Título: Diseño del Motor de Procesamiento de Sentencias para el Sistema Chronos.

Autores: Ing. Diosnel Quiñones Rosales [(1)]
Ing. Alain Arias Yanez [(2)]

1 Universidad de las Ciencias Informáticas-Facultad Regional de Granma, Cuba, dquinones@grm.uci.cu
2 Universidad de las Ciencias Informáticas-Facultad Regional de Granma, Cuba, aayanez@grm.uci.cu

Resumen

Los procesos de negocio involucran actualmente grandes volúmenes de información y la complejidad de sus relaciones se incrementa cada día, por lo que las empresas e instituciones científicas buscan alternativas eficientes y poco costosas que permitan el almacenamiento y tratamiento de la información. La replicación de los datos es una de las soluciones más utilizadas en la actualidad para este tipo de problemas. En la Universidad de las Ciencias Informáticas se desarrolla el Sistema Chronos, que pretende ser una herramienta de réplica asíncrona para entornos multimaestro. Ha sido concebida a partir de necesidades particulares de un escenario de réplica, asumiendo las principales potencialidades de las soluciones similares en la actualidad para Postgresql e implementando nuevas mejoras que la distinguen del resto. Surge la necesidad de las especificaciones del diseño de un componente que capture y replique las sentencias SQL originadas en un nodo del *cluster* de bases de datos al resto de los nodos de manera asíncrona, que recopile la información necesaria para realizar los reportes de mensajes de estados del *cluster* de bases de datos para las actividades de monitorización, que garantice la recuperación de los nodos en situaciones de *failover* y *failback* permitiendo la mayor convergencia de datos posible, que permita la replicación de datos instantánea y programada, una resolución de conflictos personalizada a partir de las exigencias de un negocio en particular, que cumpla con los tipos de sincronización del resto de las herramientas similares y que administre las conexiones a bases de datos de manera eficiente.

Palabras Claves: asíncrona, bases de datos, Chronos, multimaestro, Postgresql, réplica.

Introducción

El establecimiento del conocimiento y la información, unido al incremento de los avances en las Tecnologías de la Informática y las Comunicaciones (TIC) y su rápida expansión a través de la Internet propician las condiciones necesarias para lograr obtener un desarrollo acelerado de las investigaciones. Los procesos de negocio involucran actualmente grandes volúmenes de información y la complejidad de sus relaciones se incrementa cada día, por lo que las empresas e instituciones científicas buscan alternativas eficientes y poco costosas que permitan el almacenamiento de la información, es por esto que surge la necesidad de la replicación de los datos. La réplica de bases de datos es una de las soluciones más generalizadas para alcanzar los retos que imponen la alta disponibilidad, tolerancia a fallos y escalabilidad en los sistemas de software actuales. La amplia gama de soluciones existentes incluyen sistemas que involucran replicación síncrona y asíncrona, desarrolladas para entornos de réplica multimaestro y maestro-esclavo. La replicación síncrona es muy utilizada en *cluster* de bases de datos destinados a garantizar servicios de alta disponibilidad, muy tolerante a fallos si es configurada en un entorno de réplica multimaestro. En entornos maestro-esclavo es aplicada donde el volumen de consultas de lectura posibles es superior a las transacciones de escritura o cuando es necesario centralizar los permisos de cambios en un servidor únicamente.

La replicación asíncrona es utilizada en sistemas donde la actualización de los cambios es un

proceso independiente a la ejecución de la transacción en sí, lo que no significa que la réplica no se realice de manera instantánea. Si bien esto limita la utilidad para algunas configuraciones, la realidad es que en ocasiones se hace imprescindible su uso como la mejor solución a determinados escenarios. Las herramientas que permiten réplica de datos poseen diversas aplicaciones debido a su utilización para configurar sistemas que requieren alta disponibilidad de los datos, mayor capacidad de respuesta, balance de carga en el acceso y sistemas de copias de respaldo de datos, por solo mencionar los más importantes.

La réplica de datos supone un costo elevado en el tráfico de paquetes por la red y en la cantidad de operaciones que se ejecutan por cada transacción que realiza el usuario. Es necesario desarrollar pruebas que permitan seleccionar la mejor configuración de un entorno de réplica. Un estudio de las principales herramientas de réplica para Postgresql existentes en la actualidad demostró la existencia de inconsistencias en situaciones de *failover* y *failback* en nodos del *cluster* de bases de datos, poca utilización de interfaces gráficas de usuario para la administración y configuración, la configuración total por *Shell*, la exigencia de un conocimiento a profundidad de la herramienta por el administrador de bases de datos, mínima validación de las configuraciones por lo que no se reduce el margen de error y la gran mayoría de las herramientas no permiten la configuración de réplica programada. En el caso de los sistemas asíncronos multimaestro presentan una resolución de conflictos muy básica.

En la Universidad de las Ciencias Informáticas se desarrolla el Sistema Chronos, que pretende ser una herramienta de réplica asíncrona para entornos multimaestro. Ha sido concebida a partir de necesidades particulares de un escenario de réplica, asumiendo las principales potencialidades de las soluciones similares en la actualidad para Postgresql e implementando nuevas mejoras que la distinguen del resto. Surge la necesidad de las especificaciones del diseño de un componente que capture y replique las sentencias SQL originadas en un nodo del *cluster* de bases de datos al resto de los nodos de manera asíncrona, que recopile la información necesaria para realizar los reportes de mensajes de estados del *cluster* de bases de datos para las actividades de monitorización, que garantice la recuperación de los nodos en situaciones de *failover* y *failback* permitiendo la mayor convergencia de datos posible, que permita la replicación de datos instantánea y programada , que permita una resolución de conflictos personalizada a partir de las exigencias de un negocio en particular, que cumpla con los tipos de sincronización del resto de las herramientas similares y que administre las conexiones a bases de datos de manera eficiente .

A partir del análisis de esta **situación problemática** surge la presente investigación que se plantea el **problema** de: Inexistencia de un componente que permita la réplica asíncrona de sentencias y la recopilación de información para actividades de monitorización en un *cluster* de bases de datos Postgresql. El **objeto de estudio** en el que se enmarca la investigación lo constituyen los procesos de réplica de bases de datos objeto-relacionales. El **campo de acción** abarcado son los procesos de réplica asíncrona multimaestro de bases de datos Postgresql.

El **objetivo general de esta investigación es** elaborar el diseño del Motor de Procesamiento de Sentencias del Sistema Chronos.

Desarrollo

Réplica de datos

Para comprender el objeto de estudio del trabajo se fundamentan algunos conceptos importantes sobre el contenido de esta investigación.

¿Qué es un Sistema de Gestión de Bases de Datos (SGBD)?

“Un SGBD es la herramienta que permite interactuar los datos con los usuarios de los datos, de forma que se garanticen todas las propiedades definidas en una base de datos” [2]. Estas propiedades son la integridad, confidencialidad, seguridad y disponibilidad de los datos.

Actualmente estos sistemas presentan problemas con el procesamiento de gran volumen de información, por lo que surge la necesidad de la replicación de datos como solución, para lograr un mejor rendimiento del SGBD.

¿Qué es un cluster?

Un cluster consiste en un tipo de sistema de procesamiento paralelo o distribuido, compuesto por un conjunto de computadoras que trabajan cooperativamente como un único e integrado recurso de cómputo [3].

Los conceptos de SGBD y de cluster están estrechamente relacionados con la réplica de datos, que según [4] se define como: “En un sistema de gestión de bases de datos distribuidas, es el proceso de copiar la base de datos (o partes de la misma) a los otros lugares de la red. La replicación permite a los sistemas de bases de datos distribuidas mantener la sincronización”. Este proceso asegura que los datos estén siempre disponibles en el lugar necesario para ser utilizados en el momento indicado.

Tipos de réplica

Los tipos de replicación están relacionados con el momento en que se realiza la actualización de los datos cuando se efectúa una transacción de escritura en la Base de Datos.

Replicación Síncrona

En el caso de la replicación síncrona, si se actualiza una réplica dada, todas las demás réplicas del mismo fragmento de datos también se actualizan dentro de la misma transacción; lo que implica que (desde un punto de vista lógico) solo existe una versión de los datos. La mayoría de los productos implementan la replicación síncrona por medio de disparadores (posiblemente ocultos y manejados por el sistema). Sin embargo la replicación síncrona tiene la desventaja de que impone una sobrecarga sobre las transacciones que actualizan cualquier réplica [5].

Replicación Asíncrona

En el caso de la replicación asíncrona, las actualizaciones a una réplica son propagadas hacia los demás en algún momento posterior, no dentro de la misma transacción, por lo tanto la replicación asíncrona presenta un retardo de tiempo o latencia, durante el cuál es posible que las réplicas no sean idénticas [5].

Entornos de réplica

La réplica de datos puede aplicarse de dos maneras diferentes de acuerdo a las necesidades propias de las personas, sistemas o instituciones que la utilicen.

Multimaestro

El entorno de replicación multimaestro es conocido como “par a par o la réplica de camino de n” que permite múltiples nodos actuando como pares iguales. En este entorno cada nodo es maestro, y

se comunica con otros nodos maestros. Cada nodo permite operaciones de lectura y escritura, cuando se modifica cualquiera de los nodos el resto es actualizado, por lo que se elimina la sobrecarga de un único nodo. La replicación multimaestro es una alternativa eficaz para aquellos sistemas que realizan gran cantidad de operaciones de lectura/escritura sobre sus datos [6].

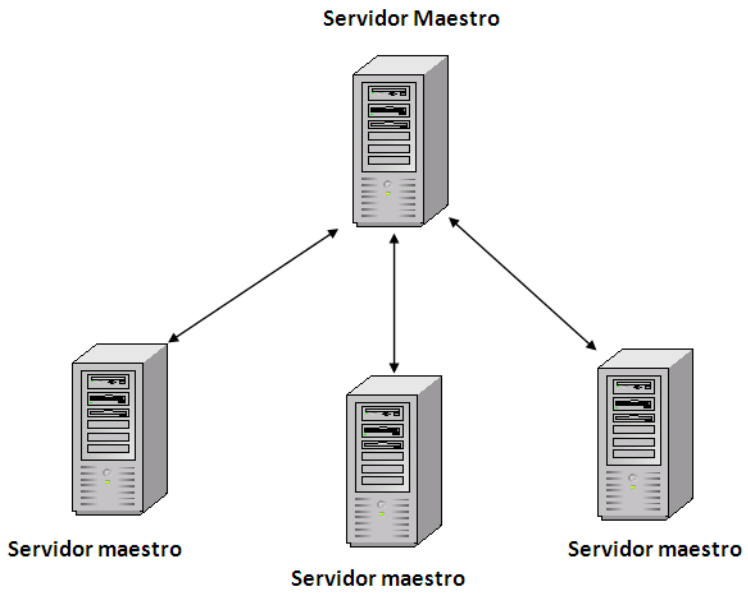


Figura 1 Entorno de Réplica Multimaestro.

Maestro-Esclavo.

Entorno de replicación maestro-esclavo es conocida también como “de solo lectura “, permite a un solo maestro recibir consultas de lectura/escritura, mientras los esclavos solo pueden aceptar consultas de lectura. Debido a que las operaciones de escritura pueden ser efectuadas en un único nodo, para sistemas que realicen modificaciones constantemente, no se logran buenos índices de rendimiento [7].

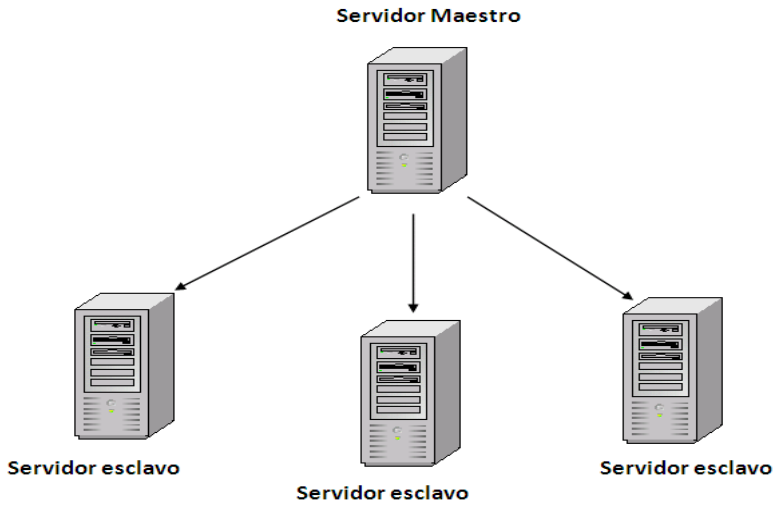


Figura 2 Entorno de Réplica Maestro – Esclavo

Técnicas de replicación asíncrona

Las técnicas de replicación asíncrona están relacionadas con el intervalo de tiempo que existe entre

un cambio realizado en la base de datos y el momento en que este se replica.

1. La replicación instantánea ocurre cuando el intervalo de tiempo entre el “commit” de una transacción SQL y el momento en que se realiza la réplica hacia el resto de las fuentes de datos esmuy pequeño. Es utilizada cuando se desea obtener la sincronización del sistema en el menor tiempo posible [8] [9].

2. La replicación programada se utiliza para sistemas donde la sincronización de los datos no tieneque ser inmediata y puede realizarse en determinados momentos en los que el cluster de bases dedatos no esté recibiendo tanto volumen de carga. Esta puede ser configurada con una periodicidaddiaria, semanal, mensual o cada N horas. Siendo una de las características del sistema que permite su flexibilidad para escenarios donde existan estaciones de trabajo locales que no necesiten de constante actividad con los servidores centrales [8] [9].

Objeto de automatización.

Los procesos a automatizar son los siguientes:

Réplica instantánea de transacciones entre bases de datos Postgresql: Este proceso permite la replicación instantánea de transacciones entre bases de datos Postgresql. A este tipo de replicación se hace referencia en el epígrafe 1.4 de esta investigación.

Réplica programada de transacciones entre bases de datos Postgresql: Este proceso permite la replicación programada de transacciones entre bases de datos Postgresql. A este tipo de replicación se hace referencia en el epígrafe 1.4 de esta investigación.

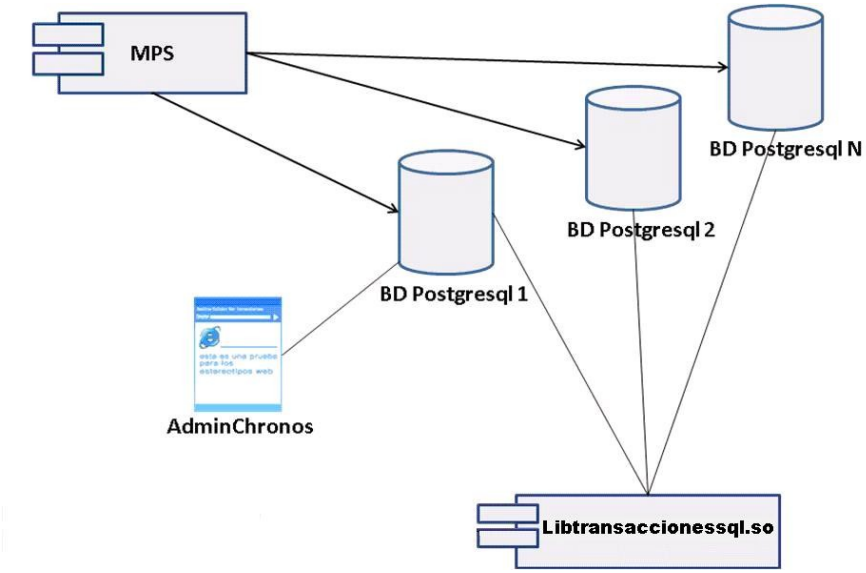
Recopilación de información del estado del cluster de bases de datos: Este proceso se encarga del envío de mensajes sobre la información de la réplica en su nivel más bajo. Se encarga del registro de la actividad de réplica en cada nodo de bases de datos mediante ficheros de log. La información es más específica y detallada y consiste en: usuarios, réplicas realizadas, estados de consistencia de las bases de datos, estado de las sincronizaciones programadas y tipos de eventos que el sistema está procesando.

Administración de Conexiones a las bases de datos del cluster: Este proceso se encarga de administrar las conexiones a las bases de datos del cluster, permitiendo el uso eficiente de las mismas y aumentando el rendimiento general de los nodos.

Carga de información de eventos de réplica: Este proceso se encarga de cargar toda la información de los eventos programados e instantáneos configurados para ejercer las actividades de réplica entre bases de datos.

Componentes del Sistema

El sistema Chronos está integrado por varios componentes que permiten el funcionamiento de la réplica y la integración con el Sistema de Gestión de bases de datos (SGBD) Postgresql. En la figura 4 se representa un diagrama de componentes generales del mismo.



BD Postgresql desde 1 hasta N: Versiones liberadas de la base de datos Postgresql que constituyen los Sistemas de Gestión de bases de datos con los que interactúa el Sistema Chronos.

Libtransaccionesql.so: Librería de enlace dinámico que extiende la funcionalidad del gestor capturando los cambios que se producen en cualquiera de las tablas contenidas, los cuales deben ser replicados según el tipo de sincronización que se configure.

Motor de Procesamiento de Sentencias (MPS): El Motor de Procesamiento de Sentencias (MPS) deberá replicar los cambios ocurridos en los Sistemas de Gestión de Base de Datos (SGBD) Postgresql configurados por la herramienta de administración (AdminChronos) y utilizará la librería Libtransaccionesql.so para capturar los cambios que se producen en cualquiera de las tablas de la Base de Datos.

AdminChronos: Aplicación Web de administración, configuración y monitorización en línea del sistema. Posee una interfaz gráfica de usuario amigable y fácil de usar, lo que le confiere un valor agregado a la aplicación. A través de esta es posible realizar todas las configuraciones de sincronización para todos los nodos del cluster de manera sencilla y rápida. Permite monitorizar el estado de la réplica haciendo interactuar al usuario con este como parte del sistema. No interviene en el rendimiento de los nodos del cluster de bases de datos puesto que puede ser utilizada desde una estación de trabajo remota que tenga acceso a la base de datos del sistema Chronos.

Características del Sistema Chronos

El funcionamiento de estos componentes debe garantizar un conjunto de características en el sistema que lo diferencian del resto de las herramientas de su tipo:

Soporte para multiprocesamiento

El Motor de Procesamiento de Sentencias (MPS) debe constituir el núcleo de réplica de la aplicación y permitir soporte para multiprocesamiento a partir de técnicas avanzadas de programación concurrente. Esto permitirá lograr una mayor cantidad de Transacciones Por Segundo (TPS) aumentando el rendimiento total del sistema a la vez que aprovecha las características multinúcleo de los servidores. Además se deberá ejecutar como un proceso independiente capturando, planificando y replicando los cambios ocurridos en los SGBD Postgresql.

Técnicas de Replicación

Las técnicas de réplica asumidas por el servidor deben ser instantáneas o programadas, cumpliendo con las exigencias de este tipo de software en la actualidad.

Resolución de conflictos personalizados

El sistema Chronos debe ofrecer un conjunto de soluciones estándares frecuentes para los conflictos que pueden surgir durante el proceso de sincronización de los datos. Si las soluciones no satisfacen las necesidades del usuario, este podrá definir sus propios métodos de solución de conflictos que pueden ser muy específicos del negocio al que responde la Base de Datos, permitiendo la adaptabilidad del sistema a escenarios específicos de réplica.

Tipos de Sincronizaciones

Las sincronizaciones entre fuentes de datos definen cómo se realizará la réplica entre una fuente origen y su destino. Estarán clasificadas en tres tipos diferentes:

1. **Adicionar Diferencias:** Permite adicionar sólo las diferencias (tuplas o registros) de una fuente dedatos origen hacia una fuente de datos destino.
2. **Intercambio:** Garantiza que los datos de una fuente origen se adicionen en la fuente destino yviceversa. Esta sincronización requiere un método de resolución de conflictos que permita tomar una decisión en caso de que ocurra una anomalía. Es muy utilizado en escenarios de réplicabidireccional.
3. **Copia Total:** Este método establece una copia total de una fuente de datos origen hacia una fuente destino. Los datos de la fuente destino se pierden y son reemplazados por la tabla origen. Suele ser bastante costoso para la réplica instantánea y es muy utilizado para recuperaciones de fuentes de datos que inician desde cero o adición de nuevas fuentes a sincronizaciones ya establecidas.

Administración de conexiones inteligente

Chronos manipulará de manera inteligente las conexiones a las bases de datos del cluster, permitiendo el uso eficiente de las mismas y aumentando el rendimiento general de los nodos.

Recuperación ante fallos

Ante la caída de un nodo el sistema debe ser capaz de detectarlo. Cuando se restablece el servicio, Chronos reiniciará una recuperación de las bases de datos inconsistentes, logrando la convergencia de los datos respecto al resto de los nodos.

Análisis comparativo de otras soluciones existentes con la propuesta

La replicación instantánea es implementada por todas las herramientas de réplica existentes en el mundo como Pgpool II, Bucardo, Reko, Réplica bidireccional basada en control de cambios, Slony-I y PgCluster y la propuesta de solución implementará la replicación programada que es totalmente nueva además de la instantánea. Destacar que Bucardo es la única herramienta de réplica que utiliza la resolución de conflictos personalizados pero de manera muy pequeña y los tipos de sincronizaciones, la resolución de conflictos personalizados que se prevee en la propuesta es bastante similar a la que utiliza Bucardo pero de manera más amplia y utilizará los tipos de sincronizaciones. Las herramientas de réplica existentes no permiten administración de conexiones inteligentes, excepto la herramienta de réplica Pgpool II y no permiten la recuperación ante fallos de manera automática, excepto la herramienta de réplica PgCluster, la propuesta de solución permitirá

ambas características. Las herramientas de réplica Reko y Réplica bidireccional basada en control de cambios, no realizan la técnica de réplica programada, poseen una resolución de conflictos muy básica y no administran de forma inteligente las conexiones a la base de datos, la propuesta de solución reúne todas estas características.

Dependencias y Relaciones con otros software

Para el correcto funcionamiento del Motor de Procesamiento de Sentencias (MPS) es necesario utilizar las librerías que se describen a continuación:

Libconfuse (Lectura de parámetros de configuración)

Libconfuse es una librería para el análisis de los archivos de configuración escritos en C. Soporta secciones, listas, valores como cadenas, enteros, reales, booleanos entre otros, así como algunas otras características tales como uno o dos cadenas entre comillas, la expansión de variable de entorno, funciones e incluye también declaraciones anidadas. Es utilizada por el Lector de Parámetros de configuración para parsear los datos configurados en el fichero chronos.conf [24].

Libpq (conexión a postgres desde C++)

Libpq es una librería para la conexión a PostgreSQL desde C++. Esta permite que los programas de usuario puedan comunicarse con la Base de Datos PostgreSQL. La Base de Datos Postgresql puede ser local o puede ser en otra máquina y acceder a través de TCP / IP. Se utiliza para la comunicación con las bases de datos en réplica desde el MPS [25].

Modelo de datos

Este modelo de datos es usado para describir la representación lógica y física de la información persistente manejada por el sistema.

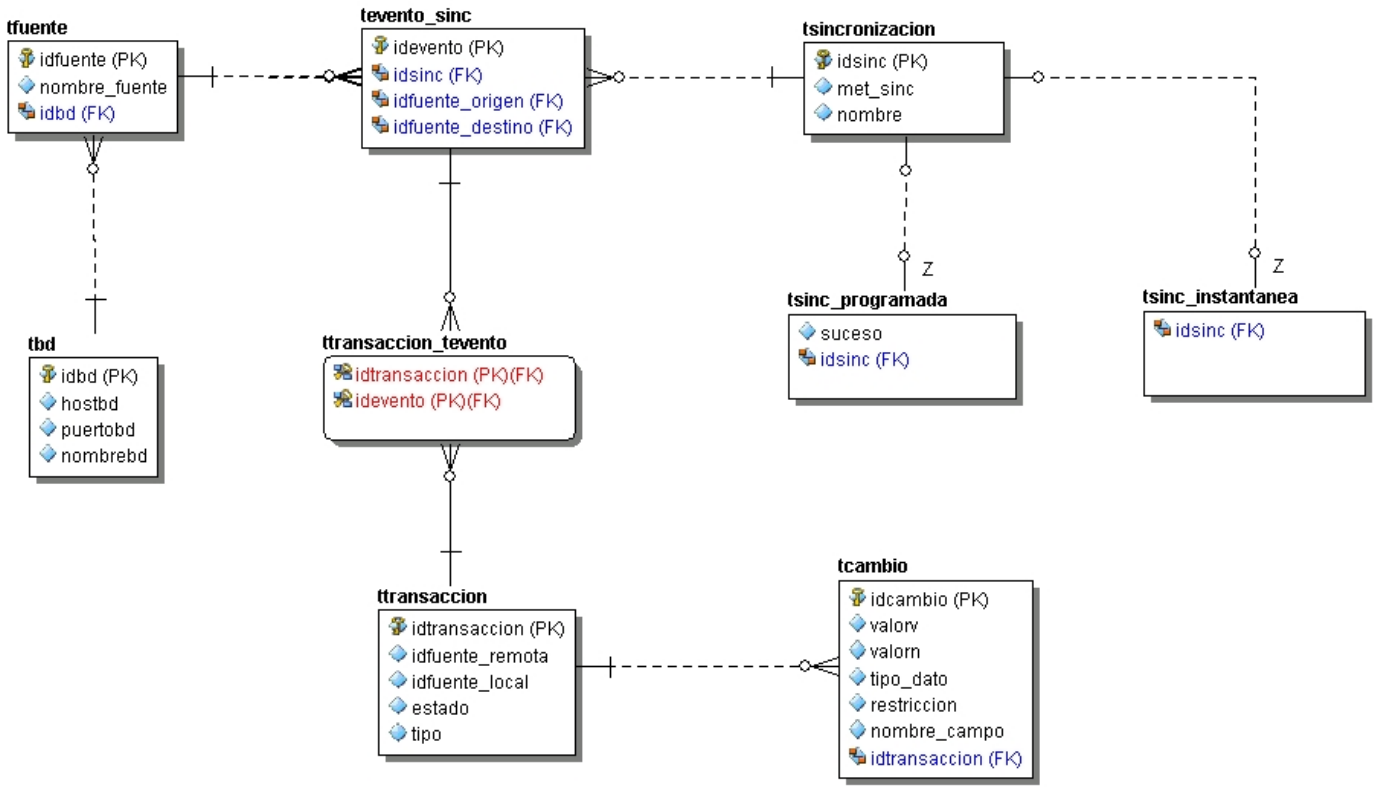


Figura 5 Diagrama Entidad Relación

En el diseño de la Base de Datos se definió un esquema (chronos). El esquema chronos contiene: tablas, funciones, disparadores y secuencias.

Dentro del esquema chronos se definieron 4 funciones, utilizando el lenguaje plpgsql:
- chronos.confirma_replica (bigint, bigint, bigint): Confirma la realización correcta de la réplica de una transacción, eliminando sus datos asociados una vez que esta se replicó correctamente.
- chronos.fregistrar_cambios_uaplicacion (bigint, integer, character varying, variadic text [], variadic text [], variadic text [], variadic text []): Utilizada por la líbrería libtransacciones.so cuando el usuario que replica es externo. Actualiza los cambios en la tabla tcambio que se realizan en una fuente de datos origen realizados por el usuario de aplicación. Actualiza los eventos asociados a la transacción en la tabla rttransaccion_tevento e inserta la transacción nueva en la tabla ttransaccion.
- chronos.fregistrar_cambios_ureplica (integer, variadic text [], variadic text [], variadic text [], variadic text []): Utilizada por la líbrería libtransacciones.so cuando el usuario que replica es chronos. Actualiza los cambios en la tabla tcambio que se realizan en una fuente de datos origen realizados por el usuario de aplicación para la última transacción realizada.

- chronos.finsertar_sql_replica (text, text, bigint, bigint) Utilizada por el MPS en la réplica de réplicas. Actualiza los eventos asociados a la transacción en la tabla rttransaccion_tevento e inserta la transacción nueva en la tabla ttransaccion. Bloquea la tabla ttransacion para que fregistrar_cambios_ureplica pueda actualizar los cambios para esta operación.

Dentro de las secuencias se definieron:

- chronos.sec_nestado_transaccion_seq: Permite incrementar en uno el valor del campo X cada vez que se modifique el estado de la transacción en la tabla Y.
-chronos.sec_tcambio_seq: Permite incrementar en uno el valor del campo X cada vez que se ejecute un cambio en la tabla Y.
-chronos.sec_tevento_sinc_seq: Permite incrementar en uno el valor del campo X cada vez que se inserta un registro en la tabla Y-chronos.sec_transaccion_seq: Permite incrementar en uno cada vez que se ejecute una transacción.

Existen tablas como tbd, tfuente, tsincronizacion, instantánea y programada que no tienen secuencias asociadas porque el valor de sus llaves primarias es establecido por la aplicación de administración y configuración AdminChronos.

Descripción de la arquitectura propuesta

Arquitectura de 3 capas

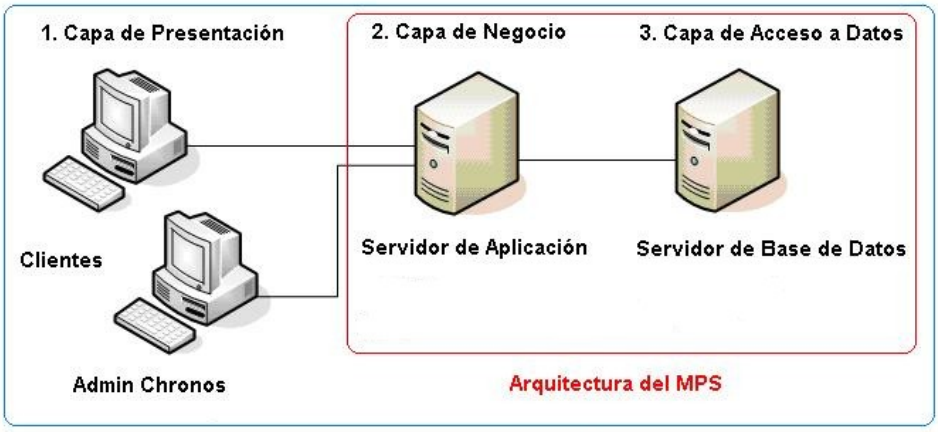
Los sistemas o arquitecturas en capas definen cómo organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Es importante definir una arquitectura única sobre la cual construir la aplicación. Para integrar todo el ambiente de desarrollo hay que definir la estructura sobre la que descansarán las próximas soluciones del negocio, esto propicia la neutralidad e independencia de los productos para adaptarse a la arquitectura y consolida la cohesión de las diferentes tecnologías existentes [28].
Las capas lógicas definidas son:

Capa de presentación: es la que presenta el sistema al usuario, comunica y captura la información

de este en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario.

Capa de negocios: Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de Acceso a Datos: Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Arquitectura en capas del Sistema Chronos

El sistema Chronos en general con todos sus componentes utiliza una arquitectura en 3 capas compuesta por la capa de presentación en donde se encuentra la interfaz de administración y configuración de la aplicación web AdminChronos, la capa de negocio que está compuesta por el Motor de Procesamiento de Sentencias (MPS) y el negocio de la aplicación AdminChronos y por la capa de Acceso a Datos que está compuesta por la Base de Datos y el esquema Chronos. El Motor de Procesamiento de Sentencias es un componente que forma parte del sistema Chronos y la arquitectura que presenta no posee la capa de presentación porque no tiene interfaz gráfica ni usuarios que interactúen con ella por lo cual solamente existe en la capa de Negocio y la capa de acceso a datos.

Conclusiones

Con este diseño se le da cumplimiento al objetivo general trazado al inicio del desarrollo del trabajo, realizando el diseño del Motor de Procesamiento de Sentencias del Sistema Chronos que permita la réplica asíncrona multimaestro de sentencias y la recopilación de información para actividades de monitorización en un cluster de bases de datos Postgresql.