**Guioteca**

**Manual de Mantenimiento**

Autor: Héctor Vergara

Última Modificacion: 11/04/2014

# Servicios

El esquema de funcionamiento de Guioteca consiste en un conjunto de servicios operando de manera simultánea, incluyendo:

**Base de Datos: MySQL**

Almacena toda la información de Guioteca, incluyendo artículos, categorías, usuarios, etc.

**Aplicaciones: Apache + PHP (mod\_php)**

Se encarga de procesar los requests y devolver una respuesta, utilizando WordPress a través de PHP.

**Caché: Varnish**

Almacena y provee de copias del contenido más utilizado, para evitar impactar al servidor de aplicaciones. Se configura a través de reglas utilizando un lenguaje llamado VCL.

**Frontend: Nginx**

Actúa como *proxy reverso* y como servidor de estáticos, gracias a su buena performance y poco uso de memoria.

(DIAGRAMA)

# Componentes

## WordPress

La aplicación utiliza el modo de *blogs multiples* para albergar a todos sus canales temáticos. Cada blog entonces, posee su propio set de tablas en MySQL además de un prefijo en su URL.

Por ejemplo, el blog **Tecnología** posee las tablas **wp\_23\_xxxxxxx** en la base de datos y el prefijo de URL **/tecnologia/** para su navegación.

Además, depende del funcionamiento de *Plugins* desarrollados específicamente para la plataforma, como:

* **Editor de Blogs y Sub-Blogs**: desarrollado por *PixelKit*, este plugin permite a los editores priorizar y ordenar los contenidos desplegados en Portada y las portadillas respectivas.
* **Varnish**: desarrollado por Héctor Vergara, permite el refresco del contenido a visualizar después de una acción de modificación (crear, editar, eliminar artículo), enviando una señal al servidor de caché.
* **Thumbnails**: desarrollado por Matías Salvadores, permite la generación de miniaturas de múltiples tamaños para ser desplegados en variados templates del sitio.
* **Assets**: desarrollado por Héctor Vergara, permite comprimir y eliminar comentarios del HTML generado, además de combinar los archivos CSS y JS para una mejor performance y menor tiempo de carga.

(DIAGRAMA)

## Buscador

Reemplazando las funciones de búsqueda originales de WordPress, fue desarrollado un buscador más avanzado, que permitiese alcanzar una mejor indexación y aumentar la encontrabilidad del contenido.

Para ello fue utilizado una instancia de SOLR, la cual almacena un índice que contiene todos los contenidos generados en Guioteca.

La integración con WordPress permite indexar y actualizar los registros en caso de *creación, edición o eliminación*, enviando la señal indicada al servidor SOLR.

La lógica y configuración de esta integración se puede encontrar en la carpeta **wp-content/plugins/solr-search**.

Dentro del archivo **solr-config.php** se debe especificar la URL del repositorio SOLR:

define('SOLR\_ENDPOINT', 'http://search.mersap.com/solr/guioteca')

## CDN

Buscando mejorar la performance del sitio (en tiempo de carga) se exploraron múltiples alternativas. Entre ellas está la de separar los archivos estáticos en un dominio separado, para evitar la creación de *Cookies* que impidan mantener en *caché* este tipo de datos.

Para ello fue configurado un *Virtual Host* llamado **g.cdn.mersap.com**, que mantiene una copia de los archivos estáticos durante un tiempo razonable (1 semana).

Esto permite mejorar el tiempo de carga, distribuyendo la carga del navegador al paralelizar la descarga de archivos necesarios para el despliegue de la página.

# Operaciones

## Instalación

Los paquetes necesarios para correr Guioteca son:

* MySQL 5+
* Apache 2+
* PHP 5+
* Módulos PHP: mysql, gd, xml, apc
* Varnish 3+
* Nginx 1.4+

En CentOS se instalan de la siguiente manera:

# Varnish Repo

rpm --nosignature -i http://repo.varnish-cache.org/redhat/varnish-3.0/el6/noarch/varnish-release/varnish-release-3.0-1.el6.noarch.rpm

# Nginx Repo

rpm --nosignature -i http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm

yum install -y mysql-server

yum install -y httpd

yum install -y php php-mysql php-gd php-mysql php-pecl-apc

yum install -y varnish

yum install -y nginx

Luego se debe clonar el repositorio de Guioteca en GitHub, el cual contiene tanto el código fuente de la aplicación, como los archivos de configuración:

git clone <https://github.com/neonline/guioteca.git> /var/www/guioteca

## Procedimientos

**Borrar caché**

Para eliminar entradas en el caché de Varnish, debemos utilizar la consola de Varnish.

varnishadm -S /etc/varnish/secret -T localhost:6082

Algunos ejemplos de utilización:

# Eliminar todo el caché

**ban.url .\***

# Eliminar todas las URLs que comiencen con “/tecnologia”

**ban.url ^/tecnologia**

# Eliminar todas las URLs que sean del host g.cdn.mersap.com

**ban req.http.Host == g.cdn.mersap.com**

**Bloquear IPs sospechosas**

Para detectar a los crawlers indiscriminados y abusadores, se creó el comando **visitop**, el cual permite ver las 10 IPs que más requests hacen a Guioteca.

Si alguna de estas IPs parece sospechosa, debe ser bloqueada para proteger la integridad del sistema. Para ello puede ejecutarse el comando:

iptables -A INPUT -s IP\_SOSPECHOSA -j REJECT

Esto impedirá cualquier conexión desde esa IP a nuestro servidor.

**Firewall**

El servidor de Guioteca contiene algunas reglas de protección contra ataques, basado en *iptables*.

Este incluye bloquear todos los accesos a puertos, excepto a los puertos 80 (http) y 22 (ssh).

La excepción es la habilitación de todo acceso desde la red Mercurio, perteneciente a las subredes:

* 200.12.16.0/21
* 200.12.24.0/22