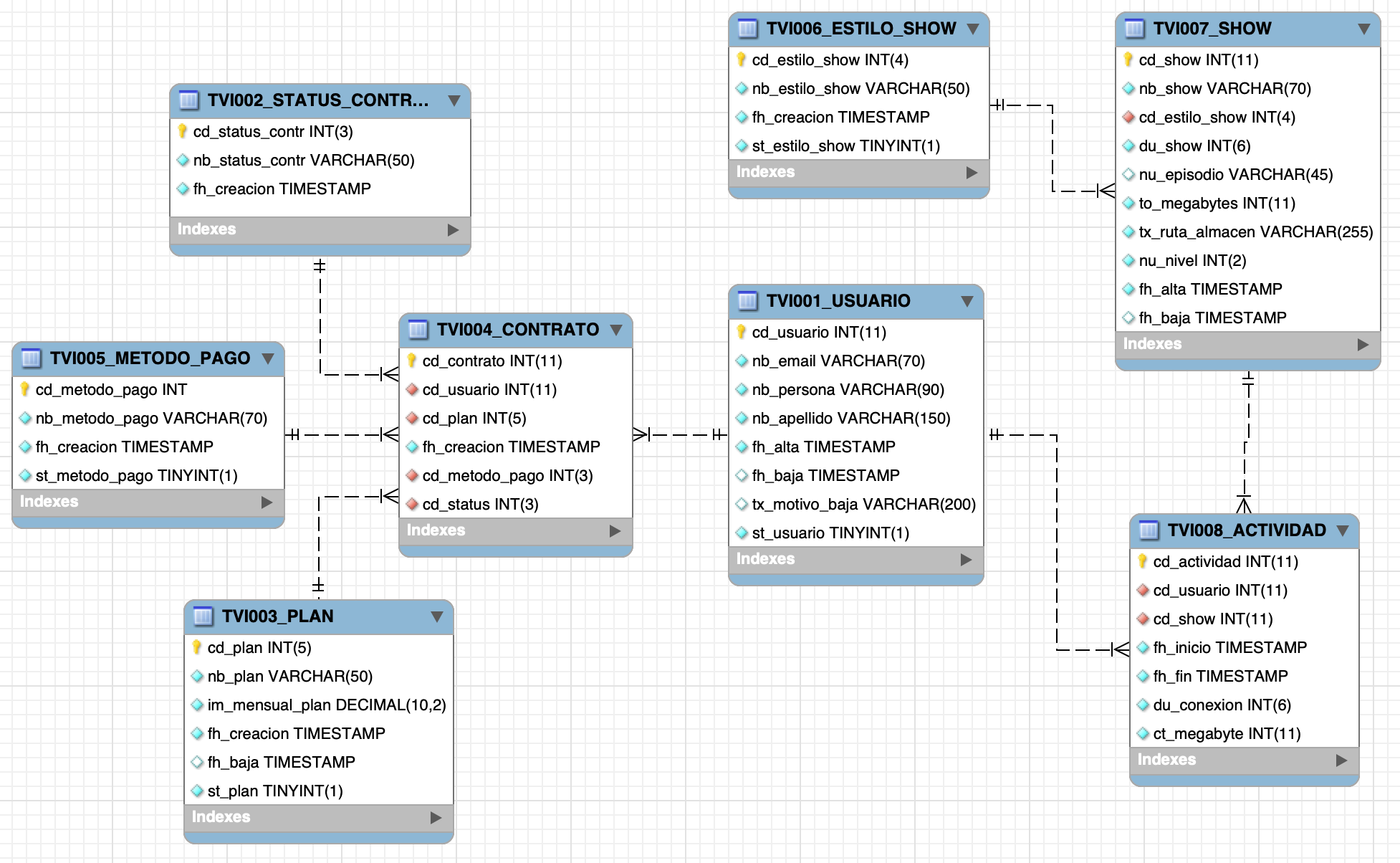
**Challenge.**

**Desarrollador: Rodrigo Salvador Murillo González.**

El modelo de datos diseñado muestra una posible solución para soportar un servicio de transmisión de video.

Se contempla un servicio de tipo Netflix únicamente para consumo de video.



Considerando que el motor de bd que se utiliza es MySql, el modelo E-R se desarrolló con la herramienta MySQL Workbench.

Este modelo plantea la solución de manejar entidades como USUARIO que desea contratar un servicio de transmisión de video mediante un CONTRATO, asociado a un PLAN, dependiendo de las opciones y precios disponibles.

Los SHOW se consideran los programas, noticieros, series de tv, películas, entre otras. La entidad ACTIVIDAD registra el tiempo que un usuario ve o consume un SHOW.

Con base en este modelo, se proponen las métricas de usuarios registrados y sus consumos, costos de los planes contratados, así como para identificar shows que no son de interés para los clientes.

Adjunto en el repositorio, existen los siguientes archivos:

* **ddl / script\_mytestdb\_ddl.sql**

Contiene las sentencias necesarias para la creación de la base de datos y las tablas mostradas en el diagrama E-R. Se desarrolló considerando estándares de modelado y nomenclatura para nombres de tablas, campos y vistas.

* **dml / script\_mytestdb\_dml.sql**

Contiene las sentencias necesarias para popular las tablas mostradas en el diagrama E-R, con datos de prueba para ejemplificar los catálogos y los datos de actividades de usuarios. El orden de inserción de los datos respeta integridad referencial entre las tablas.

Tomando en consideración que para MySQL no existen las tablas materializadas, se buscó una forma alternativa para tener esta solución. Se construyen dos tablas que simulan ser las vistas materializadas, y se les actualiza la data sin afectar el rendimiento de la base.

Adjunto en el repositorio, existe el siguiente archivo:

* **procedures / script\_mytestdb\_materializacion.sql**

Contiene la sentencia para la generación de un EVENTO que se ejecuta diariamente con una frecuencia de 2 horas, ejecutando un STORED PROCEDURE que destruye la tabla y la vuelve a construir tomando información más reciente.

Finalmente, para crear tablas agregadas, se construyen dos tablas que contienen la información agregada para el tiempo transmitido por el usuario y por show.

El repositorio se incluye el siguiente archivo:

* **procedures / script\_mytestdb\_agregacion.sql**

Contiene la sentencia para la generación de un EVENTO que se ejecuta diariamente a las 00:00 hrs, ejecutando un STORED PROCEDURE que destruye la información del mes corriente y vuelve a insertarla tomando información más reciente. Esta información tendrá un delay de 1 día, ya que por tratarse de información agregada, su uso es para analíticos, y no para consulta de información transaccional.

Nota: El Script de ***python / scala*** no se consideró necesario, ya que las soluciones propuestas se pueden resolver con las mismas herramientas internas de MySQL, aunque si fuera necesario el desarrollo de un ETL orquestado por alguna herramienta ajena, se puede realizar.