

## Gong - Correccion - Mejora #3792

### Agilizar aplicación usando cache

2019-08-27 14:16 - Santiago Ramos

<b>Estado:</b>	Resuelta	<b>Fecha de inicio:</b>	2019-08-27
<b>Prioridad:</b>	Normal	<b>Fecha fin:</b>	
<b>Asignado a:</b>	Santiago Ramos	<b>% Realizado:</b>	100%
<b>Categoría:</b>	Desarrollo	<b>Tiempo estimado:</b>	0.00 hora
<b>Versión prevista:</b>	4.06		
<b>Descripción</b>			
Intentar agilizar la respuesta de la aplicación usando un cache:			
<ul style="list-style-type: none"><li>• Caché en memoria con persistencia</li><li>• Caché compartido para permitir su uso por varias instancias de la aplicación para un único dominio</li></ul>			
La solución a probar es Redis usando el modo de bajo nivel.			
<pre># app/models/gor_conf.rb class GorConfig &lt; ActiveRecord::Base   validates_uniqueness_of :name, message: _("Nombre de parámetro repetido."), case_sensitive: false   validates_presence_of :name, message: _("Parámetro") + " " + _("no puede estar vacío.")    after_update :flush_cache    def self.getValue param_name     Rails.cache.fetch("GorConfig/#{param_name}", expires_in: 24.hours) do       gc = GorConfig.find_by_name(param_name.to_s)       gc ? gc.value : nil     end   end    private    def flush_cache     Rails.cache.delete("GorConfig/#{self.name}")   end end</pre>			

#### Histórico

##### #1 - 2019-08-27 14:18 - Santiago Ramos

- Se actualizó Descripción

##### #2 - 2019-09-27 09:34 - Santiago Ramos

- Categoría establecido a Desarrollo

- Estado cambiado Nueva por Resuelta

- Asignado a establecido a Santiago Ramos

- Versión prevista establecido a 4.06

- % Realizado cambiado 0 por 100

Cerramos el ticket con la solución de cache ya implementada y ejemplos sobre el cálculo de presupuesto total de etapa y cálculos con condiciones de presupuestos y gastos de proyecto.

Se irán abriendo tickets individuales para cada mejora de cache concreta.