Universidad Técnica Nacional

Ingeniería en Tecnologías de la Información

Prof ing. Andrés Joseph Jiménez Leandro

Fabian David Bolaños Murillo

Mariangel Artavia Rojas

Practica integral de Bases de Datos

Documentación paso a paso

2025

**Paso a Paso del Script en la base de Datos**

**1. Creación de la Base de Datos**

Se crea la base de datos principal y se establece como contexto de trabajo:

CREATE DATABASE laboratorio;

GO

USE laboratorio;

GO

**2. Creación de Tabla Aerolíneas**

Se crea la tabla para almacenar información de las aerolíneas participantes:

CREATE TABLE aerolineas (

codaerolinea varchar(10) primary key not null,

nombre nvarchar(100) not null

);

Se insertan los datos iniciales de cuatro aerolíneas: LATAM Airlines, American Airlines, Iberia y Aerolíneas Argentinas.

**3. Inserción de Datos en Tabla Ciudades**

Se insertan registros en la tabla ciudades (previamente creada) con cinco destinos principales: Buenos Aires, Santiago, Miami, Madrid y Lima, incluyendo sus códigos de aeropuerto correspondientes.

**4. Creación de Tabla Aviones**

Se establece la tabla de aviones con relación foránea hacia aerolíneas:

CREATE TABLE aviones (

id\_avion int primary key,

modelo varchar(100) not null,

capacidad int not null,

codaerolinea varchar(10) not null,

foreign key (codaerolinea) references aerolineas(codaerolinea)

);

Se registran cinco aviones de diferentes modelos con sus respectivas capacidades y aerolíneas asignadas.

**5. Creación de Tabla Vuelos**

Se crea la tabla principal de vuelos con referencia a los aviones:

CREATE TABLE vuelos (

id\_vuelo int primary key,

num\_vuelo varchar(20) not null,

origen varchar(100) not null,

destino varchar(100) not null,

fecha date not null,

avion int not null,

foreign key (avion) references aviones(id\_avion)

);

Se insertan tres vuelos iniciales y se realiza una actualización de prueba en uno de los registros.

**6. Creación de Tabla Pasajeros**

Se establece la tabla de pasajeros con información personal y de contacto:

CREATE TABLE pasajeros (

id\_pasajero int primary key,

nombre varchar(100) not null,

apellido varchar(100) not null,

pasaporte varchar(50) not null unique,

email varchar(100),

telefono varchar(20)

);

Se insertan dos pasajeros de ejemplo con datos completos.

**7. Creación de Tabla Tarifas**

Se implementa la tabla de tarifas con relación a los vuelos:

CREATE TABLE tarifas (

id\_tarifa int primary key,

id\_vuelo int not null,

clase varchar(50) not null,

precio decimal(10,2) not null,

foreign key (id\_vuelo) references vuelos(id\_vuelo)

);

Se definen tres tarifas diferentes para clases turista y ejecutiva.

**8. Creación de Tabla Reservas**

Se establece la tabla de reservas relacionando vuelos y pasajeros:

CREATE TABLE reservas (

id\_reserva int primary key,

id\_vuelo int not null,

id\_pasajero int not null,

num\_asiento varchar(10) not null,

estado varchar(20) not null,

foreign key (id\_vuelo) references vuelos(id\_vuelo),

foreign key (id\_pasajero) references pasajeros(id\_pasajero)

);

Se crean tres reservas de prueba con diferentes estados y se realiza una actualización.

**9. Creación de Roles de Seguridad**

Se implementan cuatro roles principales para el control de acceso:

CREATE ROLE rol\_admin;

CREATE ROLE rol\_operador;

CREATE ROLE rol\_consulta;

CREATE ROLE rol\_seguridad;

**10. Creación de Usuarios y Logins**

Se crean los logins y usuarios correspondientes para cada rol:

CREATE LOGIN usr\_admin\_vuelos WITH PASSWORD = 'AdminAerolinea2025';

CREATE USER usr\_admin\_vuelos FOR LOGIN usr\_admin\_vuelos;

Se repite el proceso para los usuarios operador, consulta y seguridad.

**11. Asignación de Usuarios a Roles**

Se vinculan los usuarios creados con sus respectivos roles mediante ALTER ROLE ADD MEMBER.

**12. Asignación de Permisos Generales**

Se otorgan permisos específicos a cada rol: control total al administrador, permisos CRUD al operador, y permisos de solo lectura a los roles de consulta y seguridad.

**13. Permisos Granulares**

Se implementan restricciones específicas como limitar el acceso del rol consulta solo a ciertas columnas de la tabla pasajeros y restringir las actualizaciones del operador únicamente a la columna estado de reservas.

**14. Políticas de Seguridad a Nivel de Fila**

Se crea una función de filtro para vuelos futuros:

CREATE FUNCTION fn\_vuelos\_futuros(@id\_vuelo INT)

RETURNS TABLE

WITH SCHEMABINDING

AS

RETURN SELECT 1 AS acceso

WHERE EXISTS (

SELECT 1

FROM dbo.vuelos

WHERE id\_vuelo = @id\_vuelo

AND fecha >= CAST(GETDATE() AS DATE)

);

Se aplica una política de seguridad que filtra automáticamente solo vuelos futuros.

**15. Encriptación de Datos Sensibles**

Se implementa encriptación simétrica para proteger información sensible como números de pasaporte, utilizando certificados y claves simétricas con algoritmo AES\_256.

**16. Auditoría con Triggers**

Se crea una tabla de auditoría y un trigger para registrar automáticamente las inserciones en la tabla pasajeros:

CREATE TABLE auditoria\_pasajeros (

id\_auditoria INT IDENTITY PRIMARY KEY,

id\_pasajero INT,

accion VARCHAR(50),

fecha DATETIME,

usuario VARCHAR(100)

);

**17. Funciones de Negocio**

Se implementa una función para calcular la edad de pasajeros basada en fecha de nacimiento:

CREATE FUNCTION dbo.fn\_calcular\_edad (@fecha\_nac DATE)

RETURNS INT

**18. Creación de Vistas**

Se desarrolla una vista que combina información de aviones y aerolíneas:

CREATE OR ALTER VIEW vw\_AvionesAerolineas AS

SELECT

a.id\_avion,

a.modelo,

a.capacidad,

ae.nombre AS aerolinea

FROM aviones a

JOIN aerolineas ae ON a.codaerolinea = ae.codaerolinea;

**19. Funciones de Ventana**

Se implementan consultas avanzadas utilizando ROW\_NUMBER() y SUM() OVER para ranking de aviones por capacidad y cálculo de capacidades acumuladas por aerolínea.

**20. Consultas con Múltiples JOINs**

Se desarrollan consultas complejas que integran información de aerolíneas, aviones y vuelos mediante múltiples tipos de JOIN (INNER y LEFT).

**21. Plan de Respaldos**

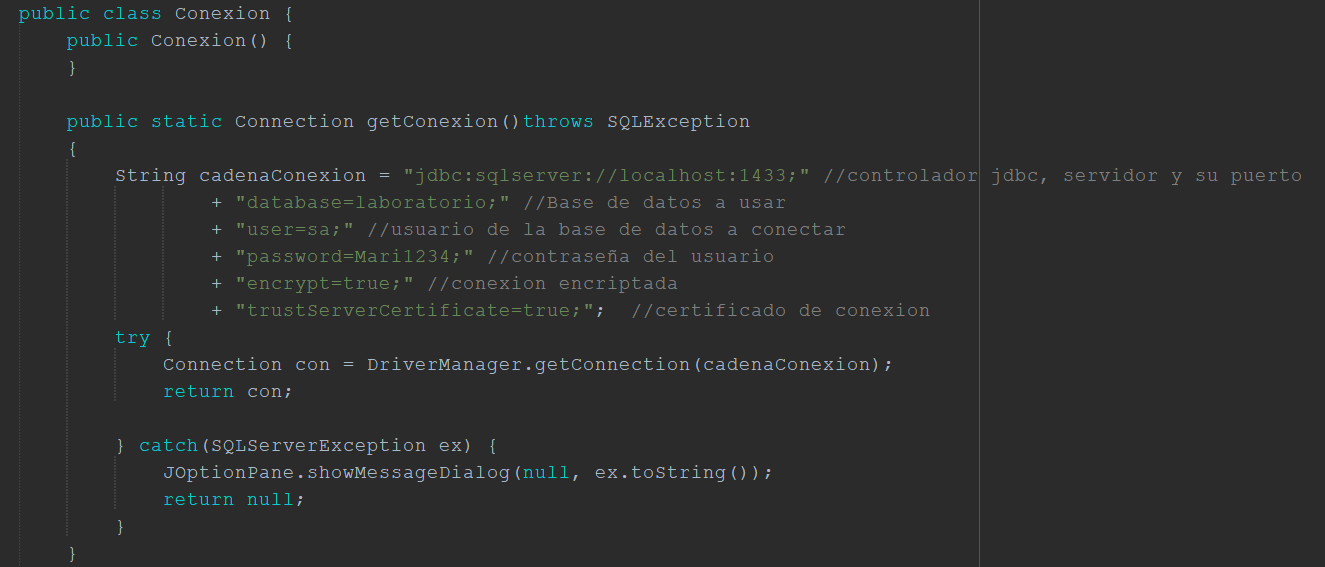
Se implementa una estrategia completa de backups que incluye:

* Backup completo de la base de datos
* Backup diferencial para cambios incrementales
* Backup del log de transacciones para recuperación punto en tiempo

Cada tipo de backup se configura con compresión para optimizar el espacio de almacenamiento.

**Pasos de la creación del proyecto en Java-Netbeans**

1. Se debe de establecer la conexión entre el motor de la base de datos, en este caso “Sql Server Express 2022” y java específicamente con “netbeans” en este caso



Nota: para poder establecer la conexión se deben de incorporar los jars a las librerías, de lo contrario no funcionará

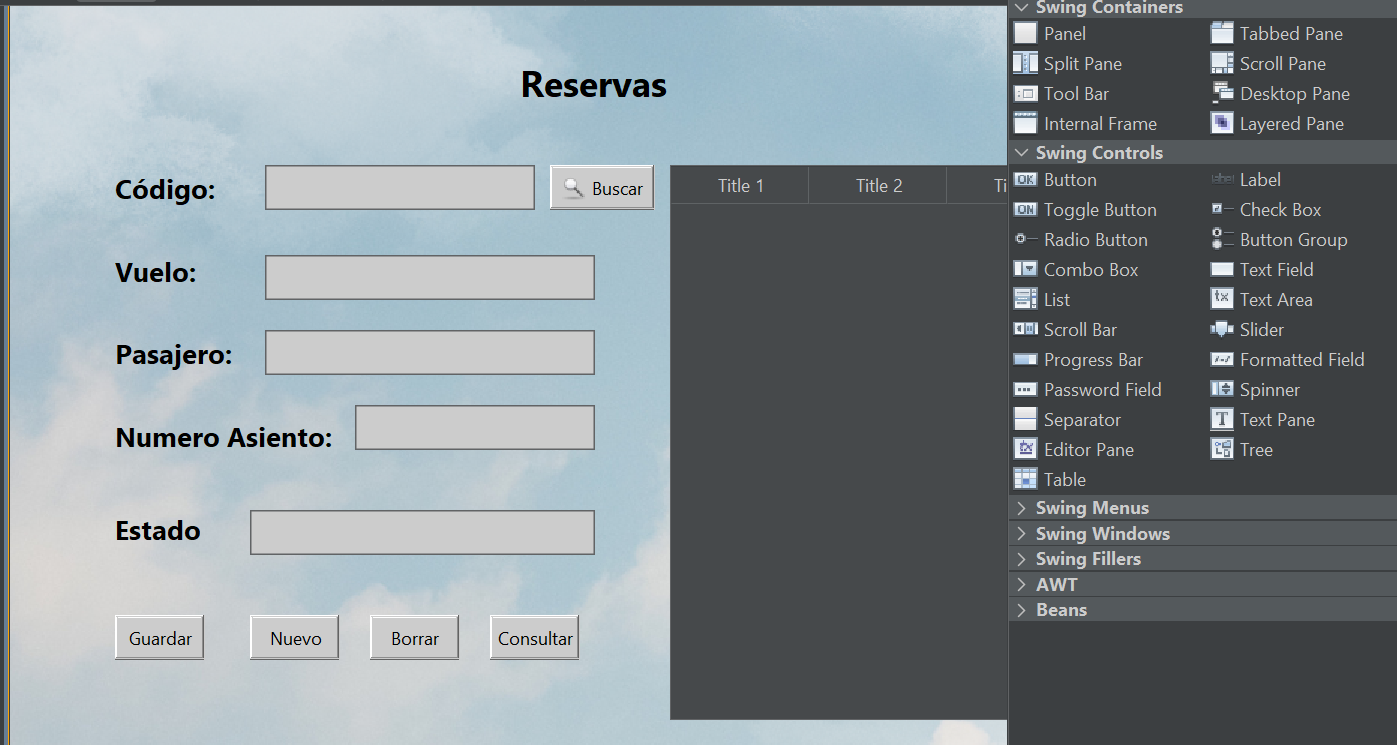
1. Creamos las clases de las tablas que queremos ver, en este laboratorio de utilizó la tabla de vuelos y reservas creadas anteriormente en la Base de datos
2. Se deben de incorporar los datos que vamos a utilizar (deben de ser igual a las tablas de la base de datos, con el mismo tipo de datos, sino enviará error) y se asignan los getters and setters de cada dato.



Texto

El contenido generado por IA puede ser incorrecto.

1. Luego se crean los Jframe o paneles para la parte gráfica, en este caso de utilizó un Jframe para la pantalla inicial y 2 paneles (uno para cada “tabla”) y se crea la parte gráfica, se añaden los elementos que vamos a necesitar



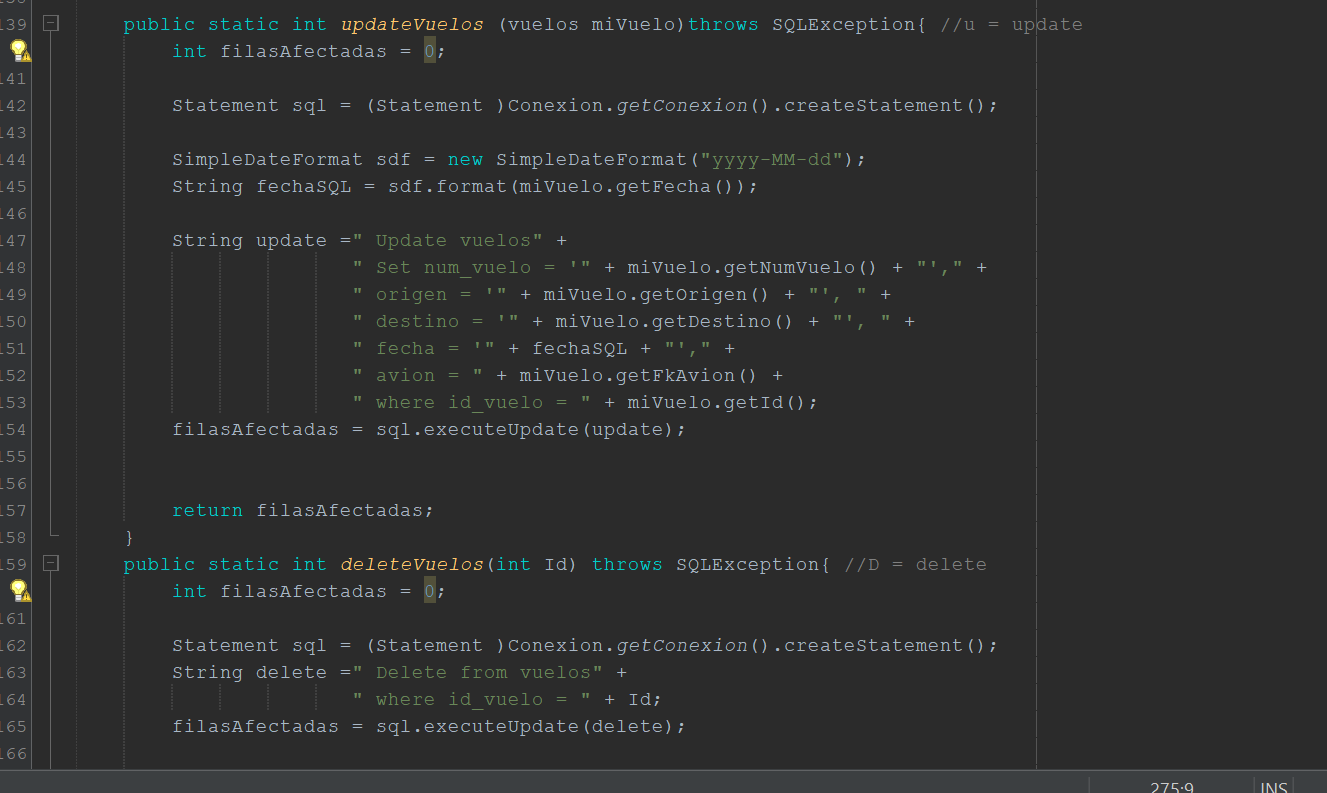
Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. En la clase de la conexión debemos crear varios métodos para poder llamar los datos de la base de datos y luego poder configurarlo con los botones de la interfaz.

Métodos creados:

* siguienteVuelo/ siguienteReserva: este funciona para traer el numero consecutivo de los vuelos y reservas, el identifica cual es el ultimo id en la tabla y trae el número que le sigue
* obtenerVuelos/ obtenerReservas: este método ayuda a cargar los datos de los vuelos y las reservas en un objeto
* listarVuelos/ listarReservas: Ayudará a mostrar la lista de los vuelos y reservas existentes en la base de datos
* insertVuelo/insertReserva: Sirve para guardar los datos ingresados en la base de datos
* updateVuelos/updateReservas: Ayuda a actualizar los datos existentes en caso de ser necesario
* deleteVuelos/deleteReservas: Sirve para eliminar algún dato ingresado en la base de datos



1. Luego se crearán los botones CRUD para cada tabla en la parte de la interfaz para que cada uno tenga su funcionalidad



Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

1. Por último, se llamará cada panel en el respectivo botón de la pantalla principal, en este caso hay 2 botones: Uno para el registro de vuelos y otro para el registro de Reservas

