RG-T3411 Transparencia y gestión de información en el sector de agua & saneamiento. CONSULTOR INDIVIDUAL PARA REALIZAR PARAMETRIZACIONES Y AJUSTES EN SISTEMAS DE INFORMACIÓN INSTITUCIONALES, DE ACUERDO CON LAS NECESIDADES DE LA COOPERACIÓN TÉCNICA.

ANDRÉS FELIPE VILLAMIZAR VECINO

Manual Técnico

2020

## Introducción

Este documento es un manual técnico, resultante de las actividades realizadas para la entrega final del desarrollo de la consultoría“*RG-T3411 TRANSPARENCIA Y GESTIÓN DE INFORMACIÓN EN EL SECTOR DE AGUA & SANEAMIENTO. CONSULTOR INDIVIDUAL PARA REALIZAR PARAMETRIZACIONES Y AJUSTES EN SISTEMAS DE INFORMACIÓN INSTITUCIONALES, DE ACUERDO CON LAS NECESIDADES DE LA COOPERACIÓN TÉCNICA)”*

Contenido

[Introducción 2](#_Toc35362056)

[Ilustraciones 4](#_Toc35362057)

[CACHE 5](#_Toc35362058)

[TIPO DE CACHE Y PROGRAMACIÓN. 5](#_Toc35362059)

[INSTALACIÓN DE APCu. 10](#_Toc35362060)

[ADMINISTRACIÓN DE APCu 17](#_Toc35362061)

## Ilustraciones

[Ilustración 1 Openlayers con proveedor OpenStreetMap 6](#_Toc27414572)

[Ilustración 2 Bing API con Bing como proveedor 7](#_Toc27414573)

[Ilustración 3 Pantalla de inicio Pentaho 8](#_Toc27414574)

[Ilustración 4 PostgreSQL desde pgAdmin 9](#_Toc27414575)

[Ilustración 5 Estructura del generador de plantillas estáticas 10](#_Toc27414576)

[Ilustración 6 Ubicación del generador de plantillas 11](#_Toc27414577)

[Ilustración 7 Generación del modelo 11](#_Toc27414578)

[Ilustración 8 Generación del CRUD 12](#_Toc27414579)

[Ilustración 9 Modelo Entidad Relación 13](#_Toc27414580)

[Ilustración 10 Tablas en la base de datos 13](#_Toc27414581)

[Ilustración 11 Mapa generado con Yii2 14](#_Toc27414582)

[Ilustración 12 Home 15](#_Toc27414583)

[Ilustración 13 Detalle del indicador 16](#_Toc27414584)

# CACHE

La memoria Cache es una colección de datos duplicada que se guarda de alguna forma, con la finalidad de accesarla de una forma más rápida que cuan esta es generada; de esta manera se puede generar la información de una manera más rápida.

En otra palabra la memoria Cache es una forma de mejorar el desempeño de una aplicación Web. La memoria cache puede ser usada en ambos extremos de la aplicación, por ejemplo, en el navegador del usuario para almacenar elementos de la aplicación como los estilos y JavaScript, o en el servidor guardando consultas en memoria, o en bases de datos temporales.

PHP y Yii manejan varios tipos básicos de cache, los cuales se describen a continuación.

## TIPO DE CACHE Y PROGRAMACIÓN.

**Cache de Datos.**

Su función básica es la de almacenar variables, consultas u objetos en algún almacenamiento para luego volver a consultarlos desde ahí. Este tipo de cache se utiliza mediante los “componentes de cache”, los cuales representan un tipo de almacenamiento diferente, como archivos, memoria o bases de datos.

Yii soporta en la versión más reciente 10 componentes de cache, de los cuales 2 están en estado “deprecated”, es decir en una próxima actualización va a ser retirado. Estos componentes son:

* yii\caching\ApcCache: Cache guardada en el segmento de memoria asignada a PHP en el servidor, es aconsejado y considerado el más rápido en los manuales de PHP y Yii2, cuando la aplicación esta centralizada en un solo servidor.
* yii\caching\DbCache: Usa una tabla de bases de datos para guardar los datos de Cache.
* yii\caching\ArrayCache: Guarda los datos de un “request” en un array.
* yii\caching\DummyCache: Cache falso, sirve para realizar la programación y se puede cambiar por otro cuando el servidor tenga algún cache seleccionado.
* yii\caching\FileCache: Utiliza archivos estandarizados para guardar los datos.
* yii\caching\MemCache: Usa los módulos memcache de PHP, considerada por el manual de Yii2 como la opción más rápida cuando la aplicación se encuentra distribuida en varios servidores.
* yii\redis\Cache: Implementa la cache en Redis, que es un manejador de bases de datos que se encuentran en estructuras de Memoria.
* yii\caching\WinCache: Cache cuando la aplicación se despliega en un servidor Windows IIS.
* yii\caching\XCache: Extensión de PHP marcada para retirar.
* yii\caching\ZendDataCache: Se utiliza el componente Zend Data, marcado para retirar.

Para empezar a usar la memoria Cache de Datos en Yii2 se debe buscar el archivo de configuración, ubicado en config/web.php para Yii2 basic, o en \*\*\*\*\*/config/main.php para Yii2 advanced (reemplazando \*\*\*\*\* por backend o forntend si solo se va usar en una sola sección, o en common si se va a usar en todo el proyecto), y poner las siguientes líneas:

'components' => [

'cache' => [

'class' => 'yii\caching\ApcCache',

'keyPrefix' => 'myapp',

'useApcu' => true,

],

],

La línea class sirve para indicar cual de lo diferentes componentes mencionados se va a utilizar, la línea keyPrefix se utiliza para diferenciar la aplicación en la cache cuando el servidor tiene más de una, y la línea useApcu indica que se va a utilizar esa versión en específico, esto último debido a que APC es usado en PHP 5.4 hacia atrás y APCu en PHP 5.6 en adelante.

Una vez realizado el ajuste en el archivo de configuración se puede utilizar las Apis de programación para incluir la memoria Cache en la aplicación. Las siguientes son las APIS de Yii2:

* get (): obtiene los datos de cache mediante el uso de una llave.
* set (): guarda los elementos en chache mediante el uso de una llave.
* add (): adiciona un elemento en la cache cuando la llave no existe, de lo contrario lo salta.
* getOrSet (): recupera un elemento existente de cache, o ejecuta una función que tenga una respuesta para almacenar en cache y devuelve la información cuando esta no existe.
* multiGet (): obtiene múltiples elementos con diferentes llaves.
* multiSet (): guarda múltiples elementos con diferentes llaves.
* multiAdd(): adiciona múltiples ítems.
* exists(): devuelve un valor indicando si el elemento con la llave especificada existe.
* delete(): borra un elemento de cache.
* flush(): vacía la memoria cache.

Yii2 utiliza las mismas Apis de programación para todos los componentes, por esta razón se puede realizar la programación de la memoria cache y actualizar el componente en cualquier momento.

Ejemplos:

1. set: el siguiente ejemplo muestra en la primera línea como se guardan datos en cache con la función set, primero se le indica cual va a ser la llave con la que se va a guardar el dato, luego los datos, y por último la duración de la cache en segundos.

$cache->set($key, $data, 45);

1. get: este ejemplo muestra cómo se puede evaluar si existe un dato en cache, primero se obtiene en una variable el dato con la llave específica del mismo, luego se pregunta si hay algún valor valido, en caso de ser falso se procede a realizar algún código en específico. Este caso puede usarse para evaluar cuando se debe poner los datos en memoria Cache.

$data = $cache->get($key);

if ($data === false) {

// $data no encontrada

}

1. getOrSet: en el siguiente ejemplo se puede ver una función de la aplicación, en esta se evalúa de donde se debe obtener los datos de retorno utilizando la función getOrSet, la cual busca si el cache llamado LoadCantones existe, en caso de ser verdadero, devuelve el valor guardado en memoria cache, en caso de ser falso, ejecuta la función interna.

protected function LoadCantones(){

$data = Yii::$app->cache->getOrSet('LoadCantones',

function () {

return $\_sqlcantones = \app\models\gisindicadores\Localizacion::find()->where(['id\_tipolocalizacion'=>'2'

}, Yii::$app->params['timeCache']);

//cachea la consulta de cantones

return $data;

}

**Cache de consultas SQL.**

Es una version especial de Cache de Datos, sirve para guardar en cache resultados de consultas a bases de datos. Tiene la limitación que no soporta campos de bases de datos que guarden datos complejos como por ejemplo los BLOB, y la cantidad de memoria en cache de la consulta por defecto apenas es de 1MB.

Las Apis de programación para este caso son:

* yii\db\Connection::cache(): Guarda el resultado de una o un conjunto de consultas SQL
* yii\db\Connection::noCache(): Indica cuales consultas de un grupo no deben guardarse en memoria Cache.

Estas Apis también pueden incluir el tiempo de duración de los datos en memoria Cache.

Ejemplos:

1. cache: en este ejemplo se obtiene el valor de resultado mediante el llamado de la función cache en el modelo de Customers, en caso de que la consulta ya esté en cache devuelve el valor de la consulta guardada, de lo contrario devuelve y guarda en memoria Cache el valor de la función interna, el cual es una consulta.

$result = Customer::getDb()->cache(function ($db) {

return Customer::find()->where(['id' => 1])->one();

});

1. noCache: siendo $db una conexión a base de datos de la configuración, se llama el api cache y dentro de esta un noCache en donde se van a aponer las consultas que no se deben guardar.

$result = $db->cache(function ($db) {

// consultas que van a estar en memoria cache

$db->noCache(function ($db) {

// consultas que no va a estar en memoria cache

});

return $result;

}, 600);

1. Una sola línea: las Apis pueden ser llamadas en una sola línea de código, se ejecuta el comando, junto con la API cache a la cual se le pasa el tiempo de duración.

$customer = $db->createCommand(

'SELECT \* FROM customer WHERE id=1')

->cache(60)->queryOne();

**Cache de Fragmentos.**

Yii2 permite guardar en memoria cache, partes de una página web, y como en cache de datos, necesita una llave única para guardarse. Sus Apis de programación se compone de dos funciones que deben usarse dentro de las vistas.

* beginCache($id): busca si existe en memoria Cache algún contenido con el id indicado, si lo encuentra renderiza el contenido, de lo contrario utiliza el código fuente de la vista que está a continuación de su invocación y antes de la siguiente función.
* endCache: marca el final del fragmento que será guardado en memoria Cache.

La API beginCache adicional al id recibe un array con diferentes opciones:

* Durartion: indica el tiempo en segundos en el que se mantiene el fragmento en cache.
* Variations: Guarda la memoria cache dependiendo de algún valor que cambia.
* Enabled: Permite que los datos se guarden en memoria Cache de acuerdo a una expresión que devuelva verdadero o falso.

EL cache por fragmentos también puede ser anidado, es decir que un fragmento en memoria cache puede ir dentro de otro, adicionalmente también se pueden indicar líneas que deben permanecer dinámicas dentro de un fragmento de cache.

Ejemplos:

* beginCache - endCache - duration: Este ejemplo imprime en pantalla los campos de un modelo enviado a la vista, y serán mantenidos en memoria cache por 3600

<?php if ($this->beginCache('cachedPersona'),

['duration' => 3600]) { ?>

<?php foreach ($models as $model): ?>

<?= $model->id; ?>

<?= $model->name; ?>

<?= $model->email; ?>

<br/>

<?php endforeach; ?>

<?php $this->endCache(); } ?>

**Cache de Paginas.**

En este caso una página completa será guardado en memoria cache en el lado del servidor, cuando se vuelva a llamar se obtendrá de la memoria cache en lugar de volver a generarla por completo.

Para este tipo de cache en Yii2 es necesario cambiar la configuración en behaviours en el controlador de la página que se desea guardar en memoria Cache, se debe cambiar la clase por yii\filters\PageCache, en la propiedad “only” indicar en que evento se especificó se va a realizar el guardado, junto con esto se debe adicionar la propiedad duration para indicar el tiempo de duración en memoria Cache; por último se agrega “dependency” en donde se debe agregar la sub propiedad “class” con el valor yii\caching\DbDependency, junto con otra subpropiedad “sql” donde se debe escribir una consulta que se tomara como base para saber si la página cambió y debe ignorarse lo que hay en memoria cache.

Ejemplo:

public function behaviors()

{

return [

[

'class' => 'yii\filters\PageCache',

'only' => ['index'],

'duration' => 60,

'dependency' => [

'class' => 'yii\caching\DbDependency',

'sql' => 'SELECT COUNT(\*) FROM cantones,

],

],

];

}

## INSTALACIÓN DE APCu.

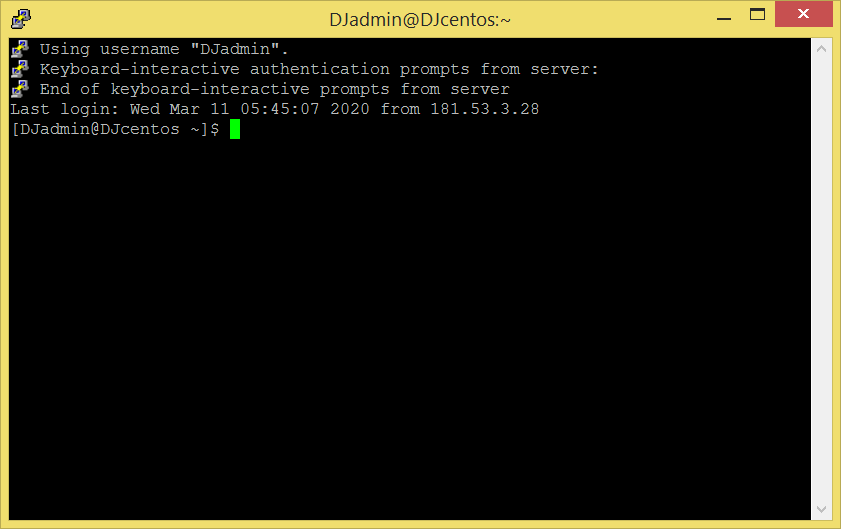
El tipo de cache que se va a utilizar en el proyecto es el APC, que para la versión de PHP es más exactamente el APCu. Para pode iniciar el trabajo con este componente de cache de Yii2 es necesario instalar los módulos correspondientes de PHP; esta instalación varía de acuerdo al sistema operativo y al ambiente (desarrollo, producción).

A continuación, se presentan dos instalaciones, primero se muestra cómo se debe instalar en el servidor de producción con sistema operativo CentOS 7, para continuar con la instalación en Windows para el ambiente de desarrollo usado en el proyecto.

**Instalación en CentOS 7.**

CentOS 7 es un sistema operativo clonado de Red Hat Enterprise, por consiguiente, los comandos van a estar basados en el de este sistema operativo, o en otras palabras utiliza “yum” como gestor de instalaciones y actualizaciones.

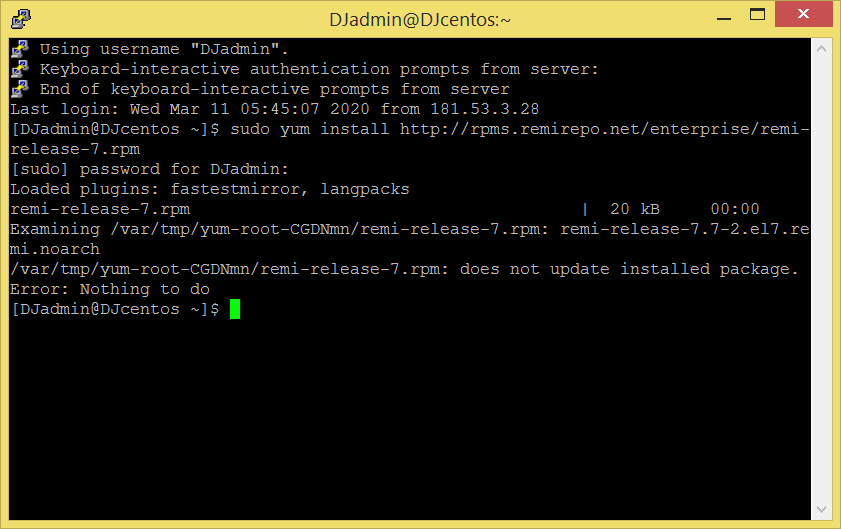
1. Para iniciar el proceso es necesario Logearse en el servidor con un usuario que tenga permisos de instalación



1. Una vez se ha ingresado es necesario verificar que se tenga instalado el repositorio Remi o instalarlo, estas dos acciones se pueden revisar con el comando:

sudo yum install <http://rpms.remirepo.net/enterprise/remi-release-7.rpm>

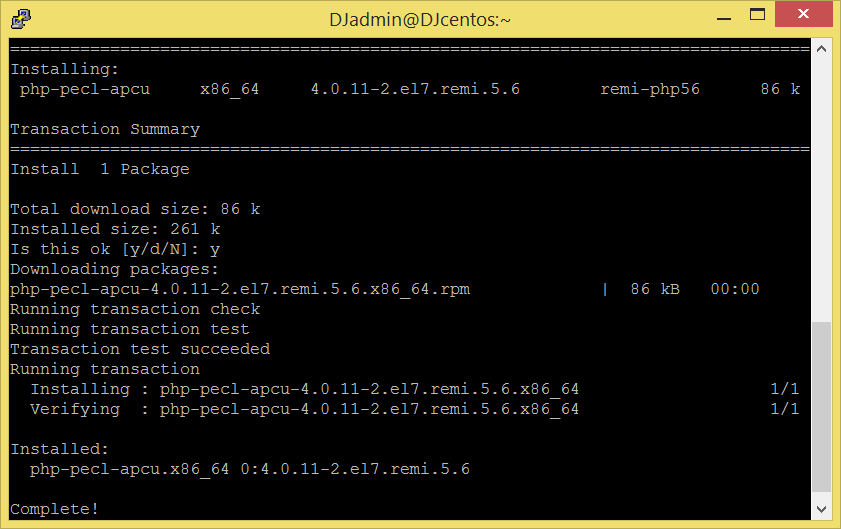
En la siguiente imagen se puede apreciar el resultado en el caso que ya estuviese instalado, en caso contrario iniciara una instalación en donde le preguntara al usuario si desea proceder.



1. Como siguiente paso se procede a instala el módulo APCu mediante el comando

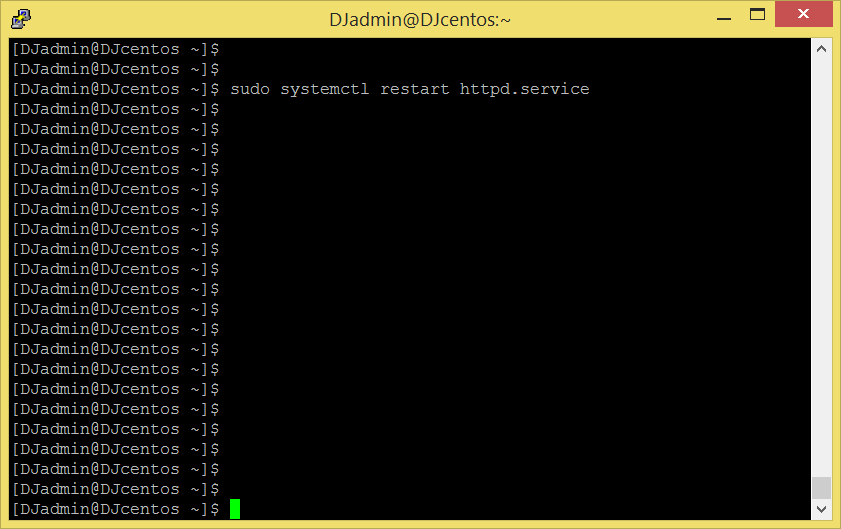
sudo yum install php-pecl-apcu

A continuación, se ve como se realiza el proceso de instalación, mostrando el paquete que se ha encontrado y preguntando si de sea continuar.

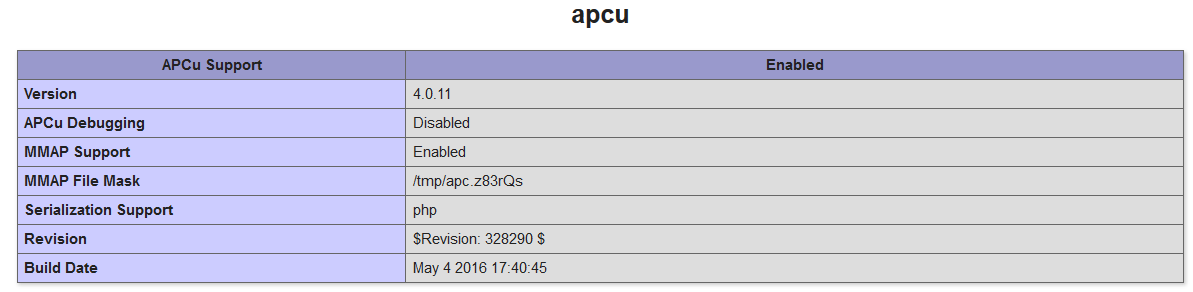


1. Lo siguiente es reiniciar el ser vicio para que la nueva instalación, junto con sus configuraciones tengan efecto, para este se escribe lo siguiente:

sudo systemctl restart httpd.service



1. Por último, se verifica que aparezca el nuevo módulo dentro de las propiedades de php.



Para ver si el módulo se encuentra instalado como se ve en la imagen anterior, se puede ubicar en algún lugar del servidor, pero de manera temporal o que solo se pueda acceder desde local, una página web con una etiqueta de php que imprima todas estas propiedades, el contenido de la etiqueta es el siguiente:

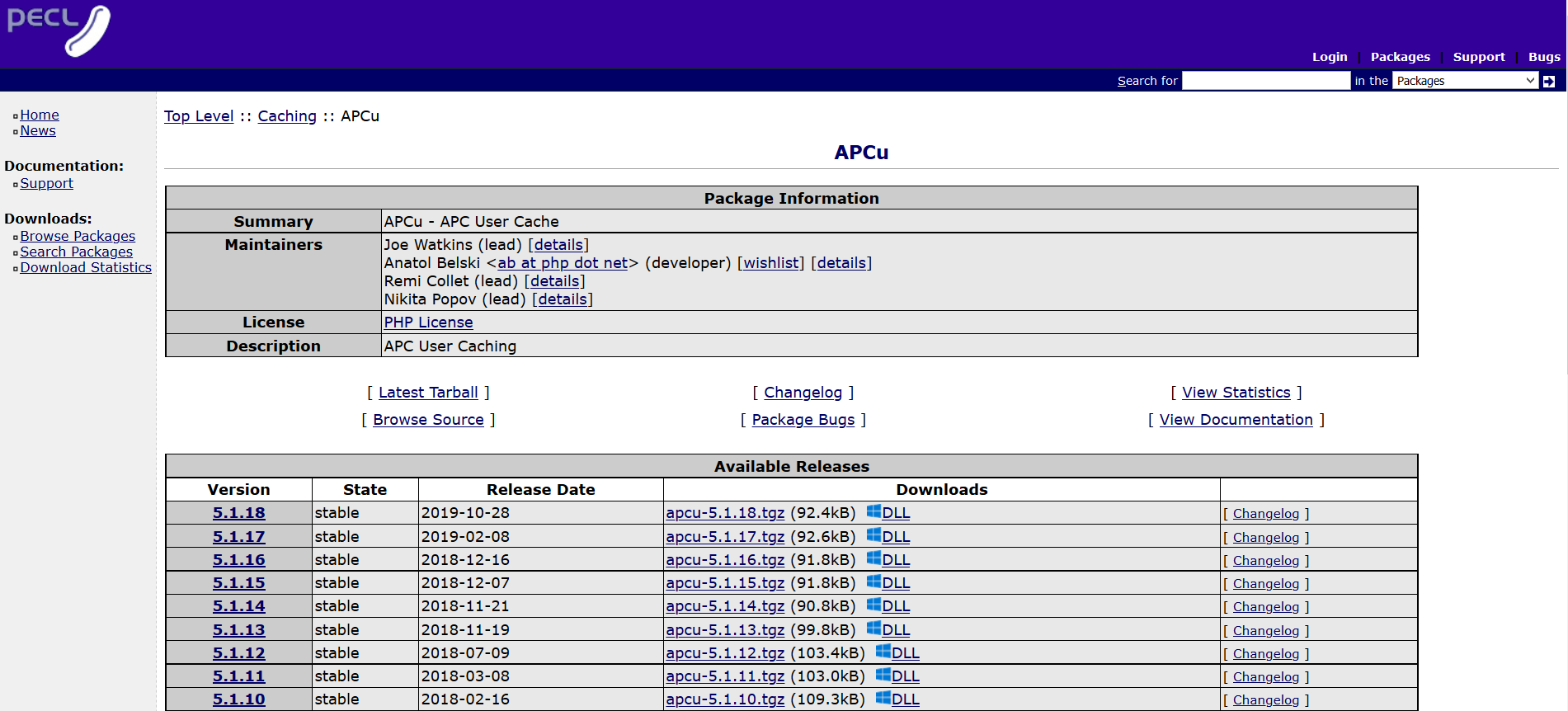
php phpinfo();

**Instalación en Windows Xampp.**

En Windows la instalación del módulo de PHP se realiza utilizando dlls en lugar de un gestor de paquetes. A continuación, se presentan los pasos necesarios para poder utilizar APCu en Windows.

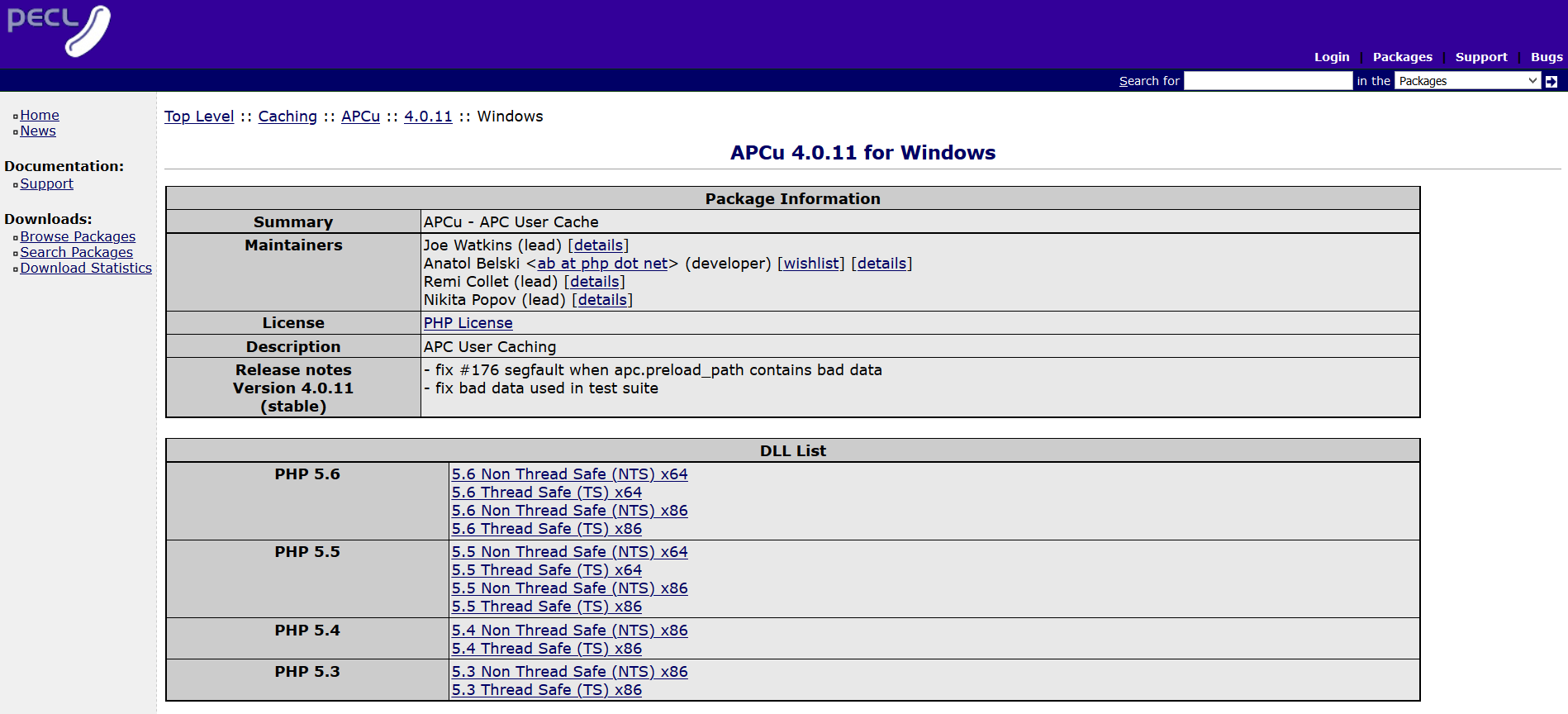
1. Lo primero que se debe hacer es la descarga de la dll de la página web oficial.

<http://pecl.php.net/package/APCu>



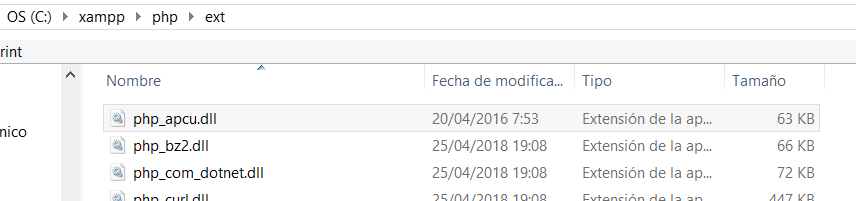
En la imagen anterior se ve que en la página aparece un listado con todas las versiones posibles de descarga, para elegir cual es la adecuada, primero necesitamos conocer que versión de PHP se tiene instalada en Windows y la arquitectura de la misma; para el caso del ejemplo se tiene 5.6, con arquitectura x86, entonces se procede a ir al enlace que dice DLL.

1. El siguiente paso es seleccionar la versión correcta, una vez se ingresa al enlace de DLL, se muestra que versiones son compatibles, para el caso de PHP 5.6 se encontró que la versión más reciente compatible es la 4.0.11, como se ve en la imagen a continuación, es compatible con PHP 5.3 al 5.6, para el caso seleccionamos arquitectura x86, y como es para un servidor apache que usa hilos, se descarga la que dice “thread safe”.



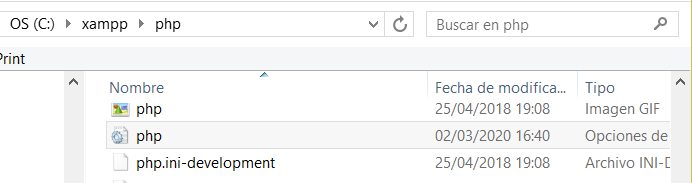
1. Una vez descargada la dll se procede a copiarla en la carpeta de módulos de php en el equipo, que para el caso de Xampp está en:

C:\xampp\php\ext



1. Luego se busca y se abre en un editor de texto el archivo php.ini, que en Xampp se encuentra en:

C:\xampp\php



1. Al final de php.ini se agregan las siguientes líneas, que indican cual dll es la de APCu y unos parámetros necesarios para el funcionamiento, ver sección de ajuste del cache.

[apcu]

extension="C:\xampp\php\ext\php\_apcu.dll"

apc.enabled=1

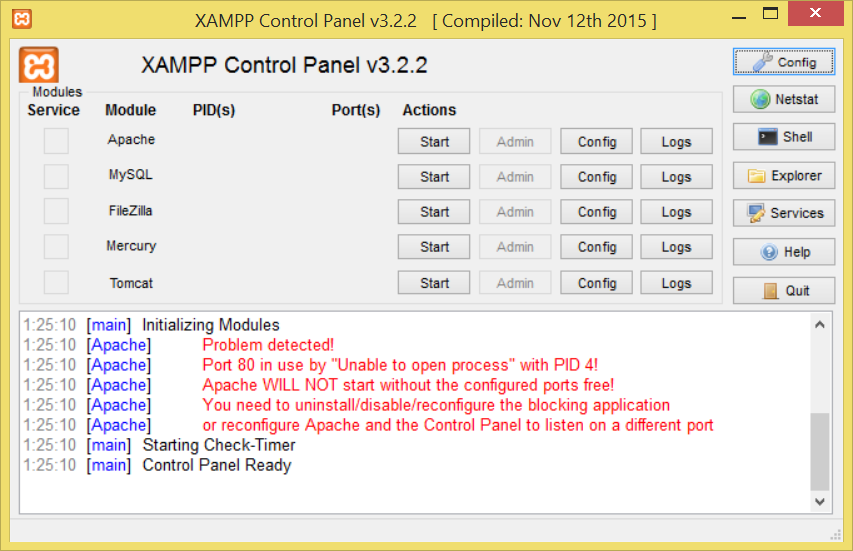
apc.shm\_size=32M

apc.ttl=7200

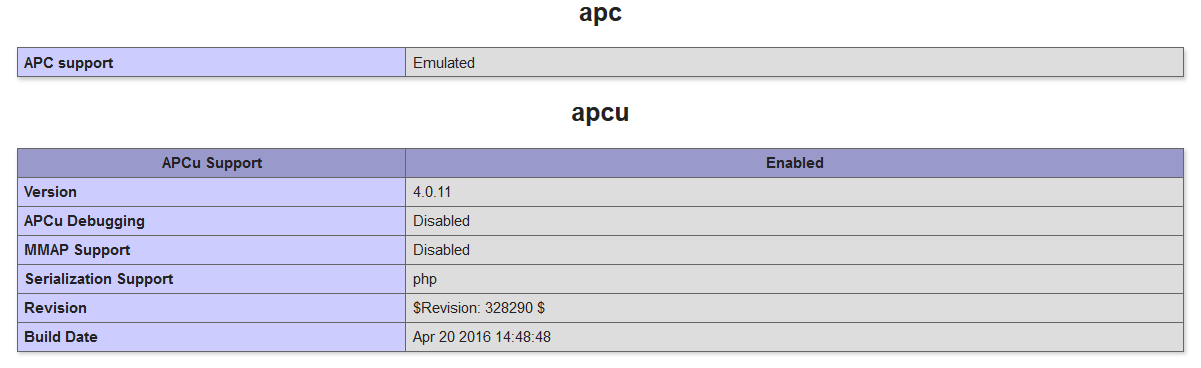
apc.enable\_cli=1

apc.serializer=php

1. Para que tomen efecto los cambios se debe reiniciar el servicio en la consola de Xampp.



1. Se verifica que mediante el acceso de una página PHP con la instrucción de phpinfo, tal cual se realiza en la instalación para Linux (ver sección anterior).



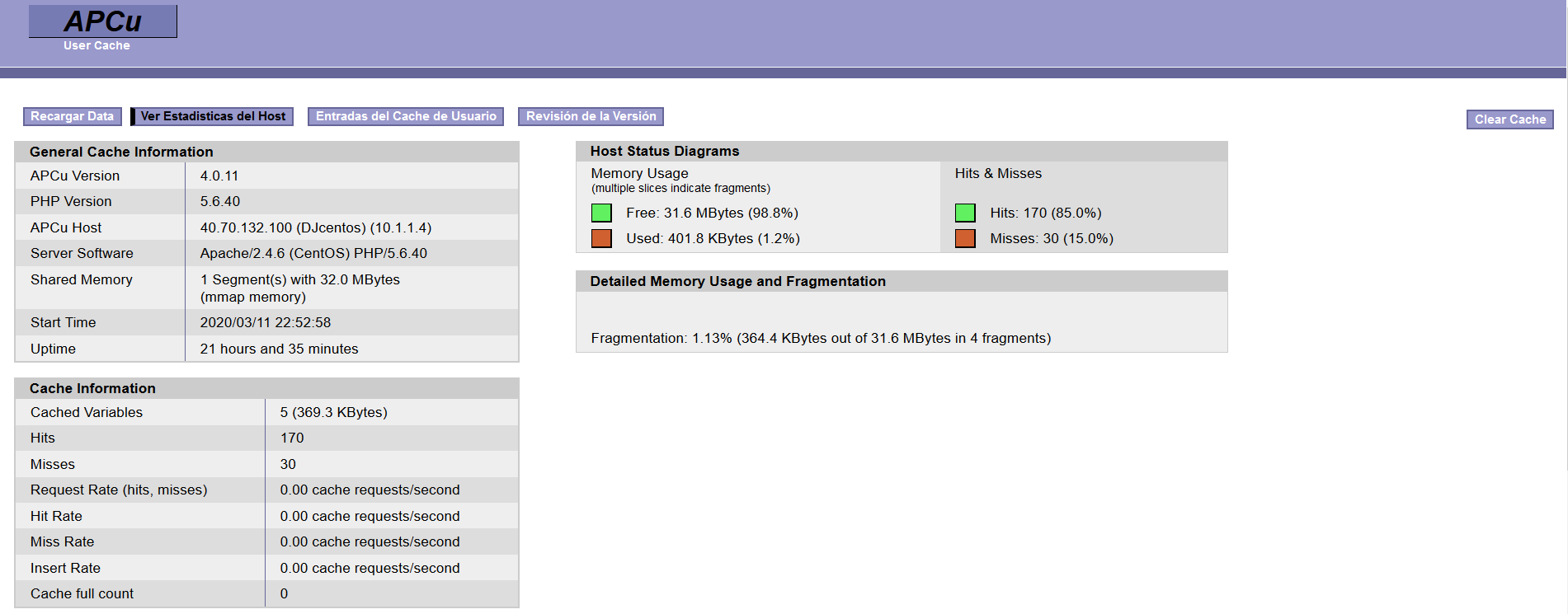
## ADMINISTRACIÓN DE APCu

**Herramienta de visualización de APCu.**

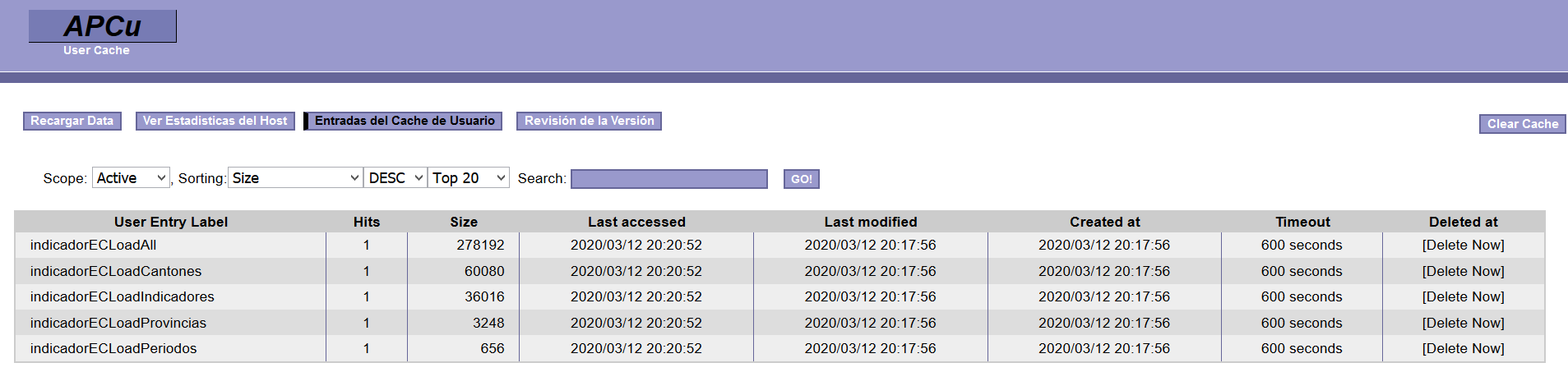
Para una mejor administración de las cache, es aconsejable el uso de herramientas que la presenten en tiempo real, estas debido a que es un módulo de PHP es posible programarlas usando el mismo lenguaje y las funciones que este ya contiene. Como muestra de esto tomamos del github (<https://gist.github.com/mdobydullah/2bc1cca5824eae59117a5a93ed947974>) un código que está distribuido con licencia PHP 3.01, que se llama apcu.php, y se modificó para que funcione con la zona horaria de Ecuador.

Esta página se compone de una cabecera con el nombre del módulo de caché APCu, luego muestra un menú con cuatro opciones:

1. Recargar Data. Esta opción sirve para actualizar la información de caché del equipo.
2. Ver estadísticas del host. En esta parte se revisan los datos generales del uso de la cache, junto con la información básica de las versiones de apache, PHP y el host



1. Entradas de Cache de Usuario. Se presenta un listado con las diferentes entradas de cache que se han hecho, el tamaño, las fechas de acceso, modificación, o creación, cuanto tiempo de vida tiene, y si desea borrar esa entrada de caché.



1. Revisión de la Versión. Un resumen de las ultimas actualizaciones de APCu, debido a que el proyecto se está desarrollando en php 5.6 solo se pueden usar las versiones 4, por esta razón aparece como desactualizada en la imagen.



1. Por último, Limpiar Cache, el cual elimina todas las entradas de cache que hay en el equipo

**Ajuste del cache en la máquina.**

Para realizar este ajuste se debe editar el archivo etc/php.ini, al cual se le deben modificar o agregar (si apenas se está instalando), las siguientes líneas según lo que se dese ajustar en Cache.

#Enable/Disable

apc.enabled=1

# Memory Segments

apc.shm\_size=512M

## PHP file cache 1 hour ##

apc.ttl=3600

## User cache 2 hour ##

apc.user\_ttl=7200

## Garbage collection 1 hour ##

apc.gc\_ttl=3600

La primera opción permite habilitar o deshabilitar el módulo de APCu, el siguiente permite definir el tamaño del segmento en memoria para el cache, luego se presenta el tiempo de vida (Time To Live) ttl del cache de archivos, después el de las entradas de datos de usuario, y por ultimo cada cuanto se ejecuta el Grabage Collector o gc, el cual se encarga de recuperar la memoria que está siendo ocupada por Cache que ya no se usa.