



Manual Técnico

eTAB

Esta página se ha dejado vacía a propósito

Índice de contenidos

Capítulo 1	Tecnologías utilizadas	7
Capítulo 2	Instalación del eTAB	11
2.1	Requerimientos	11
2.2	Instalación de Symfony2	11
2.3	Configuración	13
2.4	Configuración de Postgres	14
2.5	Instalación de RabbitMQ	16
2.6	Instalación de librería wkhtmltopdf	17
2.7	OPCIONAL: Validación de Usuarios desde directorios LDAP	17
Capítulo 3	Modelo De Datos	19

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 2015 trabajara 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
- 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
- PivotTable.js: Librería para crear tabla pivote
- Redis: Motor de base de datos en memoria.

1.0.1 Gestor de base de datos

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

Versión 9.1 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.

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 Documentación

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

1.0.9 Tabla pivote

Se utiliza la librería PivotTable.js, la cual permite crear tablas pivotes a través de una interfaz sencilla similar a la de las hojas de cálculo, su código y documentación puede consultarse desde la página de su creador [PivotTable.js] (<http://nicolas.kruchten.com/pivot-table/examples/>)

1.0.10 Redis

Redis (<http://redis.io/>) es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como

una base de datos durable o persistente. Se utilizará como memoria caché de consultas de Doctrine y también para almacenar informes, los cuales se ejecutarán y realizarán las consultas normales cuando se carguen por primera vez o cuando se hayan actualizados algunos de sus datos, y se guardará el resultado en Redis; para futuras peticiones a esos informes se tomarán de lo almacenado en Redis lo cual es mucho más rápido.

Capítulo 2

Instalación del eTAB

2.1 Requerimientos

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

2.2 Instalación de Symfony2

2.2.1 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, en ambos casos desde una consola de comandos.

```
# 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 php5-mysql php5-sybase php5-json
postgresql-contrib redis-server
# php5enmod mcrypt
```

2.2.2 Crear usuario y directorio de trabajo

El directorio y usuario a utilizar pueden variar de acuerdo a los que se deseen elegir en cada instalación, como ejemplo se usará un usuario llamado *siig* y el directorio de instalación */var/www/siig*

```
# adduser siig
# mkdir /var/www/siig
# chown siig:siig /var/www/siig
# su siig
$ cd /var/www
```

2.2.3 Obtener el código fuente

Puedes descargarlo desde: <https://github.com/SM2015/SIIG/tarball/master> o clonar el repositorio

```
$ git clone -b v1.0.0 https://github.com/SM2015/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

2.2.4 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
```

2.2.5 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. Más adelante, en la sección "Configuración de PostgreSQL" se muestra un ejemplo de cómo crear un usuario de la base de datos para utilizarlo en el Siig. Además en la sección "Configuración de Mondrian" se explica el significado de las variables: *carpeta_siig_mondrian*, *conexion_bd_pentaho* y *listado_metadata*.

2.3 Configuración

2.3.1 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.

2.3.1.1 Configurar un VirtualHost

Creamos el archivo para la definición del VirtualHost

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

El contenido será similar a esto:

```
<VirtualHost 127.0.0.7>

    ServerName siig.localhost
    DocumentRoot /var/www/siig/web

    <Directory /var/www/siig/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
```

2.3.2 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
```

2.3.3 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

2.4 Configuración de Postgres

2.4.1 Editar archivo de configuración

Como usuario root realizar:

1. Editar el archivo `/etc/postgresql/9.1/main/pg_hba.conf`
2. Cambiar la siguiente línea, sustituir la última palabra por `md5`

```
local    all                                all                                md5
```

Reiniciar PostgreSQL

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

2.4.2 Crear el usuario dueño de la base de datos

Se creará el usuario dueño de la base de datos, las opciones utilizadas dependerán de los criterios que se quieran seguir, se muestra un ejemplo, ejecutar `createuser --help` para la explicación de las opciones. El nombre utilizado y la clave debe corresponder con los parámetros especificados al ejecutar `php composer.phar install` en unas secciones anteriores

```
# su postgres
$ createuser -d -S -R -P admin;
```

Al finalizar presionar la combinación Ctrl+D 2 veces para regresar al usuario siig y continuar con la instalación.

2.4.3 Crear la base de datos

```
$ app/console doctrine:database:create
```

2.4.4 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 estructura de la base de datos

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

- 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_origen_dato integer,
    datos hstore,
    ultima_lectura timestamp,

    FOREIGN KEY (id_origen_dato) REFERENCES origen_datos(id) on update CASCADE on
    delete CASCADE

);
```

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

```
\i [directorio_instalacion]/src/MINSAL/CostosBundle/Resources/estructurasBD/
estructuras.sql
```

```
\i [directorio_instalacion]/src/MINSAL/CostosBundle/Resources/estructurasBD/
costos_rrhh.sql
\i [directorio_instalacion]/src/MINSAL/CostosBundle/Resources/estructurasBD/
costos_activo_fijo.sql
```

2.4.5 Cargar datos iniciales

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

2.4.6 Crear un usuario administrador del SIIG

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

2.5 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:15672` El usuario por defecto es **guest** y la clave **guest**
- Además es necesario configurar el CRON para que ejecute periódicamente la carga de datos, con esto se llamará al proceso `origen-dato:cargar` que verificará para cada indicador si le corresponde realizar la carga de datos según se haya configurado: diario, mensual, bimensual, trimestral o anual. Un ejemplo podría ser crear el archivo: `/etc/cron.d/carga-php-siig`

```
#Ejecutar cada dia a las 00:00
0 0 * * * www-data test -x /usr/bin/php && /usr/bin/php /var/
www/siig/app/console origen-dato:cargar
```

2.6 Instalación de librería wkhtmltopdf

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

1. Descargar wkhtmltopdf desde <http://wkhtmltopdf.org/downloads.html> elegir la versión adecuada al sistema operativo
2. Descomprimir. Ej.: `tar xjf wkhtmltox-linux-amd64_0.12.0-03c001d.tar.xz`
3. Copiar el archivo (como root): `cp wkhtmltox/bin/wkhtmltopdf /usr/bin/wkhtmltopdf`
4. Dar permisos de ejecución (como root): `chmod +x /usr/bin/wkhtmltopdf`

2.7 OPCIONAL: Validación de Usuarios desde directorios LDAP

Si fuese necesario validar usuarios contra un directorio LDAP, se deben seguir los pasos descritos en esta sección. 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 continuación se muestran las líneas relevantes para especificar que directorio usar:

```
#Fr3d_LDAP
fr3d_ldap:
  driver:
    host:          10.10.20.2 # IP del Servidor LDAP institucional
    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
```

2.7.1 Cargar la aplicación

En este punto estamos listos para cargar la aplicación desde: http://siig.localhost/app_dev.php

Capítulo 3

Modelo De Datos

3.0.1 Diagrama Entidad Relacion

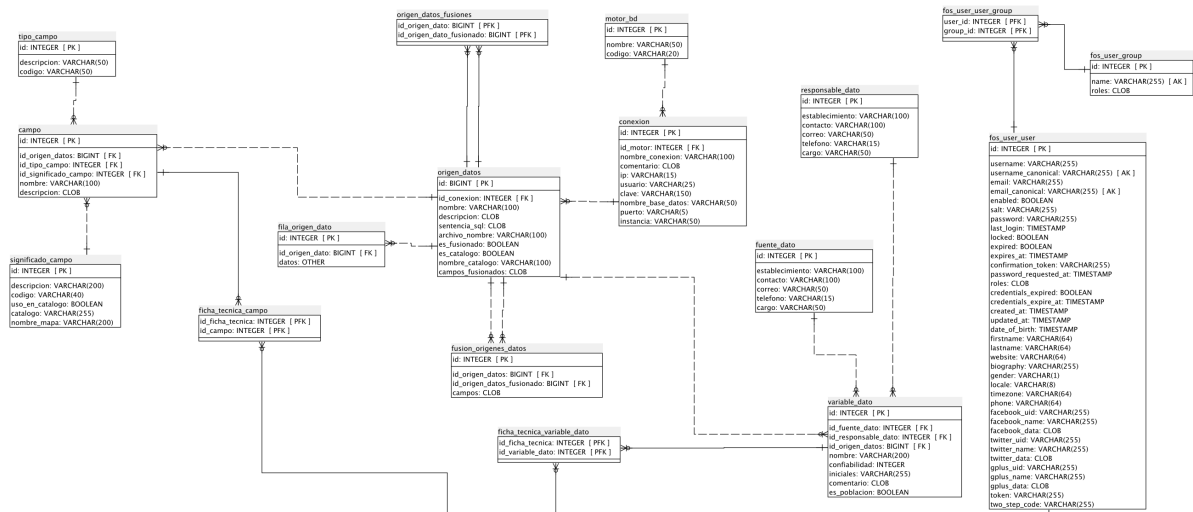


Figura 3.1 Diagrama ER1

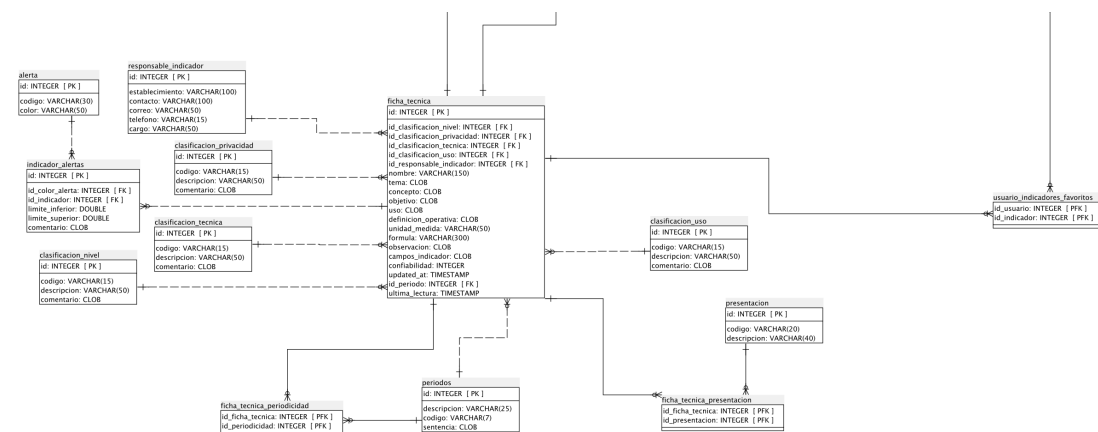


Figura 3.2 Diagrama ER2

3.0.2 Diccionario de Datos

3.0.2.1 Lista de tablas

- [alerta \(página 21\)](#)
- [campo \(página 21\)](#)
- [clasificacion_nivel \(página 22\)](#)
- [clasificacion_privacidad \(página 22\)](#)
- [clasificacion_tecnica \(página 23\)](#)
- [clasificacion_uso \(página 23\)](#)
- [conexion \(página 24\)](#)
- [ficha_tecnica \(página 24\)](#)
- [ficha_tecnica_campo \(página 26\)](#)
- [ficha_tecnica_periodicidad \(página 27\)](#)
- [ficha_tecnica_presentacion \(página 27\)](#)
- [ficha_tecnica_variable_dato \(página 27\)](#)
- [fila_origen_dato \(página 28\)](#)
- [fos_user_group \(página 28\)](#)
- [fos_user_user \(página 29\)](#)
- [fos_user_user_group \(página 31\)](#)
- [fuente_dato \(página 31\)](#)
- [fusion_origenes_datos \(página 32\)](#)
- [indicador_alertas \(página 32\)](#)
- [motor_bd \(página 33\)](#)
- [origen_datos \(página 33\)](#)
- [origen_datos_fusiones \(página 34\)](#)
- [periodos \(página 35\)](#)
- [presentacion \(página 35\)](#)
- [responsable_dato \(página 35\)](#)
- [responsable_indicador \(página 36\)](#)
- [significado_campo \(página 37\)](#)

- [tipo_campo](#) (página 37)
- [usuario_indicadores_favoritos](#) (página 37)
- [variable_dato](#) (página 38)

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](#) (página 32) 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 (página 33))	id_origen_datos	BIGINT		NOT NULL
id_tipo_campo (FK (página 37))	id_tipo_campo	INTEGER		NOT NULL
id_significado_campo (FK (página 37))	id_significado_campo	INTEGER		NOT NULL
nombre	nombre	VARCHAR(100)		NOT NULL

descripcion	descripcion	CLOB		
-------------	-------------	------	--	--

Esta tabla depende de:

- [origen_datos](#) (página 33) por medio de (id_origen_datos)
- [tipo_campo](#) (página 37) por medio de (id_tipo_campo)
- [significado_campo](#) (página 37) por medio de (id_significado_campo)

Esta tabla es usada por:

- [ficha_tecnica_campo](#) (página 26) 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](#) (página 24) 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 \(página 24\)](#) 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 \(página 24\)](#) 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 \(página 24\)](#) 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 (página 33))	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 ([página 33](#)) por medio de (id_motor)

Esta tabla es usada por:

- origen_datos ([página 33](#)) 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 (página 22))	id_clasificacion_nivel	INTEGER		NOT NULL
id_clasificacion_privacidad (FK (página 22))	id_clasificacion_privacidad	INTEGER		NOT NULL
id_clasificacion_tecnica (FK (página 23))	id_clasificacion_tecnica	INTEGER		NOT NULL
id_clasificacion_uso (FK (página 23))	id_clasificacion_uso	INTEGER		NOT NULL
id_responsable_indicador (FK (página 36))	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 (página 35))	id_periodo	INTEGER		NOT NULL
ultima_lectura	ultima_lectura	TIMESTAMP		

Esta tabla depende de:

- [periodos \(página 35\)](#) por medio de (id_periodo)
- [clasificacion_privacidad \(página 22\)](#) por medio de (id_clasificacion_privacidad)
- [clasificacion_tecnica \(página 23\)](#) por medio de (id_clasificacion_tecnica)
- [clasificacion_uso \(página 23\)](#) por medio de (id_clasificacion_uso)
- [clasificacion_nivel \(página 22\)](#) por medio de (id_clasificacion_nivel)
- [responsable_indicador \(página 36\)](#) por medio de (id_responsable_indicador)

Esta tabla es usada por:

- [ficha_tecnica_periodicidad \(página 27\)](#) hace referencia la campo (id)
- [ficha_tecnica_campo \(página 26\)](#) hace referencia la campo (id)
- [ficha_tecnica_presentacion \(página 27\)](#) hace referencia la campo (id)
- [indicador_alertas \(página 32\)](#) hace referencia la campo (id)
- [ficha_tecnica_variable_dato \(página 27\)](#) hace referencia la campo (id)
- [usuario_indicadores_favoritos \(página 37\)](#) 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 (página 24))	id_ficha_tecnica	INTEGER	PK	NOT NULL
id_campo (PK) (FK (página 21))	id_campo	INTEGER	PK	NOT NULL

Esta tabla depende de:

- [campo \(página 21\)](#) por medio de (id_campo)

- ficha_tecnica (página 24) 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 (página 24))	id_ficha_tecnica	INTEGER	PK	NOT NULL
id_periodicidad (PK) (FK (página 35))	id_periodicidad	INTEGER	PK	NOT NULL

Esta tabla depende de:

- periodos (página 35) por medio de (id_periodicidad)
- ficha_tecnica (página 24) 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 (página 24))	id_ficha_tecnica	INTEGER	PK	NOT NULL
id_presentacion (PK) (FK (página 35))	id_presentacion	INTEGER	PK	NOT NULL

Esta tabla depende de:

- ficha_tecnica (página 24) por medio de (id_ficha_tecnica)
- presentacion (página 35) 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 (página 24))	id_ficha_tecnica	INTEGER	PK	NOT NULL
id_variable_dato (PK) (FK (página 38))	id_variable_dato	INTEGER	PK	NOT NULL

Esta tabla depende de:

- [ficha_tecnica \(página 24\)](#) por medio de (id_ficha_tecnica)
- [variable_dato \(página 38\)](#) 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 (página 33))	id_origen_dato	BIGINT		NOT NULL
datos	datos	[1111]		

Esta tabla depende de:

- [origen_datos \(página 33\)](#) 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` (página 31) 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](#) (página 31) hace referencia la campo (id)
- [usuario_indicadores_favoritos](#) (página 37) 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 (página 29))	user_id	INTEGER	PK	NOT NULL
group_id (PK) (FK (página 28))	group_id	INTEGER	PK	NOT NULL

Esta tabla depende de:

- [fos_user_group](#) (página 28) por medio de (group_id)
- [fos_user_user](#) (página 29) 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 \(página 38\)](#) 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 (página 33))	id_origen_datos	BIGINT		NOT NULL
id_origen_datos_fusionado (FK (página 33))	id_origen_datos_fusionado	BIGINT		NOT NULL
campos	campos	CLOB		

Esta tabla depende de:

- [origen_datos \(página 33\)](#) por medio de (id_origen_datos)
- [origen_datos \(página 33\)](#) 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 (página 21))	id_color_alerta	INTEGER		NOT NULL
id_indicador (FK (página 24))	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 \(página 21\)](#) por medio de (id_color_alerta)
- [ficha_tecnica \(página 24\)](#) 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 \(página 24\)](#) 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 (página 24))	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 \(página 24\)](#) por medio de (id_conexion)

Esta tabla es usada por:

- [campo \(página 21\)](#) hace referencia la campo (id)
- [origen_datos_fusiones \(página 34\)](#) hace referencia la campo (id)
- [origen_datos_fusiones \(página 34\)](#) hace referencia la campo (id)
- [variable_dato \(página 38\)](#) hace referencia la campo (id)
- [fila_origen_dato \(página 28\)](#) hace referencia la campo (id)
- [fusion_origenes_datos \(página 32\)](#) hace referencia la campo (id)
- [fusion_origenes_datos \(página 32\)](#) hace referencia la campo (id)

22. origen_datos_fusiones ()

Nomre Logico de Columna	Nombre Fisico de Columna	Tipo	PK	Nullable
id_origen_dato (PK) (FK (página 33))	id_origen_dato	BIGINT	PK	NOT NULL
id_origen_dato_fusionado (PK) (FK (página 33))	id_origen_dato_fusionado	BIGINT	PK	NOT NULL

Esta tabla depende de:

- [origen_datos \(página 33\)](#) por medio de (id_origen_dato_fusionado)
- [origen_datos \(página 33\)](#) 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 \(página 27\)](#) hace referencia la campo (id)
- [ficha_tecnica \(página 24\)](#) 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 \(página 27\)](#) 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 \(página 38\)](#) 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 \(página 24\)](#) 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 \(página 21\)](#) 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 \(página 21\)](#) hace referencia la campo (id)

29. usuario_indicadores_favoritos ()

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

id_usuario (PK) (FK (página 29))	id_usuario	INTEGER	PK	NOT NULL
id_indicador (PK) (FK (página 24))	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 (página 31))	id_fuente_dato	INTEGER		NOT NULL
id_responsable_dato (FK (página 35))	id_responsable_dato	INTEGER		NOT NULL
id_origen_datos (FK (página 33))	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 \(página 33\)](#) por medio de (id_origen_datos)
- [fuente_dato \(página 31\)](#) por medio de (id_fuente_dato)
- [responsable_dato \(página 35\)](#) por medio de (id_responsable_dato)

Esta tabla es usada por:

- [ficha_tecnica_variable_dato \(página 27\)](#) hace referencia la campo (id)

<

h3>

Esta página se ha dejado vacía a propósito

Lista de figuras

3.1 Diagrama ER1	19
3.2 Diagrama ER2	19