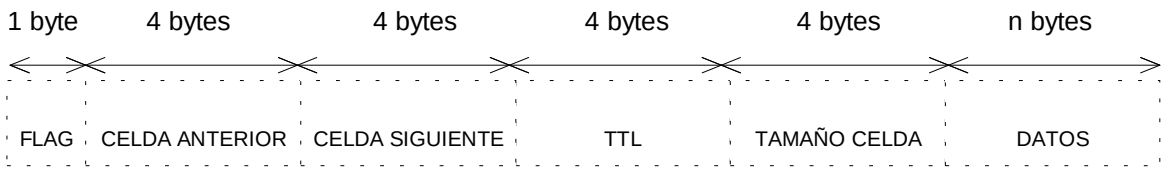
ANEXO IV

ESTRUCTURA DE LA MEMORIA CACHÉ

La zona de memoria que comprende a la memoria caché está compuesto por unidades elementales de datos a las que se le denominan CELDAS. Cada CELDA contiene los datos relativos a un registro fuente, también existen otras celdas que no contienen datos. Las celdas de pueden dividir en CELDAS llenas y CELDAS vacías.

El tiempo de permanencia de un registro fuente en caché está regido por un TTL(tiempo de vida), por lo que cada CELDA llena tendrá información acerca del TTL del registro fuente que contiene. Dentro de la memoria CACHE las CELDAS llenas van a crear una lista ordenada doblemente enlazada, y su criterio de ordenación dentro de dicha lista será el de menor a mayor TTL del registro fuente que contenga dicha celda. Las CELDAS vacías también van crear otra lista doblemente enlazada, siendo su criterio de ordenación de menor a mayor tamaño en bytes de dicha celda.

La memoria caché estará compuesta por una sucesión de celdas, cada una de ellas con el siguiente formato:



Los campos que forman una celda se describen a continuación:

- FLAG: Indica si la celda está llena(contiene datos de un registro fuente en su interior), o si por el contrario la celda está vacía(no contiene datos).

- CELDA ANTERIOR: Indica el offset relativo en bytes, desde el comienzo de esta celda en la memoria caché, hasta la celda situada anteriormente en la lista, de celdas vacías si corresponde a una celda vacía, o la lista de celdas llenas si corresponde a una celda llena.

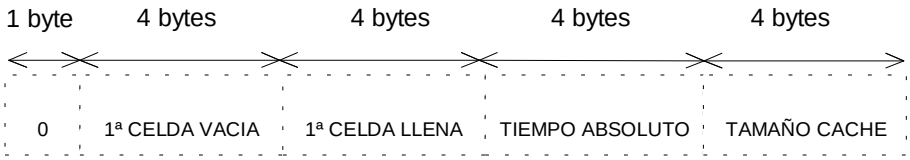
- CELDA SIGUIENTE: Indica el offset relativo en bytes, desde el comienzo de esta celda en la memoria caché, hasta la celda situada posteriormente en la lista, de celdas vacías si corresponde a una celda vacía, o la lista de celdas llenas si corresponde a una celda llena.

- TTL: Para celdas vacías su valor será nulo, y para celdas llenas indica el tiempo de vida(TTL) relativo actual de esta celda en la memoria caché(relativo a la celda situada anteriormente en la lista de celdas llenas). Para que se comprenda mejor, su tiempo de vida total será la suma de todos los TTL relativos de cada celda llena desde el inicio de la lista de celdas llenas, hasta la posición en que se encuentre esta celda.

- TAMAÑO CELDA: Contiene el tamaño total de la celda en bytes.

- DATOS: Contiene el campo DATOS del registro fuente que esté incluido dentro de esta celda

Al comienzo de la memoria caché habrá una celda de cabecera de CACHE, con estructura de una CELDA, que tendrá información acerca de la memoria caché. Esta celda de cabecera tiene los siguientes campos:



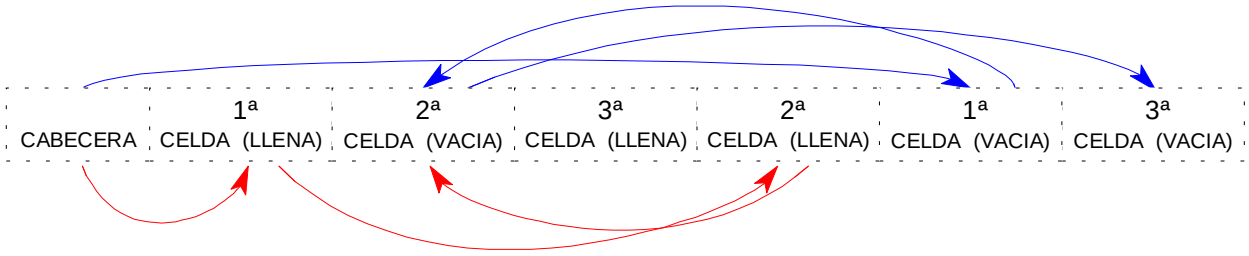
- 1ª CELDA VACÍA: Offset relativo en bytes desde la posición inicial de la caché, hasta la primera CELDA vacía que inicia la lista de CELDAS vacías.

- 1ª CELDA LLENA: Offset relativo en bytes desde la posición inicial de la caché, hasta la primera CELDA llena que inicia la lista de CELDAS llenas.

- TIEMPO ABSOLUTO: Tiempo absoluto del sistema en segundos, desde que se hizo la última actualización de TTLs dentro de la CACHE.

- TAMAÑO DE LA CACHE: Tamaño en bytes de la memoria caché.

Un ejemplo de memoria caché puede ser la siguiente:



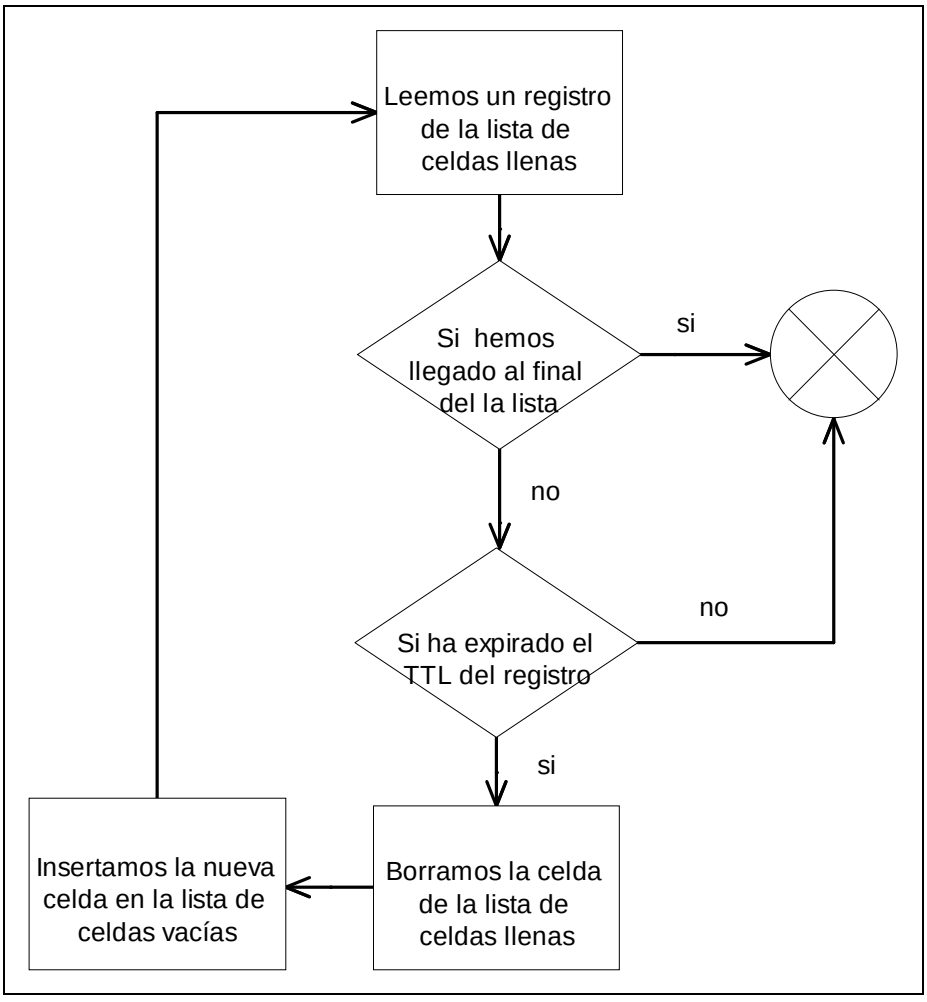
Existen dos listas dentro de la memoria caché, una de CELDAS llenas, y otra de CELDAS vacías. La lista de CELDAS llenas está ordenada según el TTL que le queda para expirar dentro de la caché, y la lista de CELDAS vacías está ordenada de menor a mayor tamaño de celda.

ACTUALIZACIÓN DE CACHÉ

La memoria caché contiene una serie de registros fuentes que dejan de ser válidos una vez transcurrido el TTL(tiempo de vida) de dicho registro fuente. Para la eliminación de estos registros fuentes la clase CACHÉ implementa el método *actualiza_cache*.

En la lista de celdas llenas, las celdas están ordenadas según el tiempo de vida(TTL) que le resta a cada registro fuente dentro de la memoria caché. Además en la cabecera de la caché existe un campo que nos indica la última vez que se actualizó la caché(eliminación de los registros no válidos cuya TTL había expirado).

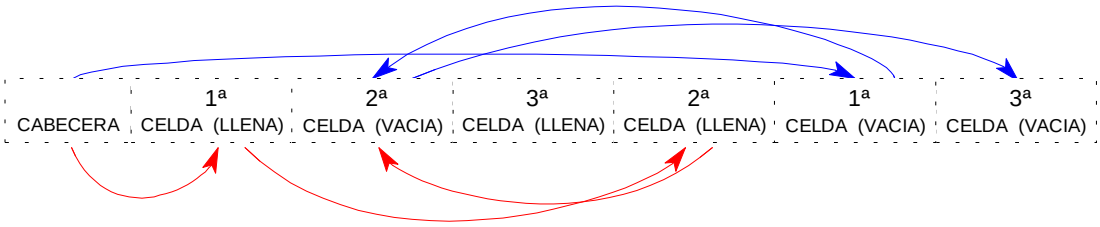
El algoritmo de actualización de la caché es el siguiente:



Desde la primera celda se recorre la lista de celdas llenas, comprobando si la TTL del registro fuente de la celda ha expirado. Si el registro en cuestión ha expirado, se procede a borrar la celda llena correspondiente de la lista de celdas llenas, e insertar esta celda como una nueva celda vacía dentro de la lista de celdas vacías. Este proceso se repetirá, hasta que se alcance un registro cuyo TTL no haya expirado, o bien se llegue al final de listas llenas.

REORGANIZACIÓN DE CACHÉ.

La memoria caché está compuesta por una sucesión de celdas, unas llenas y otras vacías, ordenadas en listas doblemente enlazadas.



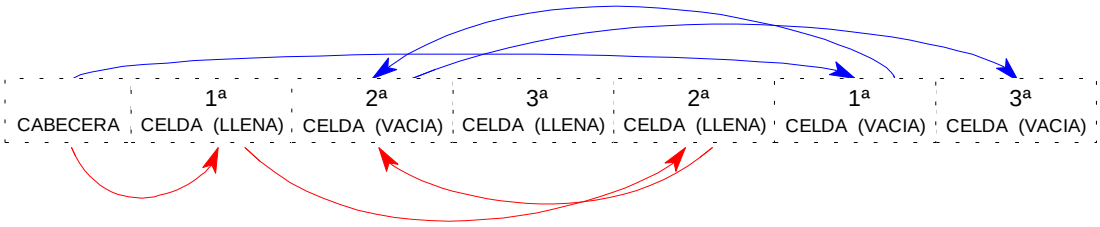
Puede llegar un momento en que existan muchas celdas vacías distribuidas dentro de la memoria caché, lo cual quita eficiencia a la hora de realizar operaciones con la misma, ya que las celdas vacías únicamente se necesitan para obtener espacio libre suficiente donde meter una nueva celda llena(con un registro fuente incluido). Incluso puede llegar un momento que, aun teniendo en conjunto, suficiente espacio libre para insertar un nuevo registro fuente, no haya ninguna celda vacía con suficiente tamaño para este nuevo registro fuente.

Para evitar los problemas anteriores, en la clase CACHE se implementa el método *reorganiza_cache*. Este método coloca, en memoria caché, las celdas llenas consecutivamente al inicio de la caché, cubriendo así huecos libres. Una vez recolocadas las celdas llenas, creará una única celda vacía con el espacio de memoria caché restante no relleno con celdas llenas, asegurando que todo el espacio que tiene libre la memoria caché se va a encontrar dentro de una única celda vacía.

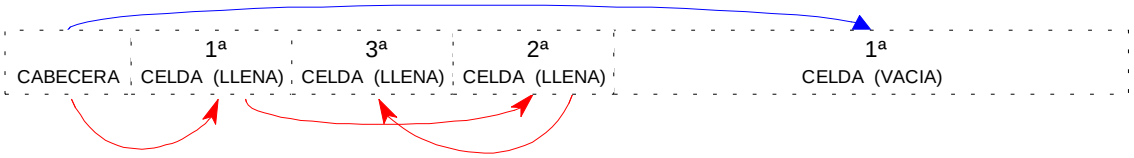
El algoritmo que implementa este método es muy simple: va recorriendo desde el principio de la memoria caché, celda por celda, viendo si es una celda llena o vacía. Si es llena, la coloca a continuación de la última celda llena reubicada, y si es vacía, se la salta continuando con la siguiente celda. Una vez recorridas todas las celdas crea una única celda vacía con todo el espacio de memoria caché que le queda libre.

Un ejemplo de reordenación se muestra a continuación:

Partimos de la siguiente caché



Tras una reorganización de las celdas, la caché queda como sigue:



Se aprecia cómo se han reubicado las celdas llenas, una a continuación de otra, y cómo se han eliminado las celdas vacías quedando una única celda vacía al final de la memoria caché.