



SERVICIO NACIONAL DE APRENDIZAJE SENA

Análisis y desarrollo de sistemas de información Plan de migración



Plan de migración				
PROYECTO		SENANOV		
DOCUMENTO		Pruebas de Humo		
VERSIÓN		V 3		
FECHA CREACIÓN		01 de junio del 2018		
FECHA CAMBIO		01 de junio del 2018		
RESPONSABLES		Mónica Díaz Juan David Ramírez Felipe Pinzón		
LÍDER		Cristian Lorenzo Ortiz		
HISTORIAL				
FECHA	NÚMERO DE VERSIÓN	OBSERVACIONES	AUTOR (ES)	VER
01/06/2018	V 3	Realización de la documentación	Felipe Pinzón	V 3



Contenido

1. Introducción

2. Justificación

3. Análisis previos del sistema actual y final

Número de registros filas

Número de atributos

Número de claves foráneas

Número de objetos de conversación no trivial

Otros aspectos a evaluar

4. Análisis de migración

5. Estrategia de migración

6. Plan de ejecución

Plan marcha atrás

Mecanismo de escalado y manejo de problemas

Plan de implantación

7. Plan de ejecución

Plan de formación

Plan de ejecución

Plan de soporte

Plan de comunicaciones



INTRODUCCIÓN

Antes de empezar a explicar los pasos que se deben tener en cuenta para realizar una migración, es importante revisar que es migrar.

La migración de BD 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 Subred Sur los beneficios que puede aportar la migración de la base de datos son Reducción del costo, en millones, en el consumo de la información, Aumento en la seguridad, 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 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.

El grado de dificultad se mide de acuerdo a algunos de los factores mencionados con anterioridad. Quizá no sea necesario cambiar de gestor de base de datos, por tal razón es muy importante analizar si se debe o no efectuar la migración, de sus ventajas y desventajas.



JUSTIFICACIÓN

Se requiere realizar una migración desde el gestor de base de datos de la subred sur de MySQL a POSTGRESQL, este ejercicio es de tipo educativo, pero es fundamental que se realice, para evaluar, analizar y entender la importancia de una migración, y de todos los aspectos relacionados con la misma.

Hoy en día, las empresas son muy variadas, y es por tal razón que en el mercado existen varios sistemas gestores de bases de datos, para pequeñas, medianas, y grandes empresas. Cada uno de ellos presenta sus desventajas y ventajas. Sin importar si se tienen datos registrados, por ejemplo en MySQL y se quieren pasar a postgresql, en este caso lo fundamental es la información, los datos contenidos en dicho gestor.

Tener en cuenta que el proceso de migración no se realiza por arte de magia, y que aunque existe software para realizar migraciones, no hay un sistema de información que sea 100% confiable.

Hay muchos factores que intervienen en un proceso de migración: tiempo, costo, escalabilidad, tamaño de la base de datos, estructura, restricciones, complejidad, soporte técnico, experiencia, planeación, control y seguimiento, compatibilidad, etc. A continuación se va a realizar un plan de migración de base de datos MySQL a POSTGRESQL. Se espera que este documento sea de gran utilidad y sirva como base para posibles migraciones en el sistema de información de la Subred Sur para la gestión de las peticiones de atención al ciudadano.



ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL.

Antes de realizar la migración, se debe realizar un análisis previo del sistema gestor de base de datos de MySql que es el actual y del sistema gestor de base de datos final Postgresql. Esto se realiza para tener una visión general de las ventajas y/o desventajas de realizar dicha migración, además se estima el nivel de complejidad de llevar a cabo dicho proceso. Estos elementos son:

Número de Registros filas:

La cantidad de registros en una base de datos es importante en este análisis previo, puesto que no es lo mismo realizar una migración de cien registros a más de mil millones de registros. Tener en cuenta que el número de registro de filas afecta tiempos de actualización, Back up, disparadores, etc.

Número de Atributos (NA)

Otro de los indicadores de complejidad se calcula en base a la cantidad de campos de las tablas. Se recomienda ejecutar consultas que permitan ver el promedio en general de las columnas, la cantidad máxima y mínima de columnas por tabla. No es lo mismo tener una tabla con 200 campos a una tabla con 2 campos, el manejo en cuanto a índices, restricciones, es diferente.

Número de claves foráneas. Referential Degree. (RD)

Es importante conocer la cantidad de llaves foráneas, primarias, índices, constraints, existes en la base de datos. Tener en cuenta que el manejo de las claves anteriores se relaciona con las tablas y la funcionalidad que estas desempeñan. Pueda que haya llaves innecesarias o por lo contrario que hagan falta índices o claves para mejor tiempos de consulta y actualización.

Número de Objetos de conversión no trivial.

En la base de datos MySql hay objetos que no tienen correspondencia directa con respecto a la base de datos Postgresql por tanto dicho proceso se vuelve un poco complejo. Es por eso que están importante conocer que tan complejo es dicha conversión y así tomar medidas para prever el impacto de la migración.

Los objetos de la base de datos cuya conversión a lenguaje Postgresql, de nivel de complejidad bajo son: tablas, índices, secuencias, vistas, sinónimos.



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 terceros 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 (ventana de corte).

ANALISIS DE MIGRACION

En este punto deben identificarse los cambios específicos que hay que realizar para transformar cada elemento de la base de datos MySQL en un elemento de PostgreSQL que funcione de la misma manera. También hay que identificar los cambios a realizar en el código de la propia aplicación para que funcione sobre una base de datos PostgreSQL.

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

Por otro lado, se va a mencionar algunos tipos de datos MySQL y su equivalente en Postgresql:

TIPOS DE DATOS	
MySQL	PostgreSQL
bigint	bigint
double, decimal, float	numeric
int, mediumint	integer
smallint, tinyint	smallint
char	character
varchar, enum, set	character varying
longtext, mediumtext, text, tinytext	text
tinyblob, blob, mediumblob, longblob	bytea
datetime, timestamp	timestamp
time, date, year	date

Ilustración 1 Tipos de datos.



ESTRATEGIA DE MIGRACIÓN (PLAN DE MIGRACIÓN).

ELABORAR PLAN DE EJECUCION

El plan de Ejecución contiene todas las tareas necesarias para pasar del sistema origen al sistema destino. Entre esas tareas está la implementación de todas las transformaciones que se hayan identificado, tanto de base de datos como de código, pero también la elaboración de todos los planes asociados a la ejecución de la migración, como pueden ser:

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 un back up 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 MySQL, para en caso de que se diga que no se realice la migración solo sea iniciar el servicio de base de datos MySQL.

Tener en cuenta que la documentación es un tema fundamental en este punto tanto de lo que había como de lo que existe actualmente. Es decir, configuraciones, usuarios, etc.

Mecanismo de escalado y manejo de problemas.

Son las posibles contingencias ante problemas externos a la migración, como podrían ser cortes de luz, etc. Estas contingencias incluyen personal de contacto, listas de correo, posibilidad de reuniones de urgencia, etc.

Plan de Implantación.

Contiene toda la información de los pasos a seguir para la realización del cambio de un SGBD al otro.

- ✓ Desglose de Tareas de cada paso (implementadores, probadores,...)
- ✓ Duración prevista de cada paso (Margen máximo de desviación)
- ✓ Flujo de los pasos (dependencias entre tareas, posibilidad de paralización de las mismas)



- ✓ Eventos o responsable que toma la decisión de éxito de la migración o marcha atrás de la misma.

PLAN DE EJECUCION

Como se ha mencionado en este documento es fundamental realizar un estudio de funcionamiento de bases de datos en MySQL y Postgresql, 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:

Definirá las actividades que permitirán la formación de las personas involucradas o afectadas por la migración. En general los usuarios de la aplicación no deberían 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:

Incluirá todas las acciones necesarias para poner en funcionamiento el sistema con la "nueva" base de datos.

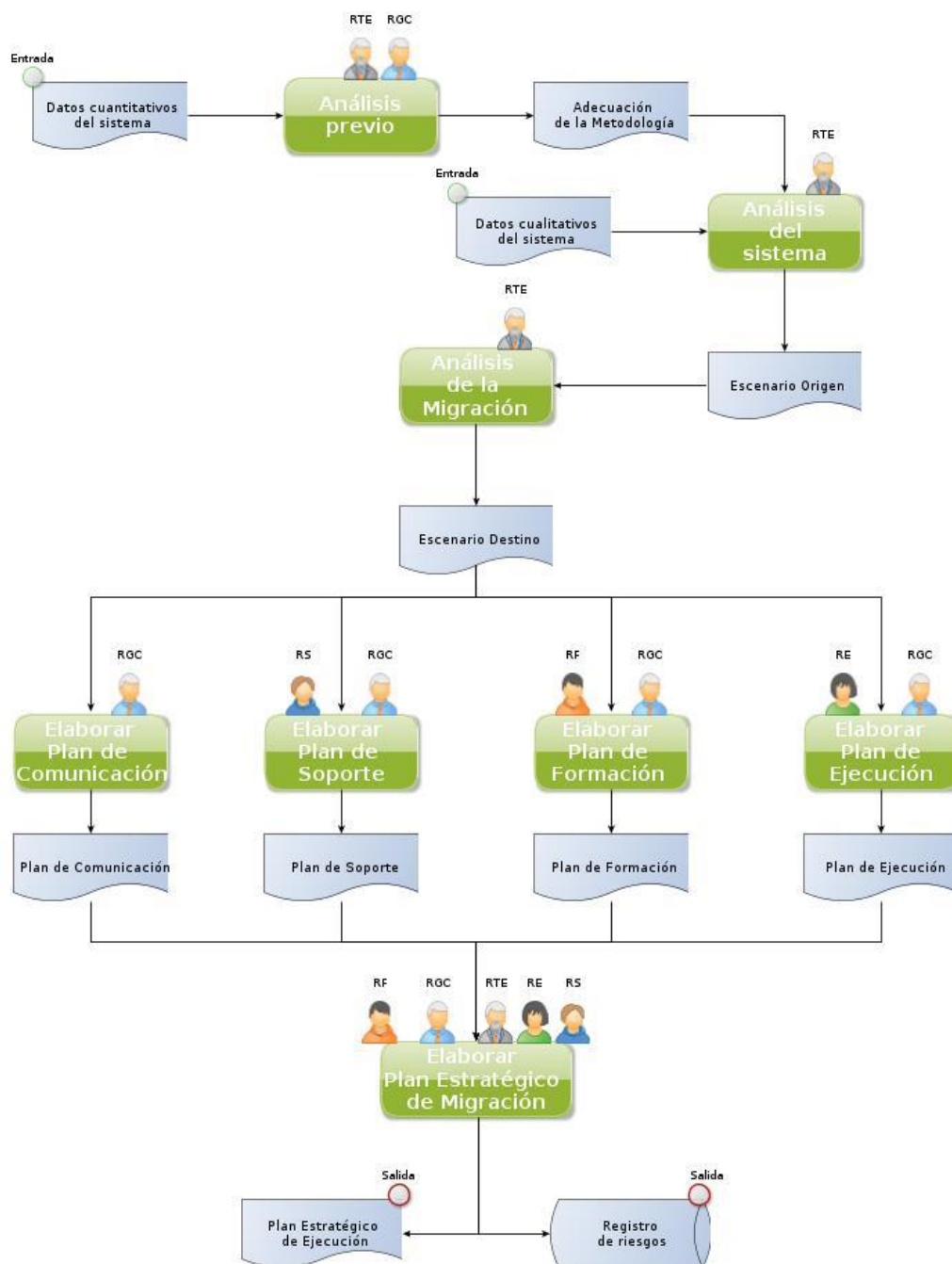
Plan de Soporte:

Incluirá todo lo necesario para gestionar las incidencias que los usuarios detecten durante la migración: detalle de los protocolos de comunicación de incidencias, niveles de servicio para la resolución, dedicación del equipo de soporte y demás factores clave en la definición de un servicio de atención a usuarios.

Plan de Comunicación:

tras la definición de los planes anteriores, es el momento de diseñar un plan de comunicación que mantenga al usuario final y a los equipos de desarrollo y soporte de la organización con un grado suficiente de información acerca del cambio que se va a producir, cómo informar de incidencias detectadas, etc. Por último, una vez definidos los Escenarios Origen y Destino y confeccionados los distintos planes, se pone toda la información en común para generar un plan global que llamaremos Plan Estratégico de Migración.

Ilustración de la migración de datos para la verificación en un plan de migración



Para la verificación en un plan de migración se pueden utilizar las siguientes consultas a la base de datos:

- Conteo de número de registros para categorizar la base de datos. (esto en caso de existir estadísticas actualizadas de MySql)

```
select sum (num_rows) from dba_tables where OWNER = 'SCHEMA_NAME';
```

- Consulta para determinar numero de atributos:

```
SELECT
AVG(COUNT(ATC.COLUMN_NAME)),STDDEV(COUNT(ATC.COLUMN
_NAME)),
MAX(COUNT(ATC.COLUMN_NAME)),MIN(COUNT(ATC.COLUMN_NA
ME))
FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE
ATC.TABLE_NAME = AT.TABLE_NAME
AND AT.OWNER = 'SCHEMA_NAME' and ATC.OWNER =
'SCHEMA_NAME' GROUP BY ATC.TABLE_NAME;
```

- Consulta para calcular cantidad de indices

```
SELECT AC.CONSTRAINT_TYPE,count(*) FROM ALL_CONSTRAINTS
AC, ALL_TABLES ALT
WHERE AC.TABLE_NAME = ALT.TABLE_NAME AND ALT.OWNER =
'SCHEMA_NAME' AND AC.OWNER = 'SCHEMA_NAME'
GROUP BY AC.CONSTRAINT_TYPE;
```

- Consultar los cantidad y tipos de objetos existentes en la base de datos

```
Select owner,object_type,count(*)from dba_objects where owner =
'SCHEMA_NAME'
Group by owner,object_type Order by 1,2,3;
```

Después de conocer las estadísticas anteriores, y por supuesto los tipos de objetos existentes en la base de datos de origen, se puede empezar a buscar cada uno de ellos en la documentación respectiva de Postgresql y así determinar las diferencias de sintaxis. En caso de que la migración se realice por medio de un programa o software, se deben realizar las consultas mencionadas anteriormente para verificar que la información se pasó correctamente.

Tener en cuenta que la migración se puede realizar de acuerdo a varios criterios, por ejemplo, se puede generar un back up solo de las estructuras, o un back up completo. Esto depende del criterio del DBA. Otra cosa que no se puede olvidar es generar el back up de la base de datos actual. Esto es indispensable por que no se conoce que situación se presente.

Si ya se tiene el back up y si ya se definió la forma en la que se va a realizar la migración es decir manual, o por medio de algún software, se tienen que desactivar o mejor no pasar los disparadores porque estos pueden ocasionar problemas o errores de migración. Así mismo con los índices, llaves o constraint.



Se recomienda hacer primero la creación de tablas, luego de vistas, índices, claves, restricciones.

Hace falta mencionar que en general una base de datos es creada para gestionar información, y casi siempre esta va enlazada a una aplicación por tal razón no olvidar realizar las pruebas de conexión respectivas, antes de empezar a realizar la migración, una vez se tenga una conexión establecida, se debe también analizar cuál es el papel de la base de datos en la aplicación, es decir si desde el desarrollo se realizan consultas o actualizaciones a la base de datos.

Después de mencionar varios aspectos a tener en cuenta en la migración de base de datos, y teniendo como tal la estructura y datos en el nuevo gestor, se empieza a pasar como tal la lógica del negocio y esto es lo correspondiente a procedimientos, funciones, disparadores y demás objetos, utilizando la sentencias correspondientes en Postgresql.