQL Server

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): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Actividades desarrolladas

### CREACIÓN DE UNA NUEVA INSTANCIA DENTRO DE SQL SERVER

Para iniciar el diseño de la base de datos distribuida simulada, se creó una nueva instancia de SQL Server denominada SITIO\_A. Esta instancia es necesaria porque permite trabajar de forma independiente con los esquemas y tablas que representan cada sede.

#### Ejecución del instalador de SQL Server

Al observar la ilustración 1 presentada a continuación logramos observar que al ejecutar el centro de instalación de SQL Server debemos escoger una carpeta en la cual se realizara la creación de la instancia.

CONSEJO: Escoger la carpeta en la que se encuentren los archivos del programa.

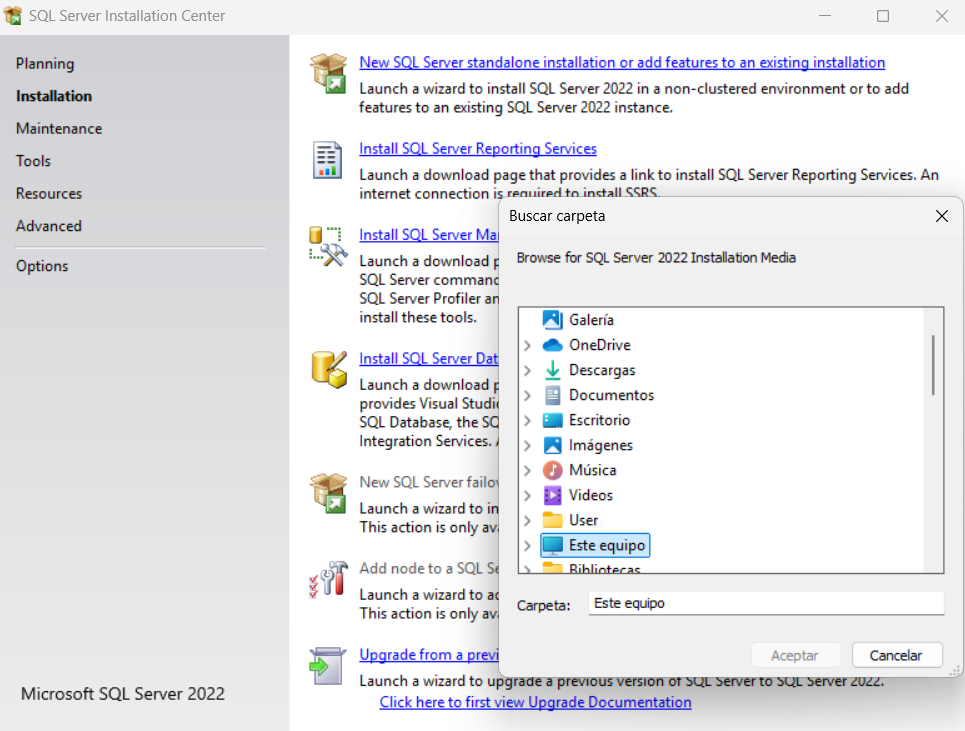
****

Ilustración 1 Selección de carpeta para la creación de la instancia

#### Configuración y Creación de la Instancia

Durante la configuración en el apartado de tipo de instalación se escoge la opción Realizar una nueva Instalación de SQL Server, como se puede observar en la ilustración 2.

Esto permitirá realizar la creación de la instancia.

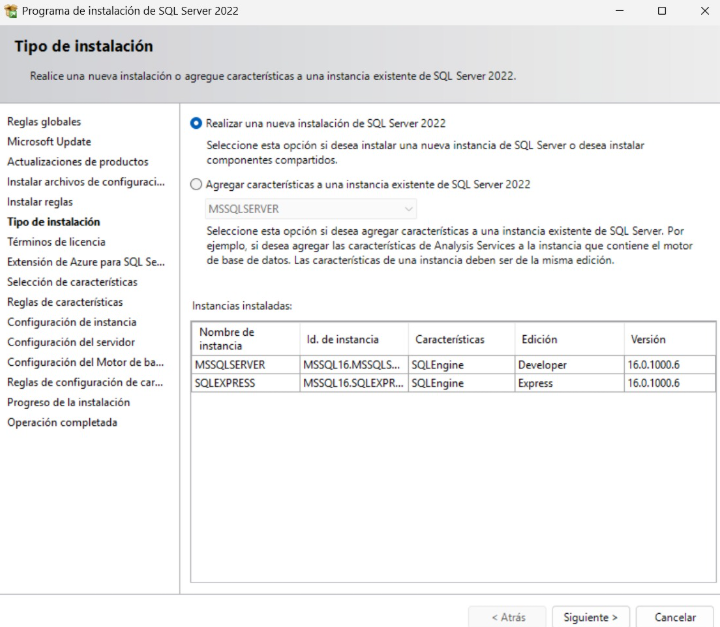


Ilustración 2 Tipo de instalación

En el apartado de **configuración de instancia** debemos seleccionar colocar un nombre propio a la nueva instancia y se colocara el nombre que se desee. En este caso SITIO\_A, como se observa en la ilustración 3.

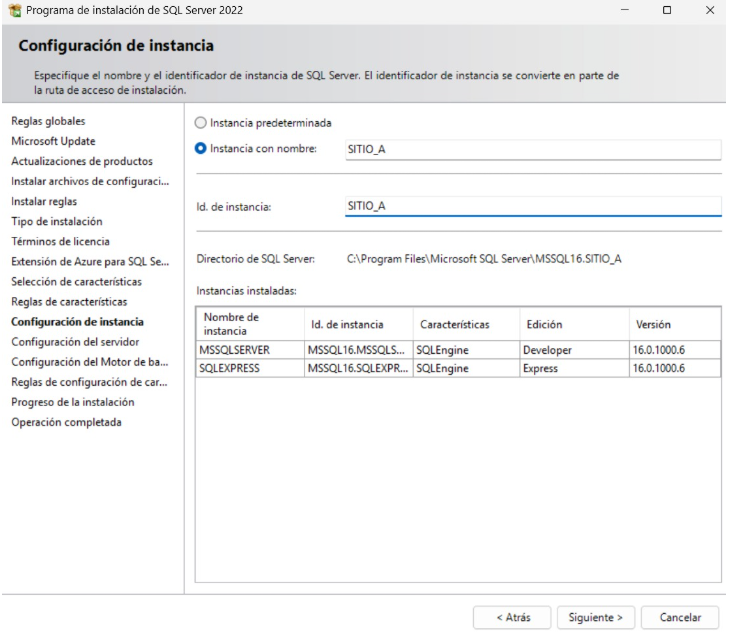


Ilustración 3 Asignar nombre a la nueva instancia

Una vez configurado todos los apartados se iniciará el proceso de instalación como se observa en la ilustración 4.

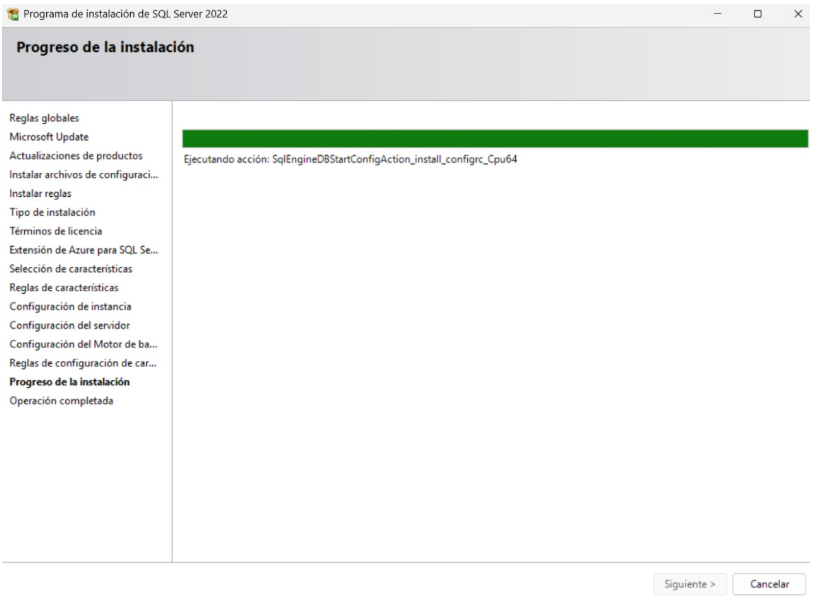
****

Ilustración 4 Progreso de instalación de la instancia

Como se puede observar en la Ilustración 5, la instancia ha sido creada correctamente y está lista para la posterior creación de la base de datos, esquemas y fragmentos que conformarán la vista global.

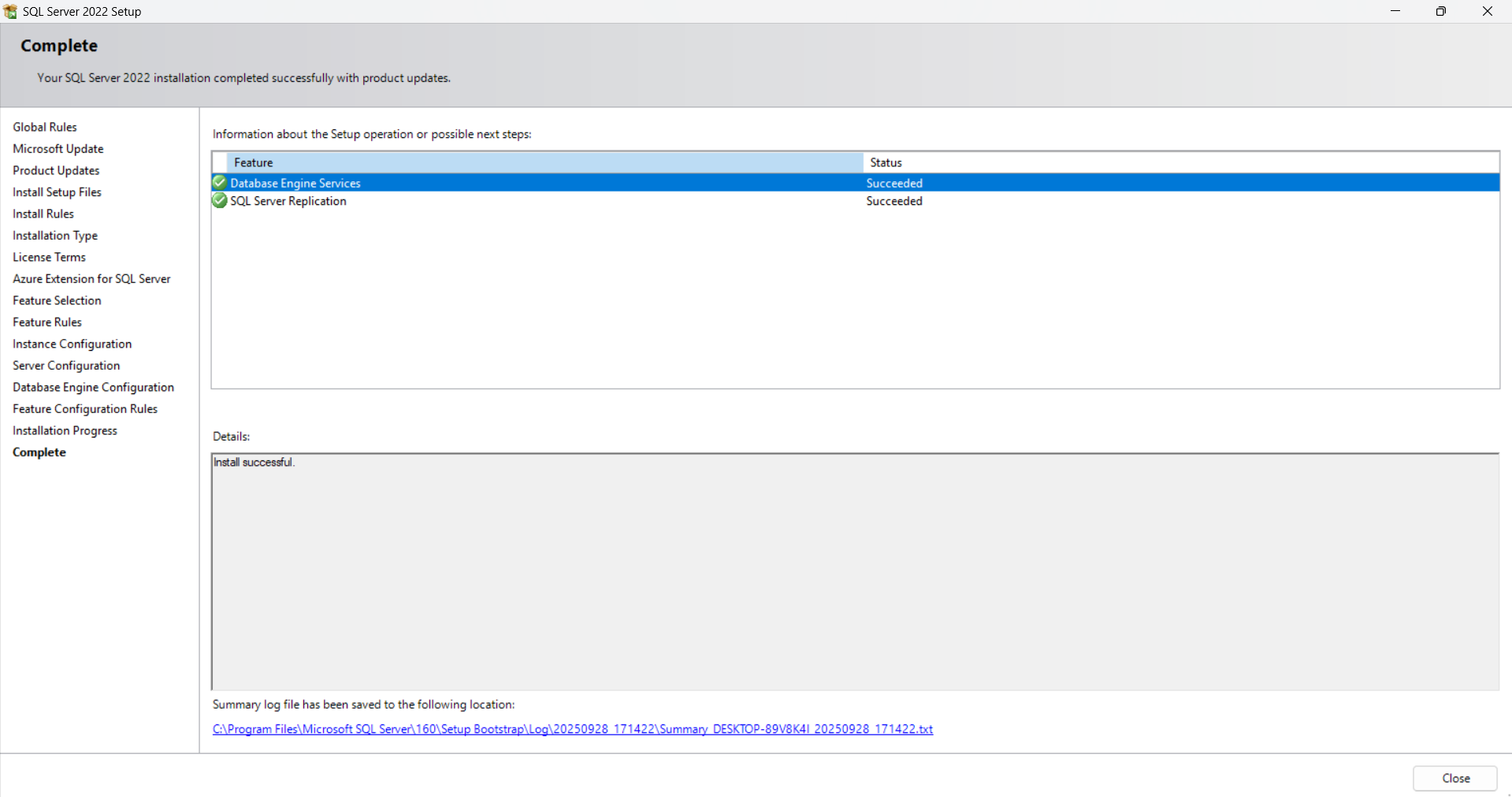
****

Ilustración 5 Creación Completa de la nueva Instancia

#### ¿Como ingresar a la instancia creada?

Para lograr ingresar en la instancia que creamos ejecutamos la aplicación de SQL Server y en la ventana en la cual debemos ingresar el nombre del servidor deberemos ingresar el nombre colocado a la instancia durante su creación, este nombre debe ir antecedido por .\

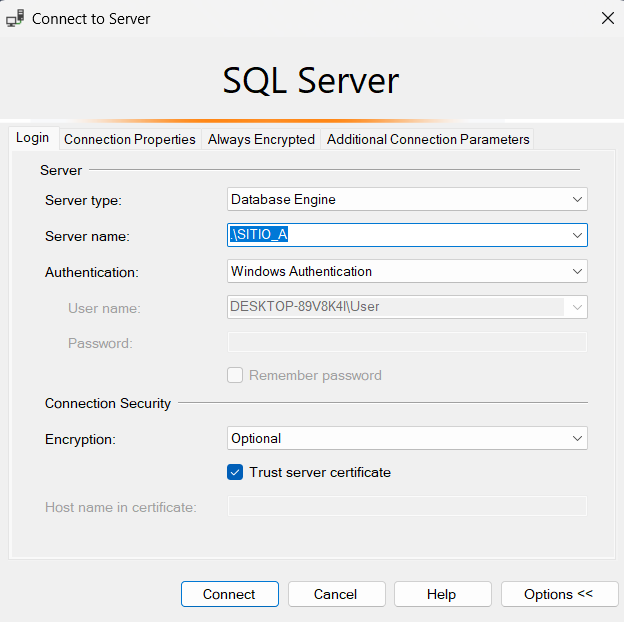


Ilustración 6 Ingreso a la Instancia creada

### DISEÑO DE FRAGMENTACIÓN MIXTA (H+V) correctamente justificado (incluye PK en fragmentos verticales)

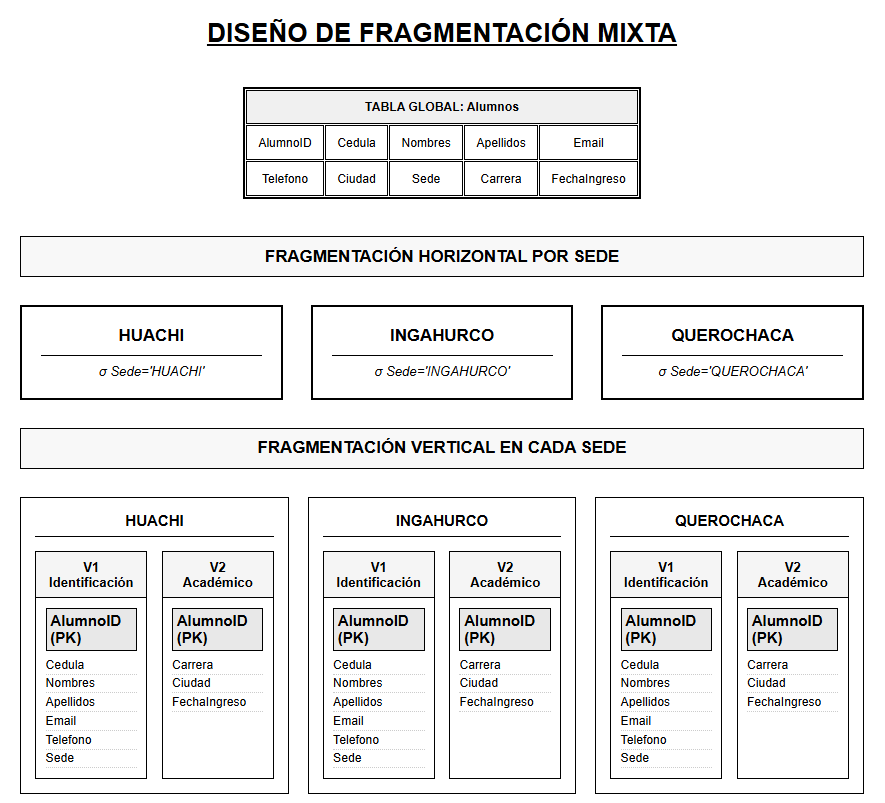


Ilustración 7 Diseño de Fragmentación Mixta

##### Justificación del diseño presentado (Ilustración 6):

La fragmentación mixta implementada divide inicialmente los datos de forma horizontal según la Sede (& Sede='X'), permitiendo que cada campus administre exclusivamente sus estudiantes y asegurando que ningún alumno se duplique entre fragmentos. Adicionalmente, cada fragmento horizontal se subdivide verticalmente en dos partes: V1 contiene información personal y de contacto (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede) orientada a la privacidad, mientras V2 almacena datos académicos institucionales (AlumnoID, Carrera, Ciudad, FechaIngreso) de uso administrativo frecuente. La clave AlumnoID se mantiene en ambos fragmentos verticales como requisito fundamental para realizar la reconstrucción mediante JOIN entre V1 y V2, y posteriormente UNION entre las tres sedes. Esta estrategia preserva la integridad total de los datos, garantiza acceso transparente mediante vistas globales, y optimiza tanto la seguridad como el rendimiento al separar información sensible de la operativa.

### IMPLEMENTACIÓN TÉCNICA (BD, esquemas, tablas, constraints, datos)

Una vez dentro de la instancia procederemos a la creación de nuestra base de datos.

1. Creación de la Base de Datos dentro de la Instancia:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 8 Creación de la Base de Datos

Ya creada la base de datos correctamente lo siguiente será la creación de esquemas:

1. Creación de Esquemas: HUACHI, INGAHURCO Y QUEROCHACA

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Ilustración 9 Creación de Esquemas

Continuando crearemos todas las tablas necesarias:

1. Creación de Tablas y Constraints:

Texto

El contenido generado por IA puede ser incorrecto.

Ilustración 10 Creación de Tabla INGAHURCO.Alumnos\_V1

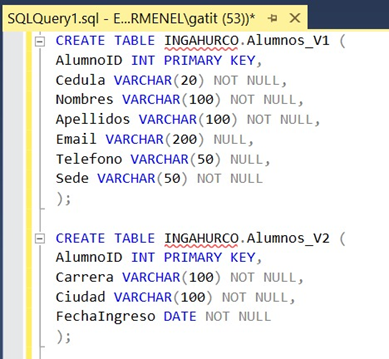


Ilustración 11 INGAHURCO.Alumnos\_V2

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 12 Creación de tablas HUACHI.Alumnos\_V1 y HUACHI.Alumnos\_V2

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 13 Creación de tablas QUEROCHACA.Alumnos\_V1 y QUEROCHACA.Alumnos\_V2

1. Verificación de la creación de tablas:

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Ilustración 14 Verificación Tablas Creadas

1. Verificación de Tablas y Constrains:

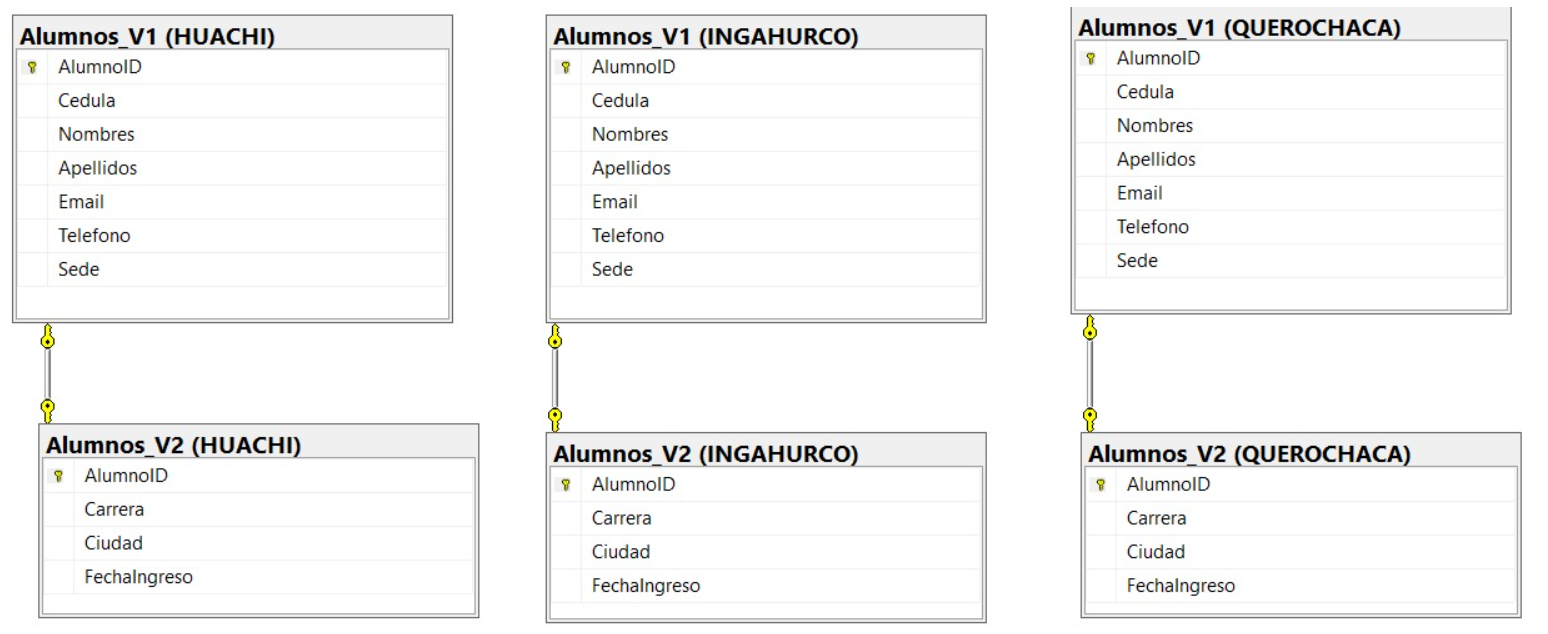
****

Ilustración 15 Diagrama de Tablas Creadas

Una vez creadas las tablas y verificado la correcta relación entre ellas gracias a los constrains procedemos a la inserción de datos:

1. Inserción de Datos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Ilustración 16 Inserción de Datos en las diferentes tablas

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 17 Inserción de Datos en las diferentes tablas

Al haber ingresado datos en nuestras tablas continuaremos con la creación de vistas. Los datos ingresados nos permitirán entender su comportamiento y además si existen perdidas de datos dentro de la base.

### VISTAS LOCALES + VISTA GLOBAL que reconstruyen sin pérdida

###### Creación de VISTAS LOCALES:

Vista para HUACHI:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Ilustración 18 Creación Vista Local (HUACHI)

Vista para INGAHURCO:

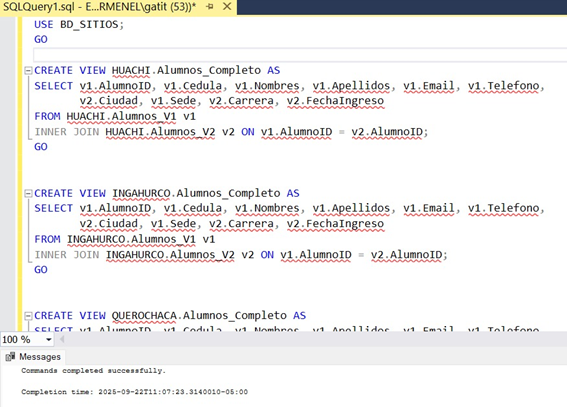


Ilustración 19 Creación Vista Local (INGAHURCO)

Vista para QUEROCHACA:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 20 Creación Vista Local (QUEROCHACA)

###### Creación de VISTA GLOBAL:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Ilustración 21 Creación Vista Global

La vista global creada en la ilustración 21, reconstruye todo sin perdidas.

### VERIFICACIÓN (consultas, conteos, filtros por sede/carrera)

1. Verificación Consultas

Consulta 1: Buscar un alumno por su Cédula en la tabla HUACHI.Alumnos\_V1.



Ilustración 22 Verificación de Consultas Ejemplo 1

Consulta 2: Consultar la carrera y ciudad de un alumno en INGAHURCO.Alumnos\_V2.

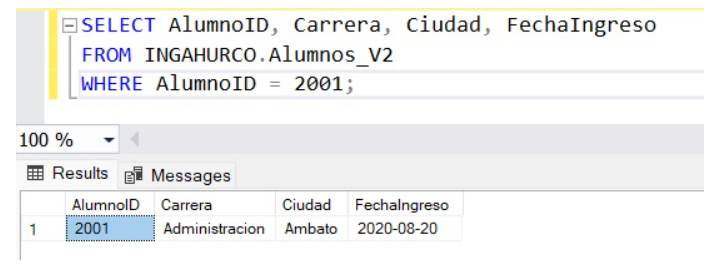


Ilustración 23 Verificación de Consultas Ejemplo 2

1. Verificación de Conteos

Ejemplo: Contar el número total de estudiantes separadas por sede.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 24 Verificación de Conteos Ejemplo 1

1. Verificación de Vistas Locales

Vistas creadas:

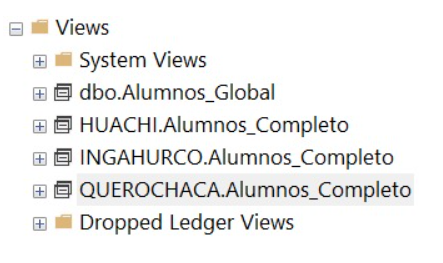
****

Ilustración 25 Todas las Vistas Creadas

Verificación Vista HUACHI:

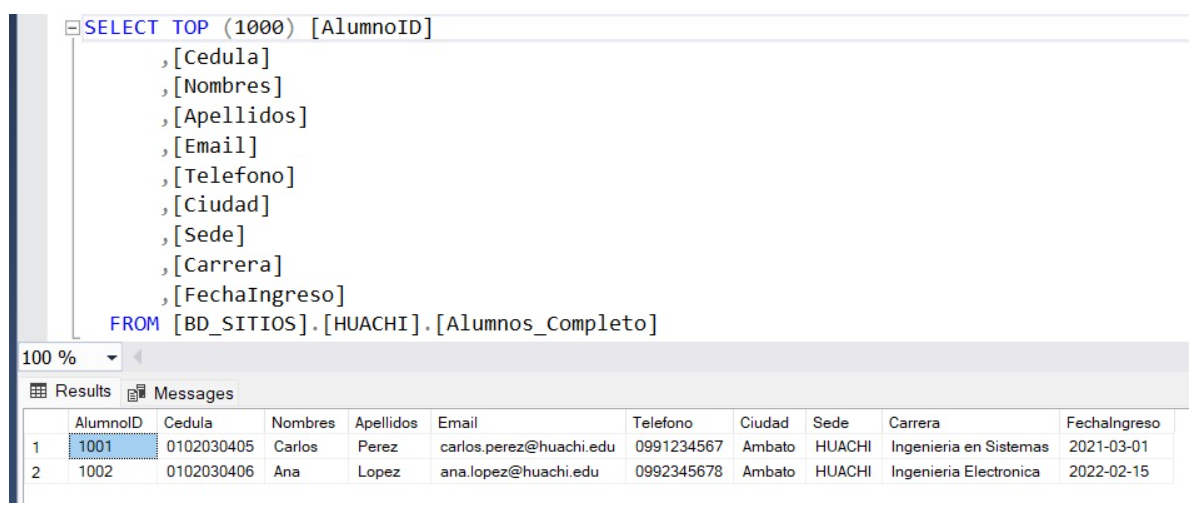


Ilustración 26 Verificación Vista HUACHI

Verificación Vista INGAHURCO:

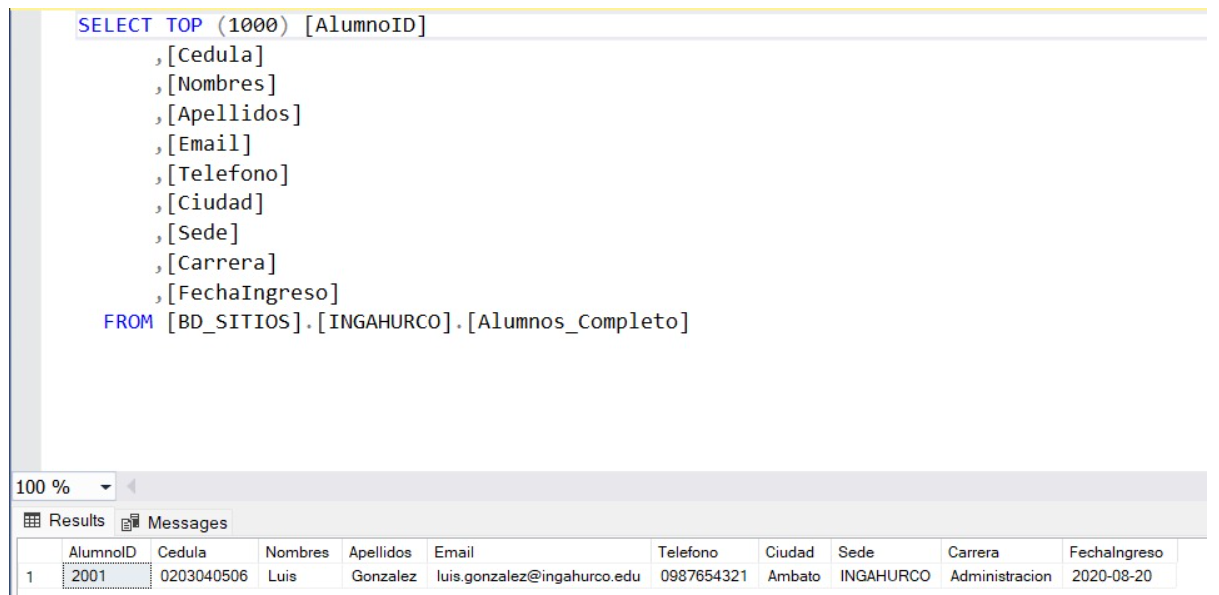
****

Ilustración 27 Verificación Vista INGAHURCO

Verificación Vista QUEROCHACA:

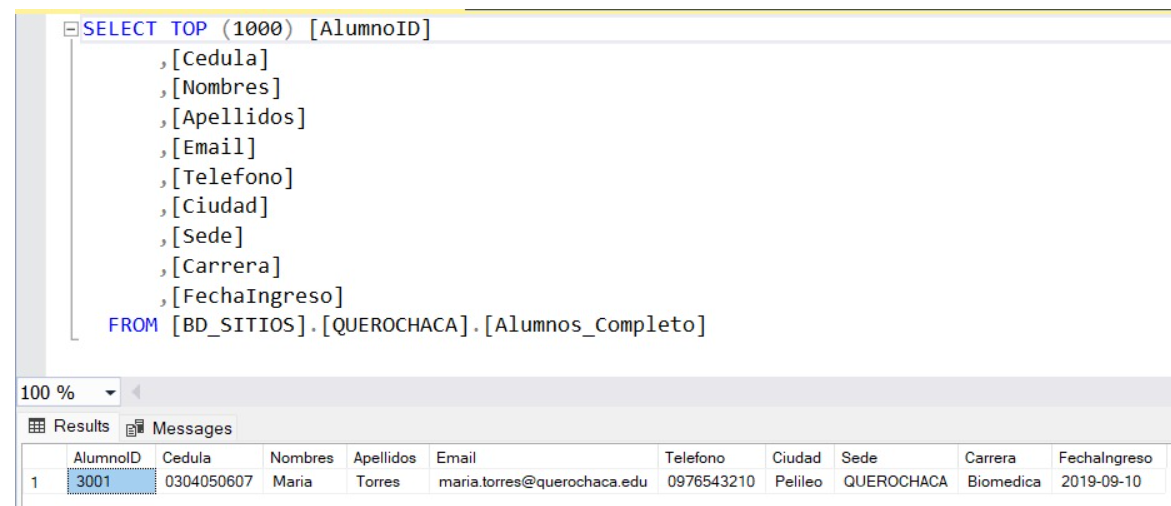
****

Ilustración 28 Verificación Vista QUEROCHACA

1. Verificación de Vista Global

****

Ilustración 29 Verificación de Vista Global

1. Verificación con Filtro por Sede

Ejemplo: Filtrar por la sede HUACHI:

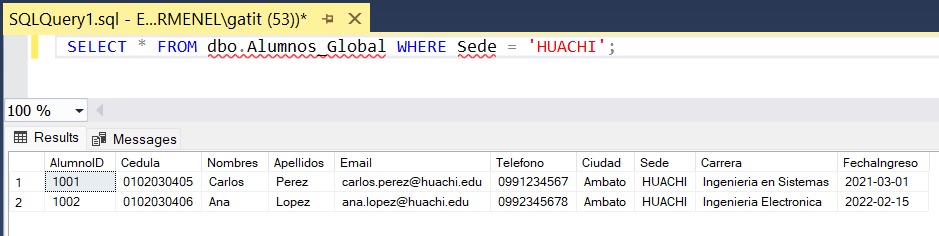


Ilustración 30 Ejemplo Filtro por sede HUACHI

1. Verificación con Filtro por Carrera

Ejemplo: Filtrar por la carrera Ingeniería en Sistemas:

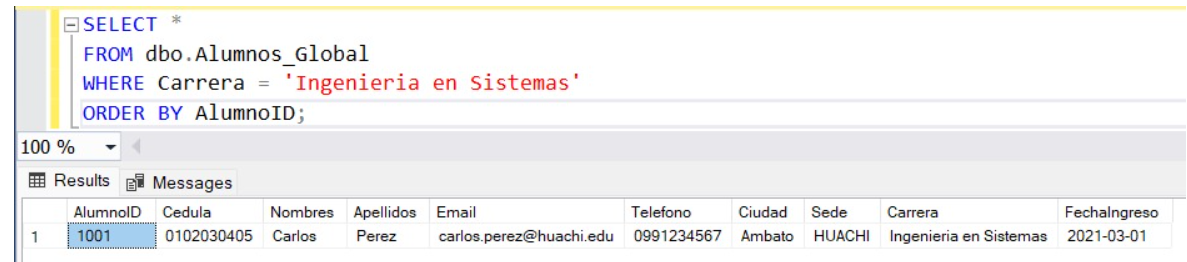
****

Ilustración 31 Ejemplo Filtro por carrera Ingeniería en Sistemas

## Resultados obtenidos

Se implementó exitosamente la fragmentación mixta con 3 esquemas (HUACHI, INGAHURCO, QUEROCHACA), cada uno con 2 tablas fragmentadas verticalmente (V1 y V2). La inclusión de AlumnoID en ambos fragmentos permitió la reconstrucción correcta mediante JOIN. Las vistas locales unieron los fragmentos verticales de cada sede, y la vista global combinó las tres sedes mediante UNION ALL. Las consultas sobre la vista global funcionaron correctamente, permitiendo acceder a todos los datos de forma transparente sin conocer la fragmentación subyacente.

Las pruebas de conteo, agrupación por sede y por carrera ejecutaron sin errores, demostrando que el diseño cumple con los principios de completitud, reconstrucción y transparencia de una base de datos distribuida.

## Habilidades blandas empleadas en la práctica

Liderazgo

Trabajo en equipo

Comunicación asertiva

La empatía

Pensamiento crítico

Flexibilidad

La resolución de conflictos

Adaptabilidad

Responsabilidad

## Conclusiones

* Se diseñó correctamente la fragmentación mixta por sede y tipo de datos, incluyendo AlumnoID en ambos fragmentos verticales para garantizar la reconstrucción completa de la información.
* Se implementaron exitosamente los 3 esquemas con sus tablas fragmentadas, constraints y vistas. La vista global permite acceder a todos los datos de forma transparente.
* Las consultas ejecutadas sobre la vista global funcionaron correctamente, confirmando que la reconstrucción mediante JOIN y UNION ALL opera sin pérdida de datos.

## Recomendaciones

* Mantener el atributo AlumnoID como clave común en todos los fragmentos verticales, ya que es fundamental para garantizar la integridad y reconstrucción de los datos.
* Documentar claramente la correspondencia entre fragmentos y vistas para facilitar mantenimiento y futuras ampliaciones del sistema.
* En entornos con múltiples servidores físicos, evaluar la replicación o federación de datos para mantener coherencia entre sedes y mejorar la eficiencia de consultas distribuidas.
* Usar UNION ALL en lugar de UNION para evitar costos innecesarios de eliminación de duplicados, siempre que se garantice que no haya solapamiento de datos entre sedes.