

PLAN DE MIGRACIÓN

SISTEMA ORGANIZACIONAL EDUCATIVO

S.O.E



JUSTIN JARNO GARZÓN CARDENAS

JAIRO STYP RODRIGUEZ PATIÑO

JUAN CAMILO ROJAS ROJAS

JUAN DAVID MERCADO TORRES

FICHA:2251585

SERVICIO NACIONAL DE APRENDIZAJE SENA

CENTRO DE ELECTRICIDAD, ELECTRÓNICA Y TELECOMUNICACIONES

**PROGRAMA DE TECNOLOGIA EN ANALISIS Y DESARROLLO DE SISTEMAS DE
INFORMACIÓN**

BOGOTÁ D.C, SEPTIEMBRE 2021

Ficha de Control de Cambios

Proyecto	S.O.E
Documento	Plan de Migración
Versión	1.0
Fecha de Acción	22/09/2021
Fecha de Modificación	22/09/2021
Responsables	Justin Jarno Garzón Cardenas Jairo Styp Rodriguez Patiño Juan Camilo Rojas Rojas Juan David Mercado Torres

Historial

Fecha	Número de versión	Observaciones	Autor	Ver
22/09/2021	1.0		Justin Jarno Garzón Cardenas Jairo Styp Rodriguez Patiño Juan Camilo Rojas Rojas Juan David Mercado Torres	2.0

Tabla de Contenidos

INTRODUCCIÓN	1
JUSTIFICACIÓN	2
ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL.	3
Número de Registros filas:	3
Número de Atributos (NA):	3
Número de claves foráneas. Referential Degree. (RD):	5
Número de Objetos de conversión no trivial:	5
Otros aspectos a evaluar:	5
ANÁLISIS DE MIGRACIÓN	7
ESTRATEGIA DE MIGRACIÓN (PLAN DE MIGRACIÓN)	9
ELABORAR PLAN DE EJECUCIÓN	9
Plan marcha atrás	10
Mecanismo de escalado y manejo de problemas	10
Plan de implantación	10
PLAN DE EJECUCIÓN	11
Plan de formación	11
Plan de soporte	11
Plan de comunicaciones	11

INTRODUCCIÓN

Antes de empezar los pasos a seguir para realizar una migración de base de datos, es pertinente explicar que es la migración en una BD.

La migración de datos es el proceso mediante el cual realizamos una transferencia de datos de unos sistemas de almacenamiento de datos a otros, de unos formatos de datos a otros o entre diferentes sistemas informáticos. Los motivos habituales para utilizar la migración de bases de datos son: Actualización a la última versión del software de base de datos para mejorar la seguridad y el cumplimiento. Mover datos existentes a una nueva base de datos para reducir costos, mejorar el rendimiento y lograr escalabilidad.

Como todo, la migración no es un proceso 100% confiable, este proceso tiene ciertas desventajas que pueden ocurrir en cualquier momento, por lo cual hay que tener siempre un plan de respaldo para en cualquier momento volver atrás y no perder datos o cosas importantes. Algunas desventajas de la migración son:

1. Corrupción de datos: Cuando existen cambios en el formato, el contenido de los datos entre el sistema heredado y el sistema de destino pueden contener anomalías, redundancias y duplicidades.
2. Pérdida de datos: Tras la transferencia de información se pueden llegar a perder datos, no estando disponibles en el lugar de destino.
3. Riesgo semántico: Existe un riesgo al tener unidades de medidas distintas en la migración.
4. Riesgo de interferencia: Se produce cuando distintas partes interesadas están utilizando la aplicación de origen simultáneamente durante el proceso de transferencia.

El ecosistema de datos de una empresa comprende una variedad de aplicaciones. Con el tiempo, una empresa puede decidir migrar desde una base de datos existente para ahorrar costos, mejorar la confiabilidad, lograr escalabilidad o cualquier otro objetivo.

1. La migración de la base de datos es importante porque ayuda a ahorrar dinero.
2. El beneficio de la migración de la base de datos es que ayuda a mover los datos de un sistema heredado obsoleto a un software modernizado.
3. La migración de la base de datos ayuda a unificar datos dispares para que sean accesibles por diferentes sistemas.

JUSTIFICACIÓN

Se requiere realizar una migración desde el gestor de base de datos PostgreSQL al gestor Microsoft SQL Server, con diferentes propósitos de forma educativa entendiendo la importancia de la migración y los pasos que son necesarios para las buenas prácticas de este proceso, culminando el conocimiento en base de datos.

Es necesario tener claro el propósito de la migración, ya sea solo trasladar la estructura de la base de datos de un gestor a otro, o solo el traslado de los datos que se tienen guardados y se quieren almacenar en otro gestor de datos con el propósito de mejorar o actualizar el software, el rendimiento, la estabilidad, etc.

Hay que tener en cuenta diferentes aspectos los cuales cumplen un valor fundamental en una exitosa migración, los cuales son: El tiempo, el tamaño de la estructura de la base de datos, la compatibilidad, planeación, escalabilidad, experiencia entre otros.

Teniendo en cuenta todo lo anterior mencionado se procede con la planeación de la respectiva migración de gestor PostgreSQL a Microsoft SQL Server

ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL.

Para poder realizar una buena migración es necesario primero analizar ambos sistemas gestores de datos, el actual en el cual está almacenado los datos y se encuentra la estructura del mismo, que en nuestro caso es PostgreSQL y el sistema de gestor final en el cual se espera trasladar la información ya sea los datos o estructura, para ello se debe tener en cuenta diferentes tipos de elementos:

Número de Registros filas:

Se tienen almacenado tres registros en PostgreSQL los cuales se encuentran en la tabla type_document, los cuales son los diferentes tipos de identificación cédula, tarjeta de identidad y cédula extranjería (CC, TI, CE).

Número de Atributos (NA):

- La tabla “person”, sus 8 atributos son:
id, document_number, type_document_id, first_name, second_name, first_last_name, second_last_name, user_id.
- La tabla “rol” su único atributo es:
name_rol.
- La tabla “matter”, sus 3 atributos son:
id, name_matter, name_subject.
- La tabla “subject”, su único atributo es:
name_subject.
- La tabla “enrollment”, sus 9 atributos son:
id, student_id, eps_id, address, home_number, state, father_id, mother_id, tutor_id.

- La tabla “tutor”, sus 4 atributos son:
id, person_id, number_tutor, address_tutor.
- La tabla “annual_final_grade”, sus 3 atributos son:
id, final_quartely_grade, annual_final_grade.
- La tabla “user” sus 5 atributos son:
id, login, password, institutional_email, image_url.
- La tabla “rol_user” sus 3 atributos son:
id, name_rol, user_id.
- La tabla “type_document” sus 4 atributos son:
id, abbreviation, document_name, document_status.
- La tabla “student” sus 5 atributos son:
id, person_id, birth_date, birth_place, course_id.
- La tabla “cycle” su unico atributo es:
number_cycle.
- La tabla “coordinator” sus 3 atributos son:
id, person_id, number_cycle.
- La tabla “student_history” sus 2 atributos son:
id, student_id.
- La tabla “mother” sus 4 atributos son:
id, person_id, number_mother, address_mother.
- La tabla “father” sus 4 atributos son:
id, person_id, number_father, address_father.
- La tabla “current_year” su atributo es:
number_year.
- La tabla “teacher” sus 4 atributos son:
id, person_id, name_working_day, matter_id.
- La tabla “secretary” sus 3 atributos son:
id, person_id, name_working_day.
- La tabla “student_history_annotation” sus 2 atributos son:

id, student_historyid, annotationid.

- La tabla "timetable" sus 3 atributos son:
id, archive_timetable, course_id.
- La tabla "course" sus 5 atributos son:
id, number_course, number_year, number_cycle, teacher_id.
- La tabla "working_day" su unico atributos es:
name_working_day.
- La tabla "annotation" sus 3 atributos son:
id, notation, notation_date.
- La tabla "eps" sus 2 atributos son:
id, name_eps.
- La tabla "final_quaterly_grade" sus 8 atributos son:
id, student_id, matter_id, number_quarter, seventy_percentage,
twenty_percentage, ten_percentage, final_grade.
- La tabla "quarter" su 1 atributos es:
number_quarter.

Número de claves foráneas. Referential Degree. (RD):

El número de llaves primarias son: 26

El número de llaves foráneas son : 33

Número de Objetos de conversión no trivial:

Número de objeto de conversión no trivial: 2 (serial- int identity), (int4, int32 - int)

Otros aspectos a evaluar:

Hay una serie de factores comunes a cualquier base de datos que influyen en la complejidad de la migración y que hay que ponderar:

- ✓ Cantidad y tipo de SQL propietario que se use.
- ✓ Calidad de datos.
- ✓ Existencia de documentación del sistema.
- ✓ Requisitos de diseño tales como la alta disponibilidad y replicación.
- ✓ Software de terceras dependencias.
- ✓ Cambio en el sistema operativo y/o cambio de la plataforma de

- ✓ hardware como consecuencia de la migración.
- ✓ Cualificación y experiencia del personal involucrado en la migración.
- ✓ La disponibilidad de un equipo dedicado para el desarrollo de la migración.
- ✓ Imposibilidad de detener los cambios y los nuevos desarrollos sobre el código a migrar.
- ✓ Tiempo máximo que se permite para la realización de la migración

ANÁLISIS DE MIGRACIÓN

En este apartado se muestra los diferentes tipos de datos que existe en cada uno de los gestores de base datos y se puede visualizar las diferencia entre ellos.

Tipo de datos de PostgreSQL	Tipo de datos de Microsoft SQL Server
BIGINT	bigint
BIGSERIAL	binario
BIT	bit
BIT VARYING	char
BOOLEAN	fecha
BOX	datetime
BYTEA	datetime2
CHARACTER	datetimeoffset
CHARACTER VARYING	decimal
CIDR	flotante
CIRCLE	geografía
DATE	geometría
DOUBLE PRECISION	imagen
INET	int
INTEGER	dinero
INTERVALO	nchar
JSON	ntext
JSONB	numérico
LINE	nvarchar
LSEG	real
MACADDR	smalldatetime
MONEY	smallint
NUMERIC	smallmoney
PATH	texto
PG_LSN	tiempo
POINT	marca de hora

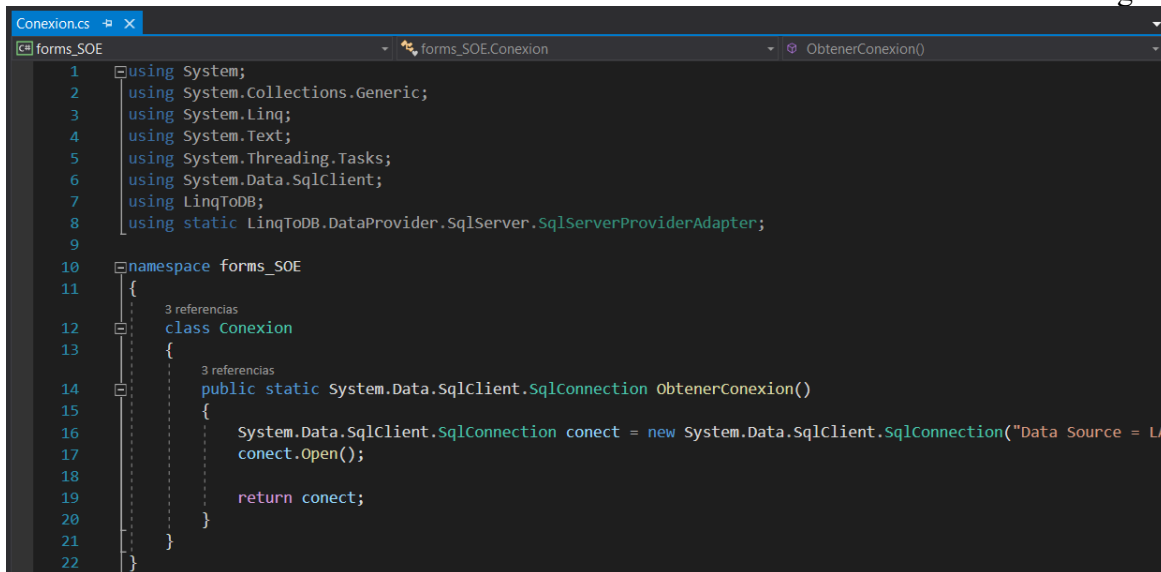
POLÍGONO
REAL
SMALLINT
SMALLSERIAL
SERIAL
TEXTO
TIME (sin zona horaria)
TIME (con zona horaria)
TIMESTAMP (sin zona horaria)
TIMESTAMP (con zona horaria)
TSQUERY
TSVECTOR
TXID_SNAPSHOT
UUID
XML

tinyint
udt
uniqueidentifier
varbinary
varchar
xml

ESTRATEGIA DE MIGRACIÓN (PLAN DE MIGRACIÓN)

ELABORAR PLAN DE EJECUCIÓN

Para poder realizar la conexión al gestor Microsoft SQL Server es necesario realizar una clase en Visual Studio la cual se le denominó el nombre Conexion.cs con el fin de conectarse mediante la variable “connect” como se muestra en el código.

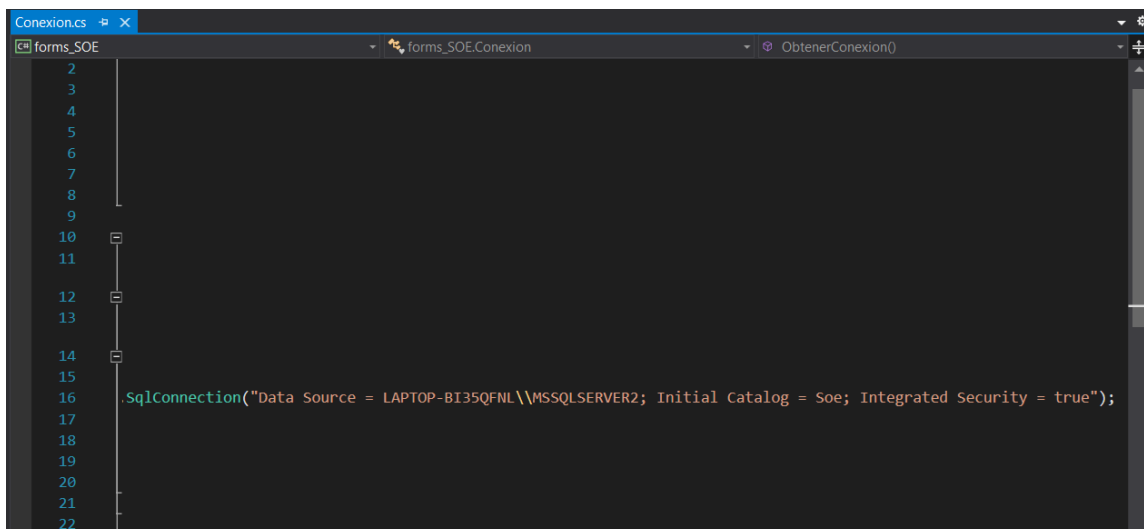


```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Data.SqlClient;
7  using LinqToDB;
8  using static LinqToDB.DataProvider.SqlServer.SqlServerProviderAdapter;
9
10 namespace forms_SOE
11 {
12     class Conexion
13     {
14         public static System.Data.SqlClient.SqlConnection ObtenerConexion()
15         {
16             System.Data.SqlClient.SqlConnection connect = new System.Data.SqlClient.SqlConnection("Data Source = L/
17             connect.Open();
18
19             return connect;
20         }
21     }
22 }

```

Después se digita el nombre de la máquina en la cual se encuentra la base de datos y por último agrega el nombre que se le asignó a la Data Base de Microsoft SQL Server.



```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 SqlConnection("Data Source = LAPTOP-BI35QFNL\\MSSQLSERVER2; Initial Catalog = Soe; Integrated Security = true");
17
18
19
20
21
22

```

Plan marcha atrás

Para realizar el plan marcha atrás es importante dejar la base de datos en el ambiente donde fue creado en este caso PostgreSQL y no realizar ninguna modificación o eliminación de ninguna tabla o dato registrado.

En caso tal de alguna modificación de la estructura de la base de datos se tiene las secuencias DML para no perder ninguna tabla, conexión o llaves primarias, esto para tener un respaldo y que la estructura no corra ningún riesgo de ser eliminada o modificada.

En caso de la modificación sea en los datos registrados en la base de datos se cuenta con las sentencias DDL las cuales tienen como objetivo al ser ejecutada volver a llenar los datos que estaban almacenados anteriormente.

Mecanismo de escalado y manejo de problemas

Para estas situaciones se tiene planteado por el grupo de trabajo, tener todas las sentencias de código en diferentes hardwares y aparte de ello tener todo estos tipos de documentos en diferentes repositorios de forma local y de forma online como podría ser Git, Github, además si es necesario subir todo tipo de archivos a la nube como lo serían OneDrive o Google Drive.

Plan de implantación

Lo siguiente será dar a entender el paso a paso que se debe seguir en el caso de una migración a la base de datos del gestor de PostgreSQL a Microsoft SQL Server.

1. Guardar la base de datos en PostgreSQL con los datos, sin ningún tipo de error.
2. Realizar una copia de seguridad de los documentos, sentencias y datos.
3. Comparar los tipos de datos de los gestores de base de datos y analizarlos.
4. Reemplazar los tipos de datos para que sean aceptados por el gestor en el cual que quiere migrar.
5. Verificar y modificar los datos en los que se presente algún tipo de error.
6. Realizar pruebas de datos para así determinar si la migración fue exitosa.

PLAN DE EJECUCIÓN

Se identifican las diversas actividades que se realizarán para la migración y se aseguran las diversas funcionalidades que serán necesarias para que la migración se construya de una manera correcta para su ejecución exitosa, estas actividades son plan de formación, plan de soporte y el plan de comunicados.

Plan de formación

Para empezar el plan de formación primeramente se tendrá que informar al sector técnico y soporte, también a los respectivos desarrolladores sobre las actividades que se realizaron en la migración. A los usuario no será necesario informarles del cambio debido a que no notarán ningún tipo de cambio en el aplicativo o página web

Plan de soporte

En el plan de soporte en caso tal de que un usuario detecte alguna incidencia en la página web y sea comunicada al sector de soporte, ellos verificarán este cambio y realizarán el debido seguimiento a este caso, realizando un informe al sector de desarrollo para que se realice el debido proceso de este problema dándole solución. Por último el sector de soporte será informado de las correcciones para brindarle la comunicación al usuario para un satisfactorio funcionamiento del aplicativo web.

Plan de comunicaciones

Mediante la interacción del usuario con el aplicativo web, el usuario podrá realizar una breve explicación sobre las incidencias que evidencia el usuario en el aplicativo, la incidencia será registrada y mediante un un informe el cual tendrá la descripción detallada del incidente que el usuario afectó en el transcurso de su interacción. Con el soporte se desarrollará los debidos cambios para un funcionamiento oportuno y óptimo en las diversas actividades que se cumplirán en el aplicativo web.