



# MANUAL TÉCNICO

ETAB

Esta página se ha dejado vacía a propósito

# Índice de contenidos

|            |  |           |
|------------|--|-----------|
| Capítulo 1 | <b>Tecnologías utilizadas</b>                              | <b>7</b>  |
| Capítulo 2 | <b>Tablero eTAB</b>  | <b>11</b> |
| 2.1        | Instalacion del Sistema Integrado de Información Gerencial | 11        |
| 2.2        | Requerimientos   | 11        |
| 2.3        | Pasos  | 11        |
| 2.4        | 1. Instalación de Symfony2                                 | 12        |
| 2.5        | Configuración  | 12        |
| 2.6        | 2. Instalación de Postgres                                 | 14        |
| 2.7        | 3. Instalación de RabbitMQ                                 | 15        |
| 2.8        | 4. Instalación de Servidor OLAP Pentaho + Saiku            | 16        |
| 2.9        | Instalación de librería wkhtmltopdf                        | 18        |
| 2.10       | Validación de Usuarios desde directorios LDAP              | 18        |
| Capítulo 2 | <b>Fr3d_LDAP</b>   | <b>19</b> |
| Capítulo 3 | <b>Modelo De Datos</b>                                     | <b>21</b> |
| Capítulo 4 | <b>Gestión de Cubos OLAP</b>                               | <b>41</b> |

Esta página se ha dejado vacía a propósito

# Introducción

El presente manual técnico describe cada uno de los componentes del sistema eTAB y los pasos necesarios para instalarlo.

El sistema de información eTAB es parte de la iniciativa Salud Mesoamérica 2015 (SM2015). Esta es una iniciativa cuyo fin es reducir las inequidades en salud que están afectando al 20 por ciento mas pobre de la población en Centro America y Mexico. Esta iniciativa también tiene como objetivo apoyar los esfuerzos de los gobiernos de la región para alcanzar los Objetivos del Milenio.

Salud Mesoamérica 2015, pone especial atención a la áreas de salud reproductiva, nutrición maternal y neonatal, inmunización, y la prevención y control del dengue y la malaria. Para este fin Salud Mesoamérica 2015trabajara en conjunto con los ministerios de salud de la región y el Sistema de Salud Publica Mesoamericano. Este proyecto es parte de la plataforma de integración regional conocido como Proyecto Mesoamerica.

Los resultados esperados de incluyen una reducción significativa en las tasa de mortalidad infantil para niños de menos de cinco años. Esta iniciativa también esta busca reducir la malnutrición crónica en la niñez y las mujeres embarazadas. Estos cambios son críticos para mejorar las estadísticas sobre partos y para ofrecer mejores condiciones para el crecimiento del recién nacido. A su vez esta iniciativa busca tener un efecto directo en comunidades pobres sobre la cobertura y calidad de vacunas, control pre y post natal y acceso a planificación familiar entre otros servicios.

Salud Mesoamérica 2015 espera generar conocimiento de relevancia global sobre como aumentar asistencia en salud de bajo costo en comunidades pobres. Para este fin el sistema de información eTAB permite analizar y dar seguimiento a los indicadores en salud con los que trabaja este proyecto.

Esta página se ha dejado vacía a propósito

## Capítulo 1

# Tecnologías utilizadas

El Tablero eTAB es un servicio Web disponible para que dependencias del sistema de salud suban sus datos para poder analizarlos, generar gráficas y reportes.

La aplicación cuenta con un módulo para efectuar la extracción, transformación y carga de datos (ETL) desde diferentes fuentes. Estos datos son agregados y almacenados en una base de datos relacional (OLTP). Los datos están organizados por catálogos de referencia e Indicadores medibles. Los usuarios del sistema pueden administrar estos indicadores y catálogos y todos sus tributos usando el las herramientas que brinda el sistema. Para efectuar consultas en línea los datos son agregados dentro de tablas optimizadas para el análisis en línea (OLAP). Las tablas para análisis son actualizadas periódicamente usando procedimientos almacenados de PostgreSQL.

La gestión de consultas a las tablas de análisis OLAP se realiza por medio de un servidor dedicado. La interacción entre el servidor OLAP y el resto de la aplicación se realiza por medio de consultas AJAX. El resultado de las consultas al servidor OLAP, es procesado usando JQuery y graficado usando la librería de gráficos D3.

Todo el software utilizado para creación del SIIG/eTAB son paquetes de software libre. Estos incluyen:

- GitHub: Gestor de control de versiones de código fuente
- Apache: Servidor de paginas web
- PostgreSQL: Gestor de bases de Datos
- Symfony: Entorno de desarrollo para PHP
- PHP: Lenguaje de desarrollo de la Aplicación eTAB
- Java/OpenJDK: Lenguaje de desarrollo del servidor OLAP

- Pentaho: Servidor OLAP desarrollado en Java
- Saiku: Interfaz de analisis para consultas a servidor Pentaho
- D3.js: Librería para la generación de gráficos
- JQuery: Lenguaje para interfaces de usuario
- RabbitMQ: Servidor de Mensajería
- EasyBook: Generador de documentos en formato PDF
- Bootstrap: Framework para interfaces de usuario

### 1.0.1 Gestor de base de datos

[PostgreSQL] (<http://www.postgresql.org/>)

Versión 9.2 Actualmente el sistema únicamente puede utilizar PostgreSQL por la siguiente razón: La aplicación debe proveer la capacidad de analizar datos para cualquier indicador. Cada indicador esta construido con varios datos y relacionados por una formula almacenada en el sistema. Es posible crear una tabla por cada grupo de datos, con la limitante de que es necesario conocer el dato antes de guardarlo, lo cual no es sostenible a futuro.

La base de datos necesita guardar datos sin conocer de antemano sus características. Esto se logra usando un esquema de datos generico EAV (entidad-atributo-valor). El manejo de esquemas EAV es implementado de diferentes formas para diferentes gestores de bases de datos. El Tablero eTAB usa la implementación de Postgres la cual crea un tipo especial de dato llamado HSTORE. Además, el modulo de cubos OLAP utiliza la función CROSSTAB de Postgres para transponer grupos de datos. El tipo de campo HSTORE y la función CROSSTAB no existen en otros gestores de bases de datos por lo cual no seria posible instalar esta aplicación usando un gestor de base de datos que no sea PostgreSQL.

Para utilizar el tipo de datos HSTORE y la función CROSSTAB es necesario instalar el paquete contrib de postgresql y luego crear las extensiones correspondientes usando:

```
~#postgres> CREATE EXTENSION hstore ;
```

```
~#postgres> CREATE EXTENSION tablefunc ;
```

### 1.0.2 Servidor Web

[Apache2] (<http://www.apache.org>)

Apache es un servidor Web de código abierto, se ha realizado sobre Apache versión 2.2

### 1.0.3 Framework de desarrollo/Servidor

[PHP] (<http://www.php.net>)



[Symfony](<http://symfony.com/>)

[GitHub](<https://github.com/>)

El lenguaje que se ha utilizado es PHP 5.3.18 dentro de una estructura de desarrollo MVC manejada Symfony versión 2.4 Cada miembro del equipo de desarrollo usa un aplicativo diferente para escribir/modificar el código fuente. Los mas populares popular es Netbeans(version libre para PHP) y Nano. Para manejar cambios y mejoras al código fuente se uso Github. La totalidad del código fuente esta disponible en <https://github.com/erodriguez-minsal/SIIG>

### 1.0.4 Framework JavaScript

jQuery (<http://jquery.com/>) versión 1.8.3 junto a jQuery UI (<http://jqueryui.com/>)

### 1.0.5 Framework para interfaces de usuario

Bootstrap (<http://twitter.github.com/bootstrap/>)

Bootstrap es un framework que hace HTML, CSS y Javascript simple y flexible para componentes e interacciones de interfaz de usuarios populares.

### 1.0.6 Librería para gráficos

D3 (<http://d3js.org/>)

Antes conocida como Protovis, D3 es una biblioteca de JavaScript para manipular documentos basados en datos. D3 ayuda dar vida a los datos usando HTML, SVG y CSS. D3 enfatiza los estándares Web ofreciendo todas las capacidades de los navegadores modernos sin ligarse a una estructura propietaria. A diferencia de otras librerías, D3 no crea imágenes, sino que interactúa la pagina para crear los gráficos usando elementos de HTML5 como Canvas y SVG.

### 1.0.7 Mensajería

RabbitMQ (<http://www.rabbitmq.com/>)

La carga de datos se apoya de las librerías de este paquete para crear una 'lista de espera' para evitar que el servidor se sature al recibir demasiadas peticiones simultaneas.

### 1.0.8 Servidor de Cubos OLAP

Pentaho (<http://community.pentaho.com/>)

La version 'comunidad' del servidor de cubos OLAP Pentaho es un proyecto de código abierto desarrollado usando Java 6, se uso la ultima version disponible 4.8, el paquete incluye el servidor de aplicaciones Tomcat.

## 1.0.9 Interfaz de Analisis de cubos

Saiku (<http://community.pentaho.com/>)

Saiku es una aplicacion de JAVA que ofrece una interfaz escrita en JQuery para analizar cubos OLAP. En este proyecto se uso la version 'plugin' para Pentaho, version 2.4.

## 1.0.10 Documentación

La mayoría de la documentación ha sido escrita en formato markdown y se ha utilizado easybook (<http://easybook-project.org/>) para la gen

## Capítulo 2

# Tablero eTAB

### 2.1 Instalacion del Sistema Integrado de Información Gerencial

### 2.2 Requerimientos

- Servidor Web
- Gestor de base de datos
- PHP 5.3.8+
- Java 6

### 2.3 Pasos

1. Instalación de Symfony2
2. Instalación de Postgres
3. Instalación RabbitMQ
4. Instalación Servidor OLAP
5. Librería wkhtmltopdf

## 2.4 1. Instalación de Symfony2

### 2.4.1 Instalación de los requerimientos desde un servidor Debian

Es muy importante poner atención al indicador "#" significa que el comando debe ser ejecutado como usuario root y "\$" que debe ser ejecutado como un usuario normal

```
# apt-get update
# apt-get install php5 php5-pgsql php5-sqlite sqlite php5-xdebug
php-apc php5-cli php5-xsl php5-intl php5-mcrypt apache2 postgresql
acl git-core curl postgresql-contrib php5-ldap
```

### 2.4.2 Obtener el código fuente

Puedes descargarlo desde: <https://github.com/erodriguez-minsal/SIIG/tarball/master> o clonar el repositorio

```
$ git clone https://github.com/erodriguez-minsal/SIIG.git siig
```

A partir de este punto todos los comandos se deben ejecutar dentro de la carpeta en que se ha descargado el código fuente

### 2.4.3 Instalar composer

Composer (<http://getcomposer.org/>) Es una librería de PHP para el manejo de dependencias. Para instalarlo, dentro de la carpeta donde descargaste el código fuente se debe ejecutar:

```
$ curl -s https://getcomposer.org/installer | php
```

### 2.4.4 Instalar todas las librerías necesarias

```
$ php composer.phar install
```

Al finalizar la instalación, se solicitará los parámetros de conexión a la base de datos, se deben ingresar los valores correspondientes.

## 2.5 Configuración

### 2.5.1 Servidor web

Esto es para una instalación de prueba en una máquina local, la instalación real en un servidor el administrador de servicios deberá realizar esta configuración con los parámetros más adecuados: ip, dominio, configuración en el DNS, etc.

### 2.5.1.1 Configurar un VirtualHost

Creamos el archivo para la definición del VirtualHost

```
# nano /etc/apache2/sites-available/siig.localhost
```

El contenido será similar a esto:

```
<VirtualHost 127.0.0.7>

    ServerName siig.localhost
    DocumentRoot /ruta_al_directorio_descargado/web

    <Directory /ruta_al_directorio_descargado/web >
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/siig-error.localhost.log
    # Possible values include: debug, info, notice, warn, error,
crit,
    # alert, emerg.
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/siig-access.localhost.log combined
</VirtualHost>
```

En el archivo /etc/hosts agregamos la línea

```
127.0.0.7          siig.localhost
```

Habilitamos el VirtualHost

```
# a2ensite siig.localhost
```

También es recomendable activar el módulo mod\_rewrite

```
# a2enmod rewrite
```

Reiniciar apache

```
# /etc/init.d/apache2 restart
```

## 2.5.2 Permisos sobre carpetas

Es necesario tener soporte para ACL (<https://help.ubuntu.com/community/FilePermissionsACLs>) en la partición en que está el proyecto y luego ejecutar

```
$ setfacl -R -m u:www-data:rwX -m u:`whoami`:rwX app/cache app/
logs web/uploads
$ setfacl -dR -m u:www-data:rwX -m u:`whoami`:rwX app/cache app/
logs web/uploads
```

- Modificar nombre\_usuario por un usuario del sistema con que se modificará el código fuente

## 2.5.3 Verificar la configuración

Entra a la siguiente dirección desde el navegador <http://siig.localhost/config.php> Si aparece algún error debe ser corregido antes de continuar

## 2.6 2. Instalación de Postgres

### 2.6.1 Crear la base de datos

```
$ app/console doctrine:database:create
$ app/console doctrine:schema:update --force
```

### 2.6.2 Cargar datos iniciales

```
$ app/console doctrine:fixtures:load
```

### 2.6.3 Crear un usuario administrador

```
$ app/console fos:user:create --super-admin
```

### 2.6.4 Instalación de HStore

HStore (<http://www.postgresql.org/docs/9.1/static/hstore.html>) es un tipo especial de campo de PostgreSQL

- Ejecutar dentro de la base de datos, con el usuario postgres

```
create extension hstore;
```

- Crear la tabla especial que no se manejará con el ORM, hacerlo con el usuario dueño de la base de datos
- (no con el usuario postgres, a menos que este mismo sea el dueño de la base de datos)

```
CREATE TABLE fila_origen_dato(  
    id serial,  
    id_origen_dato integer,  
    datos hstore,  
  
    PRIMARY KEY (id),  
    FOREIGN KEY (id_origen_dato) REFERENCES origen_datos(id) on update  
    CASCADE on delete CASCADE  
  
);
```

## 2.7 3. Instalación de RabbitMQ

### 2.7.1 Instalación de RabbitMQ

RabbitMQ (<http://www.rabbitmq.com/>) es un sistema de mensajería empresarial completo y altamente confiable basado en el estándar AMQP. Charla sobre RabbitMQ (<http://www.symfony.es/noticias/2011/07/06/desymfony-2011-reduciendo-el-acoplamiento-entre-aplicaciones-con-rabbitmq/>). En este proyecto será utilizado para la carga masiva de datos y así evitar cuelgues o saturación del servidor.

- Agregar el repositorio

```
# sh -c 'echo "deb http://www.rabbitmq.com/debian/ testing main"  
>> /etc/apt/sources.list'
```

- Agregar la clave pública

```
# wget http://www.rabbitmq.com/rabbitmq-signing-key-public.asc  
# apt-key add rabbitmq-signing-key-public.asc
```

- Ejecutar

```
# apt-get update
```

- Instalar el paquete

```
# apt-get install rabbitmq-server
```

- Verificar que el servicio de rabbitmq esté corriendo

```
# /etc/init.d/rabbitmq-server start
```

- Iniciar las colas

```
$ src/MINSAL/IndicadoresBundle/Util/iniciar_colas.sh
```

Pueden aparecer mensajes de aviso como "/usr/bin/nohup: redirecting stderr to stdout" solo debemos presionar ENTER

- Habilitar la interfaz web de administración

```
# rabbitmq-plugins enable rabbitmq_management
# /etc/init.d/rabbitmq-server restart
```

- Cargar la interfaz web: entrar a la dirección `http://server_name:55672/mgmt/` El usuario por defecto es **guest** y la clave **guest**

## 2.8 4. Instalación de Servidor OLAP Pentaho + Saiku

El sistema utiliza el servidor de inteligencia de negocios Pentaho Edicion Comunidad. Este servidor contiene tres elementos:

A- Un gestor de persistencia (Hibernate) que se conecta a nuestra base de datos

B- El servidor de aplicaciones Java/Tomcat

C- Una aplicacion (SAIKU) para procesar peticiones REST. Este componente permite hacer consultas al cubo y mostrar resultados usando un URL desde AJAX.

1- Instalar Java y soporte de Postgres:

```
apt-get install openjdk-6-jre libpg-java
```

2- Descargar la ultima version del servidor de Pentaho en:

[http://community.pentaho.com/projects/bi\\_platform/](http://community.pentaho.com/projects/bi_platform/)

Descomprimir el archivo en la carpeta que elijamos, Ejem: `/home/siig/biserver-ce/`

Configurar la base de datos, editar `/home/siig/biserver-ce/tomcat/webapps/hibernate.properties`:



```
hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect hiberna-
te.connection.driver_class = org.postgresql.Driver hiberna-
te.connection.url = jdbc:postgresql:NOMBRE_DE_BASE_DEDATOS hiberna-
te.connection.username = USUARIO hibernate.connection.password = PASS-
WORD hibernate.hbm2ddl.auto = update
```

3- Permitir usuarios anonimos, remover la seguridad interna de Pentaho segun la docu-  
mentación:

[http://infocenter.pentaho.com/help/in-  
dex.jsp?topic=%2Fsecurity\\_guide%2Ftask\\_removing\\_security.html](http://infocenter.pentaho.com/help/index.jsp?topic=%2Fsecurity_guide%2Ftask_removing_security.html)

Iniciar el servidor: `./start-pentaho.sh`

En este punto deberíamos poder abrir la aplicación sin usar credenciales usando dirección  
del servidor:

<http://myservidor:8080/pentaho>

Los errores del sistema son registrados en:

`/opt/biserver-ce/tomcat/logs/pentaho.log /opt/biserver-ce/tomcat/logs/catalina.out`

Después de reiniciar Apache, podemos usar la nueva dirección del servidor:

<http://myservidor:8080/pentaho>

4- Descargar la ultima versión de SAIKU-UI:

<http://analytical-labs.com/downloads.php>

Copiar el archivo descomprimido en `/opt/biserver-ce/pentaho-solutions/system/`. Luego  
copiar y ejecutar el instalador de librerías de CTOOLS disponible en:

<https://github.com/pmalves/ctools-installer>

Para ejecutarlo, es necesario indicar la ubicacion de la instalacion: `./ctools-installer.sh -s  
/opt/biserver-ce/pentaho-solutions`

Reiniciar Pentaho: `./stop-pentaho.sh ./start-pentaho.sh`

5- Activar el proxy de Apache/Esconder el Puerto de Pentaho

Activar módulos de Apache: `a2enmod proxy proxy_http`

editar la seccion VirtualHost dentro de /etc/apache2/sites-enabled/000-default:

```
<Location /saiku/> ProxyPass http://localhost:8080/pentaho/content/
saiku/ ProxyPassReverse http://localhost:8080/pentaho/content/saiku/
SetEnv proxy-chain-auth </Location>
```

En este punto ya tenemos SAIKU disponible en:

<http://myserver/saiku-ui/index.html?biplugin=true>

6- Agregar definición de cubos usando la plantilla para indicadores del MINSAL:

<https://github.com/erodriguez-minsal/SIIG/wiki/PlantillaIndicadorOLAP>

7- Agregar el nuevo indicador al catalogo de cubos dentro de:

[biserver-ce/pentaho-solutions/system/olap/datasources.xml](#)

El servidor OLAP/Pentaho puede ser consultado a traves de SAIKU usando su API HTTP/REST. Esta API permite obtener informacion sobre los cubos existentes en el servidor OLAP asi como efectuar consultas. La documentacion de la API puede ser consultada en:

<http://dev.analytical-labs.com/saiku/serverdocs/>

## 2.9 Instalación de librería wkhtmltopdf

wkhtmltopdf (<http://code.google.com/p/wkhtmltopdf/>) Es una utilidad de línea de comando para convertir html a pdf

1. Descargar wkhtmltopdf desde <http://code.google.com/p/wkhtmltopdf/downloads/> list elegir la versión adecuada al sistema operativo
2. Descomprimir. Ej.: `tar xjf wkhtmltopdf-0.11.0_rc1-static-amd64.tar.bz2`
3. Mover y renombrar el archivo: `mv wkhtmltopdf-amd64 /usr/bin/wkhtmltopdf`
4. Dar permisos de ejecución: `chmod +x /usr/bin/wkhtmltopdf`

## 2.10 Validación de Usuarios desde directorios LDAP

Si un usuario aun no esta creado dentro del sistema, se hara una busqueda en el directorio LDAP especificado en el archivo `app/config/config.yml`. A continuacion se muestran las lineas relevantes para especificar que directorio usar:

```
```.yml
```

# Capítulo 2 Fr3d\_LDAP

fr3d\_ldap: driver: host: 10.10.20.2 # IP del Servidor LDAP Zimbra/Minsal port: 389 # Opcional user: baseDn: ou=people,dc=salud,dc=gob,dc=sv # contenedor de usuarios filter: (objectClass=organizationalPerson) # esquema comun para todos los usuarios del directorio

## 2.0.1 Cargar la aplicación

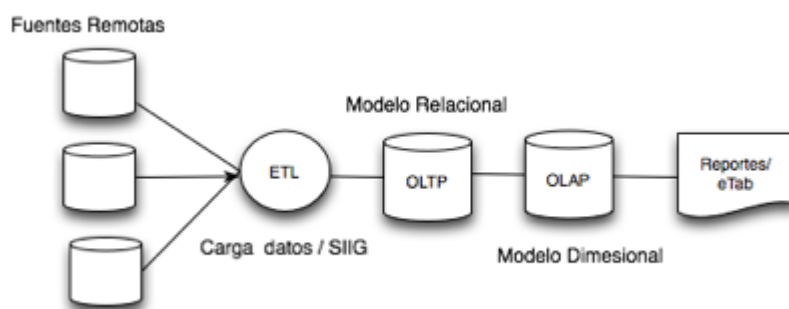
En este punto estamos listos para cargar la aplicación desde: [http://siig.localhost/app\\_dev.php](http://siig.localhost/app_dev.php)

Esta página se ha dejado vacía a propósito

## Capítulo 3

# Modelo De Datos

### 3.0.1 Esquema general de la Aplicacion



**Figura 3.1** Esquema de la aplicación

Los datos que maneja el sistema provienen de distintas fuentes y son de una naturaleza tal que es necesario utilizar el modelo de base datos sin esquema/genérico EAV. Las tabla EAV (Fila\_origen\_dato) y demás tablas auxiliares son parte del almacenamiento de datos transaccional (OLTP) de la aplicación. Esto facilita el manejo de datos de cualquier indicador sin importar sus propiedades. Los cubos de análisis multidimensional (OLAP) son generados usando estos valores genéricos y estan descritos en la seccion de Gestion de Cubos OLAP. Las tablas de los cubos OLAP usan un esquema de estrella mientras que las tablas del almacenamiento OLTP usan un modelo relacional. El Siguiente Diccionario de Datos y Diagrama ER describen la estructura del almacenamiento transaccional (OLTP) de la Aplicacion.

## 3.0.2 Diagrama Entidad Relacion

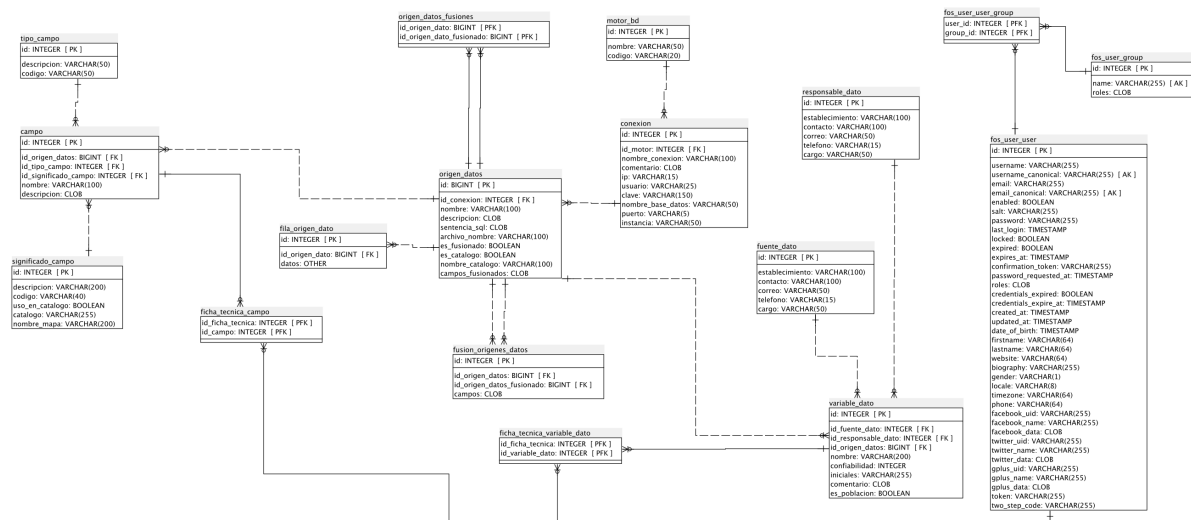


Figura 3.2 Diagrama ER1

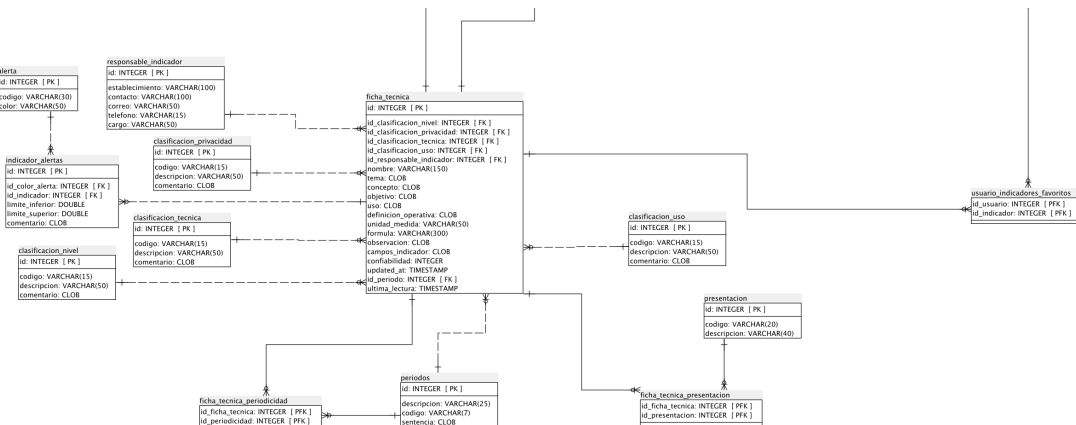


Figura 3.3 Diagrama ER2

## 3.0.3 Diccionario de Datos

### 3.0.3.1 Lista de tablas

- alerta (#alerta)
- campo (#campo)
- clasificacion\_nivel (#clasificacion\_nivel)
- clasificacion\_privacidad (#clasificacion\_privacidad)
- clasificacion\_tecnica (#clasificacion\_tecnica)
- clasificacion\_uso (#clasificacion\_uso)
- conexion (#conexion)
- ficha\_tecnica (#ficha\_tecnica)

- ficha\_tecnica\_campo (#ficha\_tecnica\_campo)
- ficha\_tecnica\_periodicidad (#ficha\_tecnica\_periodicidad)
- ficha\_tecnica\_presentacion (#ficha\_tecnica\_presentacion)
- ficha\_tecnica\_variable\_dato (#ficha\_tecnica\_variable\_dato)
- fila\_origen\_dato (#fila\_origen\_dato)
- fos\_user\_group (#fos\_user\_group)
- fos\_user\_user (#fos\_user\_user)
- fos\_user\_user\_group (#fos\_user\_user\_group)
- fuelle\_dato (#fuelle\_dato)
- fusion\_origenes\_datos (#fusion\_origenes\_datos)
- indicador\_alertas (#indicador\_alertas)
- motor\_bd (#motor\_bd)
- origen\_datos (#origen\_datos)
- origen\_datos\_fusiones (#origen\_datos\_fusiones)
- periodos (#periodos)
- presentacion (#presentacion)
- responsable\_dato (#responsable\_dato)
- responsable\_indicador (#responsable\_indicador)
- significado\_campo (#significado\_campo)
- tipo\_campo (#tipo\_campo)
- usuario\_indicadores\_favoritos (#usuario\_indicadores\_favoritos)
- variable\_dato (#variable\_dato)

## 1. alerta ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| codigo                  | codigo                   | VARCHAR(30) |    | NOT NULL |
| color                   | color                    | VARCHAR(50) |    | NOT NULL |

Esta tabla es usada por:

<

ul>

- indicador\_alertas (#indicador\_alertas) hace referencia la campo (id)

## 2. campo ()

| Nomre Logico de Columna                           | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|---------------------------------------------------|--------------------------|--------------|----|----------|
| id (PK)                                           | id                       | INTEGER      | PK | NOT NULL |
| id_origen_datos (FK<br>(#origen_datos))           | id_origen_datos          | BIGINT       |    | NOT NULL |
| id_tipo_campo (FK<br>(#tipo_campo))               | id_tipo_campo            | INTEGER      |    | NOT NULL |
| id_significado_campo (FK<br>(#significado_campo)) | id_significado_campo     | INTEGER      |    | NOT NULL |
| nombre                                            | nombre                   | VARCHAR(100) |    | NOT NULL |
| descripcion                                       | descripcion              | CLOB         |    |          |

Esta tabla depende de:

- origen\_datos (#origen\_datos) por medio de (id\_origen\_datos)
- tipo\_campo (#tipo\_campo) por medio de (id\_tipo\_campo)
- significado\_campo (#significado\_campo) por medio de (id\_significado\_campo)

Esta tabla es usada por:

- ficha\_tecnica\_campo (#ficha\_tecnica\_campo) hace referencia la campo (id)

## 3. clasificacion\_nivel ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|-------------------------|--------------------------|---------|----|----------|
| id (PK)                 | id                       | INTEGER | PK | NOT NULL |



|             |             |             |  |          |
|-------------|-------------|-------------|--|----------|
| codigo      | codigo      | VARCHAR(15) |  | NOT NULL |
| descripcion | descripcion | VARCHAR(50) |  | NOT NULL |
| comentario  | comentario  | CLOB        |  |          |

Esta tabla es usada por:

- ficha\_tecnica (#ficha\_tecnica) hace referencia la campo (id)

#### 4. clasificacion\_privacidad ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| codigo                  | codigo                   | VARCHAR(15) |    | NOT NULL |
| descripcion             | descripcion              | VARCHAR(50) |    | NOT NULL |
| comentario              | comentario               | CLOB        |    |          |

Esta tabla es usada por:

- ficha\_tecnica (#ficha\_tecnica) hace referencia la campo (id)

#### 5. clasificacion\_tecnica ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| codigo                  | codigo                   | VARCHAR(15) |    | NOT NULL |
| descripcion             | descripcion              | VARCHAR(50) |    | NOT NULL |
| comentario              | comentario               | CLOB        |    |          |

Esta tabla es usada por:

- ficha\_tecnica (#ficha\_tecnica) hace referencia la campo (id)

## 6. clasificacion\_uso ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| codigo                  | codigo                   | VARCHAR(15) |    | NOT NULL |
| descripcion             | descripcion              | VARCHAR(50) |    | NOT NULL |
| comentario              | comentario               | CLOB        |    |          |

Esta tabla es usada por:

- `ficha_tecnica (#ficha_tecnica)` hace referencia la campo (id)

## 7. conexion ()

| Nomre Logico de Columna   | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|---------------------------|--------------------------|--------------|----|----------|
| id (PK)                   | id                       | INTEGER      | PK | NOT NULL |
| id_motor (FK (#motor_bd)) | id_motor                 | INTEGER      |    | NOT NULL |
| nombre_conexion           | nombre_conexion          | VARCHAR(100) |    | NOT NULL |
| comentario                | comentario               | CLOB         |    |          |
| ip                        | ip                       | VARCHAR(15)  |    | NOT NULL |
| usuario                   | usuario                  | VARCHAR(25)  |    | NOT NULL |
| clave                     | clave                    | VARCHAR(150) |    | NOT NULL |
| nombre_base_datos         | nombre_base_datos        | VARCHAR(50)  |    | NOT NULL |
| puerto                    | puerto                   | VARCHAR(5)   |    |          |
| instancia                 | instancia                | VARCHAR(50)  |    |          |

Esta tabla depende de:

- `motor_bd (#motor_bd)` por medio de (id\_motor)

Esta tabla es usada por:

- origen\_datos (#origen\_datos) hace referencia la campo (id)

## 8. ficha\_tecnica ()

| Nomre Logico de Columna                                             | Nombre Fisico de Columna    | Tipo         | PK | Nullable |
|---------------------------------------------------------------------|-----------------------------|--------------|----|----------|
| id (PK)                                                             | id                          | INTEGER      | PK | NOT NULL |
| id_clasificacion_nivel (FK<br>(#clasificacion_nivel))               | id_clasificacion_nivel      | INTEGER      |    | NOT NULL |
| id_clasificacion_privacidad (FK<br>(#clasificacion_privacidad)<br>) | id_clasificacion_privacidad | INTEGER      |    | NOT NULL |
| id_clasificacion_tecnica (FK<br>(#clasificacion_tecnica))           | id_clasificacion_tecnica    | INTEGER      |    | NOT NULL |
| id_clasificacion_uso (FK<br>(#clasificacion_uso))                   | id_clasificacion_uso        | INTEGER      |    | NOT NULL |
| id_responsable_indicador (FK<br>(#responsable_indicador))           | id_responsable_indicador    | INTEGER      |    | NOT NULL |
| nombre                                                              | nombre                      | VARCHAR(150) |    | NOT NULL |
| tema                                                                | tema                        | CLOB         |    | NOT NULL |
| concepto                                                            | concepto                    | CLOB         |    |          |
| objetivo                                                            | objetivo                    | CLOB         |    |          |
| uso                                                                 | uso                         | CLOB         |    |          |
| definicion_operativa                                                | definicion_operativa        | CLOB         |    |          |
| unidad_medida                                                       | unidad_medida               | VARCHAR(50)  |    | NOT NULL |
| formula                                                             | formula                     | VARCHAR(300) |    | NOT NULL |
| observacion                                                         | observacion                 | CLOB         |    |          |
| campos_indicador                                                    | campos_indicador            | CLOB         |    |          |
| confiabilidad                                                       | confiabilidad               | INTEGER      |    |          |

|                                 |                |           |  |          |
|---------------------------------|----------------|-----------|--|----------|
| updated_at                      | updated_at     | TIMESTAMP |  |          |
| id_periodo (FK (#periodos)<br>) | id_periodo     | INTEGER   |  | NOT NULL |
| ultima_lectura                  | ultima_lectura | TIMESTAMP |  |          |

Esta tabla depende de:

- periodos (#periodos) por medio de (id\_periodo)
- clasificacion\_privacidad (#clasificacion\_privacidad) por medio de (id\_clasificacion\_privacidad)
- clasificacion\_tecnica (#clasificacion\_tecnica) por medio de (id\_clasificacion\_tecnica)
- clasificacion\_uso (#clasificacion\_uso) por medio de (id\_clasificacion\_uso)
- clasificacion\_nivel (#clasificacion\_nivel) por medio de (id\_clasificacion\_nivel)
- responsable\_indicador (#responsable\_indicador) por medio de (id\_responsable\_indicador)

Esta tabla es usada por:

- ficha\_tecnica\_periodicidad (#ficha\_tecnica\_periodicidad) hace referencia la campo (id)
- ficha\_tecnica\_campo (#ficha\_tecnica\_campo) hace referencia la campo (id)
- ficha\_tecnica\_presentacion (#ficha\_tecnica\_presentacion) hace referencia la campo (id)
- indicador\_alertas (#indicador\_alertas) hace referencia la campo (id)
- ficha\_tecnica\_variable\_dato (#ficha\_tecnica\_variable\_dato) hace referencia la campo (id)
- usuario\_indicadores\_favoritos (#usuario\_indicadores\_favoritos) hace referencia la campo (id)

## 9. ficha\_tecnica\_campo ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo | PK | Nullable |
|-------------------------|--------------------------|------|----|----------|
|-------------------------|--------------------------|------|----|----------|

|                                             |                  |         |    |          |
|---------------------------------------------|------------------|---------|----|----------|
| id_ficha_tecnica (PK) (FK (#ficha_tecnica)) | id_ficha_tecnica | INTEGER | PK | NOT NULL |
| id_campo (PK) (FK (#campo))                 | id_campo         | INTEGER | PK | NOT NULL |

Esta tabla depende de:

- campo (#campo) por medio de (id\_campo)
- ficha\_tecnica (#ficha\_tecnica) por medio de (id\_ficha\_tecnica)

## 10. ficha\_tecnica\_periodicidad ()

| Nomre Logico de Columna                     | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|---------------------------------------------|--------------------------|---------|----|----------|
| id_ficha_tecnica (PK) (FK (#ficha_tecnica)) | id_ficha_tecnica         | INTEGER | PK | NOT NULL |
| id_periodicidad (PK) (FK (#periodos))       | id_periodicidad          | INTEGER | PK | NOT NULL |

Esta tabla depende de:

- periodos (#periodos) por medio de (id\_periodicidad)
- ficha\_tecnica (#ficha\_tecnica) por medio de (id\_ficha\_tecnica)

## 11. ficha\_tecnica\_presentacion ()

| Nomre Logico de Columna                     | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|---------------------------------------------|--------------------------|---------|----|----------|
| id_ficha_tecnica (PK) (FK (#ficha_tecnica)) | id_ficha_tecnica         | INTEGER | PK | NOT NULL |
| id_presentacion (PK) (FK (#presentacion))   | id_presentacion          | INTEGER | PK | NOT NULL |

Esta tabla depende de:

- ficha\_tecnica (#ficha\_tecnica) por medio de (id\_ficha\_tecnica)
- presentacion (#presentacion) por medio de (id\_presentacion)

## 12. ficha\_tecnica\_variable\_dato ()

| Nomre Logico de Columna                     | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|---------------------------------------------|--------------------------|---------|----|----------|
| id_ficha_tecnica (PK) (FK (#ficha_tecnica)) | id_ficha_tecnica         | INTEGER | PK | NOT NULL |
| id_variable_dato (PK) (FK (#variable_dato)) | id_variable_dato         | INTEGER | PK | NOT NULL |

Esta tabla depende de:

- ficha\_tecnica (#ficha\_tecnica) por medio de (id\_ficha\_tecnica)
- variable\_dato (#variable\_dato) por medio de (id\_variable\_dato)

## 13. fila\_origen\_dato ()

| Nomre Logico de Columna             | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|-------------------------------------|--------------------------|---------|----|----------|
| id (PK)                             | id                       | INTEGER | PK | NOT NULL |
| id_origen_dato (FK (#origen_datos)) | id_origen_dato           | BIGINT  |    | NOT NULL |
| datos                               | datos                    | [1111]  |    |          |

Esta tabla depende de:

- origen\_datos (#origen\_datos) por medio de (id\_origen\_dato)

## 14. fos\_user\_group ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| name                    | name                     | VARCHAR(255) |    | NOT NULL |
| roles                   | roles                    | CLOB         |    | NOT NULL |

(DC2Tipo:array)

Esta tabla es usada por:

- fos\_user\_user\_group (#fos\_user\_user\_group) hace referencia la campo (id)

## 15. fos\_user\_user ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| username                | username                 | VARCHAR(255) |    | NOT NULL |
| username_canonical      | username_canonical       | VARCHAR(255) |    | NOT NULL |
| email                   | email                    | VARCHAR(255) |    | NOT NULL |
| email_canonical         | email_canonical          | VARCHAR(255) |    | NOT NULL |
| enabled                 | enabled                  | BOOLEAN      |    | NOT NULL |
| salt                    | salt                     | VARCHAR(255) |    | NOT NULL |
| password                | password                 | VARCHAR(255) |    | NOT NULL |
| last_login              | last_login               | TIMESTAMP    |    |          |
| locked                  | locked                   | BOOLEAN      |    | NOT NULL |
| expired                 | expired                  | BOOLEAN      |    | NOT NULL |
| expires_at              | expires_at               | TIMESTAMP    |    |          |
| confirmation_token      | confirmation_token       | VARCHAR(255) |    |          |
| password_requested_at   | password_requested_at    | TIMESTAMP    |    |          |
| roles                   | roles                    | CLOB         |    | NOT NULL |
| credentials_expired     | credentials_expired      | BOOLEAN      |    | NOT NULL |
| credentials_expire_at   | credentials_expire_at    | TIMESTAMP    |    |          |
| created_at              | created_at               | TIMESTAMP    |    | NOT NULL |
| updated_at              | updated_at               | TIMESTAMP    |    | NOT NULL |
| date_of_birth           | date_of_birth            | TIMESTAMP    |    |          |
| firstname               | firstname                | VARCHAR(64)  |    |          |
| lastname                | lastname                 | VARCHAR(64)  |    |          |
| website                 | website                  | VARCHAR(64)  |    |          |

|               |               |              |  |  |
|---------------|---------------|--------------|--|--|
| biography     | biography     | VARCHAR(255) |  |  |
| gender        | gender        | VARCHAR(1)   |  |  |
| locale        | locale        | VARCHAR(8)   |  |  |
| timezone      | timezone      | VARCHAR(64)  |  |  |
| phone         | phone         | VARCHAR(64)  |  |  |
| facebook_uid  | facebook_uid  | VARCHAR(255) |  |  |
| facebook_name | facebook_name | VARCHAR(255) |  |  |
| facebook_data | facebook_data | CLOB         |  |  |
| twitter_uid   | twitter_uid   | VARCHAR(255) |  |  |
| twitter_name  | twitter_name  | VARCHAR(255) |  |  |
| twitter_data  | twitter_data  | CLOB         |  |  |
| gplus_uid     | gplus_uid     | VARCHAR(255) |  |  |
| gplus_name    | gplus_name    | VARCHAR(255) |  |  |
| gplus_data    | gplus_data    | CLOB         |  |  |
| token         | token         | VARCHAR(255) |  |  |
| two_step_code | two_step_code | VARCHAR(255) |  |  |

Esta tabla es usada por:

- fos\_user\_user\_group (#fos\_user\_user\_group) hace referencia la campo (id)
- usuario\_indicadores\_favoritos (#usuario\_indicadores\_favoritos) hace referencia la campo (id)

## 16. fos\_user\_user\_group ()

| Nomre Logico de Columna                 | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|-----------------------------------------|--------------------------|---------|----|----------|
| user_id (PK) (FK<br>(#fos_user_user))   | user_id                  | INTEGER | PK | NOT NULL |
| group_id (PK) (FK<br>(#fos_user_group)) | group_id                 | INTEGER | PK | NOT NULL |



Esta tabla depende de:

- `fos_user_group` (`#fos_user_group`) por medio de (`group_id`)
- `fos_user_user` (`#fos_user_user`) por medio de (`user_id`)

## 17. fuente\_dato ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| establecimiento         | establecimiento          | VARCHAR(100) |    | NOT NULL |
| contacto                | contacto                 | VARCHAR(100) |    | NOT NULL |
| correo                  | correo                   | VARCHAR(50)  |    | NOT NULL |
| telefono                | telefono                 | VARCHAR(15)  |    | NOT NULL |
| cargo                   | cargo                    | VARCHAR(50)  |    | NOT NULL |

Esta tabla es usada por:

- `variable_dato` (`#variable_dato`) hace referencia la campo (`id`)

## 18. fusion\_origenes\_datos ()

| Nomre Logico de Columna                                          | Nombre Fisico de Columna  | Tipo    | PK | Nullable |
|------------------------------------------------------------------|---------------------------|---------|----|----------|
| id (PK)                                                          | id                        | INTEGER | PK | NOT NULL |
| id_origen_datos (FK<br>( <code>#origen_datos</code> ))           | id_origen_datos           | BIGINT  |    | NOT NULL |
| id_origen_datos_fusionado<br>(FK ( <code>#origen_datos</code> )) | id_origen_datos_fusionado | BIGINT  |    | NOT NULL |
| campos                                                           | campos                    | CLOB    |    |          |

Esta tabla depende de:

- `origen_datos` (`#origen_datos`) por medio de (`id_origen_datos`)
- `origen_datos` (`#origen_datos`) por medio de (`id_origen_datos_fusionado`)

## 19. indicador\_alertas ()

| Nomre Logico de Columna            | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|------------------------------------|--------------------------|---------|----|----------|
| id (PK)                            | id                       | INTEGER | PK | NOT NULL |
| id_color_alerta (FK (#alerta))     | id_color_alerta          | INTEGER |    | NOT NULL |
| id_indicador (FK (#ficha_tecnica)) | id_indicador             | INTEGER |    | NOT NULL |
| limite_inferior                    | limite_inferior          | DOUBLE  |    | NOT NULL |
| limite_superior                    | limite_superior          | DOUBLE  |    | NOT NULL |
| comentario                         | comentario               | CLOB    |    |          |

Esta tabla depende de:

- alerta (#alerta) por medio de (id\_color\_alerta)
- ficha\_tecnica (#ficha\_tecnica) por medio de (id\_indicador)

## 20. motor\_bd ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| nombre                  | nombre                   | VARCHAR(50) |    | NOT NULL |
| codigo                  | codigo                   | VARCHAR(20) |    |          |

Esta tabla es usada por:

- conexion (#conexion) hace referencia la campo (id)

## 21. origen\_datos ()

| Nomre Logico de Columna      | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|------------------------------|--------------------------|---------|----|----------|
| id (PK)                      | id                       | BIGINT  | PK | NOT NULL |
| id_conexion (FK (#conexion)) | id_conexion              | INTEGER |    | NOT NULL |

|                   |                   |              |  |          |
|-------------------|-------------------|--------------|--|----------|
| nombre            | nombre            | VARCHAR(100) |  | NOT NULL |
| descripcion       | descripcion       | CLOB         |  |          |
| sentencia_sql     | sentencia_sql     | CLOB         |  |          |
| archivo_nombre    | archivo_nombre    | VARCHAR(100) |  |          |
| es_fusionado      | es_fusionado      | BOOLEAN      |  |          |
| es_catalogo       | es_catalogo       | BOOLEAN      |  |          |
| nombre_catalogo   | nombre_catalogo   | VARCHAR(100) |  |          |
| campos_fusionados | campos_fusionados | CLOB         |  |          |

Esta tabla depende de:

- conexion (#conexion) por medio de (id\_conexion)

Esta tabla es usada por:

- campo (#campo) hace referencia la campo (id)
- origen\_datos\_fusiones (#origen\_datos\_fusiones) hace referencia la campo (id)
- origen\_datos\_fusiones (#origen\_datos\_fusiones) hace referencia la campo (id)
- variable\_dato (#variable\_dato) hace referencia la campo (id)
- fila\_origen\_dato (#fila\_origen\_dato) hace referencia la campo (id)
- fusion\_origenes\_datos (#fusion\_origenes\_datos) hace referencia la campo (id)
- fusion\_origenes\_datos (#fusion\_origenes\_datos) hace referencia la campo (id)

## 22. origen\_datos\_fusiones ()

| Nomre Logico de Columna                                     | Nombre Fisico de Columna | Tipo   | PK | Nullable |
|-------------------------------------------------------------|--------------------------|--------|----|----------|
| id_origen_dato (PK) ( <u>FK (#origen_datos)</u> )           | id_origen_dato           | BIGINT | PK | NOT NULL |
| id_origen_dato_fusionado (PK) ( <u>FK (#origen_datos)</u> ) | id_origen_dato_fusionado | BIGINT | PK | NOT NULL |

Esta tabla depende de:

- origen\_datos (#origen\_datos) por medio de (id\_origen\_dato\_fusionado)
- origen\_datos (#origen\_datos) por medio de (id\_origen\_dato)

## 23. periodos ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| descripcion             | descripcion              | VARCHAR(25) |    | NOT NULL |
| codigo                  | codigo                   | VARCHAR(7)  |    | NOT NULL |
| sentencia               | sentencia                | CLOB        |    |          |

Esta tabla es usada por:

- ficha\_tecnica\_periodicidad (#ficha\_tecnica\_periodicidad) hace referencia la campo (id)
- ficha\_tecnica (#ficha\_tecnica) hace referencia la campo (id)

## 24. presentacion ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| codigo                  | codigo                   | VARCHAR(20) |    | NOT NULL |
| descripcion             | descripcion              | VARCHAR(40) |    | NOT NULL |

Esta tabla es usada por:

- ficha\_tecnica\_presentacion (#ficha\_tecnica\_presentacion) hace referencia la campo (id)

## 25. responsable\_dato ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| establecimiento         | establecimiento          | VARCHAR(100) |    | NOT NULL |

|          |          |              |  |          |
|----------|----------|--------------|--|----------|
| contacto | contacto | VARCHAR(100) |  | NOT NULL |
| correo   | correo   | VARCHAR(50)  |  | NOT NULL |
| telefono | telefono | VARCHAR(15)  |  | NOT NULL |
| cargo    | cargo    | VARCHAR(50)  |  | NOT NULL |

Esta tabla es usada por:

- variable\_dato (#variable\_dato) hace referencia la campo (id)

## 26. responsable\_indicador ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| establecimiento         | establecimiento          | VARCHAR(100) |    | NOT NULL |
| contacto                | contacto                 | VARCHAR(100) |    | NOT NULL |
| correo                  | correo                   | VARCHAR(50)  |    | NOT NULL |
| telefono                | telefono                 | VARCHAR(15)  |    | NOT NULL |
| cargo                   | cargo                    | VARCHAR(50)  |    | NOT NULL |

Esta tabla es usada por:

- ficha\_tecnica (#ficha\_tecnica) hace referencia la campo (id)

## 27. significado\_campo ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo         | PK | Nullable |
|-------------------------|--------------------------|--------------|----|----------|
| id (PK)                 | id                       | INTEGER      | PK | NOT NULL |
| descripcion             | descripcion              | VARCHAR(200) |    | NOT NULL |
| codigo                  | codigo                   | VARCHAR(40)  |    | NOT NULL |
| uso_en_catalogo         | uso_en_catalogo          | BOOLEAN      |    |          |
| catalogo                | catalogo                 | VARCHAR(255) |    |          |
| nombre_mapa             | nombre_mapa              | VARCHAR(200) |    |          |

Esta tabla es usada por:

- `campo (#campo)` hace referencia la campo (id)

## 28. tipo\_campo ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo        | PK | Nullable |
|-------------------------|--------------------------|-------------|----|----------|
| id (PK)                 | id                       | INTEGER     | PK | NOT NULL |
| descripcion             | descripcion              | VARCHAR(50) |    |          |
| codigo                  | codigo                   | VARCHAR(50) |    | NOT NULL |

Esta tabla es usada por:

- `campo (#campo)` hace referencia la campo (id)

## 29. usuario\_indicadores\_favoritos ()

| Nomre Logico de Columna                 | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|-----------------------------------------|--------------------------|---------|----|----------|
| id_usuario (PK) (FK (#fos_user_user))   | id_usuario               | INTEGER | PK | NOT NULL |
| id_indicador (PK) (FK (#ficha_tecnica)) | id_indicador             | INTEGER | PK | NOT NULL |

Esta tabla depende de:

```
<ul>
  <li> <a href="#ficha_tecnica">ficha_tecnica</a> por medio de
(id_indicador)</li>
  <li> <a href="#fos_user_user">fos_user_user</a> por medio de
(id_usuario)</li>
</ul>
```

## 29. variable\_dato ()

| Nomre Logico de Columna | Nombre Fisico de Columna | Tipo    | PK | Nullable |
|-------------------------|--------------------------|---------|----|----------|
| id (PK)                 | id                       | INTEGER | PK | NOT NULL |

|                                                 |                     |              |  |          |
|-------------------------------------------------|---------------------|--------------|--|----------|
| id_fuente_dato (FK<br>(#fuente_dato))           | id_fuente_dato      | INTEGER      |  | NOT NULL |
| id_responsable_dato (FK<br>(#responsable_dato)) | id_responsable_dato | INTEGER      |  | NOT NULL |
| id_origen_datos (FK<br>(#origen_datos))         | id_origen_datos     | BIGINT       |  | NOT NULL |
| nombre                                          | nombre              | VARCHAR(200) |  | NOT NULL |
| confiabilidad                                   | confiabilidad       | INTEGER      |  |          |
| iniciales                                       | iniciales           | VARCHAR(255) |  | NOT NULL |
| comentario                                      | comentario          | CLOB         |  |          |
| es_poblacion                                    | es_poblacion        | BOOLEAN      |  |          |

Esta tabla depende de:

- origen\_datos (#origen\_datos) por medio de (id\_origen\_datos)
- fuente\_dato (#fuente\_dato) por medio de (id\_fuente\_dato)
- responsable\_dato (#responsable\_dato) por medio de (id\_responsable\_dato)

Esta tabla es usada por:

- ficha\_tecnica\_variable\_dato (#ficha\_tecnica\_variable\_dato) hace referencia la campo (id)

<

h3>

Esta página se ha dejado vacía a propósito



## Capítulo 4

# Gestión de Cubos OLAP

### 4.0.1 Introducción

La cantidad de reportes que pueden generarse depende de los catálogos que se usen en cada indicador. Al interior de una estructura para reportes (cubo OLAP), cada catalogo se convierte en dimensión. Así por ejemplo el catalogo municipio se convierte en una dimensión dentro de la estructura de reportes y posibilita hacer búsquedas usando cualquier campo que exista dentro de este catálogo. Por ejemplo: Si el catalogo municipio tiene los campos municipio, departamento y región esto nos permitiría hacer las siguientes consultas: numero de casos por municipio, número de casos por departamento, número de casos por región.

La ventaja principal de usar un gestor de cubos OLAP es aislar la lógica de las búsquedas para analizar los datos. De esta forma el sistema se enfoca en presentar al usuario la mayor cantidad de información de forma flexible sin preocuparse de la lógica para obtener los datos.

### 4.0.2 Indicadores y Cubos OLAP

El sistema cuenta con un servidor de gestion de cubos OLAP que se conecta directamente a la base de datos de Indicadores. La definición de cubos esta descrita en el archivo `biserver-ce/pentaho-solutions/system/olap/datasources.xml`, los contenidos de este archivo se muestran a continuacion:

```
```xml
```

```
Provider=Mondrian;DataSource=Pentaho
```

```
Pentaho BI Platform Datasources
```

```
http://localhost:8080/pentaho/Xmla?userid=joe&password=password
```

```
Provider=mondrian
```

```
PentahoXMLA
```

```
MDP
```

```
Unauthenticated
```

```
Provider=mondrian;DataSource=Minsal
```

```
solution:/admin/resources/metadata/indicador23.mondrian.xml
```

```
Provider=mondrian;DataSource=Minsal
```

```
solution:/admin/resources/metadata/indicador24.mondrian.xml
```

```
Provider=mondrian;DataSource=Minsal
```

```
solution:/admin/resources/metadata/indicador25.mondrian.xml
```

```
Provider=mondrian;DataSource=Minsal
```

```
solution:/admin/resources/metadata/indicador7.mondrian.xml
```

Como puede verse en este código cada indicador es un catálogo/cubo cuya descripción está contenida en otro archivo XML. Para facilitar la creación de nuevos cubos a continuación se muestra el código base de un nuevo indicador, esta plantilla está disponible en <https://github.com/erodriguez-minsal/SIIG/wiki/PlantillaIndicadorOLAP>

```
xml
```

```
<!--N.3 - Listado de dimensiones disponibles en este indicador.
```

```
Para cada dimension el formato a seguir es:
```

```
DimensionUsage name=Etiqueta source=DimensionPreDefinida
```

```
foreignKey=ColumnaTablaIndicador-->
```

```
<DimensionUsage name="Departamento" source="Departamento"
```

```
foreignKey="id_departamento"/>
```

```
<DimensionUsage name="Municipio" source="Municipio"
```

```

foreignKey="id_municipio"/>
<DimensionUsage name="Region" source="Region"
foreignKey="id_region"/>
<DimensionUsage source="Tiempo" name="Tiempo" visible="true"
foreignKey="fecha" highCardinality="false"/>

<!--N.5 - Definicion de la formula unidad de medida-->

```

```

'''

```

### 4.0.3 Crear/Actualizar Catalogo Tiempo

Cada indicador/cubo puede utilizar la dimension tiempo, esta dimension es un tabla/catalogo especial que es creada por el administrador del sistema usando la funcion especial crear\_ctl\_tiempo. A continuacion se muestra el codigo de esta funcion:

```

'''postgres CREATE OR REPLACE FUNCTION crear_ctl_tiempo(inicio integer DE-
FAULT 2006, anios integer DEFAULT 8) RETURNS VOID AS $$ DECLARE dias integer;
myquery text; BEGIN dias:=365*$2;

```

```

DROP TABLE IF EXISTS ctl_tiempo;

```

```

myquery:='CREATE TABLE ctl_tiempo AS SELECT * from (SELECT datum AS fecha,
extract(year FROM datum)::int AS Anio, extract(month FROM datum)::int AS Mes,
to_char(datum, "TMMonth")::character(12) AS NombreMes, extract(day FROM da-
tum)::int AS Dia, extract(doy FROM datum)::int AS DiaAnio, to_char(datum, "TM-
Day")::character(12) AS NombreDiaSemana, extract(week FROM datum)::int AS Sema-
naCalendario, to_char(datum, "dd. mm. yyyy")::character(12) AS FechaCorriente, "T" ||
to_char(datum, "Q")::int AS Trimestre, to_char(datum, "yyyy/Q")::character(6) AS Tri-
mestreAnio, to_char(datum, "yyyy/mm")::character(12) AS MesAnio, -- ISO calendar
year and week to_char(datum, "iyyy/IW")::character(8) AS SemanaAnioCalendario, --
Weekend CASE WHEN extract(isodow FROM datum) IN (6, 7) THEN "FinDeSemana"
ELSE "DiaDeSemana" END AS FinDeSemana, -- Feriados para El Salvador CASE WHEN
to_char(datum, "MMDD") IN ("0801", "0802", "0803", "084") THEN "Feriado" ELSE "Dia
Laboral" END AS FeriadoElSalvador, -- Periodos festivos del calendario CASE WHEN
to_char(datum, "MMDD") BETWEEN "0701" AND "0831" THEN "Vacación de Verano"
WHEN to_char(datum, "MMDD") BETWEEN "1115" AND "1225" THEN "Temporada Na-
videña" WHEN to_char(datum, "MMDD") > "1223" OR to_char(datum, "MMDD") <=
"1231" THEN "Vacación Navideña" ELSE "Normal" END AS Periodo, -- Fecha de inicio
de fin de semana datum + (1 - extract(isodow FROM datum))::integer AS IncioSemana,
datum + (7 - extract(isodow FROM datum))::integer AS FinSemana, -- Fecha de inicio
de fin de Mes datum + (1 - extract(day FROM datum))::integer AS InicioMes, (datum +
(1 - extract(day FROM datum))::integer + "1 month"::interval)::date - "1 day"::interval AS

```

```
FinMes FROM ( SELECT "||$1||"-01-01"::DATE + sequence.day AS datum FROM genera-  
te_series(0,||dias||) AS sequence(day) GROUP BY sequence.day ) DQ ORDER BY 1) as foo;;  
EXECUTE myquery;
```

```
ALTER TABLE ctl_tiempo ADD PRIMARY KEY (fecha);
```

```
RAISE NOTICE 'Se creo tabla Tiempo de % anios a partir de %',anios,inicio;
```

```
END; $$ LANGUAGE plpgsql; `` Como se puede ver en el codigo, los intervalos y periodos  
de tiempo (feriados, etc) que se quieran usar para analizar datos pueden ser con-  
figurados al crear esta funcion. Luego de que se ha agregado esta funcion, procedemos a  
crear el catalogo tiempo dentro de la base de datos indicadores:
```

```
postgres=# select * from crear_ctl_tiempo(2008,5);
```

```
NOTICE: Se creo tabla Tiempo de 5 anios a partir de 2008
```

# Lista de figuras

|     |                                |    |
|-----|--------------------------------|----|
| 3.1 | Esquema de la aplicación ..... | 21 |
| 3.2 | Diagrama ER1 .....             | 22 |
| 3.3 | Diagrama ER2 .....             | 22 |

