Documentación técnica Cross H.

**V. 5.**

# 

Última actualización

Santiago de Cali, 4 de Agosto de 2020

# 

# Alcance

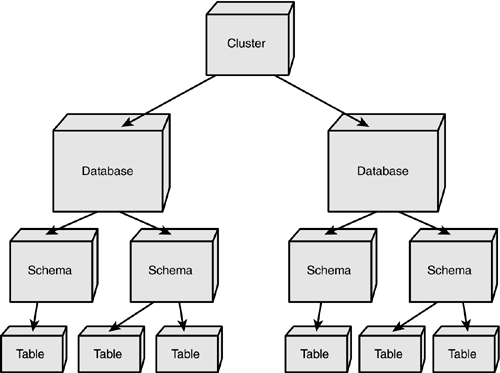
Este documento va dirigido a programadores e ingenieros de software, así como personal especializado en informática que esté a cargo del funcionamiento y mantenimiento de la aplicación ***CROSS H***, V 5.X. En este manual encontrará información acerca de los requerimientos funcionales, instalación, depuración y arquitectura de la aplicación, así como algunos ejemplos de resolución de problemas a nivel técnico. Este no es un manual de usuario del sistema ni pretende reemplazar a uno, no se encontrarán especificaciones fuertes del código fuente, sino sólo un panorama del uso de archivos y estructuras de datos del mismo.

# Requerimientos funcionales

* Servidor apache v2.0 o superior.
* Php 5.3 o superior.
* Postgresql versión 6.0 o superior.
* Sistema operativo centos 5 o superior.
* 2 GB de memoria RAM.
* Al menos 200 GB de disco duro dependiendo del volumen de documentos adjuntos.

# Estructura de la base de datos

## Esquemas



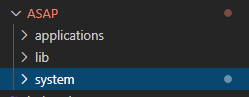
Para facilitar el funcionamiento del software en distintos ámbitos (Como por ejemplo tener en una misma instancia del software un sistema de helpdesk interno del área de tecnología y un sistema de pqrs con los clientes), CROSS hace uso de la arquitectura de esquemas de postgresql. Para cada ámbito se define un schema en donde se guardan los datos que corresponden al mismo (Por ejemplo las categorías de caso para un pqr son distintas a las categorías que usan para reportar una falla técnica),

# Arquitectura de la aplicación

La arquitectura de CROSS es MVC. Modelo Vista Controlador (**MVC**) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Las vistas de la aplicación normalmente son los archivos .tpl encontrados en cada uno de los módulos (Revisaremos esto más adelante).

# Estructura de archivos

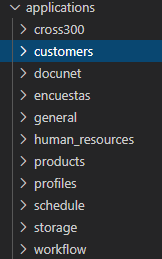


Como primera instancia de nuestra aplicación, dentro del directoria web (Para apache sobre centos usualmente var/www/html) encontraremos el directorio principal de nuestra aplicación y dentro de esta la carpeta ASAP. La cual contiene todo el código fuente de CROSS.

Dentro de ASAP encontraremos tres directorios applications, lib y system.

# Applications

Dentro del directoria de aplicaciones encontraremos toda la lógica de negocio para cada uno de los módulos de la aplicación.



Para cada módulo tenemos los siguientes subdirectorios:

* config: Contiene los archivos de configuración. Estos son utilizado en el momento de la instalación. Además, contiene archivos binarios txt con los datos generados en la instalación.
* data: Aquí se almacenan los archivos que contienen los métodos para el guardado o extracción de datos desde bases de datos, caché u otros anexos.
* Domain: Contiene la lógica de negocio que se ejecuta a través del navegador web
* rs: Archivos de manejo del servicio rs
* templates\_c: Contiene los datos de idioma para datos cargados por medio de webservices o solicitudes ajax.
* tmp: Archivos temporales, varia para cada aplicación
* web:Contiene la lógica manejada desde la aplicación al realizar acciones como presionar botones, seleccionar datos, etc. Contiene además los archivos estáticos de la aplicación como fuentes, archivos css y javascript.
  + templates: Dentro de template se encuentra las plantillas html que muestran el aplicativo.

# Lib

El directorio lib contiene todos los datos, fuentes y librerías de terceros que se usan como apoyo para el funcionamiento de la aplicación. Durante la resolución de problemas normalmente no tocaremos esta carpeta a excepción del momento en el que tengamos la necesidad de visualizar las consultas realizadas a la base de datos.

Algunas de las librerías a destacar son:

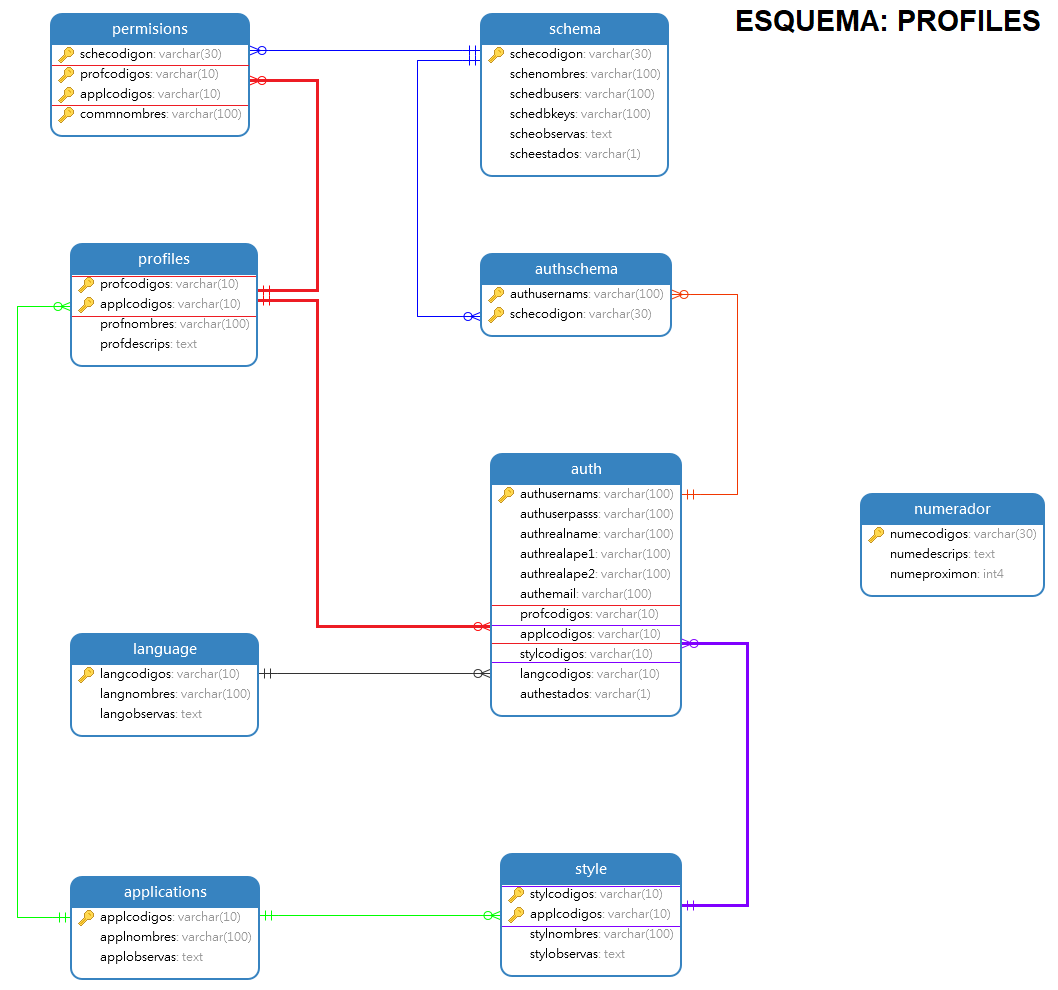
* Adodb: Maneja las conexiones con bases de datos
* nusoap: Se encarga de los webservices
* smtpmail: Se encarga del envío de correos por medio de un servidor smtp.

# System

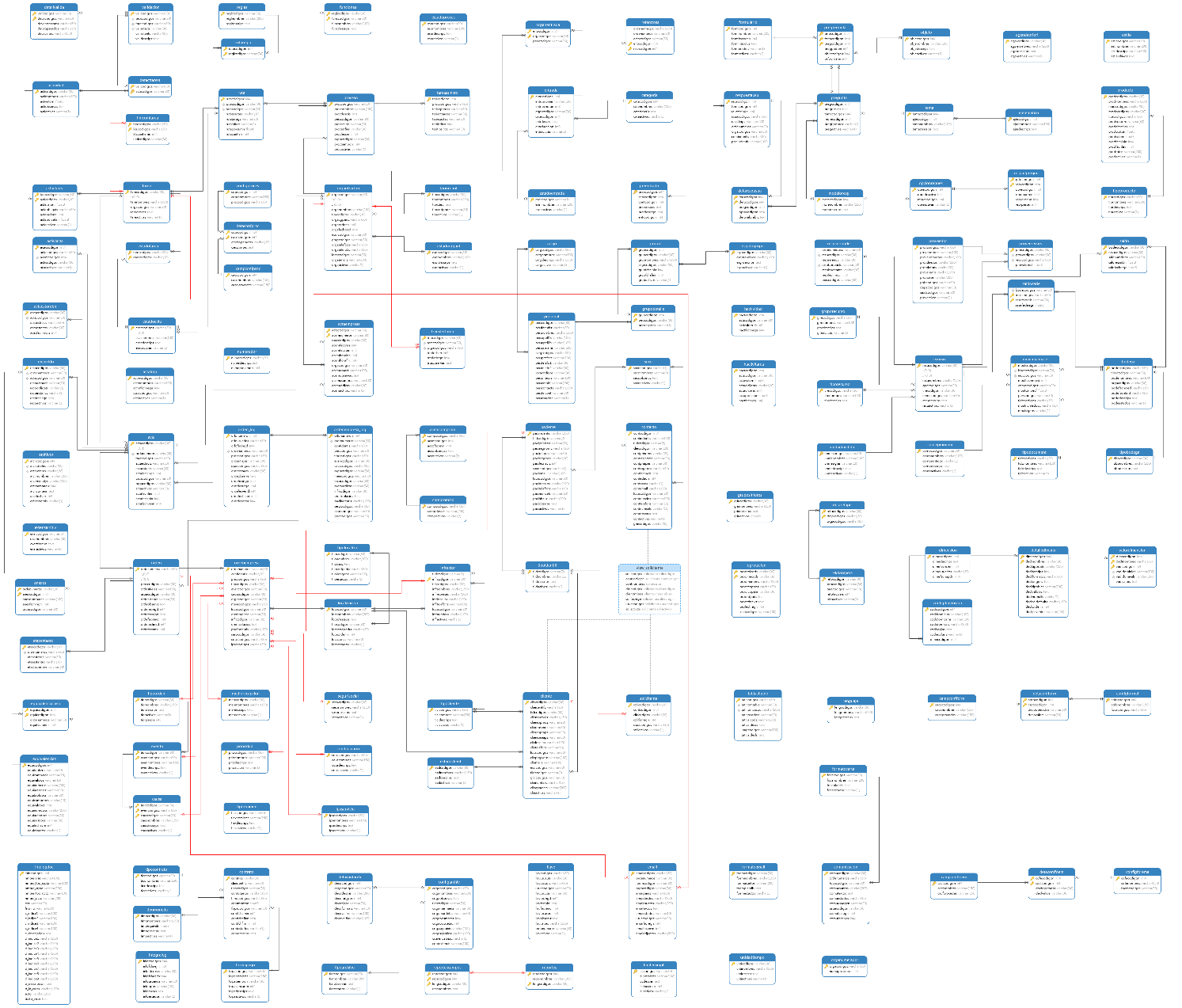
Dentro del directorio System están los archivos de la aplicación son el núcleo de la misma. A partir de system empieza la ejecución del aplicativo. La clase FrontController se encarga de manejar las solicitudes recibidas por el navegador, y dar una respuesta usualmente como un tipo de dato primitivo de cómo se debe gestionar la solicitud.

# Base de Datos

La base de datos se realizó en el SMBD ***postgresql***. El modelo de base de datos, se compone de 2 esquemas. EL primero esquema se llama ***profiles***, y en él se encuentras las tablas que están relacionadas con los usuarios de la aplicación, los perfiles, las aplicaciones y permisos de usuarios.



El segundo esquema, tiene por nombre schema2. En él se encuentran todas las tablas relacionadas a la gestión de ***CROSS***. A continuación, se puede observar la imagen con las entidades pertenecientes al esquema schema2.



# Notas

El tamaño de los archivos que se anexan vienen definidos por el máximo tamaño de la variable "upload\_max\_filesize" dentro del archivo "php.ini" que se encuentra en el directorio "/etc" y que por defecto viene en 2M.