



# MANUAL TÉCNICO

ETAB

Esta página se ha dejado vacía a propósito

# Índice de contenidos

<b>Capítulo 1 Tecnologías utilizadas.....</b>	<b>7</b>
1.1 Gestor de base de datos .....	7
1.2 Librería Contrib de PostgreSQL .....	7
1.3 Servidor Web.....	7
1.4 Framework de desarrollo/Servidor.....	7
1.5 Framework JavaScript .....	7
1.6 Framework para interfaces de usuario.....	7
1.7 Librería para gráficos .....	8
1.8 Mensajería.....	8
<b>Capítulo 2 Sistema Integrado de Información Gerencial .....</b>	<b>9</b>
2.1 Requerimientos.....	9
2.2 Instalación.....	9
2.3 Configuración.....	10

Esta página se ha dejado vacía a propósito

# Introducción

Este documento es un borrador

Esta página se ha dejado vacía a propósito

## Capítulo 1

# Tecnologías utilizadas

Todo el software utilizado para creación del SIIG/eTAB es software libre.

### 1.1 Gestor de base de datos

PostgreSQL, versión utilizada es la 9.2 Para almacenar la información obtenida de los diferentes orígenes de datos se enfrentaba ante el problema que no se puede determinar la estructura de los datos, serán diferentes para cada origen en cantidad y tipo de campos, por lo cual se eligió el tipo especial de datos HStore perteneciente a PostgreSQL para almacenar esos datos.

### 1.2 Librería Contrib de PostgreSQL

Para utilizar el tipo de datos HStore es necesario instalar el paquete contrib de postgresql

### 1.3 Servidor Web

Se ha realizado sobre Apache versión 2.2

### 1.4 Framework de desarrollo/Servidor

El lenguaje que se ha utilizado es PHP con Symfony versión 2.4

### 1.5 Framework JavaScript

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

### 1.6 Framework para interfaces de usuario

Bootstrap (<http://twitter.github.com/bootstrap/>) es un framework que hace HTML, CSS y Javascript simple y flexible para componentes e interacciones de interfaz de usuarios populares.

## 1.7 Librería para gráficos

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

## 1.8 Mensajería

La carga de datos se hace con la librería RabbitMQ (<http://www.rabbitmq.com/>)



## Capítulo 2

# Sistema Integrado de Información Gerencial

## 2.1 Requerimientos

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

### 2.1.1 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 apache2 postgresql acl
git-core curl
```

## 2.2 Instalación

### 2.2.1 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

## 2.2.2 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
```

## 2.2.3 Instalar todas las librerías necesarias

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

## 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
```

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
```

## 2.3.2 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

## 2.3.4 Configuración de la conexión

editar el archivo app/config/parameters.yml y colocar los valores correctos para las variables siguientes:

- database\_driver: pdo\_pgsql
- database\_host: localhost
- database\_port: null

- database\_name: nombre\_base\_datos
- database\_user: nombre\_usuario\_base\_datos
- database\_password: clave\_usuario

### 2.3.5 Crear la base de datos

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

### 2.3.6 Cargar datos iniciales

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

### 2.3.7 Crear un usuario administrador

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

### 2.3.8 Instalación de HStore

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

```
# apt-get install postgresql-contrib
```

- 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 RESTRICT

);
```

### 2.3.9 Instalación de RabbitMQ (<http://www.rabbitmq.com/>)

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**

### **2.3.10 Cargar la aplicación**

[http://siig.localhost/app\\_dev.php](http://siig.localhost/app_dev.php)