



Plan de migración de datos
Danca Store

Camilo Andrés Galindo Rivera
William Alejandro Bermudez Quiroga
Santiago Ninco Uribe
Davived Rayo Silva

2021

| FICHA DE CONTROL DE CAMBIOS | | | |
|-----------------------------|-------------------|---|-----|
| PROYECTO | | Danca Store | |
| DOCUMENTO | | Plan de migración | |
| VERSIÓN | | 1.0 | |
| FECHA CREACIÓN | | 23 septiembre 2021 | |
| FECHA CAMBIO | | | |
| RESPONSABLES | | Camilo Andrés Galindo Rivera William Alejandro Bermúdez Quiroga Ángel Stiven Toro Fuentes Santiago Ninco Uribe Davived Rayo Silva | |
| LÍDER | | Camilo Andrés Galindo Rivera | |
| HISTORIAL | | | |
| FECHA | NÚMERO DE VERSIÓN | AUTOR(ES) | VER |
| 23 septiembre 2021 | 1.0 | Ángel Stiven Toro Fuentes | 2.0 |

Contenido

| | |
|---|---|
| Introducción..... | 3 |
| Justificación..... | 3 |
| Análisis previos del sistema actual y final..... | 4 |
| Análisis de migración | 6 |
| Estrategia de migración (Plan de migración) | 7 |
| Plan de ejecución y soporte | 8 |

Introducción

La migración de base de datos es un proceso que se lleva a cabo para trasladar datos almacenados previamente de un gestor de base de datos a otro o desde el mismo gestor, pero a versiones diferentes.

Para la Danca Store los beneficios que puede aportar con la migración de la base de datos son:

- Seguridad de datos a largo plazo.
- Evitar suspender la aplicación.
- Modernización en la estructura.
- Aumento de la productividad en el análisis y desarrollo de las aplicaciones.

Determinar si se va a migrar la estructura de la base de datos, o solo los datos, o ambas cosas varía de acuerdo a la necesidad y objetivo de la migración en la entidad. No dejar de lado el tamaño de la base de datos y la complejidad de la base de datos, la forma en la que se va a controlar que en realidad si se migraron todos los objetos y datos de una base de datos, las restricciones, y como tal que se esté pasando toda la lógica de negocio.

Justificación

La migración se realiza de PostgreSQL a SQL Server con fines educativos, una migración de datos es un proceso para trasladar datos de un gestor de base de datos a otro, en el proceso se genera un análisis y una evaluación de datos, es un proceso que trae buenas ventajas para el mejoramiento y sostenimiento de proyecto.

Para la migración de datos se tienen que tener algunos aspectos y factores en mente, los cuales son: Tiempos, costo, estabilidad, tamaño de base de datos, estructura, restricciones, complejidad, soporte técnico, experiencia, planeación, control y seguimiento,



compatibilidad, entre muchos otros, a continuación, se mostrara la migración de base de datos de Danca Store.

Análisis previos del sistema actual y final

Antes de realizar la migración, se debe hacer un análisis previo del gestor de base de datos y la base de datos, por otro lado, se analiza también el nuevo gestor de base de datos, esto se hace para tener una visión de los 2 gestores y obtener ventajas y desventajas sobre la migración, aparte se debe estimar la complejidad de los siguientes términos.

Número de Registro filas:

La cantidad de registros (Filas) en la base de datos de PostgreSQL es actualmente de 72

Número de atributos (NA):

En la base de datos de Danca Store se conforma por 12 tablas las cuales son:

Número: #1

Nombre: user

Atributos: id, login, password, email, phone_number, address, image_url, status.

Número: #2

Nombre: payment

Atributos: id, id_client, id_method_payment, Bill_date, status.

Número: #3

Nombre: method_payment

Atributos: id, method_payment.

Número: #4

Nombre: shipping

Atributos: id, id_order, shipping_company_name, address, shipping_date, delivery_date, status.

Número: #5

Nombre: client

Atributos: id, id_document_type, document_number, name, last_name, id_user.

Número: #6

Nombre: order

Atributos: id, id_order_details, id_payment, id_client, order_date, status.

Número: #7

Nombre: products

Atributos: id, reference, name, Price, picture, stock, id_category, id_description, id_supplier, status.

Número: #8



Nombre: document_type

Atributos: id, initials, document_name, status.

Número: #9

Nombre: order_details

Atributos: id, id_client, id_product, total_value, amount, color.

Número: #10

Nombre: supplier

Atributos: id, company_name, phone_number, address, email, postal_code, status.

Número: #11

Nombre: description

Atributos: id, code_description, size, specs, information, condition, color.

Número: #12

Nombre: producto_category

Atributos: id, category_name, status.

Número de claves foráneas:

El número de llaves primarias es 12.

El número de llaves foráneas es 13.

Otros aspectos a evaluar:

Algunos aspectos comunes que se pueden evaluar para la migración:

- Calidad de los datos.
- Requisitos de diseño tales como la alta disponibilidad y replicación.
- El cambio en el sistema operativo y/o cambio de la plataforma de hardware como consecuencia de migración.
- Cualificación y experiencia del personal involucrado en la migración.
- La disponibilidad de un equipo dedicado para el desarrollo de la migración.
- Tiempo máximo que se permite para la realización de la migración.

Análisis de migración

En este punto deben identificarse los cambios específicos que hay que realizar para transformar cada elemento de la base de datos PostgreSQL en un elemento de MSSQL que funcione de la misma manera.

Identificando tipos de datos, funciones, triggers, elementos DML, procedimientos, etc., no soportados en SQL Server.

Por otro lado, se va a mencionar una comparativa entre PostgreSQL y MSSQL:

| | PostgreSQL | SQL Server |
|--------------------------------------|---|---|
| Modelo de Base de datos | DBMS relacional | DBMS relacional |
| Desarrollador | PostgreSQL Global Development Group | Microsoft |
| Lanzamiento | 1996 | 1989 |
| Versión más reciente | 12.1 | SQL Server 2019 |
| Licencia | Open Source | Commercial |
| Lenguaje de implementación | C | C++ |
| Plataformas | FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows | Linux, Windows |
| APIs y otros métodos de acceso | ADO.NET, JDBC, Librería de C nativa, ODBC, streaming API | ADO.NET, JDBC, ODBC, OLE DB, TDS |
| Lenguajes de programación soportados | .Net, C, C++, Delphi, Java, JavaScript (Node.js), Perl, PHP, Python, Tcl, R | C#, C++, Delphi, Go, Java, JavaScript (Node.js), PHP, Python, R, Ruby, Visual Basic |
| Lenguajes de Procedimiento | PGSQL, V8, Python, R, Perl, Java, Lua, sh, Tcl, Ruby, PHP | Transact SQL, .NET, R, Python, Java |
| Métodos de Particionamiento | Partición por Lista, Rango, Hash | Partición Horizontal |
| Replicación | Maestro-Esclavo | Transaccional, Merge, Snapshot, Peer-to-peer, Bidireccional |
| Transacciones | ACID | ACID |
| In-memory | No | Sí |



Estrategia de migración (Plan de migración) Elaborar plan de ejecución

Hay que identificar los cambios a realizar en el código de la propia aplicación para que funcione sobre una base de datos MSSQL.

Conexión a base de datos desde PostgreSQL en un archivo JavaScript llamado database:

```
1  const { Pool } = require('pg');
2
3  const db = new Pool({
4    host: 'localhost',
5    user: 'postgres',
6    password: process.env.DB_PASS,
7    database: process.env.DB_NAME,
8    port: process.env.DB_PORT,
9  });
10
11 module.exports = { db };
12
```

Conexión a base de datos desde MSSQL en donde nuestra base de datos se llama danca_store, el puerto es 1433 y el usuario en este caso es dancastoreadmin

```
1  const sql = require("mssql");
2
3  const dbSettings = {
4    user: process.env.DB_USER,
5    password: process.env.DB_PASS,
6    server: process.env.DB_SERVER,
7    database: process.env.DB_NAME,
8    options: {
9      encrypt: true,
10     trustServerCertificate: true,
11   },
12 };
13
14 const getConnection = async () => {
15   try {
16     const pool = await sql.connect(dbSettings);
17     return pool;
18   } catch (error) {
19     console.error(error);
20   }
21 };
22
23 module.exports = { dbSettings, getConnection };

```



Plan de marcha atrás

Si en el proceso de migración llega a suceder algo imprevisto o por si el cliente decide no realizar la migración, es importante realizar una copia de seguridad antes de empezar a desarrollar el plan de tramitación, esto es por si pasa cualquier evento que no esté contemplado, y así dar una solución rápida y óptima. Es importante tratar de no modificar el ambiente en el que está instalada la base de datos PostgreSQL, para el caso de que se decida no realizar la migración solo sea iniciar el servicio de base de datos PostgreSQL.

Tener en cuenta que la documentación es un tema fundamental en este punto tanto de lo que había como de lo que existe actualmente.

Plan de ejecución

Como se ha mencionado en este documento es fundamental realizar un estudio de funcionamiento de bases de datos en PostgreSQL y MSSQL, para establecer que transformaciones se deben realizar.

A continuación, una vez identificados los trabajos necesarios para realizar la migración y asegurar la funcionalidad del sistema, se identifican las necesidades que generará la migración y se planifican las distintas actividades. En ese momento comenzará la construcción de los planes que se ejecutarán durante el resto del proceso de migración.

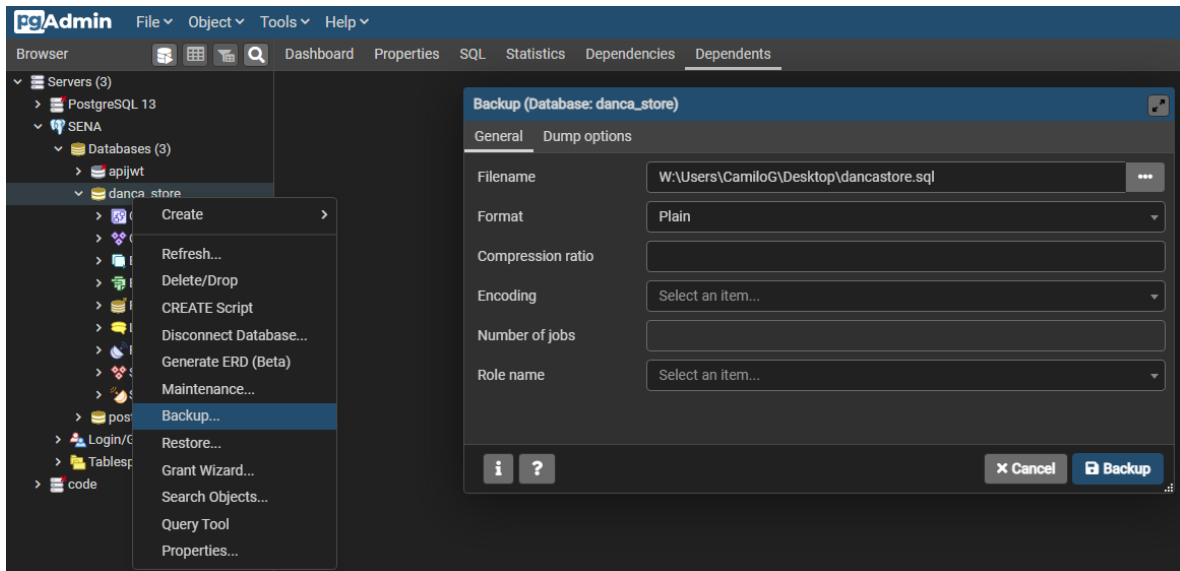
Plan de formación

Las actividades que permiten la formación de personas involucradas o afectadas por la migración. En general los usuarios de la aplicación de novedades no deberán notar ningún cambio en este tipo de migraciones, pero sí debería formarse a los equipos de soporte y mantenimiento y a los desarrolladores de la organización, para que conozcan los cambios que se han producido y cómo enfrentarse a ellos en sus tareas durante y tras la migración.

Plan de ejecución y soporte

Paso 1. Generar un respaldo de la estructura de la base de datos PostgreSQL

El primer paso para migrar nuestra base de datos PostgreSQL a MSSQL es generar una copia de seguridad en PgAdmin de la siguiente forma:



La aplicación nos producirá un archivo sql con el código para la creación de una nueva base de datos:

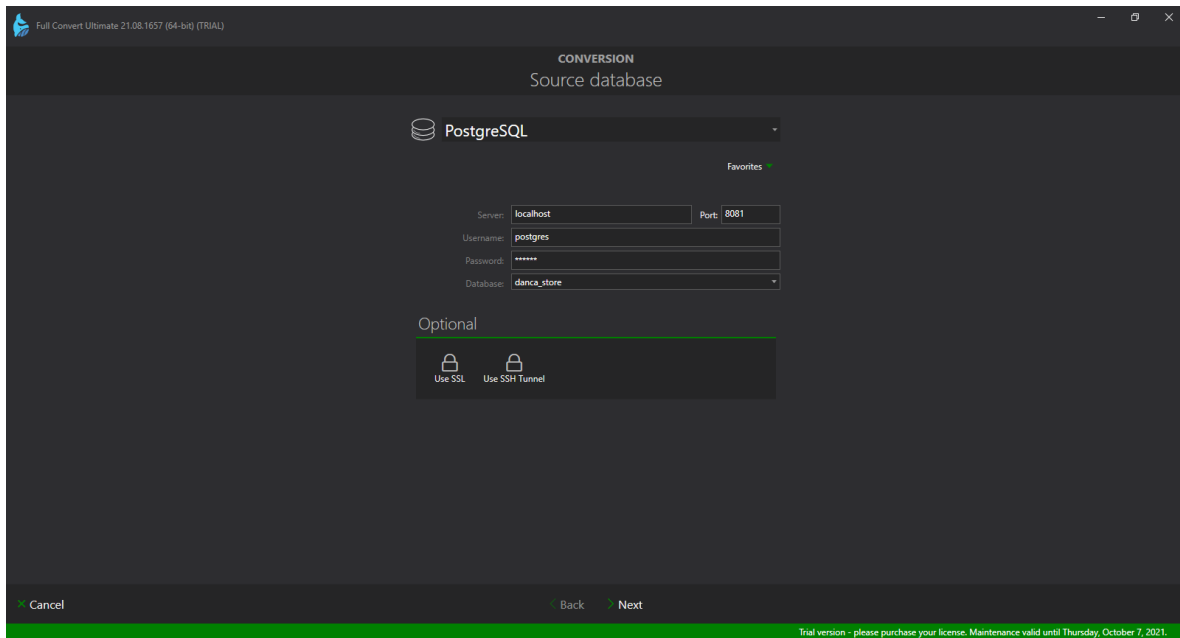
```

1 CREATE TABLE rol (
2     name VARCHAR(40),
3     CONSTRAINT pk_rol PRIMARY KEY (name)
4 );
5 CREATE TABLE user_ (
6     id serial NOT NULL,
7     login VARCHAR(90) NOT NULL,
8     password VARCHAR(90) NOT NULL,
9     email VARCHAR(90) NOT NULL,
10    phone_number BIGINT NOT NULL,
11    town VARCHAR(100) NOT NULL,
12    address VARCHAR(90) NOT NULL,
13    image_url VARCHAR,
14    status VARCHAR(100) NOT NULL,
15    CONSTRAINT pk_user PRIMARY KEY (id),
16    CONSTRAINT uk_user UNIQUE (login, email)
17 );

```

(Parte del código)

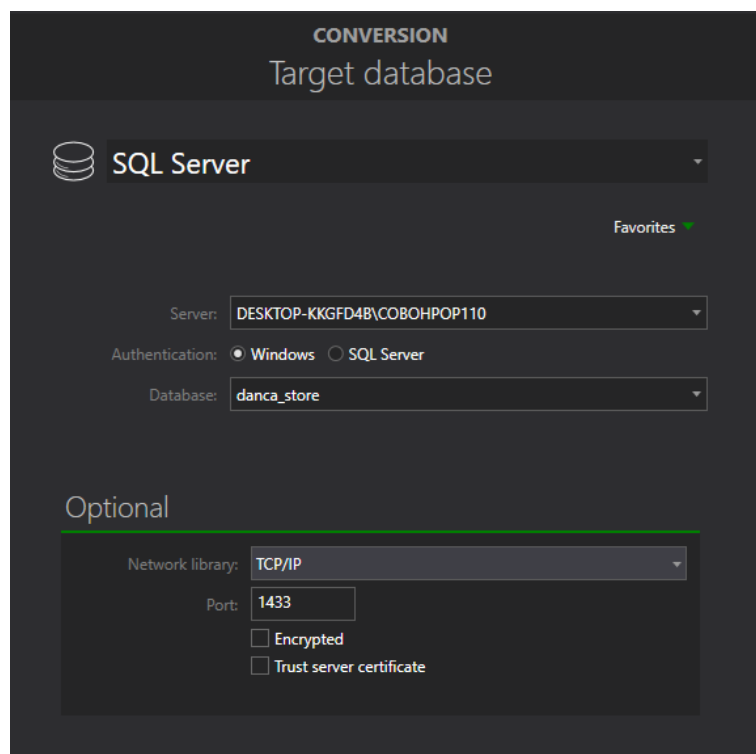
Paso 2.



Ingresar los datos de nuestra base de datos.

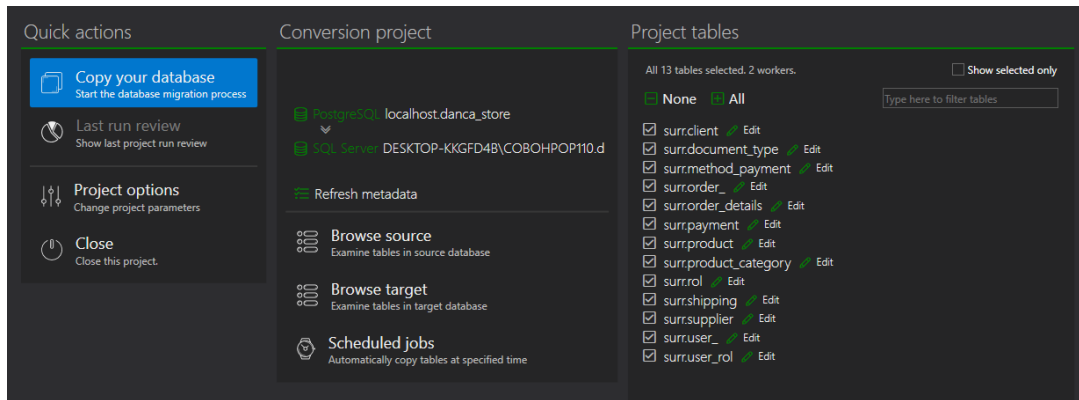
Para descargar y/o comprar este software: [Overview](#) | [Full Convert](#)

Paso 3.



Elegimos el destino de nuestra base datos SQL Server

Paso 4.



Daremos click en Copy your database y la aplicación empezara a hacer la migración automáticamente a SQL Server

Finalmente, el proceso se ha completado y tenemos todas nuestras tablas e información en SQL Server.

